Sonja Klingert
Marta Chinnici
Milagros Rey Porto (Eds.)

LNCS 8945

Energy Efficient Data Centers

Third International Workshop, E^2DC 2014
Cambridge, UK, June 10, 2014
Revised Selected Papers

Springer

Lecture Notes in Computer Science 8945

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

More information about this series at http://www.springer.com/series/7411

Sonja Klingert · Marta Chinnici
Milagros Rey Porto (Eds.)

Energy Efficient Data Centers

Third International Workshop, E^2DC 2014
Cambridge, UK, June 10, 2014
Revised Selected Papers

Springer

Editors

Sonja Klingert
University of Mannheim
Mannheim
Germany

Milagros Rey Porto
Gas Natural Fenosa, Technological Projects
Barcelona
Spain

Marta Chinnici
ICT-Division Italian National Agency
for New Technologies Energy
Rome
Italy

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Computer Science
ISBN 978-3-319-15785-6 ISBN 978-3-319-15786-3 (eBook)
DOI 10.1007/978-3-319-15786-3

Library of Congress Control Number: 2015932621

LNCS Sublibrary: SL5 – Computer Communication Networks and Telecommunications

Springer Cham Heidelberg New York Dordrecht London

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

Preface

The spread of new Data Center facilities around the world is an accompanying phenomenon of the twenty-first century always-on-connected-everywhere lifestyle. New Data Centers are being opened up both in Iceland and in urban conglomerations like Frankfurt in Germany and New York in the USA. Legacy Data Centers are being enlarged and continually updated with new equipment and management frameworks. Globally, this evolution results in an ever-increasing energy consumption of Data Centers, stipulating climate change and human impact on the earth's surface. The concept of energy efficiency in Data Centers that only a few years ago was restricted to enhancing IT equipment and cooling, is today addressed to a variety of system level technologies and associated services that will improve energy and environmental performance. The attention is furthermore focused on software running in Data Centers and the way that workload is being processed. However, as power consumers Data Centers additionally need to be viewed as part of a greater system. This applies for instance to the role Data Centers play in the context of Smart Cities. Data Centers form an important part of cities and play a leading role as an enabler of city services, but they are also huge power consumers. This pertains also to Data Centers as major players in the power grid. Reducing the carbon footprint of Data Centers worldwide is therefore a huge challenge considering the pressure of rocketing data amounts. However, promising starting points can be found both in academic and commercial research projects, as the International Workshop on Energy Efficient Data Centers E^2DC 2014 was able to show once again. For the third time, researchers from around the world met in order to commonly advance knowledge and experience of reducing Data Center energy and power consumption and aligning Data Center power profiles to the availability of renewable power resources or constraints from the power grid. The workshop was collocated with the ACM SIGCOMM e-Energy 2014 conference on June 10, 2014 in Cambridge, UK and organized by the EU FP7 project DC4Cities[1].

These proceedings of the workshop give an account on high quality papers from a huge range of relevant technologies within Data Centers as well as regarding the interaction of a Data Center with its environment aimed at saving energy and integrating renewable energy sources.

The first part of the proceedings contains four papers devoted to energy optimization algorithms and models. Yi and Singh proposed a greedy algorithm capable to find a near optimal flow assignment for large-scale Data Center networks. The suggested approach of traffic merging can reduce energy consumption of active switches. With very light load, this kind of traffic merging can save 20–40% energy cost compared to the well-established elastic tree approach. Kuehn introduced a novel method to reduce task graphs with generally distributed task processing times to a single virtual job processing time. Looking at a very different problem, i.e., the challenge of dealing with

[1] FP7 STREP # 609304, www.dc4cities.eu.

frequent blackouts from an unstable power grid, Al-Salim et al. proposed a cyclic blackout mitigation through shifting of HVAC loads by means of queuing optimization. Finally, in a work by Postema and Haverkort a set of stochastic petri net models was applied to the analysis of trade-offs between performance and power consumption of Data Center. This modeling approach is meant to support decisions in the early design stage of a Data Center.

The second part of the proceedings contains four papers focused on the future role of Data Centers in Europe. In this session Anghel et al. presented the European project GEYSER. This project is aimed at integrating Data Centers into Smart Grids and Smart Cities and its scope is to realize an optimized intelligent pervasive sensing and monitoring infrastructure. Gribaudo et al. in their paper proposed an analysis of the influence of application deployment on energy consumption based on the European project ECO2Clouds. The authors investigated different ways to deploy an application in clouds and analyzed simultaneously energy consumption and system performances for each deployment configuration. In their paper "Minimization of Costs and Energy Consumption in Data Centers by a Workload Based Capacity Management" Da Costa et al. proposed a holistic view to Data Center modeling including workloads and cooling. They introduced dynamic power capping to Data Center energy management. Dupont's objective for Data Center energy management was to give a contribution for making Data Centers more energy aware with regard to the availability of renewable energy. To this purpose an energy-aware virtual machines manager based on Constraint Programming (Plug4Green) was applied.

The third part of the proceedings discusses energy efficiency metrics for Data Centers. Capozzoli et al. presented a critical review of performance metrics for energy efficiency in Data Centers aiming to demonstrate the crucial role of thermal management for energy saving. Schlitt et al. in their paper suggested new metrics beyond PUE, capable to consider the adaptability of infrastructure to IT power: the infrastructure power adaptability (IPA) metric representing the power adaptability of the Data Center infrastructure in combination with the power variability (PVar).

The workshop also included three additional presentations: an introduction to the EU projects All4Green by Sonja Klingert (University of Mannheim) and DC4Cities by Marta Chinnici (ENEA, Italian National Agency for New Technologies, Energy and Sustainable Economic Development, Italy), as well as a keynote speech by Ian F. Bitterlin (CTO Emerson Network Power Systems, Visiting Professor at University of Leeds) about the problem of mushrooming data growth which is spurring the energy growth of the global Data Center industry and can only partly be offset by technical evolution and innovation.

We would like to thank Ian F. Bitterlin and all authors for their contributions to the third volume of the E^{2}DC proceeding, and also the reviewers for their effort: they both helped in selecting the best papers and improving the initial submissions. Also, thank you to the Session Chairs Hermann de Meer (University of Passau), Jaume Salom (IREC), and Alfonso Capozzoli (Politecnico di Torino). And we are grateful for an interested and interesting audience, who with lively discussions helped in turning the workshop yet again into a successful event.

Finally, we are grateful for the strong support from the European Commission and the ICT FP7 All4Green project.

October 2014

Sonja Klingert
Marta Chinnici
Milagros Rey Porto

Organization

Workshop Chairs

Sonja Klingert	University of Mannheim, Germany
Marta Chinnici	ENEA, Italy
Milagros Rey Porto	Gas Natural Fenosa, Spain

Technical Program Committee

Hermann de Meer	University of Passau, Germany
Daniel Gmach	HPLabs, USA
Jorjeta Jetcheva	Fujitsu Laboratories of America, USA
Paul Kuehn	University of Stuttgart, Germany
Eric Madeleine	Inria, France
Maria Perez Ortega	GFI, Spain
Mary Ann Piette	Lawrence Berkeley National Laboratory, USA
Barbara Pernici	Politecnico di Milano, Italy
Gunnar Schomaker	OFFIS, Germany
Shaolei Ren	Florida International University, USA
Tomasz Siewierski	Technical University of Lodz

Sponsoring Institutions

EU FP7 Project DC4Cities (#609304)
University of Mannheim, Germany
ENEA, Italy
Gas Natural Fenosa, Spain

Contents

Energy Optimization Algorithms and Models

The Future Role of Data Centres in Europe

Energy Efficiency Metrics for Data Centres

Energy Optimization Algorithms and Models

Agile Traffic Merging for DCNs

Qing Yi[✉] and Suresh Singh

Department of Computer Science, Portland State University,
Portland, OR 97207, USA
{yiq,singh}@cs.pdx.edu

Abstract. Data center networks (DCNs) have been growing in size and their power consumption is becoming a matter of concern. Many recent papers, including ElasticTree and CARPO, propose new near-energy-proportional DCNs, aiming at reducing the power consumption by dynamically powering off idle network switches and links. In this paper, we examine the power optimization model for DCNs, and present a scalable heuristic algorithm that finds a near-optimal subset of network switches and links that satisfies a given traffic load and consumes minimal power. Furthermore, we apply merge networks to each switch in order to power off the idle interfaces of the active switches, thus further reducing the energy consumption of active switches and achieving greater energy savings than ElasticTree. We finish by simulating large-scale fat-tree DCNs and comparing the energy cost of our techniques versus the ElasticTree method. The results demonstrate that our solution is more energy efficient.

Keywords: Data center · Routing · Merging · Fat-tree

1 Introduction

Data center networks (DCNs) are designed to support high communication bandwidth between servers. However, since many data centers have light loading for significant lengths of time or localized loading, large parts of these networks remain under-utilized. Networking equipment continues to consume energy even when sitting idle, and therefore contributes significantly to the overall operating costs of the data center over time. Two notable approaches have been studied to address this problem. In ElasticTree [12], the network forces traffic to the leftmost switches in a fat-tree topology to allow powering off unused switches. An orthogonal approach [16] replaces larger switches with many smaller ones in a fat-tree DCN to enable better packing of traffic into fewer switches compared with ElasticTree, hence achieving greater energy savings. Additionally, there are other approaches that primarily focus on changing link rate in response to load.

All these approaches achieve the goal of saving energy, but they have not proved to be able to adapt to changes in loading patterns efficiently. For instance,

This work was funded by the NSF under award No. 1217996.

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 3–16, 2015.
DOI: 10.1007/978-3-319-15786-3_1

the approach proposed in [16] is static since the switch sizes and topology is fixed at design time. As a result, the design is only energy efficient for the specific loads that the network was designed for. Indeed, as shown in the paper, only when loads are smaller than 30 % is the topology using many small switches more energy efficient than the one using large switches. ElasticTree is a more adaptive mechanism for saving energy since it computes routes every second. However, there is still considerable amount of energy wasted by switches that are powered on with light traffic on all its interfaces. Indeed, the number of such lightly loaded switches is significant, and, as a result, the overall energy savings are sub-optimal.

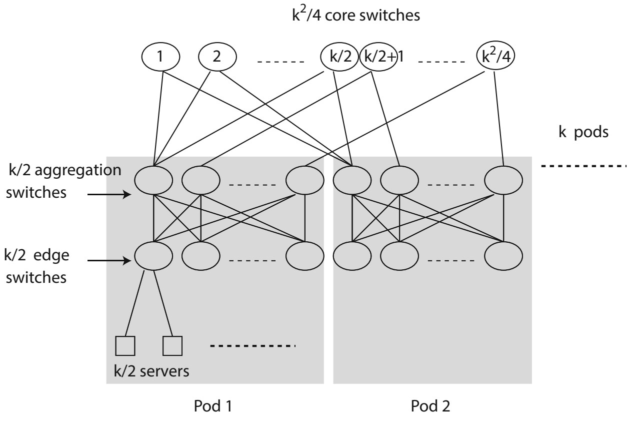


Fig. 1. Fat-tree model

To explain the deficiency of the prior approaches as well as to motivate our contribution, we consider a 3-layer fat-tree DCN shown in Fig. 1. The fat-tree network is divided into k pods, each of which has two layers of switches. The bottom layer of $k/2$ switches are called edge switches, while the upper layer of $k/2$ switches are called aggregation switches. $k/2$ servers are attached to each edge switch and each edge switch is connected to each of the aggregation switches in the same pod. The leftmost aggregation switch in each pod is connected to the first $k/2$ core switches, and so on. Thus there are $k^2/4$ core switches in total.

In previous approaches such as ElasticTree, the $k^2/2$ edge switches are always fully powered on as they are connected to servers. Although link rate adaptation at low loads will reduce energy consumption, the reduction is only a very small fraction of the interface energy cost. At the aggregation layer, switches that are powered on in ElasticTree do not fully load their interfaces (facing the edge switches) because each interface is connected to an edge switch. Even if the edge switch has very little traffic going to the aggregation switch, the link is fully powered on but very lightly loaded. *Our contribution in this paper is to enable powering off a subset of interfaces in active switches. This is accomplished by merging traffic carefully.*

1.1 Our Approach: Merging

Consider the case of an edge switch connected to $k/2$ servers, each of which offers a load of λ (expressed as a fraction of link rate). Then the total traffic to this switch from the servers is $k\lambda/2$. If $k = 8$, then for $\lambda \leq 0.25$, one switch interface will suffice to handle the traffic from all four servers. In other words, if there was a way to *merge* the traffic from the four servers, we could potentially power off three of the four switch interfaces connected to the servers. In a previous paper [14], we provided a design of a hardware device called *merge network*. Rather than repeating that discussion here, we provide a *functional* model of what such a network does, and then use it in the remainder of this paper. Figure 2 shows a $\frac{k}{2} \times \frac{k}{2}$ merge network connected to $k/2$ servers on one side and to the $k/2$ ports of an edge switch on the other side.

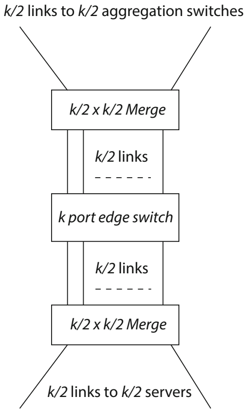


Fig. 2. Merge networks applied to a switch

1. The merge network is a fully analog device with no transceivers and, as a result, its power consumption is below one watt. The merge network can be visualized as a train switching station where trains are re-routed by switching the tracks (rather than store-and-forward).
2. Consider the uplink from the servers to the merge network. All traffic coming into the merge network is output on the *leftmost* $m \leq k/2$ links connected to the m leftmost interfaces of the switch, where $m = \lceil k\lambda/2 \rceil$ (assuming a normalized unit capacity for links). This is accomplished internally by sensing packets on links and automatically redirecting them to the leftmost output from the merge network that is free.
3. On the downlink to the servers, traffic from the switch to the $k/2$ servers is sent out along the leftmost $m \leq k/2$ switch interfaces to the merge network. The packets are then sent out along the $k/2$ links attached to the servers from the output of the merge network. The manner in which this is accomplished is described in [14] (note that the challenge is to correctly route the packets flowing through the merge network to the appropriate destinations).

To apply merge networks to a fat-tree network, we add two $\frac{k}{2} \times \frac{k}{2}$ merge networks to each edge switch as shown in Fig. 2. The connections are similar for each aggregation switch. For the core switches, we connect a $k \times k$ merge network.

1.2 Contributions and Paper Organization

In this paper, we revisit the problem of reducing energy consumption in fat-tree DCNs by attaching merge networks to each switch. In addition to the savings we obtain by forcing traffic to the left as in ElasticTree, we achieve significant additional savings by powering off unused interfaces in active switches which is made possible by merge networks.

The remainder of the paper is organized as follows. In the next section, we present an optimization model for computing routes with the goal of minimizing energy consumption. This model is different from those developed in previous papers because we also consider merge networks and our minimization function includes the number of active interfaces as a parameter. In Sect. 3, we present an algorithm that computes routes every second based on traffic load. The results of the optimization are compared against the simulated algorithm for a variety of loading scenarios, which show good agreement between the two. Finally, in Sect. 4, we present the results of simulating more realistic larger fat-tree networks and analyze the energy savings obtained when using merge networks. Section 5 presents related work and Sect. 6 summarizes the main contributions and future work.

2 Minimizing Energy Consumption

To compute the minimal power required by a DCN, we formulate a power model for all network elements including switches and links. A network $G(V, E)$ is given, where V is the set of nodes in the network and E is the set of links. We consider both the end hosts and the switches as network nodes and thus we have $V = V_1 + V_2$, where V_1 is the set of end hosts and V_2 is the set of switches. Link $(u, v) \in E$ connects node u and node v $(u, v \in V)$. Assuming each switch consumes power P_s and each link consumes power P_l, the total power consumed by the entire network can be expressed as

$$P_{total} = \frac{1}{2} \sum_{u \in V_2} k_u \times P_l + n \times P_s + \frac{\epsilon}{2} \times \sum_{u \in V, w \in V_u} f_{u,w} \qquad (1)$$

where n is the number of active switches and k_u is the number of active interfaces of switch u. V_u is the set of nodes connecting to node u. ϵ is the dynamic energy consumption factor representing the power consumption per unit data transmitted through a link. $f_{u,v}$ is amount of traffic flow assigned to link (u, v). We use binary variables y_u and $x_{u,v}$ to represent the power state of node u and link (u, v), respectively. For instance, if $x_{u,v} = 1$, link (u, v) is active; if it is 0, link (u, v) is idle and can be powered off. Therefore, k_u and n can be written as

$$n = \sum_{u \in V_2} y_u \qquad (2)$$

$$\forall u \in V_2, \quad k_u = \sum_{w \in V_u} x_{u,w} \qquad (3)$$

2.1 Optimization Model

Based on the power model defined above, we define an optimization problem in order to find the optimal flow assignment that involves a minimum subset of active network elements, (n, k_u), with the minimal total power consumption P_{total} for a given network topology and a traffic load. This optimization problem is a Mixed Integer linear Programming problem (MIP), and is an extension to the capacitated Minimum-Cost MultiCommodity Flow problem (MCMCF). A classical MCMCF problem is subject to three constraints - *capacity constraint*, *flow conservation* and *demand satisfaction*, which are written as

$$\forall (u,v) \in E, \;\; f_{u,v} \leq c x_{u,v} \tag{4}$$

$$\forall u, \;\; u \notin S \; and \;\; u \notin D, \;\; \sum_{w \in V_u} f_{u,w} - \sum_{w \in V_u} f_{w,u} = 0 \tag{5}$$

$$\begin{cases} \forall s \in S, \sum\limits_{w \in V_s} g^i_{s,w} - \sum\limits_{w \in V_s} g^i_{w,s} = t^i_{s,d} \\ \forall d \in D, \sum\limits_{w \in V_d} g^i_{w,d} - \sum\limits_{w \in V_d} g^i_{d,w} = t^i_{s,d} \end{cases} \tag{6}$$

where c is the capacity for each link. S is the set of source nodes and D is the set of destination nodes. V_s and V_d is the set of switches that connect to source node s and sink node d, respectively. $f_{u,w}$ is the total flow assigned on link (u, w) and $f_{u,w} = \sum\limits_i g^i_{u,w}$, where $g^i_{u,v}$ represents the flow of the ith traffic demand $t^i_{s,d}$ routed through link (u, v).

Capacity constraint (4) takes account of maximum link utilization and ensures that the total traffic flow assigned to a link does not surpass the link capacity. The *capacity constraint* also forces flows to go through active links only. For example, inactive link (u, v) has $x_{u,v} = 0$, which causes $f_{u,v} = 0$ meaning no traffic flow is assigned to this link. *Flow conservation* (5) ensures that traffic entering an intermediate node equals to traffic exiting from it. *Demand satisfaction* (6) describes that the overall traffic departing a source node or entering a destination node equals to the traffic demand.

Besides these three constraints, the *bidirectional link* rule ensures that both directions of a link are powered on if there is a flow assigned to either direction of the link. The *bidirectional link* constraint is expressed as

$$\forall (u,v) \in E, \quad x_{u,v} = x_{v,u} \tag{7}$$

Additionally, we include constraints that correlate the power states of switches and links. For each node u and the connected links (u, w) and (w, u), we have

$$\forall u \in V, \; \forall w \in V_u, \; x_{u,w} \leq y_u \;\; and \;\; x_{w,u} \leq y_u \tag{8}$$

$$\forall u \in V, \;\; y_u \leq \sum_{w \in V_u} (x_{u,w} + x_{w,u}) \tag{9}$$

Constraint (8) makes sure that a switch is powered off only when all its connected links are powered off, and constraint (9) ensures that a switch be powered off when all its connected links are powered off. Optionally, we can include a *non-splitting* constraint as follows to prevent flow splitting:

$$\forall i, \forall (u,v) \in E, \;\; g^i_{u,v} = t^i \times r^i_{u,v} \tag{10}$$

where $r^i_{u,v}$ is a binary decision variable that indicates whether the traffic demand t_i is assigned to link (u, v). Constraint (10) ensures that $g^i_{u,v}$, the flow assignment to link (u, v), is either equal to the ith traffic demand t_i or equal to zero.

Furthermore, we define heuristic constraints to reduce the problem size. For example, since a k-ary fat-tree network has $5k^2/4$ switches and each switch has at most k active links, we explicitly apply an upper bound and a lower bound to k_u and n as $0 \leq k_u \leq k$ and $0 \leq n \leq \frac{5}{4}k^2$, which can greatly improve convergence time for the problem.

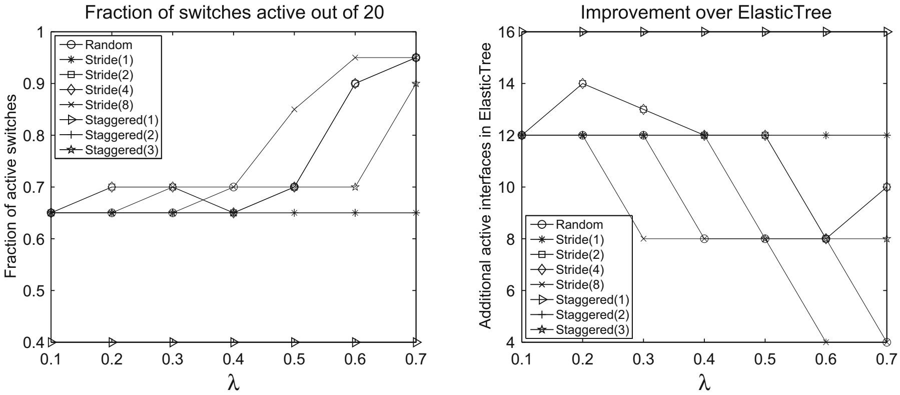
We implement the power optimization model using CPLEX, which is an optimization solver for integer programming problems. For a given traffic matrix, the optimization model outputs the numbers of active switches and links, and the flow assignment to each link corresponding to every traffic flow demand. Our model is implemented with both flow-splitting and non-flow-splitting options.

2.2 Energy Savings Due to Traffic Merging

A primary contribution of this paper is to illustrate the additional energy savings achieved by merge networks when compared with approaches such as ElasticTree. To quantify this benefit, we run the optimization problem on several different types of network loadings for a small fat-tree topology of size $k = 4$. In this topology, there are 8 edge switches, 8 aggregation switches and 4 core switches. For each edge switch, there are 2 servers connected for a total of 16 servers in 4 pods. We assume that there is a 2×2 merge network connected to either side of each edge and aggregation switch and there is a 4×4 merge network connected to each core switch.

Traffic patterns in data centers can vary greatly, and to ensure our results are widely applicable, we run the optimization algorithm on the following types of traffic: *Random*, $Stride(n)$, $Staggered(n)$ [4]. In *Random*, the source and destination are randomly selected from among the servers. For $Stride(n)$, the destination of a flow from server i is server $[(i + n) \mod 16]$, where servers are numbered left to right as $0, 1, \cdots, 15$. For example, in a $k = 4$ fat-tree network, $Stride(1)$ has almost half of the traffic goes between servers connected to the same edge switch and the other half traffic goes to aggregation and core switches. On the other hand, $Stride(4)$ sends all traffic between pods, resulting in a larger number of switches to participate in forwarding traffic. The *Staggered* traffic model assigns a probability p_1 for traffic going to a server in the same subnet (i.e., connected to the same edge switch), a probability p_2 for traffic going to a server in the same pod but different subnet, and a probability $1 - p_1 - p_2$ where the flow is destined to a server in a different pod. By varying these probabilities, we can generate a large number of different loading patterns.

Figure 3a plots the percentage of active switches for our approach as well as for ElasticTree for different loading patterns and different loads. As we have expected, the number of active switches for $Stride(1)$ does not vary with λ. This is because almost all the traffic goes to the server in the same subnet or in the same pod and therefore, the active switches required are always the eight edge switches, one aggregation switch per pod and one core switch. $Stride(8)$ shows



(a) Number of active switches

(b) Total number of active interfaces

Fig. 3. Difference in number of active switches and active interfaces network-wide

the highest number of active switches because all the traffic is inter-pod traffic and hence more core switches are used.

In order to illustrate the potential benefits of traffic merging, we take a difference between the total number of active interfaces when using ElasticTree and using traffic merging with the above optimization. The results, shown in Fig. 3b, clearly illustrate the benefits of merging. In the case of *Stride*(1), ElasticTree uses 12 more interfaces than merging. The reason is that one aggregation switch is active per pod. In ElasticTree, all the four interfaces to this switch are active (albeit with very low traffic). In our approach, in contrast, we merge the traffic using a merge network and use only a single interface of the switch.

The overall energy cost of a switch can be roughly partitioned into the cost of the chassis and the cost of the interfaces. As described in [8,16], a reasonable approximation to the cost of a switch is

$$\text{Switch Cost } = C + m \log m + m$$

where m is the number of active switch ports. The constant C accounts for static costs of a switch such as fan, etc. The second term corresponds to the cost of the interconnection fabric within the switch, which is a significant contributor to energy consumption (typically $30\,\% \sim 40\,\%$). This cost scales as $m \log m$ for a switch with m active ports. The last term is the cost contribution from the active interfaces. This term folds into itself the cost of the line cards that the interfaces are on. For the purpose of comparing the *overall cost reduction* of traffic merging relative to ElasticTree, we set C to $50\,\%$ of the maximum switch cost and express it as

$$C = m_{\max} \log m_{\max} + m_{\max}$$

where $m_{\max}$ is the number of switch ports. If the traffic load fraction going to a switch is λ, the merge network will switch the traffic to the leftmost $k = \lceil \lambda m \rceil$ ports. Thus, the cost of a switch with merge networks is written as

$$\text{Traffic Merging Switch Cost } = C + k\log k + k$$

Therefore, the fraction of cost savings of traffic merging over ElasticTree is calculated as

$$\text{Cost Savings} = \frac{m\log m - k\log k + m - k}{C + m\log m + m}$$

Figure 4 plots the fraction of reduction of network cost using traffic merging over ElasticTree. It is noteworthy that, for all traffic patterns and across all loads, the traffic merging reduces the overall energy cost even for a small-sized network consisting of 20 switches. These savings are more substantial when we consider realistic DCNs as we do later in this paper.

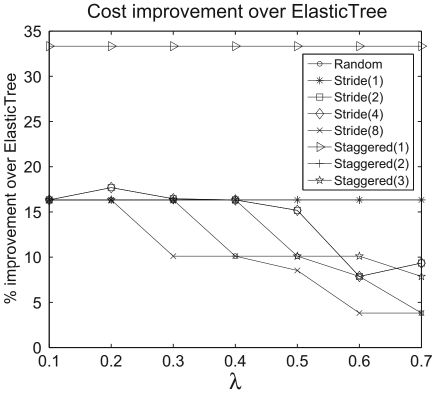


Fig. 4. Reduction in total cost when using traffic merging

3 Greedy Flow Assignment

The optimization model can find the optimal flow assignment for a given network topology and traffic loading. However, since a MCMCF problem is NP-hard, the optimization problem for a large-sized DCN cannot be solved within a reasonable time frame. To address this problem, we propose a heuristic greedy algorithm to find a near-optimal flow assignment.

3.1 Algorithm

Our greedy flow assignment algorithm is based on Dijkstra's algorithm that solves the shortest path problem. For a given network topology and a given traffic flow, our algorithm finds a route between the source node and the destination node with sufficient bandwidth and the lowest cost. We define the cost of a route as the sum of the cost of the nodes and links along the route. By carefully defining the value of the cost of each node and each link, our greedy algorithm finds the lowest-cost route for each traffic flow incrementally and ultimately obtains the

Algorithm 1. Flow assignment algorithm

```
function FLOWASSIGN($s, d, t$)
    for each vertex $v$ in $Graph$ do
        $dist[v] \leftarrow Infinity$
        $previous[v] \leftarrow nil$
    $dist[s] \leftarrow 0$
    insert $(s, dist[s])$ to $Q$
    while $Q$ is not empty do
        $u \leftarrow$ first pair in $Q$
        remove $u$ from $Q$
        if $u == d$ then
            break
        for each neighbor $v$ of $u$ do
            if $capacity(u, v) > t$ then
                $alt \leftarrow dist[u] + cost(v) + cost(u, v)$
            else
                $alt \leftarrow Infinity$
            if $alt < dist[v]$ then
                erase $(v, dist[v])$ from $Q$
                $dist[v] \leftarrow alt$
                $previous[v] \leftarrow u$
                insert $(v, dist[v])$ to $Q$
    $v \leftarrow s$
    while $v! = nil$ do
        insert $v$ to $route$
        $v \leftarrow previous[v]$
    return $route$
```

optimal routing for all the traffic flows that uses the minimum number of switches and links. The greedy algorithm is described as in Algorithm 1.

Each link in the network has a *fixed capacity*. We only assign a flow to a link when there is available capacity in that link. Once a flow is assigned to a link, the corresponding amount of capacity is subtracted from the available capacity of the link. The $cost(u, v)$ is a constant value of 2 for all links, which counts each link along the route, no matter whether it was used previously or not. Therefore, we will always find the shortest route that involves a minimum number of links. $Cost(v)$ is the cost of node v and the value is initialized as 1 for all nodes. *Once a node was used for a route once, its cost value will be updated to* 0. This will make sure that a switch that has been used in a previous route has a higher priority to be reused. As a result, we can minimize the overall number of active switches. We set higher cost for links than for switches to avoid routing loops.

3.2 Validation of Greedy Algorithm

The greedy algorithm is not optimal but, as we show below, the routes produced by the algorithm are very close to those produced by solving the optimization formulation in Sect. 2. We use the same fat-tree topology as in Sect. 2.2 with $k = 4$. For a given traffic load, we generate a number of packet traces following certain DCN traffic patterns [5]. The packet traces in each one-second interval are organized as a traffic matrix and is fed into the CPLEX optimization model and the simulated greedy algorithm. We obtain the number of active switches and active interfaces for the eight traffic patterns and seven traffic loads shown in Table 1.

The results we get from the simulated greedy algorithm are very close to those get from the CPLEX optimization model, especially for the lighter loads. Since the optimization model can only scale to a fat-tree DCN with $k = 6$, we use the greedy algorithm to simulate the optimization of a large-scale fat-tree network in the next part of this paper.

Table 1. Number of active switches and interfaces from optimization vs. from simulated greedy algorithm

load	Staggered(1)				Staggered(2)				Staggered(3)				Random			
	act SW		act I/F		act SW		act I/F		act SW		act I/F		act SW		act I/F	
	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim
10%	8	8	16	16	13	13	40	40	13	13	40	40	13	13	40	40
20%	8	8	16	16	13	13	40	40	13	13	40	40	13	13	40	40
30%	8	8	16	16	13	13	40	40	13	13	40	40	14	14	48	48
40%	8	8	16	16	13	13	40	40	13	13	40	40	14	14	48	48
50%	8	8	16	16	13	13	40	40	14	14	48	47.2	14	14	48	48
60%	8	8	16	16	13	13	40	40	14	14	48	53.4	18	19	64	72
70%	8	8	16	16	13	13	40	40	18	19	64	72	19	19	72	72

load	Stride(1)				Stride(2)				Stride(4)				Stride(8)			
	act SW		act I/F		act SW		act I/F		act SW		act I/F		act SW		act I/F	
	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim	opt	sim
10%	13	13	40	40	13	13	40	40	13	13	40	40	13	13	40	40
20%	13	13	40	40	13	13	40	40	13	13	40	40	13	13	40	40
30%	13	13	40	40	13	13	40	40	14	14	48	48	13	14	40	44
40%	13	13	40	40	13	13	40	40	14	14	48	48	14	14	48	48
50%	13	13	40	40	17	17	58	56.2	17	17	60	60.8	17	17	60	63.6
60%	13	13	40	40	18	18	64	64	19	19	72	72	19	20	72	75.2
70%	13	13	40	40	19	18	66	64	19	19	72	72	19	20	72	75.6

4 Simulation Results

We simulate a $k = 12$ fat-tree network which supports 432 servers and 180 12-port switches. In this network, there are 12 pods and each of which has six edge switches and six aggregation switches. We assume that each of the core switches has extra ports to be connected to external Internet through border routers. We assign 1 Gbps capacity to each link. We experiment with synthetic traffic data from a traffic generator and real packet traces from a university data center. *Since flow splitting will incur packet reordering cost, which is not a desirable practice in real data centers, we implement our simulation using non-splitting flow assignment.*

4.1 Synthetic Traffic Data

We generate network traffic following ON/OFF patterns derived from many production data centers [5,6]. The duration of the ON and OFF periods and the packet interarrival time follow the lognormal distribution. Like in Sect. 2.2, we study traffic patterns *Random*, *Stride*(n) and *Staggered*(n). In a $k = 12$ fat-tree network, every edge switch is connected to six servers. For *Stride*(1), flows sourcing from the first five servers of the edge switch go to servers in the same subnet, and flows from the sixth server travel to the server in the next subnet or in the next pod. In contrast, all the flows in *Stride*(6) go to the neighboring subnet, and all the traffic in *Stride*(36) and *Stride*(216) is inter-pod traffic.

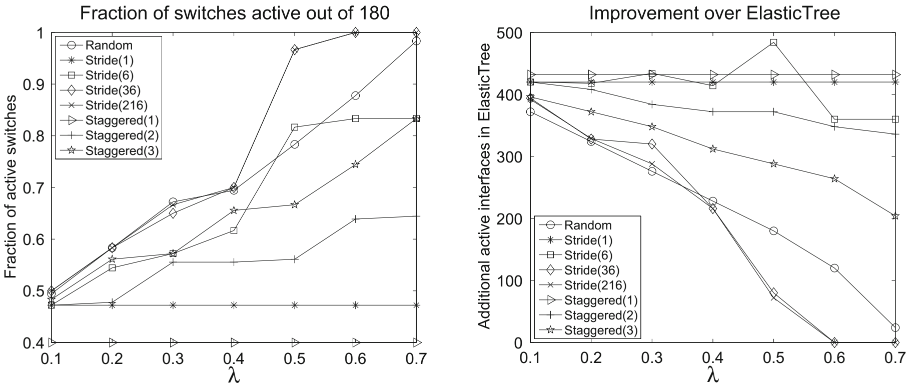
Table 2. Probabilities of flows going to the same subnet (p_1), to other subnets in the same pod (p_2), and to different pods ($1 - p_1 - p_2$) for all traffic suites studied.

Traffic Suite	p_1	p_2	$1 - p_1 - p_2$	Traffic Suite	p_1	p_2	$1 - p_1 - p_2$
Staggered(1)	100 %	0 %	0 %	*Stride(1)*	83.3 %	13.9 %	3 %
Staggered(2)	50 %	30 %	20 %	*Stride(6)*	0 %	83.3 %	16.7 %
Staggered(3)	20 %	30 %	50 %	*Stride(36)*	0 %	0 %	100 %
Random	1.2 %	7 %	91.8 %	*Stride(216)*	0 %	0 %	100 %

For *Staggered*(n), it has fixed values for p_1 and p_2 as the probabilities of flow going to the same subnet and other subnets of the same pod, respectively. Table 2 shows these values for all traffic suites studied.

The load fraction λ offered by each server varied from 0.1 to 0.7. Our simulation outputs the number of active switches (Fig. 5a) and the number of active interfaces of each switch with varies traffic loads and patterns. In general, the number of active switches increases with the traffic load. However, both *Stride*(1) and *Staggered*(1) have constant number of active switches and active interfaces. This is because, for *Stride*(1), all loads can be satisfied by using a minimum spanning tree. For *Staggered*(1), only edge switches are used since all the traffic flows are local traffic within the same subnet.

Figure 5b illustrates the difference of total numbers of active interfaces of a DCN using merge networks versus ElasticTree. It shows that more interfaces of the active switches become idle when the traffic is light, which demonstrates that traffic merging can save more energy with lighter traffic (Fig. 6). *Stride*(1) achieves the most energy savings over ElasticTree (around 42 %) because, for each active edge switch, the energy consumed by the five idle interfaces is wasted. *Staggered*(1) saves 30 % energy consumption since for the entire network, only half of the interfaces (facing the severs) of the edge switches are used.



(a) Number of active switches.

(b) Total number of active interfaces.

Fig. 5. Number of active switches and active interfaces network-wide for a $k = 12$ fat-tree network

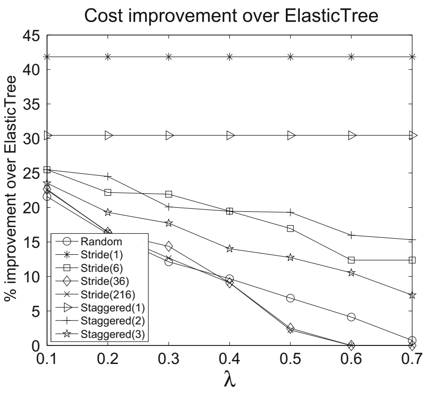


Fig. 6. Reduction in total cost when using traffic merging

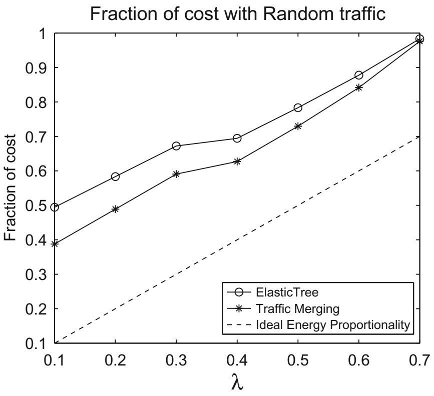


Fig. 7. Compare the total cost of ElasticTree and traffic merging

ElasticTree provides an energy-efficient solution for DCNs. However, the drawback of ElasticTree is that, a DCN still consumes a large amount of power with light load [12]. In contrast, as shown in Fig. 7, our approach reduces energy consumption when the network is lightly loaded, which demonstrates that traffic merging achieves better energy proportionality than ElasticTree.

4.2 Empirical Traffic Data

We use packet traces from a university data center published by Benson *et al.* [5]. This university data center has about 500 servers providing services for campus users. 60 % of the traffic is for Web services and the rest is for other applications such as file sharing services. Traffic traces are captured by a sniffer installed at a randomly selected switch in the data center. Figure 8 illustrates the total load of the packet traces within 50 min. The overall load is very small for a high-bandwidth fat-tree topology. We observe that power cost decreases from 30 % to 17 % when applying merge networks compared with ElasticTree (Fig. 9).

5 Related Work

The development of Internet communication and service applications requires increased bandwidth support and more powerful routing protocols for a DCN. For the past few years, many new DCN topologies have been proposed, including fat-tree [4], Clos [7] and flattened butterfly [13]. These hierarchical interconnection topologies are designed to maximize cross-section bandwidth and optimize the cost-effect ratio. Alternatively, some server-centric DCN architecture, such as DCell [10], BCube [9] and CamCube [2], use simple switches and push network routing to the servers, thus obtaining better scalability and fault-tolerance.

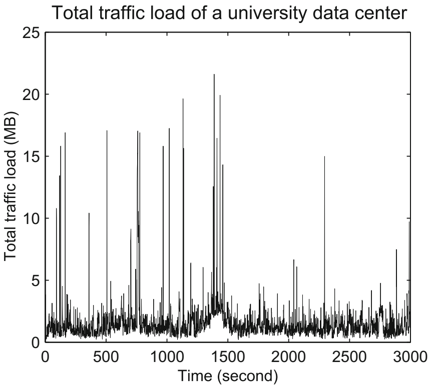


Fig. 8. Traffic load of a university data center

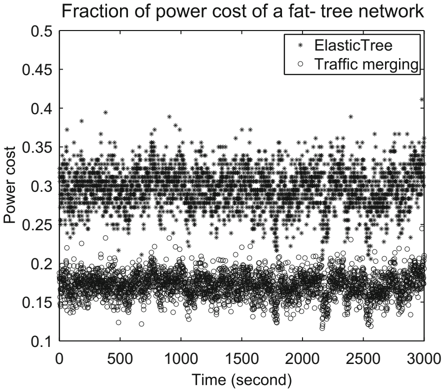


Fig. 9. Energy savings when using traffic merging

In general, these proposed network topologies are intended to support increasing number of servers and provide high capacity for bandwidth-hungry services. However, the rising energy cost of DCNs has attracted the attention of many researchers. New designs of energy-efficient network devices have been examined. For example, Abts *et al.* [1] explore dynamically tuning the link rate according to traffic intensity to save energy. Inspired by the earlier work of Gupta *et al.* [11], other researchers propose energy-proportional DCN topologies through powering off idle interfaces or devices. For example, Heller *et al.* proposed ElasticTree [12] that adapts the network topology to varying traffic loads. *CARPO* [15] examines the dynamic topology by consolidating timely-negative-correlated flows into a smaller set of links and shutting off unused ones. More recently, Adnan and Gupta propose an online path-consolidation algorithm to right-size network dynamically [3]. Widiaja *et al.* [16] compare the energy savings of optimizing fat-tree networks deployed with different sizes of switches and conclude that, with the same number of servers, it is more energy efficient to use more smaller-sized switches than using less large-sized switches when the traffic is highly localized.

Our work complements prior work by utilizing a universal greedy flow assignment algorithm to find the optimal network subset. The greedy bin-packing algorithm used in ElasticTree leverages the regularity of hierarchical DCNs and uses left-most heuristics to find the shortest route. Our greedy algorithm can find flow assignments close to the MIP model, for not just hierarchical network topologies, but also random or irregular DCN topologies. Furthermore, we apply merge networks to each switch and scale switch energy cost to the number of busy interfaces of each switch.

6 Conclusions

This paper addresses the power optimization problem of DCNs. We present a greedy algorithm that is applicable to all types of DCN topologies. We demonstrate that this algorithm can find near-optimal flow assignments comparable to solutions achieved from optimization model. In addition, by applying merge networks to each switch, we further reduce power consumption of active switches. With very light load, our approach saves 20 % $\sim$ 40 % energy cost compared with ElasticTree, depending on the traffic types. Traffic with small number of inter-pod and inter-subnet flows can benefit even more from traffic merging. In the future, we will apply merge networks to switches in different ways to explore methods that further reduce energy consumption of DCNs.

References

1. Abts, D., Marty, M.R., Wells, P.M., Klausler, P., Liu, H.: Energy proportional datacenter networks. In: ISCA (2010)
2. Abu-Libdeh, H., Costa, P., Rowstron, A., O'Shea, G., Donnelly, A.: Symbiotic routing in future data centers. In: SIGCOMM, pp. 51–62 (2010)
3. Adnan, M.A., Gupta, R.: Path consolidation for dynamic right-sizing of data center networks. In: Proceedings IEEE Sixth International Conference on Cloud Computing (2013)
4. Al-Fares, M., Loukissas, A., Vahdat, A.: A scalable, commodity data center network architecture. In: SIGCOMM, pp. 63–74 (2008)
5. Benson, T., Akella, A., Maltz, D.A.: Network traffic characteristics of data centers in the wild. In: IMC (2010)
6. Benson, T., Anand, A., Akella, A., Zhang, M.: Understanding data center traffic characteristics. In: WREN (2009)
7. Clos, C.: A study of non-blocking switching networks. Bell Syst. Tech. J. **32**(2), 406–424 (1953)
8. Eramo, V., Germoni, A., Cianfrani, A., Miucci, E., Listanti, M.: Comparison in power consumption of MVMC and BENES optical packet switches. In: Proceedings IEEE NOC (Network on Chip), pp. 125–128 (2011)
9. Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., Lu, S.: BCube: a high performance, server-centric network architecture for modular data centers. In: SIGCOMM, pp. 63–74 (2009)
10. Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y., Lu, S.: DCell: a scalable and fault-tolerant network structure for data centers. In: SIGCOMM, pp. 75–86 (2008)
11. Gupta, M., Singh, S.: Greening of the internet. In: Proceedings of ACM SIGCOMM (2003)
12. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S., McKeown, N.: ElasticTree: saving energy in data center networks. In: NSDI (2010)
13. Kim, J., Dally, W.J., Abts, D.: Flattened butterfly: a cost-efficient topology for high-radix networks. In: ISCA, pp. 126–137 (2007)
14. Singh, S., Yiu, C.: Putting the cart before the horse: merging traffic for energy conservation. In: IEEE Communications Magazine, June 2011
15. Wang, X., Yao, Y., Wang, X., Lu, K., Cao, Q.: CARPO: correlation-aware power optimization in data center networks. In: INFOCOM, pp. 1125–1133 (2012)
16. Widjaja, I., Walid, A., Luo, Y., Xu, Y., Chao, H.J.: Switch sizing for energy-efficient datacenter networks. In: Proceedings GreenMetrics 2013 Workshop (in Conjunction with ACM Sigmetrics 2013), Pittsburgh, PA, June 2013

Performance and Energy Efficiency of Parallel Processing in Data Center Environments

Paul J. Kuehn(✉)

Institute of Communication Networks and Computer Engineering,
University of Stuttgart, Stuttgart, Germany
paul.j.kuehn@ikr.uni-stuttgart.de

Abstract. A novel approach is presented for the analysis of parallel processing of stochastic workload by multi-processor/multi-core processing resources in data center environments. The method is based on job workload descriptions by task graphs with generally-distributed task execution times and task scheduling under consideration of prescribed precedence and synchronization constraints. For the analytic performance evaluation, task graphs are restricted to the analysis of directed acyclic graphs which are reduced by stepwise aggregations of tasks. The reduction allows to aggregate the whole task graph under a given number n of processing elements to a single virtual job processing time with average value h_v and coefficient of variation c_v. By this way, the whole multi-processor system can be modeled by a queuing system of type GI/G/1 from which the response time T_R and the speedup factor S(n) is derived. Finally, the influence of the stochastic properties of the workload on the performance and on energy efficiency of parallel computing will be studied and compared with serial computing on a multi-processor system modeled by a queuing system of the type M/G/n.

Keywords: Parallel processing · Task graph · Graph reduction · Queuing system · Performance evaluation · Energy efficiency

1 Introduction

Parallel processing has received enormous attention in the last 50 years, c.f. fundamental presentations in the books as [1–3]. Most of the studies in the early times of computer science addressed problems of scheduling tasks with given task processing times to be processed on a single or few processing elements under various scheduling strategies based on constant processing times, order of arrival, priority classes or deadlines for the execution. Many results are known from this research on optimum scheduling with respect to the shortest possible execution duration until completion of a given workload. For the description of more complex systems with precedence and synchronization constraints, Petri Nets (PN) [4] have proved as an excellent modeling methodology to guarantee the correct execution and to detect deadlock situations by the control of state transitions using places and tokens but were not able to express performance phenomena as a result of the absence of time. This deficiency was later corrected by the introduction of timed Petri Nets and stochastic Petri Nets (SPN) where

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 17–33, 2015.
DOI: 10.1007/978-3-319-15786-3_2

state transitions were extended by deterministic or stochastic durations. For the processing of such generalized Petri Nets, powerful tools were developed either for the simulation or for an analytical evaluation under Markovian process assumptions [5]. Simultaneously to the developments of scheduling parallel computing as described before, queuing network theory has progressed extensively within the last 5 decades which is expressed by the phenomenon of "product-form" queuing networks and efficient algorithms for their numerical performance evaluation. Queuing networks allow for modeling of parallelism at large but are severely limited with respect to synchronization constraints and generalized stochastic arrival and service processes beyond Markovian assumptions [6]. These deficiencies have partly been overcome by approximate evaluation methods and powerful computer tools for queuing network analysis and simulations, see, e.g., [7–9]. Apart from the state-of-the-art reached in queuing theory and through SPN, main problems remained open as decomposition methods to reduce complexity in the evaluation and how to apply the results practically as, e.g., to detect parallelism in the program execution path (at instruction or task level) or in the data automatically as a basis for scheduling and program execution. More recent developments in microelectronics and in program languages give rise to a re-thinking of parallel processing: Through microelectronics powerful multi-core processors with 16 or 32 cores are integrated on chip-level; multi-processor computer racks provide thousands of processors within a cloud data center. Developments in high-level programming languages allow for parallel program constructs which can be explicitly expressed by the program developer and which support compilation and scheduling by the operating system, in computing and communication.

In this paper, a novel and practical approach to the evaluation of parallel processing will be presented which is based on processing jobs described by reducable task graphs. A task graph models all possible execution paths of a program (computation job) and can be described by a directed acyclic graph (DAG) with generally-distributed task execution times, precedence conditions and synchronization constructs for parallel executable tasks [9, 10]. From the viewpoint of analysis it is important to derive task graphs automatically, to generate task graphs synthetically and to reduce the complexity by graph reduction methods [11–14]. For the analytic performance analysis of this paper it is important that the task graph can be reduced stepwise by elementary aggregations of two tasks at each step. By this approach, it is possible to reduce the whole task graph for a given number n of processing elements to one "virtual" task with a corresponding generally-distributed virtual processing time. Thus, the execution of a specified job stream on a multi-core or multi-processor system can be modeled by a virtual queuing system of type GI/G/1 where GI represents the job arrival stream with arrival rate λ, G represents the virtual task execution time on the multi-processor system, and where $n = 1$ server represents a "virtual processor". Jobs are served by the virtual processor in a batch processing mode, i.e., one at a time only, to avoid context switching overhead and cache splitting in case of simultaneous processing of multiple jobs in a time-sharing mode. Temporally idle processors are turned in a low-power sleeping mode to save energy consumption during enforced "slack times" for concurrent processes or idle periods which can be accomplished by dynamic voltage and frequency scaling (DVFS).

The remaining part of this paper is structured as follows: In Sect. 2, the problem of parallel execution of a job is defined by a task graph which is composed by four generic modeling constructs for serial processing, parallel processing, alternative task and repeated task executions. The description and reduction of the task graph follows in principle the modeling approach reported in [12, 14] and is illustrated by an example graph. Task execution times are described by generally distributed random variables and their probability density functions, respectively. For numerical analyses the generally distributed task execution times are represented by a phase-type model with 3 parameters only which allows the adaption to arbitrary mean values and coefficients of variation. The four generic modeling constructs are described generally through mathematical operations on random variables by distributions as well as by a two-moment characterizations. In Sect. 3, the reduction of the whole task graph by stepwise aggregation of tasks according to the 4 principal modeling constructs is discussed generally and for the example graph of Sect. 2. Section 4 addresses the performance of parallel processing in terms of the speed-up factor achieved by parallel processing and by the job response time by queuing analysis as well as an analysis of the energy consumption. Both performance and energy consumption are compared for two fundamentally different operation modes for multi-processor systems, serial processing of jobs on one processing element each and parallel processing of jobs on all available processing elements. The paper concludes by summarizing the current state of the project and gives an outlook on ongoing further work based on the presented methods.

2 Multi-processor Job Execution

2.1 Multi-processor Queuing Model

In Fig. 1, the considered queuing model is shown consisting of n processing elements representing processors of a multi-core or a multi-processor system. Jobs arrive

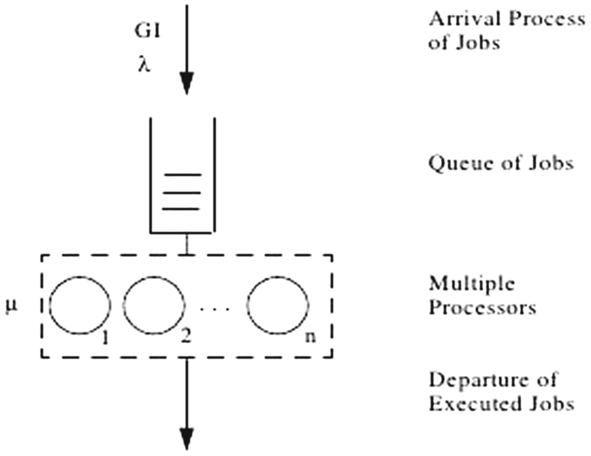


Fig. 1. Principal model of a multi-core/multi-processor processing system for parellel processing

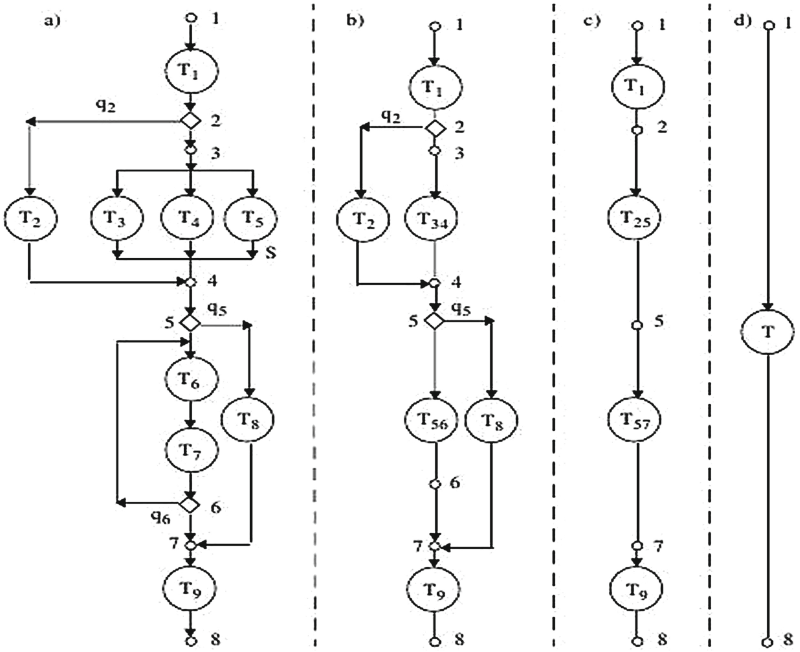


Fig. 2. Example of a task graph and its stepwise redution (a) Original task graph (b, c) Intermediate reduction steps (d) Results of aggregation

according to a general stochastic arrival process of type GI (generally- and independently-distributed arrivals) at an arrival rate of λ jobs/time unit. The model represents, e.g., the physical resources provided by a data center to a tenant (company) or to a group of users for a particular service. The model can be considered as a simplified model for an "Infrastructure as a Service" (IaaS) providing physical resources upon which a workload is processed defined by a Virtual Machine. The workload accompanied with each arrival is defined by a stochastic task graph, see e.g., Fig. 2. Jobs are processed on the multi-processor model according to the batch mode, e.g., by FIFO (First-In, First-Out) scheduling sequence and each job uses the n processors exclusively to avoid program context switching during a job being in execution.

2.2 Task Graph Job Model

Each job is represented by a task graph with stochastic task processing times. A simple model of a task graph is shown in Fig. 2(a) consisting of altogether 9 different tasks 1, 2, …, 9 and 4 generic task constructs. Terminals 1 and 8 represent the initial and final points of a job execution. The task graph belongs to the class of Directed Acyclic Graphs (DAG). Figures 2(b, c, d) represent reduced task graphs of the DAG of Fig. 2(a) and will be discussed later on in Sect. 3. Any execution path between Terminal 1 and Terminal 8 is a feasible production for a job execution.

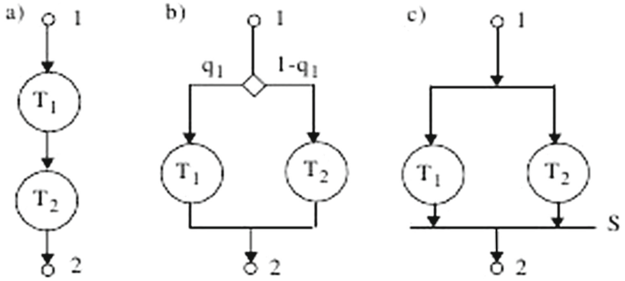


Fig. 3. Principal task graph elements (a) Sequential processing of two tasks 1 and 2 (b) Alternative split (Or-Split) of two tasks 1 and 2 (c) Parallel processing (Concurrency, And-Split) of two tasks 1 and 2 with synchronization S (And-Join) (d) Iteration loop for task 1 with parameter q

Figure 3 represents the four generic or principal constructs (task graph elements).

The execution time T_i of a task i will be represented by its cumulative probability distribution function (DF) $F_i(t) = P\{T_i < t\}$ and its probability density function $f_i(t) = dF_i(t)/dt$, respectively. Individual task execution times T_i of a job are considered as being statistically independent of each other. The aggregated execution times T of the four principal task graph elements of Fig. 3, measured between the Terminals 1 and 2, can be mathematically expressed by their PDF f(t) as follows:

(a) **Sequential processing** of two tasks (concatenation)

$$f(t) = f_1(t) \otimes f_2(t) \tag{1}$$

where the symbol $\otimes$ indicates the mathematical convolution operator

(b) **Alternative Split** of two tasks (Or-Split)

$$f(t) = q_1 f_1(t) + (1 - q_1) f_2(t) \quad \text{(Or-Split followed by Or-Join, Choice)} \tag{2}$$

Remark: The OR-Split is the basic function for tree-structured execution paths.

(c) **Parallel processing** of two tasks with synchronization (Concurrency, And-Split followed y And-Join)

$$T = \max(T_1, T_2)\text{: } f(t) = f_1(t)F_2(t) + f_2(t)F_1(t) \tag{3a}$$

$$T = \min(T_1, T_2)\text{: } f(t) = f_1(t)[1 - F_2(t)] + f_2(t)[1 - F_1(t)] \tag{3b}$$

Remark: The maximum operator is applied if both tasks 1 and 2 have to be completed before continuation. The minimum operator applies, e.g., for a parallel search.

(d) **Iteration loop** for Task 1 (Repetition)

$$f(t) = \sum_{i=0}^{\infty} q^i (1-q) \cdot f_1(t) \otimes \underbrace{[f(t) \otimes \ldots \otimes f_1(t)]}_{ifactors} \tag{4a}$$

for a probabilistic iteration with probability q and

$$f(t) = f_1(t) \otimes \underbrace{[f_1(t) \otimes \ldots \otimes f_1(t)]}_{qfactors} \tag{4b}$$

for q deterministic iterations

2.3 Generic Task Execution Model

The operations are generally too complex to be programmed for arbitrary PDFs. Therefore, a generic task execution model will be used which is defined by 3 parameters only, see Fig. 4.

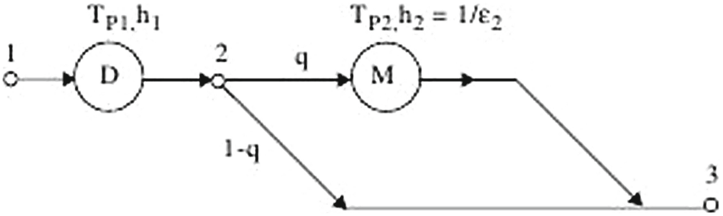


Fig. 4. Mixed phase-type model for task excution times

The substitute model of Fig. 4 consists of a series of a deterministic phase T_{P1} (D) with duration h_1 between Terminals 1 and 2 followed by a degenerated hyperexponential phase between Terminals 2 and 3 realized by a probabilistic alternative between a negative-exponentially distributed phase T_{P2} (M) with mean $h_2 = 1/\varepsilon_2$ chosen with probability q and a zero-phase chosen with probability (1−q). This model will be defined by two parameters for each task execution time T_P, its mean h_P and its coefficient of variation c_P:

$$h_P = E[T_P] \tag{5a}$$

$$c_P{}^2 = VAR[T_P]/E[T_P]^2. \tag{5b}$$

Parameters h_P and c_P can be arbitrarily prescribed, where $0 \leq c_p < \infty$. This model allows to represent any task with arbitrary mean h_P and coefficient of variation c_P by a PDF $f_P(t)$ and a DF $F_P(t)$, respectively:

$$f_P(t) = (1-q)\delta(t-h_1) + q\varepsilon_2 \exp(-\varepsilon_2[t-h_1])u(t-h_1) \tag{6a}$$

$$F_P(t) = (1-q)u(t-h_1) + q[1-\exp(-\varepsilon_2[t-h_1])]u(t-h_1), \tag{6b}$$

where $\delta(t)$ indicates the impulse function (delta function) and u(t) the unit-step function, with

$$h_P = h_1 + qh_2 \tag{6c}$$

$$c_P{}^2 = q(2-q)h_2{}^2/(h_1+qh_2)^2. \tag{6d}$$

The model has 3 parameters h_1, h_2 and q to be derived from Eq. (6c, d), i.e., there is one degree of freedom. For $h_1 \to 0$, we find $c_p \to \infty$ for $q \to 0$. The degree of freedom can be used by choosing $h_1 > 0$ to avoid trivial task execution times. Fixing h_1, parameters h_2 and q follow formally from Eq. (6c, d):

$$q = 2/\left[1 + \left(\frac{c_P h_P}{h_P - h_1}\right)^2\right] \tag{7a}$$

$$h_2 = \frac{1}{2}[h_P - h_1 + c_{P^2} h_{P^2}/(h_P - h_1)] \tag{7b}$$

For a feasible solution, $0 < q < 1$ has to be regarded as compatibility condition. A quite simple solution of the parameter fitting follows from (7a, b) by subdivision of the c_P-range:

(a) $0 \le c_P \le 1$ (**hypoexponential** characteristic)

$$q = 1,\ h_1 = h_P(1-c_P),\ h_2 = c_P h_P \tag{8a}$$

(b) $1 \le c_P < \infty$ (**hyperexponential** characteristic)

$$h_1 = 0,\ h_2 = h_P\left(1 + c_P{}^2\right)/2,\ q = 2/\left(1 + c_P{}^2\right) \tag{8b}$$

The solution (8a) represents a series of a deterministic phase and an exponential phase while (8b) represents a degenerated hyperexponential phase.

2.4 Performance of the Principal Task Graph Elements

Applying the mixed phase-type model for all task execution times, represented by the PDF $f_P(t)$ and DF $F_P(t)$ acc. to Eq. (6a, b), the execution times T for the principal task graph elements of Fig. 3 can be expressed explicitly by their PDF f(t) from Eqs. (1–4). These results are too voluminous and will be reported in detail in a forthcoming companion paper. The results for a two-moment representation are much easier to be derived by elementary moment operators on independent random variables (cases a, b and d) or on PDFs (case c). Tasks 1 and 2 are represented by the phase-type model parameters h_{i1}, h_{i2} and q_i, i = 1, 2. For the 4 principal task graph elements we find:

(a) **Sequential Processing** of two tasks 1 and 2 (Concatenation)

$$E[T] = E[T_1] + E[T_2] = h_{11} + q_1 h_{12} + h_{21} + q_2 h_{22} \tag{9a}$$

$$\begin{aligned} VAR[T] &= VAR[T1] + VAR[T_2] = q_1(2-q_1)h_{12}{}^2 + q_2(2-q_2)h_{22}{}^2 \\ c^2 &= VAR[T]/E[T]^2. \end{aligned} \tag{9b}$$

(b) **Alternative Split** of two tasks 1 and 2 followed by Or-Join (Choice)

$$\begin{aligned} E[T_i] &= q\,E[T_{1i}] + (1-q)\,E[T_{2i}],\ i = 1,2 \\ E[T] &= q(h_{11} + q_1 h_{12}) + (1-q)(h_{21} + q_2 h_{22}) \end{aligned} \tag{10a}$$

$$E[T2] = q\left(h_{11}{}^2 + 2q_1 h_{12}{}^2 + 2q_1 h_{11} h_{21}\right) + (1-q)\left(h_{21}{}^2 + 2q_2 h_{22}{}^2 + 2q_2 h_{21} h_{22}\right) \tag{10b}$$

$$VAR[T] = E\left[T^2\right] - E[T]^2 \tag{10c}$$

$$c^2 = VAR[T]/E[T]^2. \tag{10d}$$

(c) **Parallel Processing** of two tasks 1 and 2 with synchronization (Concurrency, And-Split)

In this case, the PDF f(t) of the aggregated random variable $T = \max(T_1, T_2)$ or $T = \min(T_1, T_2)$ acc. to Eqs. (3a, b) has to be derived first from which the moments

$$E[T^i] = \int_{t=0}^{\infty} t^i f(t)dt_{,\,i=1,2} \tag{11}$$

follow by integration. The variance VAR[T] and the coefficient of variation *c* follow acc. to Eqs. (10c, d). The explicit results are too voluminous and will be reported in a forthcoming paper.

(d) **Iteration Loop** for task 1 (Repetition)

Be J the RV of the number of executions of task 1 with average E[J]. Then we get for the aggregated RV T in general

$$\mathrm{E[T]} = \mathrm{E[J]} \cdot \mathrm{E[T_1]}\,. \tag{12a}$$

In case of a **geometrically-distributed** number of iterations with feedback probability q we get

$$E[J] = \sum_{j=0}^{\infty} (j+1)q^j(1-q) = 1/(1-q) \tag{12b}$$

$$\mathrm{E}\left[\mathrm{J}^2\right] = \sum_{\mathrm{j}=0}^{\infty} (\mathrm{j}+1)^2\mathrm{q}^\mathrm{j}(1-\mathrm{q}) = \frac{1+\mathrm{q}}{(1-\mathrm{q})^2} \tag{12c}$$

$$\mathrm{VAR[J]} = \mathrm{E}\left[\mathrm{J}^2\right] - \mathrm{E[J]}^2 \tag{12d}$$

$$\mathrm{VAR[T]} = \mathrm{E[J]} \cdot \mathrm{VAR[T_1]} + \mathrm{VAR[J]} \cdot \mathrm{E[T_1]}^2 \tag{12e}$$

$$\mathrm{c}^2 = \mathrm{VAR[T]/E[T]}^2. \tag{12f}$$

If J is a **constant** E[J] = J and VAR[J] = 0, then

$$\mathrm{VAR[T]} = \mathrm{J} \cdot \mathrm{VAR[T_1]}\,. \tag{12g}$$

E[T] and c follow from Eqs. (12a) and (f).

Remark: If J is an RV, the statistics of T follow from the compound distribution of T_1 and J.

3 Task Graph Reductions

3.1 General Aspects and Application Cases

As outlined above, the workload by a specific job will be described by a directed acyclic graph (DAG) consisting of elementary structural elements expressing arbitrary workflows. Any DAG can be processed on a **single-processor system** by following any possible execution path through the DAG from the initial terminal to the final terminal. Any feasible execution path through the DAG can be considered as a thread which is scheduled by the operating system and processed on the single-processor system without any stop (except for memory I/0 which is not considered here). Processing of parallel executable instruction paths have to be executed serially in arbitrary sequence but continuation after the parallel paths depends on the synchronization condition. Parallel processing executed on a single processor does not cause any “slack times”.

In a **multi-processor environment**, parallel executable paths can be scheduled such that the thread is split into multiple parallel threads which can be scheduled by the operating system and processed simultaneously as long as the synchronization point is not reached. The degree of processing simultaneity depends on the individual execution path durations: The best case is if all parallel threads are of identical durations; however, with increasing variability, the degree of processing simultaneity decreases and some processing elements will become idle for a "slack time" until the next synchronization point. This reduces the performance by increasing the total job execution time. The idea of this paper is to quantify the efficiency of parallel processing by modeling task execution times by stochastic processes in order to express the influence on the performance as well as on the energy efficiency.

This can be achieved best if the DAG could be reduced stepwise by aggregating parallel or serially executable tasks to a single virtual task where the corresponding execution times of aggregated paths are obtained by application of the basic aggregation operations introduced in Sect. 2. Reducibility of graphs is a fundamental problem of graph theory. Graphs resulting out of the use of "go to" statements, e.g., jumps out of a loop or into another program branch, cause quite complex graph structures which cannot be reduced stepwise. Modern programming languages and programming styles result into well-structured programs which support graph reducibility.

From the application point of view, many problems are adequate for parallel execution and reducibility. Examples are:

(1) Parallel execution of a program for multiple input parameter sets. Each parameter set can be executed in parallel by a multi-processor system providing results of a whole parameter range instantaneously.

(2) Batch simulation methods where a simulation is subdivided in (typically) 10 "batches", each for a certain number of "events", e.g., 100.000 events. For such programs, we find a simple program structure of one task at the beginning to configure the "batch" programs and initializing counters for statistic data, then executing all "batch" simulations in parallel, and one task after the execution of all "batches" for processing of the final results out of each "batch" execution. Such a program consists of a simple DAG-structure of the form fork and join and allows a speed-up factor close to the number of "batches".

(3) Searching within large unstructured data sets, a typical problem of "Big Data". In such cases, the data sets can be partitioned and each partition can be executed in parallel. This approach can be repeatedly applied on reduced data sets, etc.

In most of such applications, execution times of parallel threads may depend on the properties of data or may affect the number of iterations for a certain precision; these execution time variations are modeled best by random variables.

3.2 Example of a Task Graph Reduction

A simple example of a model task graph as shown in Fig. 2(a) will be considered. In a first step, m serially or parallel executable tasks are combined successively by aggregation of two tasks in each step either in a linear sequence by (m−1) iterations or in a binary-tree fashion by aggregating each time 2 tasks resulting in log m iterations; in the latter case, the aggregations themselves can be executed in parallel, too. By these steps, the task graph Fig. 2(a) results in a reduced task graph shown in Fig. 2(b). In a second step, all loops are aggregated and finally replaced by one task resulting in the further reduced task graph shown in Fig. 2(c). In the final step, only a series of sequential tasks has to be aggregated in a single resulting “virtual task” for the whole job, see Fig. 2(d).

In a tree-structured task graph, the above outlined reduction strategy is applied at first on all branches of the tree or of subtrees, followed by combining the tasks representing the whole branch at the root point of subtrees, repeatedly in bottom-up direction, starting at the leaf-level.

Remark 1: All described steps are performed on two parameters of each task (the mean value and the coefficient of variation) as outlined in Sect. 2 of this paper. For this, we need only to program the basic operations which are implemented in a procedure and are applied repeatedly.

Remark 2: As outlined above, the virtual tasks include automatically the parallel execution on multiple processors. If the degree of task parallelism exceeds the number of available processors, the task graph has to be restructured first by combining maximally possible parallel aggregations and repeat these results sequentially for the remaining parallel tasks which (of course) adds to an increased task graph execution time.

4 Performance Evaluation and Energy Efficiency

Task graph processing on a multi-processor system will be considered under two different aspects, performance and energy efficiency. For comparison, we will distinguish between two modes in each case:

Mode PP: Parallel processing of each job on the n-processor system.
Mode SP: Serial processing of each job on one processor of the n-processor system.

4.1 Performance Evaluation

The classical performance criterion for parallel computing was formulated by (13a) (Amdahl’s Law [15]): If α is the fraction of non-parallelizable parts of a program, the ideal speed-up factor is

$$\mathrm{S(n)} = 1/[(1-\alpha)/\mathrm{n} + \alpha]\,. \tag{13a}$$

As a consequence of the introduced job description by a DAG, the speed-up factor has to be re-defined as the fraction of job processing times for n = 1 (single processor) and n, i.e.,

$$\mathrm{S(n)} = \frac{\mathrm{E[T|n=1]}}{\mathrm{E[T|n]}} \tag{13b}$$

Note, that this approach is more general as individual task execution time variations and limitations in parallelization are taken into consideration, where (13a) holds for an idealized case of constant parallelization degree of n only. Under the special case of a constant parallelization degree n we get for α the result $\alpha = (nE[T|n] - E[T|1])/(n-1)$; inserting this in Eq. (13b), the result coincides with Amdahl's Law Eq. (13a).

As a second performance metric, we will consider the response time T_R measured between the arrival instant of a job and the instant when the job is executed, i.e.,

$$T_R = T_W + T_v \tag{14a}$$

where T_W is the waiting time and T_v the virtual processing time of the job.

Mode PP: Parallel Processing

The parallel processing system is represented by a virtual single-server system upon completion of the graph reduction method outlined in Sect. 3.2. T_W follows from a GI/G/1 queuing model, e.g., for Poisson job arrivals according to the Pollaczek-Khintchine formula for the M/G/1 delay system [6]:

$$E[T_W] = \rho_V \cdot \frac{(1 + c_V^2)}{2(1-\rho_V)} E[T|n], \text{with load factor } \rho_V = \lambda \cdot E[T|n]. \tag{14b}$$

Note, that the maximum capacity of this system is reached for

$$\lambda_{\max,PP} = \frac{1}{E[T|n]} \tag{15a}$$

The average processing time of a job under Mode PP is E[T|n]. The average of the slack time T_S follows from the balance equation $E[T_S] = n \cdot E[T|n] - E[T|1]$ and reduces the capacity for high loads to $\lambda_{\max,PP}$ acc. to (15a).

Mode SP: Serial Processing

If each job is assigned to be processed by one processor, the n-processor system is modeled by a GI/G/n system, where G describes the job processing time T on a single processor which follows from the original task graph by adding all processing phases resulting in a mean processing time E[T|1]. The response time T_R follows from (14a), where T_V is represented by two moments h = E[T|1], c from a task graph reduction for 1 processor, and T_W from queuing system GI/G/n; for Poisson arrivals from the delay system M/G/n (exact closed-form solutions and tabled results are known for M/M/n and M/D/n only).

The maximum capacity of this n-server system is reached for

$$\lambda_{\max,SP} = \frac{n}{E[T|1]} = \frac{n}{nE[T|n] - E[T_S]} > \lambda_{\max,PP} \tag{15b}$$

Note, that in Mode SP the full capacity of the n servers is available, where PP suffers from enforced idle times (slack times).

4.2 Energy Efficiency

As outlined in Sect. 2, we will assume for both Modes PP and SP that all jobs are processed on an n-processor machine in batch processing mode. Under low load, parallel processing results in shorter job execution times and is superior to job processing serially on a single processor. Idle phases of a processing element due to a limited parallelization degree $(1 - \alpha)$ will be considered as sleep phases ("slack times") with reduced power P_0; $P_1 > P_0$ denotes the power consumption of a running processor executing a task. Parallel processing of tasks and serial processing of tasks are neutral with respect to energy consumption as both modes require the same amount of energy in total, whereas parallel processing and serial processing differ with respect to the maximum job rate for saturation as well as with respect to the response time, i.e., there is a trade-off between parallel and serial processing. This effect has been observed in other energy-efficiency studies as well, where energy efficiency and performance reduction behave reciprocally [16, 17]. The energy consumptioncan, however, can be reduced by low-power operation of idle resources, e.g., by Dynamic Voltage and Frequency Scaling (DVFS) [18].

The energy consumption in an arbitrary large interval t_0 of time amounts for Modes PP and SP as follows:

$$E_{PP} = t_0 \cdot \rho_V \left[n \frac{E[T|1]}{nE[T|n]} \cdot P_1 + n \frac{E[T_S]}{nE[T|n]} \cdot P_0 \right] + t_0(1 - \rho_V) \cdot nP_0 \tag{16a}$$

$$E_{SP} = t_0[AP_1 + (n - A)P_0], \tag{16b}$$

where $\rho_V = \lambda \cdot E[T|n]$ denotes the utilization factor of the virtual GI/G/1 delay system and $A = E[X] = \lambda \cdot E[T|1]$ the offered (and carried) traffic value of the GI/G/n delay system, respectively. With these relationships both expressions for E_{PP} and E_{SP} of Eq. (16a, b) are identical $E_{SP} = E_{PP} = E$.

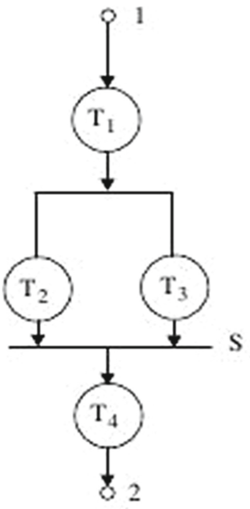
The energy efficiency η will be defined as the fraction of energy saved by DVFS relative to the energy consumption without DVFS:

$$\eta = 1 - \frac{E}{E_0} = 1 - \frac{A}{n} - \frac{n - A}{n} \cdot \frac{P_0}{P_1},$$

where $E_0 = nP_1t_0$ is the energy consumption without DVFS.

4.3 Trade-off Between the Operation Modes PP and SP

The observation that both operation modes differ in their maximum capacities gives rise for a trade-off discussion between them. This will be exemplified in the following by a numerical example for the generic task graph example for concurrency between two tasks 1 and 2 acc. to the model of Fig. 3c extended by constant common tasks at the beginning and end of the task graph with total length h_0. Two special cases of concurrent task execution times will be considered with identical average task execution times D (deterministic) and negative-exponentially distributed times (M) with parameters



Parameters

T_1, T_4	Type D, $E[T_1] = E[T_4] = h/2$
T_2, T_3	Type D, (Case 1), Type M (Case 2) $E[T_2] = E[T_3] = h = 1/\varepsilon$
h	Unit of Task Execution Times
PP	Parallel Processing of Job Tasks on both Processors (Mode 1)
SP	Serial Processing of Job Tasks on one Processor (Mode 2)
n	Number of Processors $n = 2$ (P_1, P_2)

Fig. 5a. Generic taskgraph for numeric example

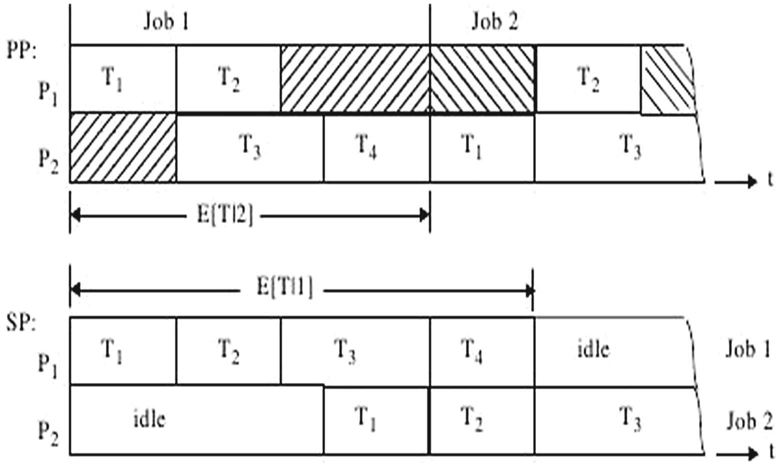


Fig. 5b. Grant-charts for job PP and SP models handed fields are slack times in PP

$h_1 = h_2 = h = 1/\varepsilon$ and c = 0 (deterministic) and c = 1 (Markovian), respectively. Jobs arrive in both cases according to a Poisson process with arrival rate λ. Figures 5a and 5b shows the example task graph and two Gantt-Charts for PP and SP schedules.

The performance analysis of Mode 1 (PP) follows the procedure of stepwise task graph reduction resulting in a virtual queuing system of the type GI/G/1. The analysis of Mode 2 (SP) follows from a standard queuing system of the type GI/G/n.

Table 1. Queuing system parameters

Mode	Case	Av. Serv. Time	SCOV	Utilization	λ_{max}	Qu.System
1 (PP)	1	$h_v = 2$ h	$c_v^2 = 0$	$\rho_v = 2\lambda h$	1/2 h	M/D/1
	2	$h_v = 2.5$ h	$c_v^2 = 0.2$	$\rho_v = 2.5\lambda h$	1/2.5 h	M/G/1
2 (SP)	1	$E[T\|1] = 3$ h	$c^2 = 0$	$\rho = 3\lambda h$	1/1.5 h	M/D/2
	2	$E[T\|1] = 3$ h	$c^2 = 2/9$	$\rho = 3\lambda h$	1/1.5 h	M/G/2

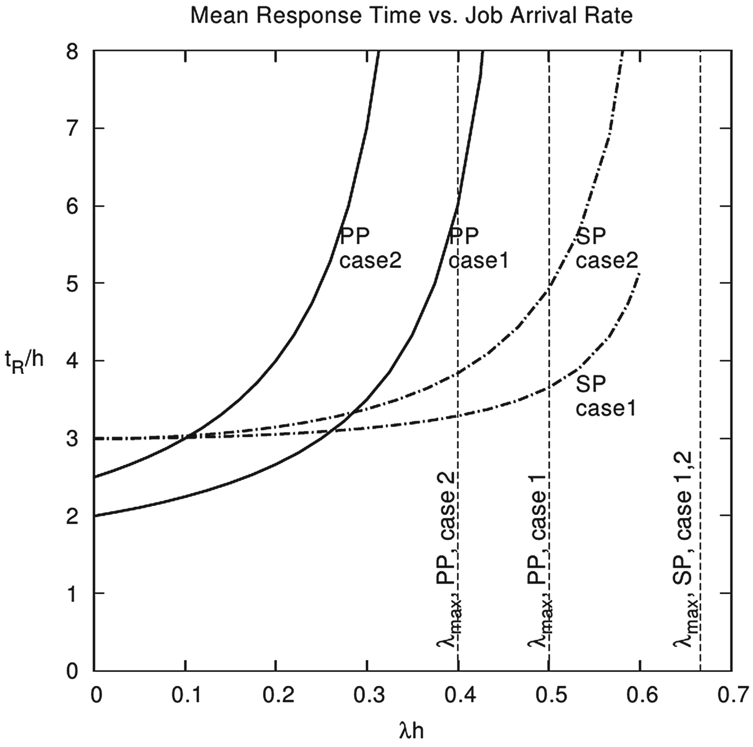


Fig. 6. Mean Response Time vs. Job Arrival Rate

The parameters of the corresponding queuing models are summarized in Table 1. Data center job arrivals are assumed to follow a Poisson distribution (Type M).

The results for the normalized average response times t_R/h are shown for both scheduling Modes 1 (PP) and 2 (SP) for the two parameter cases of constant task execution times (Case 1) and negative-exponentially distributed task execution times (Case 2) of Tasks 2 and 3, respectively (Fig. 6).

Note first the different maximum load levels for PP and SP and the different maximum load levels for PP for constant and for exponentially distributed virtual task times acc. to Case 1 and Case 2 which define the load limits of stationary system operation where the response times increase to infinity asymptotically.

The results underline the following general properties

- The maximum capacities for PP are lower than for SP.
- The maximum capacity for PP reduces with increasing slack times caused by task execution time variations.
- Trade-off of the performance results between PP and SP with smaller response times for PP in the low-load region and for SP in the high-load region.

5 Conclusions

The main contribution of this paper is a novel method by which parallel and serial processing of jobs on a multi-core/multi-processor system can be analyzed for generally-distributed task execution times by stepwise reduction of directed acyclic task graphs. The reductions are performed by task aggregations for four principal structure elements of computation programs: concatenation, alternative splitting, iterative repetitions, and concurrency of tasks. The mathematical operations are based on generally-distributed random task execution times. The principal four structure elements are used for the exact aggregation based on the theory of functions of random variables. For an efficient computational implementation, the generally-distributed task execution times are represented by a mixed phase-type model for the first and second order moments. The method allows to represent the multi-core/multi-processor system to standard queuing models of the types GI/G/1 and GI/G/n, where the service times are represented by their averages and coefficients of variation. From the application's point of view, the new method allows the extension of Amdahl's Law to more realistic conditions of random task execution times and arbitrary degrees of parallelization as well as real-time performance metrics as the average job response times. The trade-off between the two major schedules for parallel and serial processing of jobs on an n-server system leads to the most important conclusion, that parallel processing is only superior for low- and medium-load ranges, while serial processing outperforms parallel processing for high loads with respect to the maximum capacity and response times. Both job execution modes are neutral with respect to energy efficiency; the only way to increase energy efficiency is by low-power operation of idle processors through Dynamic Voltage and Frequency Scaling (DVFS). The current paper reflects the status of "work in progress"; ongoing work addresses the development of general analysis tools for the analytical solution as well as for simulations.

References

1. Conway, R.W., Maxwell, W.L., Miller, L.W.: Theory of Scheduling. Addison-Wesley Publ. Comp., Reading (1967)
2. Coffman Jr., E.G., Denning, P.J.: Operating Systems Theory. Prentice-Hall Inc., Englewood Cliffs (1973)
3. Shirazi, B.A., Hurson, A.R., Kavi, K.M. (eds.): Scheduling and Load Balancing in Parallel and Distributed Systems. IEEE Computer Society Press, Los Angeles (1995)

4. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall, Englewood Cliffs (1981)
5. Ajmone Marsan, M.: Stochastic Petri nets: An elementary introduction. In: Rozenberg, Grzegorz (ed.) APN 1989. LNCS, vol. 424, pp. 1–29. Springer, Heidelberg (1990)
6. Kobayashi, H., Mark, B.L.: System Modeling and Analysis: Foundations of System Performance Evaluation. Pearson/Prentice-Hall Inc. (2009)
7. Reiser, M., Lavenberg, S.S.: Mean-value analysis of closed multichain queuing networks. J. ACM **27**(2), 313–322 (1980)
8. Kuehn, P.J.: Approximate analysis of general queuing networks by decomposition. IEEE Trans. Commun. **27**(1), 113–126 (1979)
9. Whitt, W.: The queuing network analyzer. Bell Syst. Techn. J. **62**(9), 2779–2815 (1983)
10. Adve, V., Sakellariou, R.: Compiler synthesis of task graphs for parallel program performance prediction. In: Proceedings of 13th International Workshop on Languages and Compilers for High-Performance Computing (LCPC 2000), Yorktown Heights, N.J (2000)
11. Adve, V.S., Vernon, M.K.: Parallel program performance prediction using deterministic task graph analysis. ACM Trans. Comput. Syst. (TOCS) **22**(1), 94–136 (2004)
12. Ajwani, D., Ali, S., Morrison, J.P.: Application agnostic generation of synthetic task graphs for streaming computing applications. IBM Research Report RC 25181 (D 1107-003), 5 July 2011
13. Sadiq, W., Orlowska, M.E.: Applying graph reduction techniques for identifying structural conflicts in process models. In: Jarke, M., Oberweis, A. (eds.) CAiSE 1999. LNCS, vol. 1626, pp. 195–209. Springer, Heidelberg (1999)
14. Simon, J., Wierum, J.-M.: Accurate performance prediction for massively parallel systems and its applications. In: Fraigniaud, Pierre, Mignotte, A., Robert, Y., Bougé, Luc (eds.) Euro-Par 1996. LNCS, vol. 1124, pp. 675–688. Springer, Heidelberg (1996)
15. Sahner, R.A., Trivedi, K.S.: Performance and reliability analysis using directed acyclic graphs. IEEE Trans. Softw. Eng. **SE-13**(10), 1105–1114 (1987)
16. Sun, X.-H., Chen, Y., Byna, S.: Scalable computing in the multicore Era. In: Proceedings of International Symposium on Parallel Algorithms, Architectures and Programming (PAAP 2008) (2008)
17. Kuehn, P.J., Mashaly, M.: Performance of self-adapting power-saving algorithms for ICT systems. In: Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2013), Ghent, Belgium, 27–28 May 2013 (IEEE XPlore)
18. Mashaly, M., Kuehn, P.J.: Modeling and analysis of virtualized multi-service cloud data centers with automatic server consolidation and prescribed service level agreements. In: International Conference on Computer Theory and Applications (ICCTA 2013), Alexandria, Egypt, 29–31 October 2013
19. Wang, L., von Laszewski, G., Dayal, J., Wang, F.: Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In: Proceedings of 10th IEEE/ATM International Conference on Cluster, Cloud and Grid Computing (CCGrid 2010), pp. 368–377, 17–20 May 2010

Cyclic Blackout Mitigation Through HVAC Shifted Queue Optimization

Kasim Al-Salim(✉), Ivan Andonovic, and Craig Michie

Electronic and Electrical Engineering Department, University of Strathclyde,
Glasgow G1 1XQ, UK
{kasim.al-salim, i.andonovic, c.michie}@strath.ac.uk

Abstract. The increasing global demand for power has resulted in frequent blackouts in many geographies. The cost of domestic standby generation is prohibitive and novel strategies to provision measures that manage blackouts are becoming much sought after. In some scenarios certain amounts of surplus power can be identified, with the mix of available generation not being fully utilized. The paper presents a strategy that harnesses the aggregated superfluous power to fulfil essential demand in residential areas during cyclic blackouts. The solution has at its foundation, a multi-agent distributed demand management system with a supply-demand matching capability. Power is not distributed fairly to each user, and appliances which consume the most significant levels of power such as air conditioners are serviced according to the available superfluous power. The approach is evaluated through an extensive emulation framework and results show that the proposed system is capable of providing an acceptable Quality-of-Service (QoS) level during cyclic blackout periods and at the same time succeeds in smoothing demand profiles.

Keywords: Blackouts · Multi-Agent systems · Demand side management · Supply demand matching · Power management

1 Introduction

Modern technological breakthroughs have resulted in the extensive use of a spectrum of electrical devices which in turn have created an escalating demand for power. In addition, the repetitive nature of routine daily life features bursts of activities forming demand peaks which in many cases result in exceeding the available power and blackouts. Many countries around across the globe have witnessed large scale blackouts, the most striking being in India in 2012 which affected 670 million users [1]. Building more power generation capability is becoming prohibitively costly with a negative impact on the environment. One alternative is to redistribute demand evenly throughout the day, in so doing eliminating the need for peak load generators and satisfying the demand with base load only. This motivation has stimulated a plethora of load management techniques, measures and strategies, Demand Side Management (DSM) being the most noteworthy. Demand is modulated using various principles such as peak clipping, valley lifting, load shifting, demand conservation and build-up. Customers are essential participants in all solutions; all successful deployments rely on end user approval and collaboration.

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 34–51, 2015.
DOI: 10.1007/978-3-319-15786-3_3

Blackouts can be precipitated by a range of events, the most obvious being escalating demand, referred to as 'peak time' blackouts. The severity of the power shortage specifies how deep or long a blackout is. If the power shortage continues for extended periods of time, the blackout migrates into a cyclic mode in which power is provided following an ON/OFF ratio. Finally, if the power system reaches a steady state at which the generation always lags demand for an extended period of time (years). The cyclic blackout evolves to the most severe chronic mode. This mode is as a result of a wide scale degradation of the installed power generation, transmission and/or distribution sub-systems owing to post-war or post-disaster periods; Iraq is a recent example.

A significant body of research has focused on developing effective ways to manage blackouts e.g. [2] used intentional islanding to prevent a blackout DSM and its derivatives have been proposed extensively such as; controlling the set point of thermostatic loads [3, 4, 5]; optimizing different demand parameters [6]; converting power meters into more intelligent modes of operation through interconnected controller boards [7]; dual-measure DSM (2DSM) schemes utilizing Direct Load Control (DLC) scheduling and Dynamic Power Allocation (DPA) [8]; purchasing day ahead power and applying ON/OFF control to certain appliances [11. 9] segregates household appliances from air conditioners (ACs), powering the former continuously and distributing the remaining power according to AC thermostat setting governed by the customer. Different power distribution strategies have been evaluated: fair and temperature-based allocation.

Multi-Agent Systems (MAS) have recently emerged as a promising technique due to their flexibility and ability to manage distributed scenarios and their suitability for distributed control and management. [10] proposes a multi-agent, intelligent DSM system operating on rationed utility whilst [11] proposes a multi-agent system based on a supply-demand matching (SDM) based on the power market within a renewables-dominated generation mix.

Energy efficiency principles have also been adopted; Evaporative Air Coolers (EACs) are a low cost, energy efficient HVAC alternative to ACs, consuming lower power at effective cooling. EACs are used widely in many countries such as USA, China and Iraq and invariably result in a significant lowering in the power demand. EACs provide considerable energy savings (3 kW/air cooler) [12], an adequate level of cooling and are environmentally friendly [13]. Yamada et al. [14] evaluated the cooling effect of water sprayers and studied the design process of mist sprayers and the effects of water droplets diameters on the cooling process. [15] studied mist sprayer performance in Japanese humid weather through controlling droplet diameter.

The wide spectrum of available research with all its techniques and strategies can be classified into two approaches; 'blackout prevention' and 'blackout containment'. The first can be defined as a set of DSM measures undertaken to prevent a blackout from occurring based on shedding some load either directly or indirectly in order to sustain the remaining load negotiating solely with the utility. The second can be defined as a set of measures undertaken to contain a blackout from impacting the entire grid through (say) islanding; blackout containment also negotiates with the utility only. Alternatively, blackout mitigation can be defined as a set of DSM measures undertaken during a blackout to enable users to retain a certain Quality-of-Service (QoS) through utilizing part of or all available standby or private generation sources. Blackout mitigation neither changes nor uses direct utility power and does not correct the blackout.

The acquisition of a standby generator is an obvious option but operational challenges such as noise, polluting emissions, fire hazard of stored fuel, frequent costly maintenance, and oil disposal problem are factors which prohibit deployments. Renewable options are more attractive in this respect but are tied tightly to weather and although they are environment-friendly they still remain costly especially if they are used to operate air conditioning systems.

The contribution of the paper is to provide a solution to chronic cyclic blackouts using variable superfluous standby power inherent in existing scattered generation capabilities enabling customers to operate appliances including HVAC in environments where air conditioning is a must. The approach presented here relies on the polling of all neighborhood standby generation facilities - regardless of ownership - to map the level of superfluous power. This surplus is then used to power the ‘basic set’ of appliances such as lights, refrigerators and targets the need to meet the ‘air conditioning’ demand. The basic appliance set does not include high power consumption appliances such as electric cookers nor washing machines whilst confining the provision to one HVAC appliance (either an (AC), an air cooler (CO), or a mist fan (MF)) creating an acceptable living environment. The type of HVAC appliance used depends on the level of available surplus power. Thus the principle is that each dwelling is allocated a level of AC, CO, and MF operational time following a fair distribution referred to as the ‘usage right’ i.e. the right to use a certain HVAC appliance for a specified duration of time. The optimum solution is when there is sufficient power to allocate an AC to every family throughout the day. If this criterion cannot be satisfied, the solution finds the optimum blend of ACs, COs, and MFs that matches available surplus. The system ensures that during the 24 h of the day each family has at least one HVAC appliance operational.

The organization of the paper is as follows: Sect. 2 introduces the approach and its operation. Section 3 describes the implementation, the integrated development environment and the detail of a representative case study. Section 4 provides an evaluation of the performance of the approach together with a discussion of results for three mitigation mechanisms. Section 5 contains conclusions and suggestions for future work.

2 The Proposed System

2.1 System Structure

The proposed system hardware comprises local controllers deployed in each dwelling of a residential area. Each controller is equipped with temperature and humidity sensing capabilities. Controllers are connected to a main controller located at the local substation with connections to all neighbouring generation facilities. Figure 1 shows a typical layout of the system. A number of agents are defined to carry out key negotiation tasks and manage the re-distribution of power resources. Each dwelling is assigned a ‘House Agent’ (HA) responsible for the management of all appliances, including ON/OFF and proportional control of each appliance, sensing ambient temperature and humidity, and performing data exchange operations with other HAs and with the ‘Administrative Agent’ (AA). Houses are clustered and cluster heads are assigned. The mix of generation facilities is managed through ‘Generation Agents’ (GAs), clustered with a cluster head. GAs map the amount of power available from a certain generator and at what time. A ‘Utility Agent’ (UA) is tasked with providing utility power delivery schedules. All agent activities are orchestrated by the AA (Fig. 1).

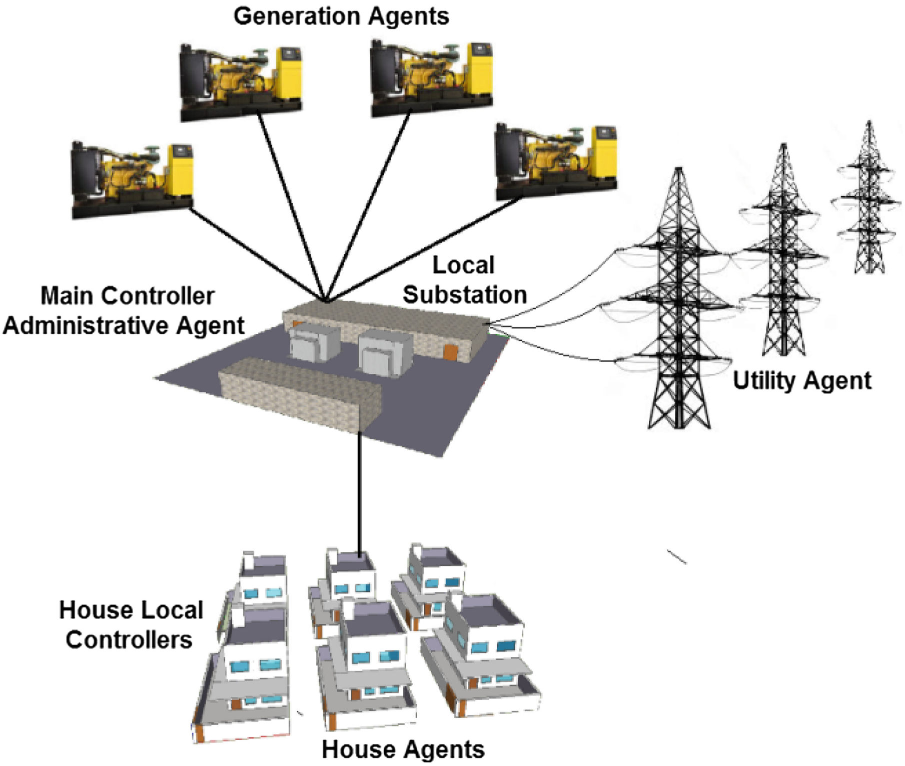
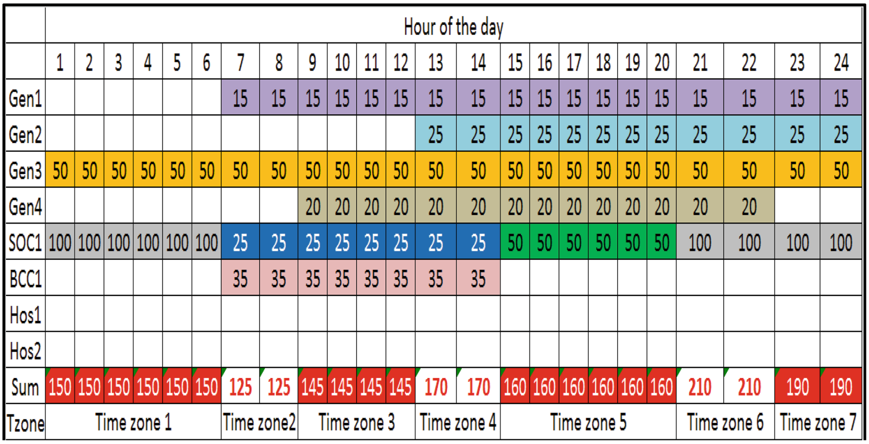


Fig. 1. Typical system layout.

2.2 System Operation

The system provisions power to basic household appliances and one HVAC appliance whose type depends on available power. All appliances are connected to the Low-Voltage distribution network. The system HVAC appliance selection pool contains three types viz. AC, CO, MF, each differing in power consumption. At the outset the AA interrogates the generation cluster head to determine the available surplus power within its cluster. The generation cluster head reports the amounts, availability times, and periods of any surplus power stream. The AA aggregates all streams into one daily schedule of surplus power availability, (Table 1).

Table 1. Surplus power aggregation matrix.



	Hour of the day																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Gen1							15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
Gen2													25	25	25	25	25	25	25	25	25	25	25	25
Gen3	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50
Gen4									20	20	20	20	20	20	20	20	20	20	20	20	20	20		
SOC1	100	100	100	100	100	100	25	25	25	25	25	25	25	25	50	50	50	50	50	50	100	100	100	100
BCC1							35	35	35	35	35	35	35	35										
Hos1																								
Hos2																								
Sum	150	150	150	150	150	150	125	125	145	145	145	145	170	170	160	160	160	160	160	160	210	210	190	190
Tzone	Time zone 1						Time zone2		Time zone 3				Time zone 4		Time zone 5						Time zone 6		Time zone 7	

In this case eight generators are listed; six have surplus power at different times and durations whilst the other two - which are hospital generators - have no surplus power to offer. The latter, for security of supply, are powered continuously and their standby generation is used as a backup for other generators in case no other alternative is available. The resultant surplus power is indicated in the 'Sum' row showing seven different levels of surplus power distributed over seven time zones. This is not a fixed distribution but changes as the generation facilities change their donation patterns.

During the utility OFF periods, the AA interrogates the power aggregation table securing a snapshot of the amount of available surplus power and its duration, calculates the basic household demand, calculates the amount of remaining power to activate HVAC devices, and allocates the best mix of HVAC devices to match the needs of each dwelling using an optimization technique developed for this purpose referred to as Shifted-Queue Optimization (Sect. 2.3). The result is the provision of some level of cooling in an optimum number of houses. For example, for 20 houses if the Shifted Queue Optimization yields (AC = 5, CO = 15, MFs = 0) the system then establishes a single HVAC cluster containing the selected HVAC appliances (5 ACs and 15 COs). In this case 5 dwellings have the usage right to operate ACs, 15 houses COs, and there is thus no need for the use of MFs. The number and type of HVAC appliances in the cluster is governed by the amount of available surplus power; thus a cluster is subject to dynamic changes in composition depending on the amount of donated surplus power. In essence the system also executes supply-demand matching. Figure 2 shows the HVAC composition for different snooped power level.

Each dwelling is then able to exchange HVAC usage right until the end of the interval at which the surplus power is fixed.

Table 2 shows an HVAC cluster containing 20 HVAC devices for 20 houses powered by a certain surplus power level lasting 4 h. Each house enjoys one hour of air conditioning and three hours of air cooling. If the number of ACs, COs, and MFs is not dividable as in the previous example, the system readjusts the number and mix of ACs, COs, and MFs so that every dwelling is allocated a fair share of cooling.

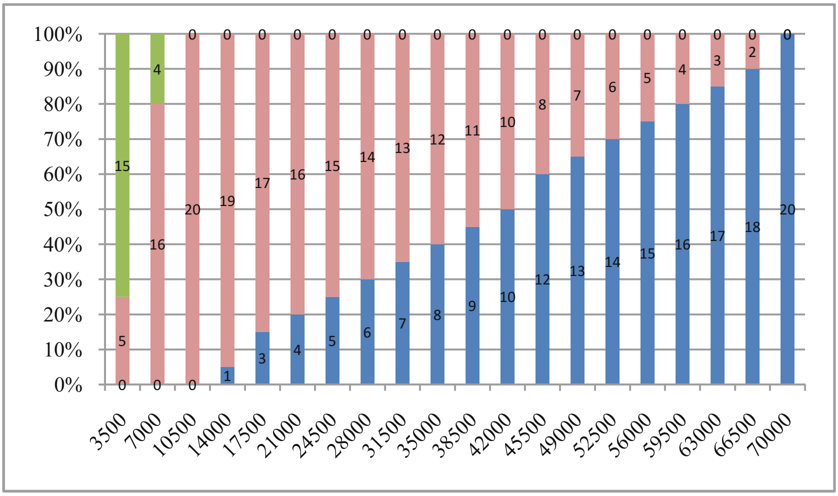
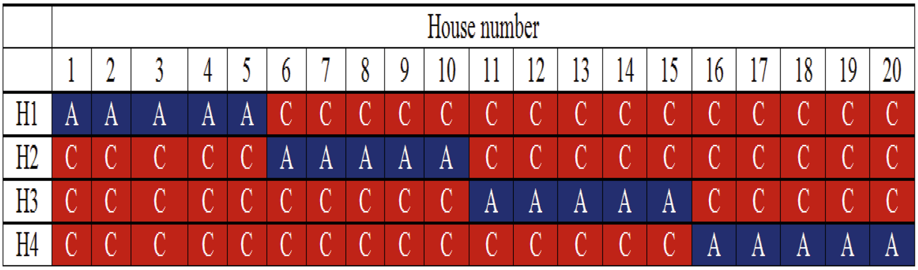


Fig. 2. HVAC cluster composition change according to the available surplus power.

Table 2. HVAC usage right distribution.



	House number																			
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
H1	A	A	A	A	A	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C
H2	C	C	C	C	C	A	A	A	A	A	C	C	C	C	C	C	C	C	C	C
H3	C	C	C	C	C	C	C	C	C	C	A	A	A	A	A	C	C	C	C	C
H4	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	A	A	A	A	A

A = air conditioner, C = air cooler, HVAC = hour

The proposed system is semi-central in the sense that some activities are central under the supervision, control, and coordination of the AA while the rest are distributed, carried out through command-based packetized inter-agent negotiation. All communication between agents is carried out through packet transfer; Fig. 3 shows the packet structure.

Figure 4 shows the functional block diagram of agents recruited in the system and their interactions; solid lines indicate intra-agent data transfer and the dotted lines indicate inter-agent data transfer.

HAs capture all data related to all appliances and have the ability to control them. Once a shortfall in allocated power needed to operate base load and HVAC appliance is identified, then the HAs manage it through inter-agent negotiation. All HA operations are orchestrated by the AA which gathers operational data from all other agents and sensors. In addition to this responsibility, aggregating power from different sources and powering appliance the AA constructs the HVAC cluster and maintains its composition

Source agent	Destination agent	Main command	Data payload	Sub command

Fig. 3. Inter-agent communication packet structure.

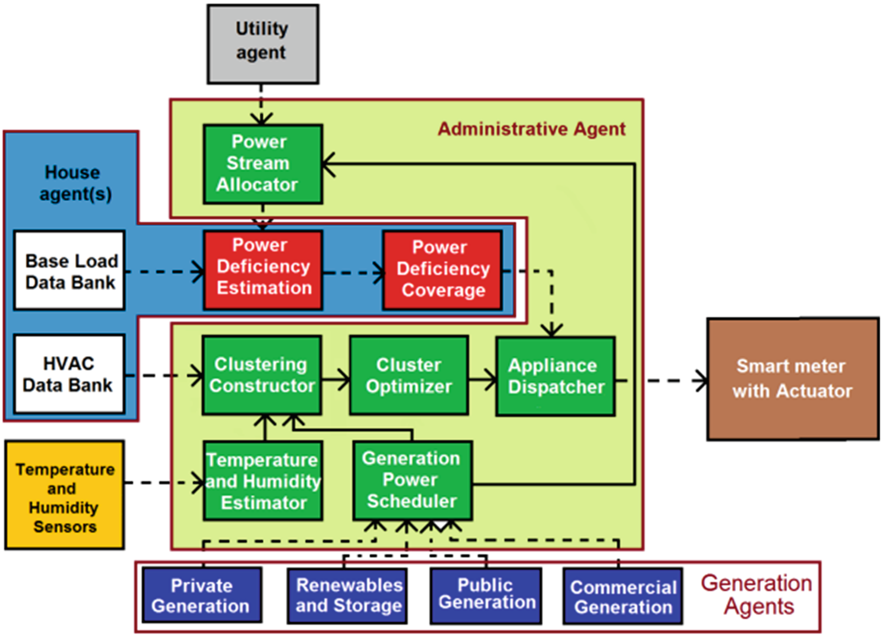
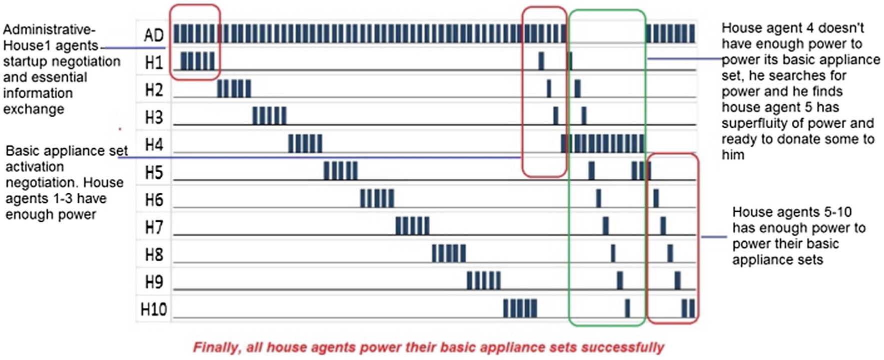


Fig. 4. Agents' functional block diagram.

in order to allocate HVAC appliance usage rights to dwellings through the appliance dispatcher. Generation and utility agents have much simpler roles confined to supplying operational data since they deal with entities owned and controlled by other parties. The system requires smart meters to have the ability to receive the permissible amount of power allocated to each house and cut the power if the dwelling exceeds its power allowance. This is marked in the figure with the 'and actuator' phrase added to the smart meter block.

Figure 5 shows the inter-agent communications during a power shortage negotiation between HAs. In this negotiation HA4 has a power shortage and it is negotiating with the other 9 HAs to identify a power donor. AA fully supervises all operations. HA could draw on surplus power if the dwellers are not home or some appliances are not operational.

Lastly, it is worth mentioning that the system is effective in all weather conditions, not limited to summer and air conditioners; it can control heaters in the same manner.



Administrative-House1 agents startup negotiation and essential information exchange

Basic appliance set activation negotiation. House agents 1-3 have enough power

House agent 4 doesn't have enough power to power its basic appliance set, he searches for power and he finds house agent 5 has superfluity of power and ready to donate some to him

House agents 5-10 has enough power to power their basic appliance sets

Finally, all house agents power their basic appliance sets successfully

Fig. 5. Agent activity chart showing inter-agent negotiation.

2.3 Shifted Queue Optimization

The goal is to maximise the number and quality of active cooling appliances on a one per dwelling basis for a certain housing cluster size. This is an optimization case in which the number of HVAC appliances that can be operated with the available surplus power is to be maximized and the number of other types of HVAC appliances is to be minimized as demonstrated by the following:

- If there is sufficient power to operate all COs or more then the aim is to maximize number of ACs and the problem is formulated as:

$$\left[\sum_{G=1}^{ng} p_G^{Snoop} \geq \sum_{m=1}^{nac} P_m^{AC}\bigg|_{\max(\mathrm{nac})} + \sum_{h=1}^{nco} P_h^{co}\bigg|_{\min(\mathrm{nco})}\right] \Leftrightarrow \left[\sum_{G=1}^{ng} P_G^{Snoop} \geq \sum_{h=1}^{nh} P_h^{co}\right] \tag{1}$$

and

$$\mathrm{n_h} = \mathrm{n_{ac}} + \mathrm{n_{co}} \tag{2}$$

- If there is insufficient power to operate all COs then the aim is to maximize number of COs and the problem is formulated as:

$$\left[\sum_{G=1}^{ng} p_G^{Snoop} \geq \sum_{m=1}^{nco} P_m^{CO}\bigg|_{\max(\mathrm{nco})} + \sum_{h=k+1}^{nmf} P_h^{MF}\bigg|_{\min(\mathrm{nmf})}\right] \Leftrightarrow \left[\sum_{G=1}^{ng} P_G^{Snoop} \geq \sum_{h=1}^{nh} P_h^{MF}\right] \tag{3}$$

and

$$\mathrm{n_h} = \mathrm{n_{co}} + \mathrm{n_{mf}} \tag{4}$$

where

P_G^{Snoop}	Surplus power
P^{AC}	AC power
P^{CO}	CO power
P^{MF}	MF power
n_g	Number of generation sources
n_{ac}	Number of air conditioners
n_{co}	Number of air coolers
n_h	Number of houses

The constraints are:

1. $n_{AC} + n_{CO} + n_{MF} =$ Number of dwellings
2. $P^{AC} \geq P^{CO} \geq P^{MF}$

Figure 6 shows a functional block diagram of an optimization. The power consumption of each HVAC option is firstly placed in a HVAC queue starting with ACs to the right for each house, COs in the middle in the same sequence as ACs, and MF at the leftmost. The power of the rightmost 10 air conditioners are added and compared with the amount of surplus power available. If the power available is sufficient then every dwelling can operate an AC; if not the queue is shifted to the right one place and the counter is incremented. The operation is repeated until the amount of surplus power is larger than the required HVAC cluster total power; at this point the final combination of HVAC appliances is indicated by the shift counter. If the count equals 10 (for 10 houses calculations) then every dwelling is allocated an AC; between 1 and 9, the number of ACs is 7 and the COs is 3 and so on. In Fig. 6 the shift right line is used to

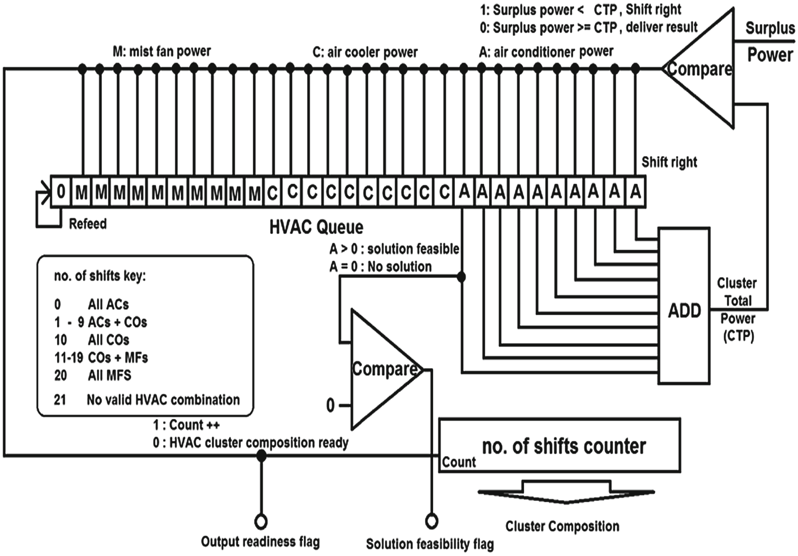


Fig. 6. Shifted Queue Optimization for 10 houses.

shift the queue to the right, filling the leftmost cell with zeroes while the two flag line at the bottom is used simply as a status indicator.

3 Implementation

3.1 Application

The solution is applied to resolve the Iraqi chronic cyclic blackout. Due to the long war (1980–2003), the bulk of the Iraqi generation, transmission, and distribution network was destroyed which has resulted in a severe power shortage and severe chronic cyclic blackouts. In parallel the extensive refurbishment of the Iraqi housing infrastructure (and thus deployment of household appliances) has resulted in increasing the power demand many fold. In addition to that Basra is subject to very hot weather, in summer days temperature reaching 56 °C; Fig. 7 shows such a day, Friday 12/7/2013 [15].

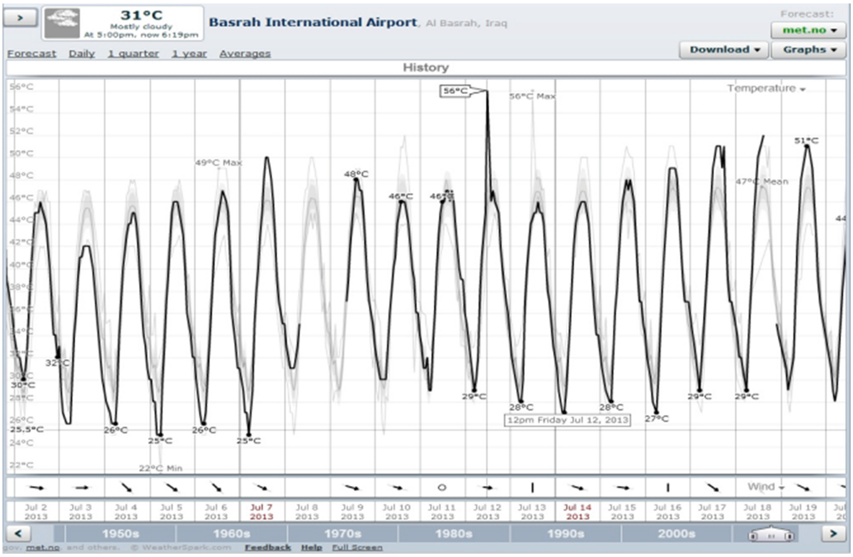


Fig. 7. Basra summer temperature

A detailed field survey was conducted to assess the problem, available resources, and the factors affecting it through a series of meetings with power administration personnel in Basra. A residential quarter -‘Kafaat’ - was selected as a suitable test environment. Figure 8 is the Kafaat residential quarter map where (1) is the medical complex, (2) Basra main bus station, (3) Ibn-Bitar hospital, (4) Kafaat residential quarter, and (5) is the Southern Oil Company (SOC).

The field study identified the availability of a mix of public, commercial and private standby generation surplus throughout Kafaat e.g. the largest being a 10 MW standby generation facility located at SOC. Figure 9 summarizes the commercial and public

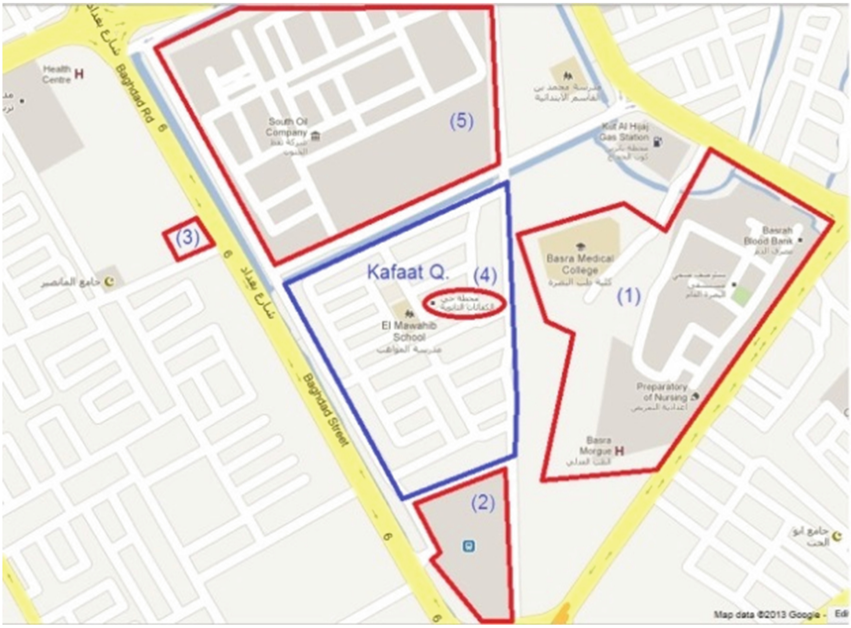


Fig. 8. Kafaat residential quarter.

standby generation facilities in Kafaat. Detailed LV distribution network maps with other technical support data were secured. to mark the boundaries, loads, LV transformers, house connection of the area needed to setup the 20 house test environment.

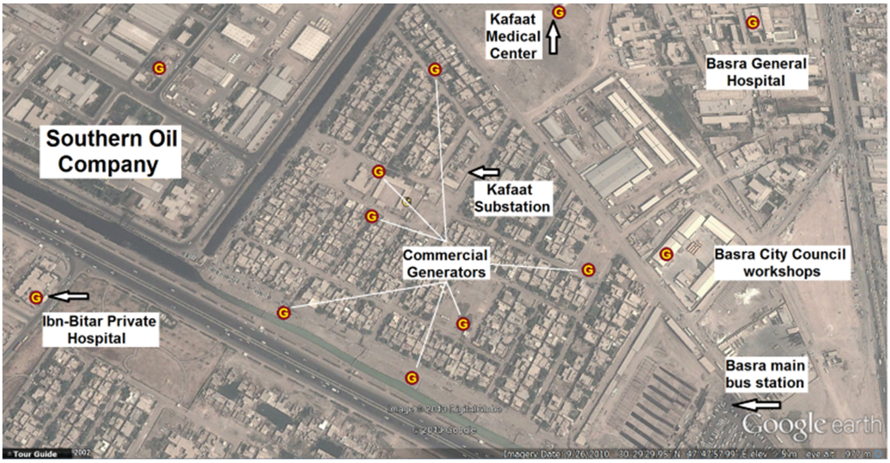


Fig. 9. Standby generation facilities in Kafaat residential quarter

Figure 10 is a schematic of the area showing houses in the left side, market, school, Mosque and substation at the right top corner.. The selection of these houses was made based on connection to the same LV distribution transformer and feeder.

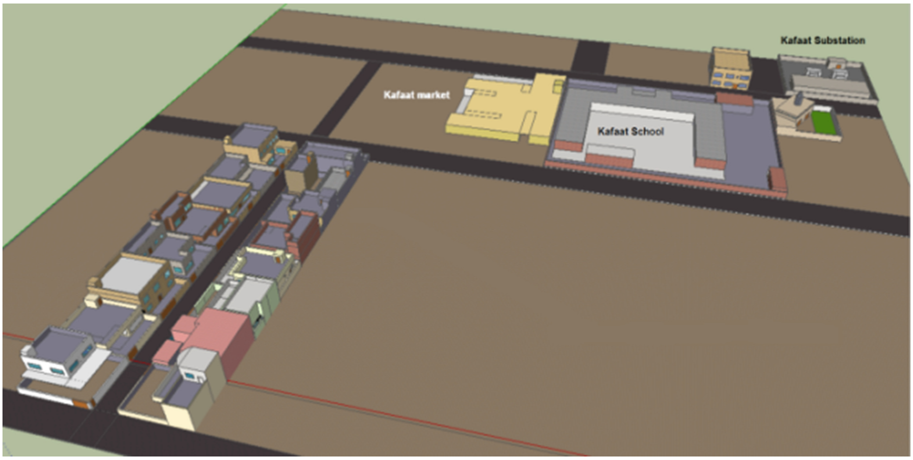


Fig. 10. Kafaat test-bed.

Figure 11 shows the detailed LV distribution network for the residential quarter; the left white rectangle shows the selected 20 houses and the right one shows the local substation.

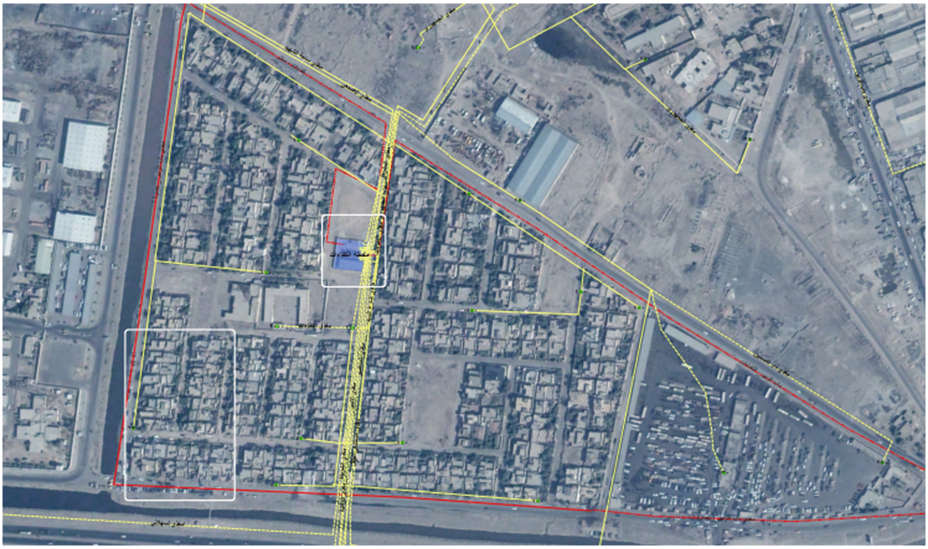
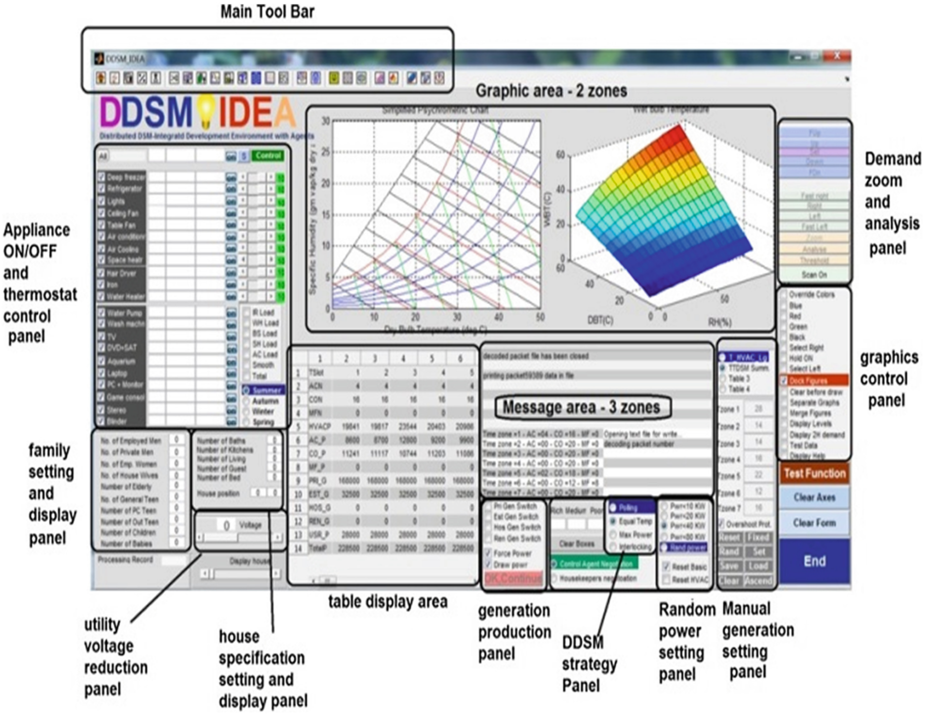


Fig. 11. LV distribution network.

In an attempt to alleviate the power shortage burden, the local authorities in Basra have proposed a two part incentive scheme offered to all commercial and private standby generation facilities participating in providing power to residential areas. The local government will supply fuel to these facilities free of charge and will also pay half the generation fees for commercial users while the other half of is charged to consumers.

3.2 System Simulation

In order to verify the proposed solution, an Integrated Development Environment (IDE) (Fig. 12) was designed. IDE comprises demand generation capabilities based on house, family, and individual daily routine data gathered from the field survey. informational data were embedded into the IDE and used to generate residential demand profiles to test the performance of the proposed solution. It is assumed that the controller that hosts the HA is connected wirelessly to the central controller at local substation which houses the AA. IDE includes ON/OFF and proportional appliance control, demand analysis unit, house and family setting profiles, generation selection mechanisms, a graphics area, table display zone, message display zone and graphics control panel.



	1	2	3	4	5	6
1	TSlot	1	2	3	4	5
2	ACN	4	4	4	4	4
3	CON	16	16	16	16	16
4	MFN	0	0	0	0	0
5	HVACP	19841	19817	23544	20403	20986
6	AC_P	8600	8700	12800	9200	9900
7	CO_P	11241	11117	10744	11203	11086
8	MF_P	0	0	0	0	0
9	PRI_G	168000	168000	168000	168000	168000
10	EST_G	32500	32500	32500	32500	32500
11	HOS_G	0	0	0	0	0
12	REN_G	0	0	0	0	0
13	USR_P	28000	28000	28000	28000	28000
14	TotalP	228500	228500	228500	228500	228500

decoded packet file has been closed

printing packet59389 data in file

Time zone =1 - AC =04 - CO =16 - MF =0
Time zone =2 - AC =00 - CO =20 - MF =0
Time zone =3 - AC =00 - CO =20 - MF =0
Time zone =4 - AC =00 - CO =20 - MF =0
Time zone =5 - AC =02 - CO =18 - MF =0
Time zone =6 - AC =00 - CO =12 - MF =8
Time zone =7 - AC =00 - CO =20 - MF =0

Opening text file for write ...
decoding packet number

Fig. 12. The integrated development environment (IDE).

4 Results and Discussion

Figure 13 shows a normal summer demand profile for 20 residential dwellings; no commercial, industrial, agricultural, religious, social, or sport areas are considered. Air conditioning is the dominating component in this demand and its footprint during night hours' is evident. Hour 8–14 are characterised by a low demand since the majority of people are at work or school.

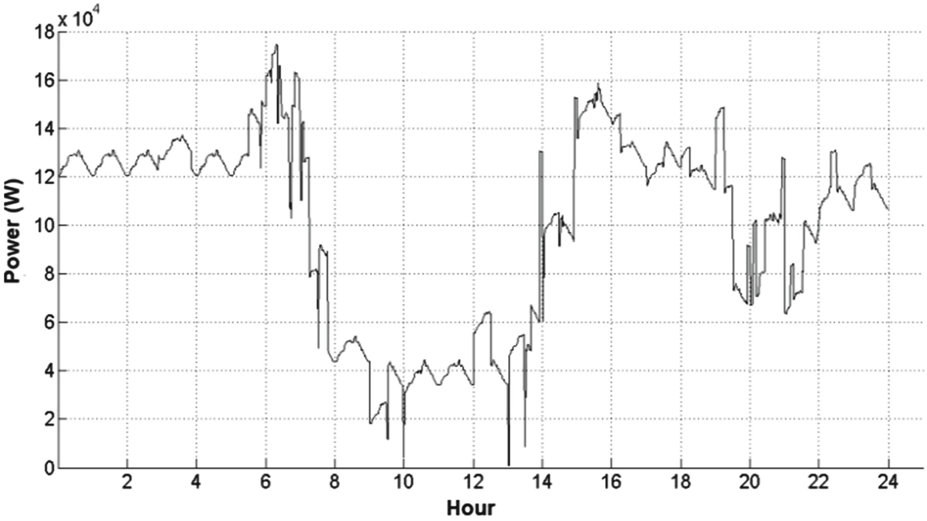


Fig. 13. Summer demand profile for 20 residential dwellings.

Figure 14 depicts the same demand profile but under a 4 × 2 cyclic blackouts viz. power is provided for 4 h and cut off for 2 h.

Figure 15 illustrates the manner in which the system manages the demand of 30 houses. For illustration purposes and to cover all possible cases, it is assumed that no utility supply is available and the base load was excluded to focus just on HVAC demand coverage. The 'blue' shows the collected surplus power while the 'red' shows the amount of power required for HVAC appliances only. The time zones are as follows; within hour 1–6 there is sufficient power to operate 15 ACs and 15 COs, so the time zone period is divided into two sub zones; within hour 7–8 no surplus power is available; within hour 9–12, a modest amount of power has been secured, sufficient to power 6 ACs and 24 COs. Thus the time is divided into (24 + 6)/6 = 5 sub-zones and since the time zone is 240 min long, then each sub-zone is 240/5 = 48 min long. Each house will have 1 × 48 min ACs usage time and 4 × 48 min CO usage time. Within hour 13–14, 5 (hour 15–20), and hour 23–24 there is sufficient power to operate all ACs (one for each dwelling).

Within hour 21–22 the power level is such that 25 ACs and 5 COs are active; the sub-zone period is 120/6 = 20 min i.e. every dwelling has 5 × 20 min AC usage time

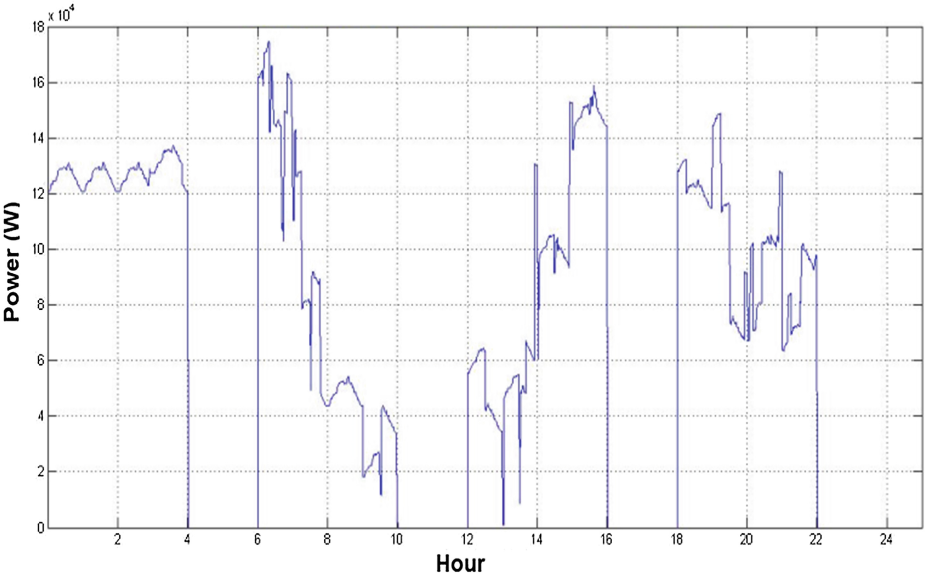


Fig. 14. Twenty houses summer demand under 4 × 2 cyclic blackouts.

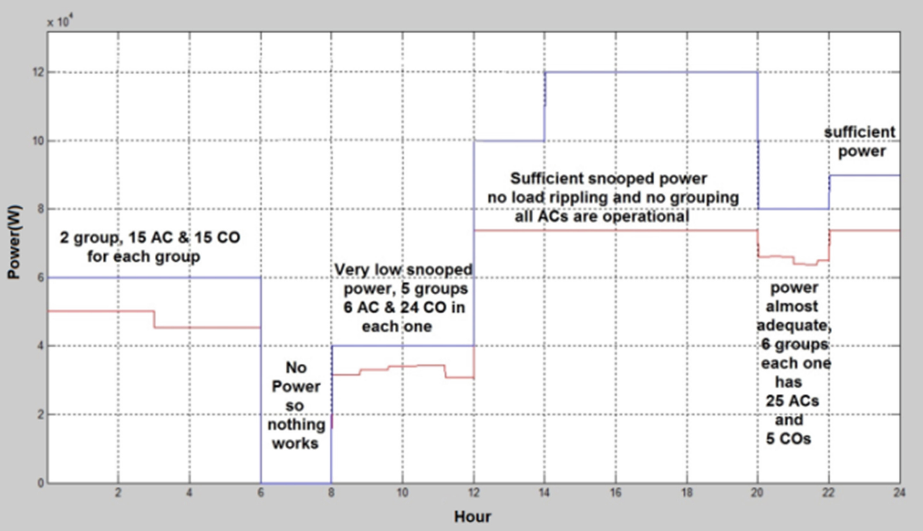


Fig. 15. SnP and HVAC demand for 30 houses (Colour figure online).

and 1 × 20 CO usage time. Table 3 summarises dwelling AC and CO usage rights for all sub-zones. It is worth noting that a level of underutilisation of surplus power occurs.

Figure 16 shows the summer demand for 20 dwellings under 4 × 2 cyclic blackouts (‘black’) and the bridging of the no-power gaps in the demand by the proposed system

Table 3. Air conditioning usage rights for time zone 7.

Sub zone	Number of dwellings with AC usage rights	Number of dwellings with CO usage rights
1	1–25	26–30
2	6–30	1–5
3	1–5, 11–30	6–10
4	1–10, 16–30	11–15
5	1–15, 21–30	16–20
6	1–20, 26–30	21–25

('blue') consisting of two components, the base load and HVAC demands. The latter is also plotted on the same figure ('red'). Both demand components are powered from surplus power ('purple').

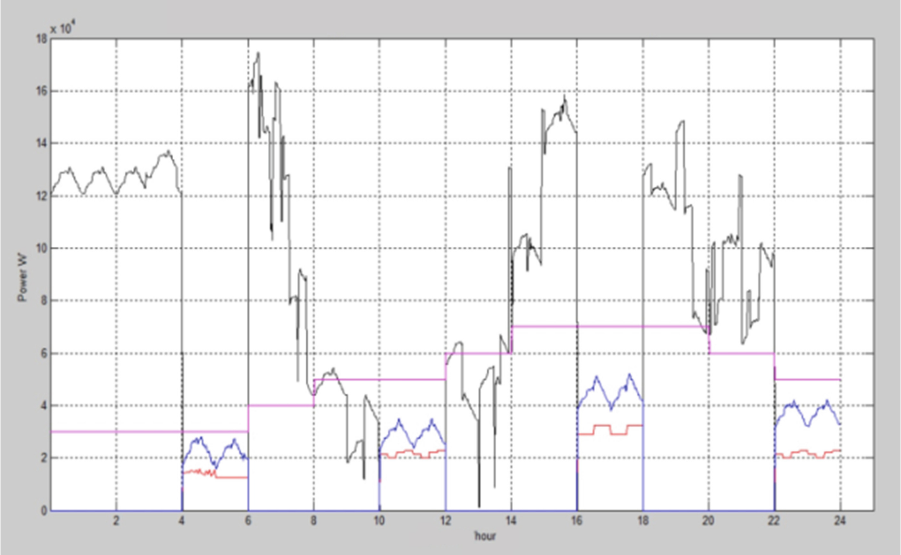


Fig. 16. Base load, surplus power, total and HVAC demands (Colour figure online).

The overall demand is shown in Fig. 17. During cyclic blackout periods, the proposed system covers the base load demand from surplus power and also uses the surplus to power HVAC appliances after optimization. There are no limits or constraints on the demand during utility on hours, so customers can sustain their normal power usage patterns. The amount of unused surplus power varies depending on base load demand and the optimization and equal distribution of HVAC appliances. The surplus power during utility on hours is not used. As is evident, the system has sustained base load and HVAC demand efficiently and has smoothed the demand profile.

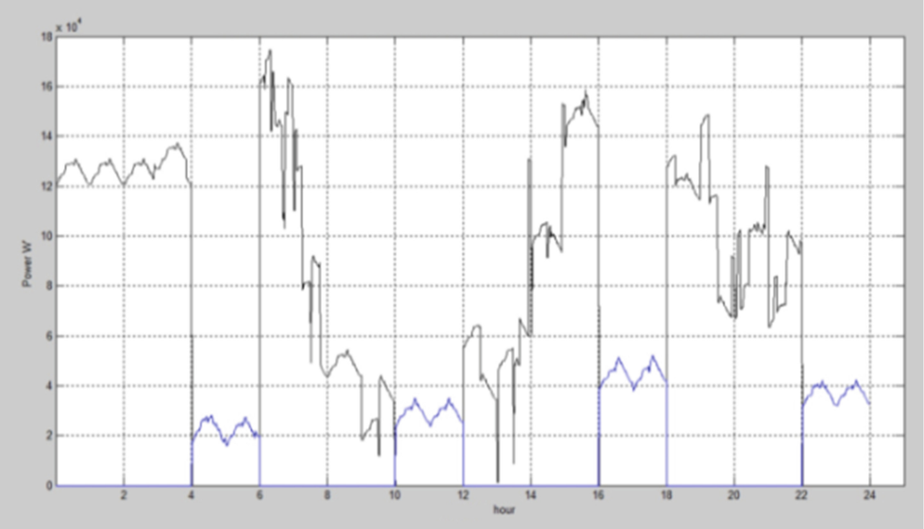


Fig. 17. The mixed demand profile.

5 Conclusions

A multi-agent distributed DSM-based system with a supply-demand matching capability has been used to sustain residential base load and HVAC demands during cyclic blackouts. A combination of air conditioners, air coolers, and/or mist fans are clustered in a dynamically updated pool of cooling resource used to service customers on a one-HVAC appliance per family basis. The mix and number of air conditioners, air coolers, and mist fans depends on the amount of available power. The HVAC appliances usage rights are distributed fairly amongst all customers. To power these appliances during a cyclic blackout, the system interrogates all nearby generation for any power surplus, aggregates all excess power and deploys it judiciously to provision power to better meet demand. The system clusters dwellings and assigns cluster heads to each, and recruits different types of agents for the negotiation of resources. Results show that the approach has succeeded in sustaining the power to basic household appliances and one HVAC appliance per dwelling. The resulting demand profile was also smoothed.

Acknowledgement. The researchers would like to thank Basra Governorate Council, Basra Power authority in all its branches that have participated effectively in supporting this research; University of Basra, Southern Oil Company, Basra Public Hospital, Ibn-Bitar Private Hospital, Kafaat Medical Center, and finally the people of Kafaat.

References

1. Wikipedia the free encyclopedia, November 2013. http://en.wikipedia.org/wiki/List_of_major_power_outages

2. Enacheanu, B. et al.: New control strategies to to prevent blackout: intentional islanding operation in distributed networks. In: 18th International Conference on Electricity Distribution, Turin, (2005)
3. Bashash, S., Fathy, H.K.: Modeling and control insights into demand-side energy management through setpoint control of thermostatic loads. In: American Control Conference, San Francisco, USA (2011)
4. Zhang, B., Baillieul, J.: A packetized direct load control mechanism for demand side management. In: IEEE 51st Annual Conference on Decision and Control (CDC), Maui, Hawaii (2012)
5. Qureshi, J.A., Gul, M., Qureshi, W.A.: Demand side management through innovative load control. In: IEEE Region 10 Conference (TENCON), Fukuoka, Japan (2010)
6. Ha, D.L., de Lamotte, F.F., Huynh, Q.H.: Real-time dynamic multilevel optimization for demand side management. In: IEEE International Conference on Industrial Engineering and Engineering Management, Singapore (2007)
7. Baba, M.F.: Smart grid with ADSL connection for solving peak blackout in west bank. In: First International Conference on Renewable Energies and Vehicular Technology (REVET), Hammamet, Tunsia (2012)
8. Shafer, M.G., Bakar, K.A., Ramadhani, F.: Novel dual demand side management (2DSM) scheme in optimizing utilization of available power. In: IEEE Symposium on Computational Intelligence in Control and Automation (CICA), Singapore (2013)
9. Chen, Y.-W., Chen, X., Maxemchuk, N.: The fair allocation of power to air conditioners on a smart grid. IEEE Trans. Smart Grid **3**(4), 2188–2195 (2012)
10. Amato, A., Calabrese, M., Di Lecce, V., Piuri, V.: An intelligent system for decentralized load management. In: IEEE international conference on computational intelligence for management systems and applications, La Coruna, Spain (2006)
11. Kok, J.K., Warmer, C.J., Kamphuis, I.G.: PowerMatcher: multiagent control in the electricity infrastructure. In: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS 2005), pp. 75–82, New York, USA (2005)
12. de Almeida, A.T., Yokoe, J.M.: Residential cool storage: peak load reduction alternatives. IEEE Trans. Power Syst. **3**(3), 837–843 (1988)
13. Bom, G.J., Foster, R., Dijkstra, E., Tummers, M.: Evaporative Air − Conditioning Applications for Environmentally Friendly Cooling - World Bank Technical Paper No. 421, Washington, USA (1999)
14. Yamada, H., Yoom, G., Okumiya, M., Okuyama, H.: Study of cooling system with water mist sprayers: fundamental examination of particle sizedistribution and cooling effects. Build. Simul. **1**(3), 214–222 (2007)
15. WeatherSpark, April 2014. http://weatherspark.com/#!graphs;ws=32872

Stochastic Petri Net Models for the Analysis of Trade-Offs in Data Centres with Power Management

Björn F. Postema[✉] and Boudewijn R. Haverkort

Centre for Telematics and Information Technology,
University of Twente, Enschede, The Netherlands
{b.f.postema,b.r.h.m.haverkort}@utwente.nl
http://www.utwente.nl/ewi/dacs/

Abstract. Due to the growth in energy consumption of data centres, the demand for optimal usage of servers has become a relevant topic. This paper contributes to the early design phases of data centres by providing insight into the power-performance trade-off that arises from power management. This paper proposes a flexible set of stochastic Petri net models which can be used easily to study the trade-off between performance and power consumption.

Keywords: Data centre · Power management · Power · Performance · Trade-offs · Stochastic Petri nets · Numerical models · Efficiency

1 Introduction

Energy consumption in data centres is still increasing [1]. As a consequence, CO_2 emission and energy bills increase all the more. So, it is worth to consider energy efficiency measures. In [2], ten ways to improve energy efficiency are elaborated. One of these ways is *power management* (PM), which is turning off servers completely or switching them into a lower power state, while trying to keep performance intact. Additionally, the cascade effect, as described in [3], which is the profit in the infrastructure that is obtained from reduction of the energy consumption of the IT equipment, strengthens the effect of efficient control strategies for PM. Hence, even small reductions in percentages of energy consumption through efficient PM may have large impact on reduction of CO_2 emission and decrease overall cost. In this paper, the simplest PM strategies considers only turning off idle servers, which can already save up to 20 to 60 % according to [4–6].

Efficient PM strategies can be explored by simulating and analysing the performance of various scenarios. Most techniques for exploring efficient PM use

Björn F. Postema—The work in this paper has been supported by the Dutch national STW project Cooperative Networked Systems, as part of the program “Robust Design of Cyber- Physical Systems”.

S. Klingert et al. (Eds.): E^2DC 2014, LNCS 8945, pp. 52–67, 2015.
DOI: 10.1007/978-3-319-15786-3_4

monitoring and testing. However, this reduces the number of applicable scenarios, since unpredictable behaviour has a low probability of occurrence and some scenarios cannot be easily tested. So, simulation and analysis of models provide a low-risk method that often result in remarkable insights into the performance of these systems. *Stochastic Petri nets* (SPNs) form a modelling technique which is convenient for performance analysis of systems. In this paper, various SPNs are introduced that model PM of servers in a data centre, and that can be efficiently analysed using numerical techniques; for background information on SPNs, we refer to [7]. For our analyses, we use a well-known software tool *Möbius* [8] that allows to model *Stochastic Activity Networks* (SANs), an SPN extension.

The goal of this paper is to obtain better insight in power consumption and performance, especially in the design phase of data centres. The contribution of this paper lies in presenting high-level theoretical models, that allow us to calculate both power consumption and performance for a given data centre configuration and a given stream of jobs. However, these high-level models might include aspects that can be considered unrealistic. The paper therefore also shows that extending the models is easy, in order to provide more realistic results.

In this paper, two SPN models for PM in a data centre are analysed: (a) a basic model with a *single server*; (b) an extended model with *multiple servers.* Even though the basic model could be called unrealistic, since it has only one server, it is still useful, because it provides fundamental insight in the trade-offs.

This paper is further organised as follows. First, the basic model for a single server with PM is explained and elaborated in Sect. 2, followed by a description of the multiple server model in Sect. 3; both are analysed for various scenarios. Section 4 discusses the complexity and scalability of the approach, and Sect. 5 presents related work. The discussion and conclusion are described in Sect. 6.

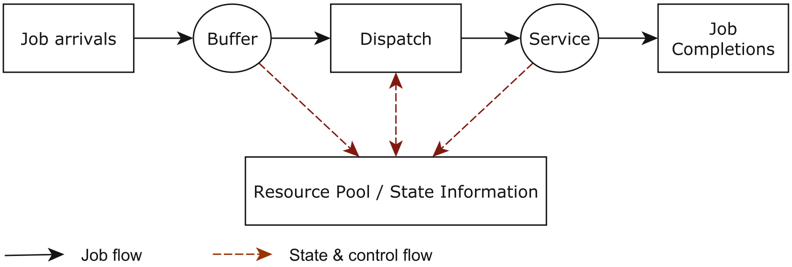


Fig. 1. The basic model of job flow in data centre with PM

2 Single Server Model

2.1 Basic Model

The basic idea of our models is illustrated in Fig. 1, which is based on [2]. A data centre serves a job stream from the outside world, buffers the incoming jobs and subsequently schedules and executes them, on the basis of the job requirements and the system-internal state information that is available. What that state information exactly is, largely depends on the data centre. It might involve only information on job queue lengths or server utilisation, but can also include information on temperature and humidity in parts of the data centre, or information in networking bottlenecks, to give just a few examples.

2.2 Single Server Stochastic Petri Net Model

Starting from the basic model for a data centre in Fig. 1, an SPN model is depicted in Fig. 2. In the model, jobs arrive to the server with rate λ via the transition `Arrivals` to the place `Buffer`. Notice that arrivals can only take place as long there is buffer capacity available, i.e., as long as there are tokens in place `BufferCap`. In that sense, the arrival process forms a truncated Poisson process. The server can be in four states: the server is either turned off (a token in place `Off`), booting (a token in the place `Booting`), processing a job (a token in the place `Processing`) or idle, i.e., not processing a job, (a token in the place `Idle`). The corresponding rewards `Po = 0`, `Pb = 200`, `Pp = 200` and `Pi = 140` signify the power consumption (in W) when a token is present in such a place. Once jobs arrive in the buffer the server needs to be booted or the idle server can directly start processing the job. The server is booted with a delay, which in our model is fixed to an exponentially distributed amount of time with mean 100 s ($\alpha = 100^{-1}$ jobs/s). After booting, the server processes the job with rate μ, where the size of the jobs determines the delay. For instance, a job size can be a web-request ($\mu = 1.00$ jobs/s, mean duration 1 s), a database request ($\mu = 0.10$ jobs/s, mean duration 10 s) or an upload/download of a file ($\mu = 0.01$ jobs/s, mean duration 100 s). The power consumption rates of servers and the job sizes are taken from [9–11]. After a job is processed, the server becomes idle. Then, either a new job is processed by the server or it is shut down after a idle time-out with rate β, i.e., as soon as a server becomes idle, a timer starts, that shuts down the server after, an average $1/\beta$ time is expired.

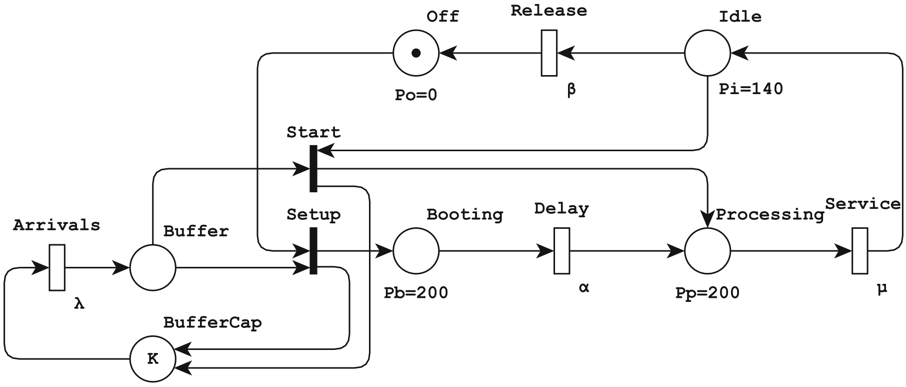


Fig. 2. Basic single server Petri net

In order to allow for a numerical solution (which is faster than simulation) the model should have a finite state space. For this reason, the place `BufferCap` is added, which is connected to the `Arrivals` transition with an output arc and the immediate transitions `Start` and `Setup` with input arcs. By setting the initial marking of `BufferCap` to `K` tokens, we make sure that there are never more than `K` jobs in the system. By setting `K` to a large value, we make sure that this finite state space does not have an impact on the performance measures of interest; we check this by making sure that the probability that place `Buffercap` would be empty, thus disabling the arrival stream of jobs, is very small in all our models. The computed probability throughout the paper is always almost zero (at least up to four digits).

To keep the models relatively simple, we have to make various assumptions. For our models to be able to obtain the underlying Markov chain (which is automatically derived and can be solved numerically), only models with exponentially distributed firing times (next to so-called immediate transitions) are allowed. In more detail, the current models assume a Poisson job arrival process, a server that has a fixed power consumption for the power states and an service time durations that follow an exponential distribution (which depends on the processing speed of the server and size of jobs offered to the server). Power management shuts down the server after a time-out expires. The value of the timer is drawn according to an exponential distribution. In practice, power management will work with deterministic timers, hence, our model is approximate in that sense. Our assumptions can be relaxed in various ways. For some of these relaxations, the underlying model is still a Markov chain, which can still be solved efficiently numerically. For others, e.g., for fully deterministic time-outs, we will need to resort to discrete-event simulation.

2.3 Power-Performance Trade-Off

The PM strategy in this paper allows to turn off idle servers and turn these servers back on, which can decreases the power consumption, but has often a negative effect on the performance, i.e., there is a so-called *power-performance trade-off*. In this section, the power-performance trade-off is explored with several performance measures, such as utilisation and mean response time. In [7] measures to obtain from SPNs are discussed, which can be applied for finding the mean response time and the throughput. The other performance metrics are found via SPN properties and reward variables. Assume that $\mathtt{J} \in \{\mathtt{Off}, \mathtt{Idle}, \mathtt{Processing}, \mathtt{Booting}\}$ and $m(\mathtt{J})$ denotes the number of tokens in place `J` for the following metrics:

The **mean number of jobs** $E[N]$ in the system is sum of the expected number of tokens in places that represent a job in the system: $E[N] = E[m(\mathtt{Buffer}) + m(\mathtt{Booting}) + m(\mathtt{Processing})]$; $E[N]$ can be easily computed from the SPN.

The **throughput** X of jobs in the system is equal to $\lambda \cdot \Pr\{\mathtt{BufferCap} \neq 0\}$, such that a stable system, i.e., with sufficient buffer capacity, has a throughput of λ jobs per second, otherwise the throughput is capped, since jobs that do not fit in the buffer are discarded.

The **power-state utilisation** is the expected number of tokens in a place corresponding to that power-state, which is computed as follows:

$$\rho_{\mathtt{J}} = E[m(\mathtt{J})]. \tag{1}$$

The **mean response time** $E[R]$ is the mean delay a job perceives from the moment it enters the buffer until the time it is finished with processing in a server. It is computed via Little's law using the average number of jobs in the system ($E[N]$) and the throughput of jobs (X), as follows:

$$E[R] = \frac{E[N]}{X}. \tag{2}$$

The **mean power consumption** of the server $E[P]$ (in W) is computed with a reward variable that depends on the state of the server, as follows:

$$E[P] = \sum_{\mathtt{J}} P_{\mathtt{J}} \cdot \Pr\{m(\mathtt{J}) = 1\} \tag{3}$$

2.4 Results

In this section, the utilisation and trade-off between the mean power consumption and mean response time are explored.

Table 1 shows parameters values (partially taken from Sect. 2.2) for the single server model; it refers to the set B which is taken such that a good spread of parameter values for β is found, as follows: $B = \{10^{-2} \cdot 1.28^i \mid i \in \mathbb{Z} \wedge i \in \{0, \ldots, 19\}\}$.

Table 1. Parameter assignments for single server model

Parameter	α	β	λ	μ	K
Assigned value	0.01	$\beta \in B$	0.007	0.01, 0.025, 0.1, 1.0	300

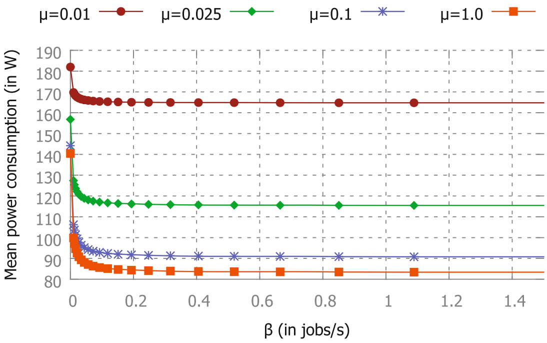


Fig. 3. Mean power consumption against β, using the simple data centre model, for four different service rates

The **mean power consumption** $E[P]$ (y-axis) against a changing β rate (x-axis) is depicted in Fig. 3 (from bottom to top: 1.0 (orange), 0.1 (blue), 0.025 (green) and 0.01 (red)). For small β, i.e., a long time-out value for an idle server, the mean power consumption is large, since the server is nearly always on and only needs to be booted very rarely. When β increases, i.e., a shorter shut down time for an idle server, the mean power consumption decreases and appear to converge to a fixed value. This is the case where a server is shut down nearly every time a job has been processed. Furthermore, the curves show that more power is consumed on average with a larger job size, i.e., when μ is smaller.

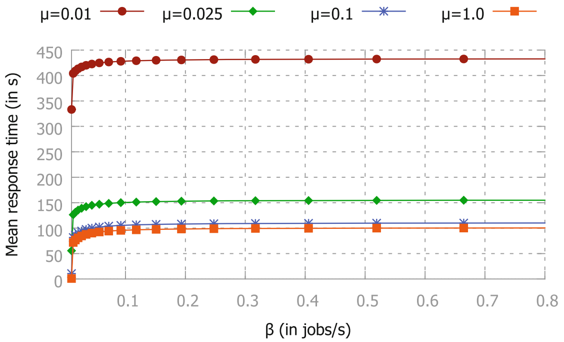


Fig. 4. Mean response time against β, using the simple data centre model, for four different service rates

Figure 4 shows the **mean response time** $E[R]$ against a changing β rate. The mean response times are rather large for one server due to the impact of booting servers, caused by PM. For this reason, when β is low, the mean response time is also low and when β increases the mean response time grows to a higher value.

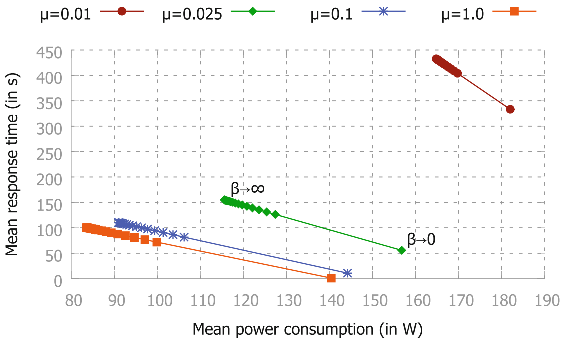


Fig. 5. Parametric curve for trade-off between mean response time and mean power consumption, using the simple data centre model, for four different service rates

Power-performance trade-off. Figure 5 illustrates the trade-off that exists between the mean response time and the mean power consumption. The mean response time is depicted on the x-axis and the mean power consumption is on the y-axis, for four different service rates μ. For each service rate μ, we clearly see that we can trade a lower mean power usage (to the left on the x-axis) for a higher mean response time (go up on the y-axis).

Note that the curve for $\mu = 0.01$ is much steeper than the other curves. This shows that the impact of changing β is relatively high for the mean response time and relatively low on the mean power consumption, compared to other service rates. The reason this curve is much steeper is that the number of jobs waiting for a server in the buffer is larger due to the relatively low processing rate ($\mu = 0.01$); this results in higher mean response times, especially when the server needs to boot more often. Hence, in this case the smallest increase in performance per Watt is obtained for shutting down a server.

3 Multiple-Server Model

3.1 Stochastic Petri Net Model

The next step is to extend the single server Petri net model from Fig. 2 with multiple servers, leading to the model illustrated in Fig. 6. First, new servers are added to the model by increasing the number of tokens in place `Off`. So, the number of tokens $n \in \mathbb{N}$ in place `Off` is equal to the number of available servers.

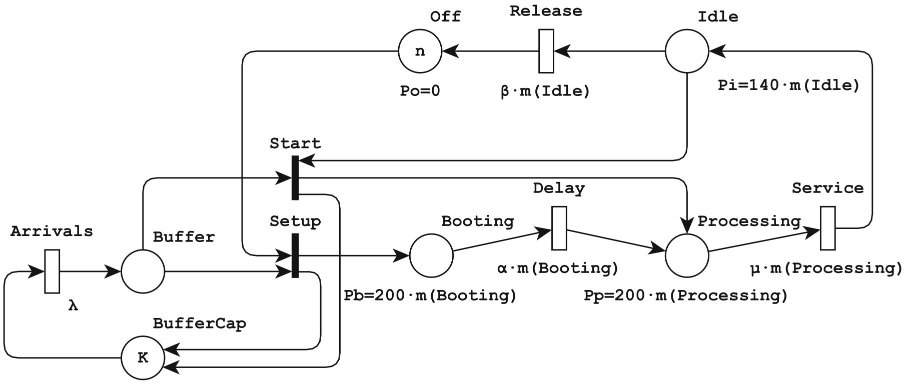


Fig. 6. Multiple server extension of basic Petri net

The depicted model is very similar to an $M|M|m$ multi-server queue as in [7], where m is the number of servers. In such a multi-server queue, the arrival rate λ remains the same when adding servers, however, the service rate μ depends on the number of jobs i in the buffer, as follows:

$$\mu_i = \begin{cases} i\mu, & i = 0, 1, \dots, m \\ m\mu, & i = m+1, m+2, \dots \end{cases} \tag{4}$$

The same principle is applied to the rates of the transitions `Delay`, `Service` and `Release` in Fig. 6, which are, respectively, $\alpha \cdot m($`Booting`$)$, $\mu \cdot m($`Processing`$)$

and $\beta \cdot m(\texttt{Idle})$, where $m(\texttt{P})$ represent the marking in place P. Our model is structured in such a way that a new server can only be activated when there is a token in the place Off. So, there are always n tokens spread over the places Off, Booting, Processing and Idle. The arrival rate λ from transition Arrivals is not changed, since multiple servers do not have an effect on this.

Note that the consequence of this approach is that PM is on a per-server level; there is no global timer for all the servers. So, each server starts a timer when it becomes idle, and turns itself off when the timer reaches its threshold, which is, as before, exponentially distributed with rate β.

In the current models, the servers are indistinguishable, since we assume homogeneous servers. As a consequence, all servers process jobs with rate μ, boot with rate α and release servers with rate β.

Furthermore, the power rewards now also depend on the marking. Obviously, if two servers are booting, i.e., two tokens are present in the place Booting, the power consumption for booting should be doubled. Note that in the model all servers have the same (distribution for) the booting time and for the processing jobs. So, for the places Booting, Processing and Idle the power rewards are adjusted to respectively $200 \cdot m(\texttt{Booting})$, $200 \cdot m(\texttt{Processing})$ and $140 \cdot m(\texttt{Idle})$. The power reward for a server that is off does not change, since the power consumption of turned off servers is exactly 0 W.

3.2 Power-Performance Trade-Off

All of the power and performance measures from Sect. 2.3 can again easily be computed. Additionally, the mean power consumption *per server* is also computed, which allows us to reason about each server individually and to make comparisons of various scenarios more informative.

As before, a large buffer capacity ($K = 300$) is chosen, such that the computed mean values remain accurate with larger numbers of jobs in the system.

3.3 Results

Two scenarios are elaborated by adjusting at most two variables in the model.

Scenario 1. The first scenario addresses the impact of scaling the number of servers in a data centres with and without PM. For scenario 1, Table 2 presents the used parameters. Every parameter is fixed, except for the number of tokens in place Off. In this scenario, the case with PM is compared to the case without PM (for which we remove transition Release from the model, so that the servers are always on).

Table 2. Parameters assignments for multiple servers scenario 1

Parameter	α	β	λ	μ	n	K
Scenario 1	0.01	0.005	1.0	1.0	2–10	300

Since the number of servers increases and the arrival rate is fixed, the processing utilisation $\rho_{\texttt{Processing}}$ is expected to drop. The cumulative utilisation plot, as depicted in Figs. 7 and 8, show from bottom to top the computed processing utilisation $\rho_{\texttt{Processing}}$ (blue), booting utilisation $\rho_{\texttt{Booting}}$ (green), idle utilisation $\rho_{\texttt{Idle}}$ (orange) and off utilisation $\rho_{\texttt{Off}}$ (red), confirms this expectation. The 2-servers case (top-left) has $\rho_{\texttt{Processing}}{\approx}50\,\%$ with PM and without PM and the 10-server case (bottom-right) has $\rho_{\texttt{Processing}}{\approx}10\,\%$ with PM and without PM. The greatest impact of shutting servers down is found with the lower processing utilisation with PM. For instance, in the 10-server case, the servers are expected to spend ≈35 % of the time off, while still ≈55 % of the time is wasted on booting a server and waiting as an idle server on jobs. In contrast to the case without PM, where servers are always on, no servers are turned off or need to boot and spend those moments idle.

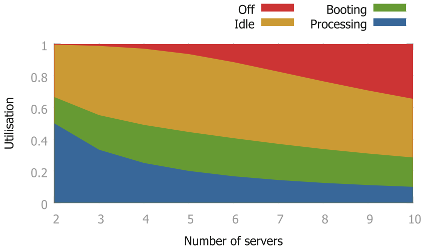


Fig. 7. Cumulative utilisation plot **with** PM when scaling the number of servers

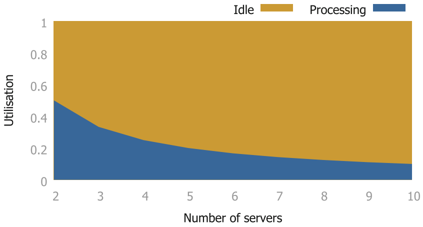


Fig. 8. Cumulative utilisation plot **without** PM when scaling the number of servers

Next, the mean response time is depicted in Fig. 9. Note that lines are added for better visibility. The plot shows the impact of PM on the mean response time for multiple servers. The 2-server case shows a mean response time of approximately 1.85 s with PM and 1.33 s without PM (top-left). The 10-server case shows a mean response time of approximately 1.25 s with PM and 1 s without PM (bottom-right). An interesting minimal mean response time is found with the 4-server case with PM of ≈1.16 s. When servers are booting and no servers are available to be booted or idle, new incoming jobs have to wait in the buffer for a server. Since booting takes 100 s and processing only 1 s on average, the impact on the mean response time is the greatest with small number of servers, which explains the high mean response time for 2 servers. In the plot, an increase in the mean response time is recorded from 4 to 10 servers. The reason for this is that servers are only shut down when the server is idle and no jobs are waiting to be processed. Recall from Fig. 7, that the 4-server case spends the same amount of time per job on booting as processing, whereas the 10-server case spends much more time per job on booting compared to processing, i.e., the 10-server case is much more often in power-state `Off` than the 4-server case. Therefore, the relatively long booting time compared to a short processing time slightly increases the mean response time.

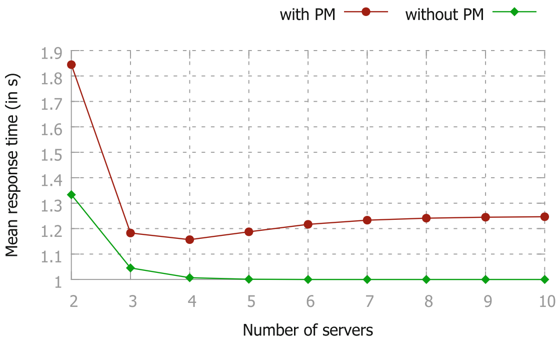


Fig. 9. Impact of PM on mean response time for various number of servers

Having seen the mean response time, next the mean power consumption is considered in Fig. 10. The mean power consumption for 2 and 10 servers is respectively 340 W and 1460 W, without PM, in comparison to the case with PM, where 2 and 10 servers respectively have a mean power consumption of 342 W and 939 W. The slightly higher cost with PM for 2 servers is caused by booting servers. However, the energy reduction, which is caused by turning off idle servers, increases with the number of servers and has a larger impact on the mean power consumption than booting servers. The power consumption is reduced when the number of servers is larger than 4 compared to the case where servers are always on for this workload. If you use 1 to 4 servers, it is more efficient to *not* use PM at all for this load in this scenario.

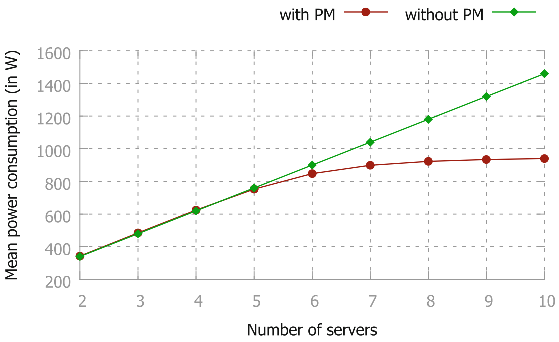


Fig. 10. Impact of PM on mean power consumption for various number of servers

Scenario 2. During the peak hours of a data centre, the number of arriving jobs increase. In this scenario, the impact of β and the number of servers are discussed by comparing two small data centres, one with 5 servers and one with 10 servers, where in both data centres the mean inter-arrival time ($1/\lambda$) between jobs and release rate (β) vary. Table 3 shows all parameters of scenario 2, which all are fixed, except for λ, β, and the number of servers, where β is taken from B as follows: $B = \{10^{-5} \cdot 1.28^{i} \mid i \in \mathbb{Z} \wedge i \in \{0, \ldots, 27\}\}$.

Table 3. Parameters assignments for multiple servers scenario 1

Parameter	α	β	λ	μ	n	K
Scenario 2	0.01	$\beta \in B$	0.5, 1, 2	1.0	5, 10	300

Figure 11 shows the trade-off between the mean response time and the mean power consumption for 5 servers, three job arrival rates, $\lambda = 0.5$ (left curve), $\lambda = 1$ (middle curve) and $\lambda = 1.5$ (right curve), and varying β. The utilisation is expected to be really low for 5 servers. The data points close to $\beta = 0$ are in the lower part of the curves, and the data points for larger β are in the top of each of the curves. Again, there is a power-performance trade-off, which can be regulated by β.

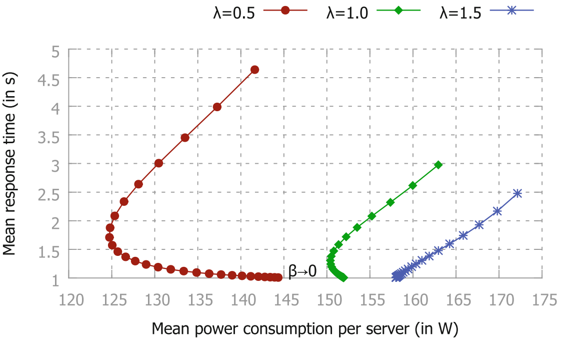


Fig. 11. Parametric curve for trade-off between mean response time and mean power consumption per server, using the multiple server data centre model with **5** servers, for three different job arrival rates

For instance, consider the curve for $\lambda = 0.5$ for 5 servers (red). Values for β higher than 0.001 ($E[R] \approx 1.8$ and $E[P] \approx 124$) are all bad choices for β, since the mean response time and mean power consumption both increase. However, for β below 0.001, the power-performance trade-off exists. In more detail, when β decreases the mean power consumption increases and the mean response time decreases. The other curves show similar behaviour, only the curve for $\lambda = 1.5$ (blue) does not show the power-performance trade-off any more, which basically suggests that with the current PM strategy it is better to always keep the servers on.

Figure 12 shows the same trade-off for 10 servers with similar curves. Note that λ is the same as with 5 servers, so the 10 server case has even smaller load.

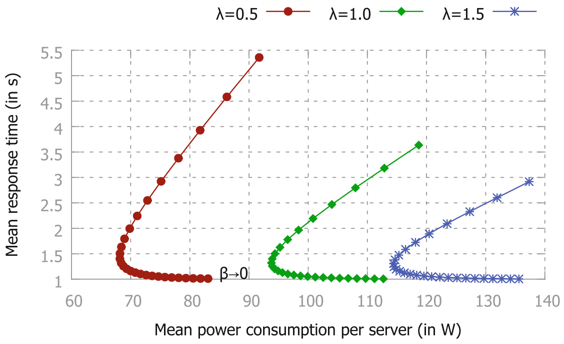


Fig. 12. Parametric curve for trade-off between mean response time and mean power consumption per server, using the multiple server data centre model with **10** servers, for three different job arrival rates

4 Computational Cost and Scalability

For the evaluation of all the models, the provided SPN is translated by Möbius into an underlying Markov chain; the number of states and the time for *state-space generation* (SSG) for various cases is depicted in Table 4. As can be seen, even though the models have thousands of states, generating these does not cost much time at all. The generated Markov chain is subsequently solved numerically using *Successive over-relaxation* (SOR). For smaller models, roughly, below 1000 states, a direct *LU-decomposition* (LUD), can also be used. As becomes clear from the table, the solution times are all very small, so that easily many scenarios can be studied. The longest computation times were reported for model instances with very small β, apparently leading to so-called stiff models. All Möbius settings were kept default or as recommended; the accuracy was set to 9 decimal places. All measurements have been performed on a machine equipped with a 2.70 GHz Intel® Core™ i7-4800MQ CPU, 8 GB of RAM and Windows 7 64-bit.

Table 4. Computation time and size of state space

	Single-server	Multi-server (1)	Multi-server (2)
Number of states	604	910—3586	1856—3586
CTDP[a] SSG (in s)	0.04	0.07—0.27	0.15—0.30
CTDP[a] LUD (in s)	0.01	n.a	n.a
CTDP[a] SOR (in s)	0.03	0.05—0.70	0.06—15.30
Total CTDP[a] (in s)	0.04—0.07	0.12—0.97	0.21—15.60

[a] CTDP = Computation time per data point.

Having seen the computation times and sizes of the state space for Scenario 1 (Sect. 3.3) and Scenario 2 (Sect. 3.3), the size of the state space for Scenario 1 is computed to examine how scalable the models are, as follows:

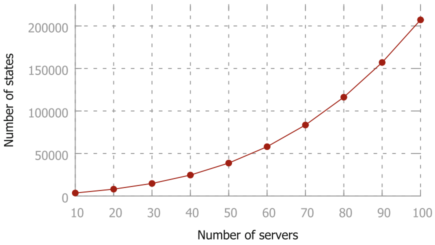


Fig. 13. Number of states when scaling the number of servers from 10 to 100 servers computed with Möbius

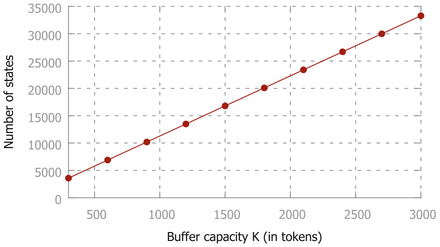


Fig. 14. Number of states when scaling the buffer capacity from K = 300 to K = 3000 computed with Möbius

Figure 13 shows the effect of scaling the number of servers on the number of states in Möbius. The 10-server case shown (bottom-left) has 3586 states and the 100-server case (top-right) has 207151 states.

As a consequence of scaling the number of servers, the number of tokens in buffer capacity often needs to be adjusted when the arrival rate is increased. Figure 14 shows the effect of scaling the number of tokens for the buffer capacity for the 10-server case from Scenario 2.

5 Related Work

The work of [12] focuses on power-performance trade-offs with PM, similar to this paper. Also, PM is conducted in [13], which focusses on power-saving algorithms with hysteresis for adapting to the load of the system. In contrast to our paper, both papers elaborate their models for virtualisation and/or multiple servers directly at the level of *Markov chains*, while we propose simpler SPN models, which are more generally applicable to data centres.

A similar remark applies to the papers [14, 15], which both focus on modelling consolidation of virtual machines with the aid of numerical analysis of *Stochastic Reward Nets* (SRNs). Their main goal is to propose and analyse virtual resource allocation strategies, which differs from our approach that focuses more on analysis of the power-performance trade-off for data centres with PM.

Power-performance trade-offs for various policies are also explored in [16], in which an exact analysis of an $M|M|k|setup$ multi-server queue (k servers and a *setup* time) using the new *recursive renewal reward* (RRR), for solving Markov chains with repeating structures, is applied. They focus on demonstrating the new RRR technique with a data centre case study, while our focus is on the analysis of power-performance trade-offs with PM.

The papers [17,18], which are related to the *All4Green* project, explore performance trade-offs at the server level, such that hardware components and energy-aware mechanisms for these components are taken into account. Other approaches, such as [19,20], focus on uncovering fundamental trade-offs (power, capacity, performance and dependability) with *Disk Power Management* (DPM), which is used for energy-aware file and storage systems. Also, here models for the power state of the system are analysed to discuss these trade-offs. In contrast to their work, our SPN models are defined on a much higher level, such that the overall impact on power-performance can be discussed.

Our approach differs from all the above in proposing a flexible set of convenient and extendible SPN models, which are numerically solvable via Möbius in order to analyse important power-performance trade-offs. Furthermore, the models are useful for data centres during the design phase to provide insight into trade-offs for various designs. In short, this paper differs from other works in the application of an other modelling technique and/or aim for a different goal.

6 Discussion and Conclusion

In this paper, simple models for single server and multiple servers data centres with PM are analysed, to study the power-performance trade-off. For that purpose, SPN models have been defined from which we can easily and quickly derive important performance and power-usage measures, such as utilisation, mean response time and mean power consumption. To do so, we used the tool Möbius. Interesting trade-offs between the mean power consumption and the mean response time are presented, which show that with PM reduction in the power consumption can be obtained at the cost of a higher mean response time.

For data centre analysts, an advantage of our approach is that the proposed models are reasonably high-level; this allows them to easily describe different configurations. Furthermore, the available analysis tools allow for the easy computation of relevant power and performance measures, thereby hiding mathematical details from the analysts.

The proposed SPN models can easily be extended towards models with (i) different PM strategies, e.g., with some form of hysteresis; (ii) dynamic PM, e.g., dynamic voltage and frequency scaling; (iii) multiple server types (speeds) and a mixtures of job sizes and inter-arrival times; (iv) sleep and hibernate states for the servers; (v) virtualisation; and (vi) thermal-aware data centres.

The models proposed in this paper still allow for the usage of efficient numerical methods. However, future model extensions might require discrete-event simulation, but the tool Möbius also supports this. Furthermore, validation of the models with actual measurements in data centres or a small measurement set up with actual servers is intended as future work.

Acknowledgements. The authors would like to thank the anonymous reviewers for their constructive feedback.

References

1. Koomey, J.G.: Growth in data center electricity use 2005 to 2010. Analytics Press, Oakland (2011)
2. Haverkort, B.R., Postema, B.F.: Towards simple models for energy-performance trade-offs in data centres. In: Fishbach, K., Grossmann, M., Krieger, U.R., Staake, T. (eds.) Proceedings of International Workshop on Demand Modeling and Quantitative Analysis of Future Generation Energy Networks and Energy Efficient Systems, pp. 113–122. University of Bamberg Press (2014)
3. Emerson Network Power: Energy logic: Reducing data center energy consumption by creating savings that cascade across systems. White paper of Emerson Electric Co, pp. 1–19 (2009)
4. Ohara, D.: Sustainable computing: Is it time to turn off your servers? TechNet Magazine (2008)
5. Narayanan, D., Donnelly, A., Rowstron, A.: Write off-loading: practical power management for enterprise storage. ACM Trans. Storage **4**(3), 1–23 (2008)
6. Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L., Zhao, F.: Energy-aware server provisioning and load dispatching for connection-intensive internet services. In: Proceedings of 5th USENIX Symposium on Networked Systems Design and Implementation, pp. 337–350 (2008)
7. Haverkort, B.R.: Performance of Computer Communication Systems: A Model-Based Approach. Wiley, New York (1998)
8. Clark, G., Courtney, T., Daly, D., Deavours, D., Derisavi, S., Doyle, J., Sanders, W., Webster, P.: The Mobius modeling tool. In: Proceedings of 9th International Workshop on Petri Nets and Performance Models, pp. 241–250. IEEE Computer Society (2001)
9. Barroso, L.A., Hölzle, U.: The case for energy-proportional computing. Computer **40**(12), 33–37 (2007)
10. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., Vogels, W.: Dynamo: Amazon's highly available key-value store. ACM SIGOPS Oper. Syst. Rev. **41**(6), 205–220 (2007)
11. Gandhi, A., Harchol-Balter, M., Kozuch, M.A.: Are sleep states effective in data centers? In: Proceedings of International Green Computing Conference, pp. 1–10. IEEE (2012)
12. Ghosh, R., Naik, V.K., Trivedi, K.S.: Power-performance trade-offs in IaaS cloud: a scalable analytic approach. In: Proceedings of 41st International Conference on Dependable Systems and Networks Workshops, pp. 152–157. IEEE (2011)
13. Kuhn, P., Mashaly, M.: Performance of self-adapting power-saving algorithms for ICT systems. In: Proceedings of International Symposium IFIP/IEEE on Integrated Network Management, pp. 720–723 (2013)
14. Bruneo, D., Lhoas, A., Longo, F., Puliafito, A.: Analytical evaluation of resource allocation policies in green IaaS clouds. In: Proceedings of 3rd International Conference on Cloud and Green Computing, pp. 84–91 (2013)
15. Bruneo, D., Longo, F., Puliafito, A.: Modeling energy-aware cloud federations with SRNs. In: Jensen, K., van der Aalst, W.M., Ajmone Marsan, M., Franceschinis, G., Kleijn, J., Kristensen, L.M. (eds.) Transactions on Petri Nets and Other Models of Concurrency VI. LNCS, vol. 7400, pp. 277–307. Springer, Heidelberg (2012)
16. Gandhi, A., Doroudi, S., Harchol-Balter, M., Scheller-Wolf, A.: Exact analysis of the M/M/k/setup class of Markov chains via recursive renewal reward. In: Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 153–166 (2013)

17. Basmadjian, R., Niedermeier, F., De Meer, H.: Modelling and analysing the power consumption of idle servers. In: Proceedings of 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability, pp. 1–9 (2012)
18. Lovász, G., Niedermeier, F., Meer, H.: Performance tradeoffs of energy-aware virtual machine consolidation. Cluster Comput. **16**(3), 481–496 (2012)
19. Bostoen, T., Mullender, S., Berbers, Y.: Power-reduction techniques for data-center storage systems. ACM Comput. Surv. **45**(3), 1–38 (2011)
20. Bostoen, T., Mullender, S., Berbers, Y.: Analysis of disk power management for data-center storage systems. In: Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, pp. 1–10 (2012)

The Future Role of Data Centres in Europe

GEYSER: Enabling Green Data Centres in Smart Cities

Ionut Anghel[1], Massimo Bertoncini[2], Tudor Cioara[1], Marco Cupelli[3], Vasiliki Georgiadou[4](✉), Pooyan Jahangiri[3], Antonello Monti[3], Seán Murphy[5], Anthony Schoofs[6], and Terpsi Velivassaki[7]

[1] Universitatea Tehnica Cluj-Napoca, Cluj-Napoca, Romania
{ionut.anghel, tudor.cioara}@cs.utcluj.ro
[2] Engineering - Ingegneria Informatica Spa, Rome, Italy
massimo.bertoncini@eng.it
[3] Rheinisch-Westfaelische Technische HochSchule Aachen, Aachen, Germany
{MCupelli, PJahangiri, AMonti}@eonerc.rwth-aachen.de
[4] Green IT Amsterdam, Amsterdam, The Netherlands
vgeorgiadou@greenitamsterdam.nl
[5] Zurcher Hochschule Fur Angewandte Wissenschaften, Winterthur, Switzerland
murp@zhaw.ch
[6] Wattics Ltd, Dublin, Ireland
anthony.schoofs@wattics.com
[7] SingularLogic, Athens, Greece
tvelivassaki@ep.singularlogic.eu

Abstract. Information Technology is a dominant player of our modern societies; Data Centres, lying at the heart of the IT landscape, have attracted attention, with their increasing energy consumption being a constant topic of concern, especially when it comes to the negative impact on the quality of their surrounding environment. Nevertheless, recent technological and societal advances are paving the way for DCs to change their role from passive energy consumers into *prosumers*, thus, transforming themselves into leading players within their smart district surroundings. This paper describes the innovative GEYSER approach to enabling green networked DCs to monitor, control, reuse, and optimize both their energy consumption and production, and in particular from renewable resources, towards becoming active participants within Smart Grids and Smart Cities.

Keywords: Green data centres · Energy efficiency · Energy *prosumers* · Renewable energy sources · Smart cities · Smart grids

GEYSER is co-funded by the European Commission as part of the 7th Research Framework Programme (FP7-SMARTCITIES-2013). The authors wish to acknowledge the Commission for its support, the efforts of the partners, and the contributions of all experts involved in GEYSER. The GEYSER consortium is: Engineering - Ingegneria Informatica Spa (IT), ASM Terni Spa (IT), Green IT Amsterdam (NL), SURFSARA BV (NL), ABB Limited (IE), Zurcher Hochschule fur Angewandte Wissenschaften (CH), SingularLogic (GR), Wattics Limited (IE), Rheinisch-Westfaelische Technische HochSchule Aachen (DE), Universitatea Tehnica Cluj-Napoca (RO).

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 71–86, 2015.
DOI: 10.1007/978-3-319-15786-3_5

1 Introduction

As our lives become heavily digitized, the role and significance of Information Technology grows steadily. Not surprisingly, Data Centres, lying at the heart of the IT landscape, have attracted attention over the last decade. The workload directed to DCs around the globe increases in volume and density, reinforced by recent technological advances, such as cloud computing and mobile networking. Such trends are actually a driving force towards keeping DCs close to, if not within, urban agglomerations, and are expected to rise in the upcoming years. On the downside, however, they also raise concerns about increased energy consumption and, in general, detrimental environmental footprint, especially under the prism of the climate change. Thus, energy and ecological efficiency are important factors to consider while designing, operating, or even decommissioning a DC, especially within urban areas.

Energy consumption of DCs significantly affects operational costs and, consequently, business sustainability and competitiveness; according to Gartner, Inc. [1] "energy-related costs account for approximately 12 % of overall data centre costs and are the fastest rising cost in the data centre". Besides, the sector's increasing energy needs pose risks of causing supply shortage and potential electricity network instabilities or blackouts. Introducing such instabilities has also a dire effect on the integration of (distributed) renewable energy sources, an essential element of Smart Grids. In a world of Smart Grids and Smart Cities, energy inefficient practices of the past such as for example ignoring the potential use of waste heat, cannot be sustained. For the DC sector to continue its operation, energy efficiency and seamless integration with Smart City infrastructures are mandatory steps towards environmental, business and social sustainability.

In the recent years, serious efforts have been taken up by consortia involving the industry, academia and public authorities to address the increasing energy demand of the DC sector [2–5]. So far efforts were focused on treating DCs as isolated islands: coordinated cooling and load management to reduce energy consumption [6], energy efficiency using virtualization techniques [7] or load (re)distribution [8], and of course big players of the industry, such as Google, adopting novel techniques for their back-up storage [9]. Although such efforts do provide valuable tools and practices towards reducing energy consumption and environmental footprint, they should be considered as just the beginning of the journey since: (a) the energy demand is still on the rise, so obviously they are not enough, and (b) they are missing out positive synergistic effects that emerge from considering DCs as connection hubs within both data and energy (including both electricity and heat) networks. The latter becomes of special interest when looking into DCs specific load profile which actually qualifies them as potentially leading players in offering support services to the power grid distributors within the market of the so-called ancillary services[1]. Also, in anticipation of game-changing transformations of this market due to the creation of local markets of services very much in connection with a Smart City vision.

[1] https://www.entsoe.eu/about-entso-e/market/balancing-and-ancillary-services-markets/.

Moving a step forward in this direction, the GEYSER project no longer considers DCs as pure energy consumer "silos" with no or limited relationships with the surrounding context and stakeholders, but paves the way for the next generation of green sustainable DCs, turning them into active connection hubs at the crossroads of two bidirectional, interwoven network overlays:

- The Data Network, characterized by computational and storage workload and communications flows, placed on different servers and DCs, and
- The Energy network, characterized by both unified energy and control and information flows

This duality can provide DCs with a unique leverage to become the enablers of maximizing the overall use of renewable energy while ensuring network stability, by offering energy consumption flexibility to Smart Cities (Districts) and delivering support services specifically tailored to the distribution grid operators. Thus, GEYSER future green, networked DCs will be able to monitor, control, reuse and optimize their energy consumption and production, from renewable energy in particular, within the framework of a holistic representation of energy and along the underway roadmap towards acting as both consumers and producers that is, *prosumers*, of energy.

The remainder of this paper is structured as follows: Sect. 2 discusses the innovative GEYSER approach to providing DCs with the right tools for transforming themselves into energy *prosumers* within a Smart City environment; Sect. 3 outlines the general GEYSER architecture; Sect. 4 presents the simulation environment where GEYSER vision is to be validated in preparation for the pilot activities; finally, Sect. 5 concludes this paper.

2 GEYSER Visionary Scenarios

The GEYSER project aspires to go a step beyond current efforts by providing DCs with the conceptual, business and software framework that enables them to become leading actors within their Smart City (District) environments. It should be noted that within GEYSER the concepts of Smart City and Smart Grids are not fully interchangeable. On the contrary, a GEYSER-compliant DC has the option of either offering energy flexibility to support Smart City (District) energy management and consumption optimization carried out at holistic level through the integration of power, heating, and cooling, or offering directly to the smart power grid operator(s) specific support and regulation services. In particular, the GEYSER vision is built on the following pillars:

- In the context of a Smart City environment
  - Energy is considered as yet another type of service, bringing under its umbrella electricity, heating, and cooling, thus introducing the term *unified energy*.
  - The role of a Smart City Energy Manager is introduced, being the one in charge of overseeing the optimal operations of energy distribution and management networks in order to ensure that energy demands on a Smart City level are met at all times.

  - DCs may have setup specific collaboration agreements with local energy distributor(s) (of power, district heating, or both) for participating to demand side management so as to provide short term regulating capability or as local energy supplier with near real time balancing flexibility. This provides DCs with the possibility of, for example, migrating their loads at peak time of renewable energy. Besides, by integrating DCs with the neighbourhood's local thermal grids, the waste heat generated by usual DCs operations can be transferred and used for meeting heating needs of associated business offices or neighbourhood (residential or otherwise) buildings.
- In the context of DC operations
  - DCs may have the possibility of in-house renewable energy source, for example solar panels on their roof or windmills in their premises. They may also have the possibility of storing energy temporarily, such as compressed air storage facilities, flywheel, ice storage tanks, and UPS. Naturally, back-up solutions of brown energy sources, for example diesel electric generators, may be available as well.
  - DCs implemented operation procedures should make continuous efforts for optimizing their local operations aiming at decreasing the energy consumption thus allowing to offer a greater flexibility on energy demands. For increasing the DC operation efficiency, two types of optimizations can be continuously running: (i) DC facility operation optimization by dynamically monitoring, controlling and adjusting the cooling, heating, humidity and lighting of the DC and (ii) DC workload execution optimization by energy aware IT workload migration, deployment, consolidation, and execution.

Bearing in mind the above, the GEYSER project is to also set the scene for a Unified Green Energy Marketplace, as a tool for enabling interaction between the DC sector and Smart Cities, while effectively optimizing the integration of DCs with a Smart City's physical energy infrastructure such as, for example, smart electricity grids and smart heating grids. By actively participating within the Unified Green Energy Marketplace, a DC may see an opportunity rising to meet the energy demand on the Smart City level while making profit and contributing to the overall efforts towards energy conservation, efficiency, and maximizing use of renewable energy sources. In such cases, the DC may choose to internally rearrange its operation so as to meet the underlying flexibility requirements by, for example:

- Falling back to their own energy storage reserves to manage in-house heating demands, for example in the form of precooling such as ice storage tanks
- Considering IT workload migration
- Cogeneration (make use of waste heat)

Thus, the GEYSER innovative conceptual and software framework will provide DCs with the means to interact primarily on the Smart City (Smart District) level by actively participating within the Unified Green Energy Marketplace to also fulfil energy demands; especially taking into account their unique load profile that allows them to offer specific ancillary services such as voltage regulation and load smoothening.

As the first step in realizing the ambitious GEYSER vision, various scenarios are identified and analysed within the project as a combinatory exercise alongside the following two dimensions:

- Is there the possibility of IT workload migration, either within the same DC (temporal migration, Service Level Agreements (SLAs) relaxation) or within a network of DCs (including therefore spatial migration as well[2])?
- Are there multiple DCs connected to the grid with the possibility of coordinated energy management?

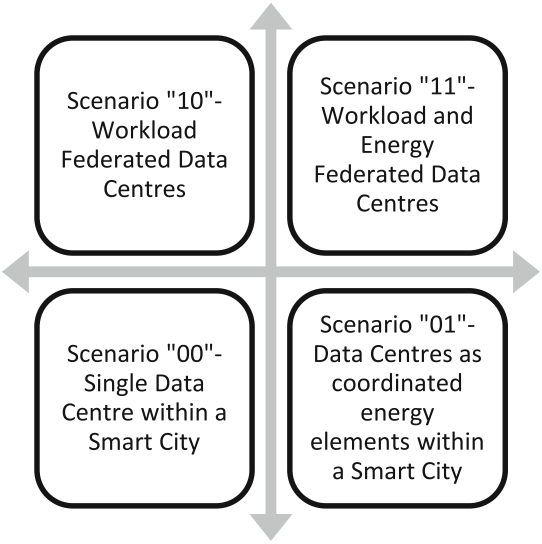

Scenario "10"- Workload Federated Data Centres
Scenario "11"- Workload and Energy Federated Data Centres
Scenario "00"- Single Data Centre within a Smart City
Scenario "01"- Data Centres as coordinated energy elements within a Smart City

Fig. 1. Scenarios 2-dimensional space

The 2-dimensional scenario space can be visualized as shown in Fig. 1. The horizontal axis refers to the option of coordinated energy management and the vertical one to the option of IT workload migration, with special focus on load relocation. The four reference scenarios are thus defined as follows:

- Scenario “00”: Single DC within a Smart City
- Scenario “01”: DCs as coordinated energy elements within a Smart City
- Scenario “10”: Workload Federated DCs
- Scenario “11”: Workload and Energy Federated DCs

Within the GEYSER framework, of high importance is the investigation of the synergistic effects on the overall energy consumption and sustainability efforts that emerge when multiple DCs become active players in the Unified Green Energy Marketplace.

[2] In the case of spatial migration the cost of IT workload transportation is an important factor to consider.

Adoption of the GEYSER framework is expected to lead to (i) DCs increasing the share of locally produced renewable energy in their operation, and, in general, contribute towards maximising the use of renewable energy sources, (ii) DCs contributing to reducing CO_2 emissions on system level, (iii) DC operations being optimised in terms of facility and IT resources energy efficiency and across a network of DCs, (iv) DCs acting as energy *prosumers* in the context of Smart City and Smart Grids, (v) DCs taking advantage of the smart grid flexibility for energy demand management, and finally (vi) DCs exchanging energy with the city through the Unified Green Energy Marketplace.

3 GEYSER System Design

3.1 General Architecture

The GEYSER conceptual architecture, as shown in Fig. 2 uses a hierarchical approach based on real time monitoring and sensing, assessing the energy efficiency of the system and taking optimization decisions to improve the energy efficiency from the lower layer individual components, all the way up to the network of DCs.

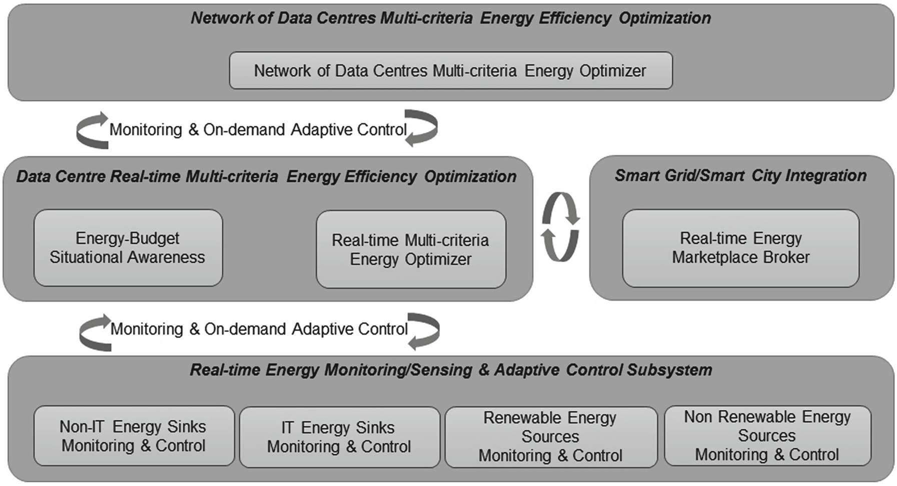

Network of Data Centres Multi-criteria Energy Efficiency Optimization
Network of Data Centres Multi-criteria Energy Optimizer
Monitoring & On-demand Adaptive Control
Data Centre Real-time Multi-criteria Energy Efficiency Optimization
Energy-Budget Situational Awareness
Real-time Multi-criteria Energy Optimizer
Smart Grid/Smart City Integration
Real-time Energy Marketplace Broker
Monitoring & On-demand Adaptive Control
Real-time Energy Monitoring/Sensing & Adaptive Control Subsystem
Non-IT Energy Sinks Monitoring & Control
IT Energy Sinks Monitoring & Control
Renewable Energy Sources Monitoring & Control
Non Renewable Energy Sources Monitoring & Control

Fig. 2. GEYSER Conceptual Architecture

Real-time Energy Monitoring/Sensing and Adaptive Control Subsystem. This subsystem aims at collecting data by means of sensors to (i) determine the initial status and energy (both electricity and heat) consumption of the various subsystems and provide a rough classification and categorisation of hosted applications, and (ii) identify an initial trade-off of schedule and allocation of workload to virtualised resources, which optimizes energy consumption, while respecting existing SLAs. The real-time measurements allow us to identify significant deviations from the optimal plan, which could be minimised. On this side GEYSER innovates by introducing a layer of software intelligence on top of physical measurements, trading off extra cost for incremental hardware with the level of information provided. Non-intrusive appliance load monitoring algorithms (NIALM) are used within GEYSER as a solution for disaggregating loads and for tracking information not possible to be provided by physical meters, such as for example, the current power state of a cooling unit or which lighting units are powered on, when they are all fed from a same circuit. Uncovering the periods of power activity of individual loads also allows for a finer investigation of DC components' consumption patterns that are currently unknown to DC operators.

Data Centre Real-time Multi-criteria Energy Efficiency Optimization. This component deals with two types of optimizations at different levels: continuous optimization at the level of the data centre and on-demand optimization at the level of the network of data centres.

The *continuous optimization* aims at improving the DC energy efficiency in the context of the smart city by: (i) optimizing the usage and sharing of DC locally produced renewable energy, (ii) optimizing the interaction and energy exchange with the smart city, and finally (iii) optimizing DC facility operation and workload execution. It also takes into account a number of new criteria including, for example, the overall DC energy consumption (electricity load, electricity transformations and geothermal), the overall energy production (renewable and heat), the energy price and flexibility aspects, the energy gain and the performance penalties by time and spatial relocation and migration of processing, internal network characteristics. The continuous optimization will be addressed by re-using the Green Cloud Scheduler OpenNebula ecosystem [10] component developed in the EU FP7 GAMES project and enhancing it to consider the new criteria of the GEYSER project.

The *on-demand optimization* aims at energy efficient relocation of workload across multiple interconnected DCs and automatically renegotiation and decreasing the SLA levels contracted by customers, for limited periods of time, when the DC cannot sustain the SLAs. In the energy optimizer a number of criteria are added including the overall network of DC energy consumption, the overall network of DCs energy production (renewable and heat), the energy gain and the performance penalties by time and spatial relocation and migration of processing, storage and networking load though a network of DCs, as well as the network and transmission costs.

The multi-criteria optimization problem requires specific search strategies capable of identifying the optimal or near optimal solutions. In consequence we plan to approach and solve the problem by means of evolutionary techniques that combine the strength elements of different bio-inspired meta-heuristics. We plan to combine population-based algorithms (such as Evolutionary Algorithms, and especially Genetic Algorithms, Bee Colony Optimization, and Particle Swarm Optimization) with trajectory-based algorithms (such as Simulated Annealing or Tabu Search) aiming to find the perfect balance between intensification and diversification aspects of the optimization problem. The advantage of using bio-inspired techniques is that they require modelling data structures with low processing overhead while by defining a proper fitness function, the optimal workload placement can be found in relatively short convergence times without processing the entire search space as opposed to classical exhaustive search strategies. For example, the energy efficient relocation of workload across multiple interconnected DCs can be approached exploiting the bees foraging behaviour [11, 12]. The GEYSER Optimizer may implement scout mobile agents (similar to scout bees) which randomly migrate from a source DC (the hive) to the interconnected DCs (food sources) aiming at gathering information regarding their operation context such as load levels, SLAs, cross-DCs network capabilities, energy consumption, power source type, and so on. When the scout mobile agents return, their findings are analysed and the most appropriate DC for workload relocation will be selected and the relocated workload activities will follow the path of the scout mobile agent.

To take optimization decisions the GEYSER optimization component relies on the sensed data describing the DC internal and external context for inferring, in real-time, the energy system status and the detection of contingencies or anomalies.

This is archived by means of the *Energy-Budget Situational Awareness (at the level of a DC and a Network of DCs)* components which estimate the current and foreseen energy consumption and production, taking into account IT energy consumption, cooling, heating, resources usage, internal network characteristics, heat dissipation, and so on. The Energy-Budget Broker components are based on semantic annotations of the sensed data to achieve contextualisation. The annotations are done using the semantically enhanced GEYSER data model (see Sect. 3.2). Reasoning techniques will be applied on the semantically enhanced data aiming at assessing the current energy budget of the DC. Prediction techniques will be used to forecast future energy consumption and renewable energy production trends both at DC and smart city and grid levels. Based on the prediction outcome, the GEYSER optimizer may proactively take actions to decrease the energy consumption, so as to take advantage of smart grid flexibility options or to sell the surplus energy to the city. GEYSER will develop prediction techniques based on time series analysis and neural networks that use the knowledge extracted by means of data mining to determine future energy consumption trends, frequent patterns, and so on. The constructed time series models will be used to extract information regarding short and long term trends of the energy consumption and production, cyclical, seasonal and irregular components. For example, during night time the workload and energy consumption levels of a DC are decreasing (a predicted cyclical component). Using this knowledge the GEYSER Multi-criteria Real-time Energy Efficiency Optimizer will proactively prepare the DC to save enough energy to power up a smart neighbourhood (if requested). Also, during summer time, the energy consumption levels of a smart city are at very high levels (a predicted seasonal component) mostly due to the extensive usage of air conditioners. As a result, GEYSER Optimizer may proactively shift most of the DC workload to partner DCs located in colder regions.

The IT Load Migration Broker and Network of DC Adaptive IT Load Migration subcomponents (not visible in Fig. 2) are in charge of defining the load migration strategy, passing the decision taken to the Virtual Machine scheduler, and estimating the best way to migrate the IT load among the network of DCs.

Customers' Quality of Service (QoS) & SLA Negotiation Broker subcomponent (not visible in Fig. 2) is responsible for monitoring the QoS of the offered services and re-negotiate SLAs between DCs. Since the workload to be migrated within the limits of a DC corresponds to a particular service that is provided to a specific customer under certain conditions and guarantees, it is of primary importance to maintain the SLA guarantees after the reallocation. GEYSER aims to make sure that the established SLAs will not be affected by the migration process. However, in case this might not be possible or in case an adjustment of the SLA is required so as to achieve a better pricing mode for the end-customer, GEYSER automatically re-negotiates SLAs and decreases the customers contracted SLA levels, within acceptable and agreed limits for limited periods of time. SLA negotiation is dynamic, in the sense that a new SLA offered by the DC provider may be followed by modified suggestions by the customer.

3.2 Semantically Enhanced GEYSER Data Model

A central component of the GEYSER framework is the semantically enhanced data model, whose goal is to define the energy efficiency semantics for DCs in the context of urban smart environments. The model will be implemented by means of ontology and will be used for: (i) defining the main energy-enhanced business vocabulary together with energy-performance complex dependencies, (ii) representing and sharing data and knowledge among GEYSER modules, and (iii) enacting reasoning techniques for assessing the energy efficiency of the DC and forecasting the energy consumption and production future values.

In that respect, EU FP7 GAMES data model will be extended to cope with the DC flexibility aspects and interoperability with the smart city and to ensure semantic integration with the prominent EU eeBuilding Models including the leading-edge BEMS model and a semantic mapping with the EU FP7 COOPERATE [13] project Neighborhood Information Model (NIM).

Concepts related to identifying the DC energy efficiency are classified in two main branches:

- Concepts describing the DC internal context – used to assess and improve the energy efficiency of a single isolated DC data;
- Concepts describing the DC external context - used to assess and improve the energy efficiency of the DCs in the context of the smart grid and smart city.

Data Centres Internal Context. DC internal context semantic model design is built upon the Energy Aware Context Model [14] defined in the EU FP7 GAMES project and GEYSER will use a Component – Action – Indicator based approach. As a result the relevant concepts are classified within three main categories: (i) DC Components, (ii) DC Energy Efficiency Optimization Actions, and (iii) DC Energy Efficiency Indicators.

DC Components are describing active elements inside the DC that can be monitored, controlled, and optimized. DC Components are further classified in two main sub classes: ***components that consume energy*** and ***components that produce energy***. In the DC we have identified two classes of components that consume energy, as shown in Fig. 3: (i) non-IT components which do not run workload but may be used to assure the proper conditions for workload execution or improve the quality of personnel working conditions and (ii) IT components which run the DC workload.

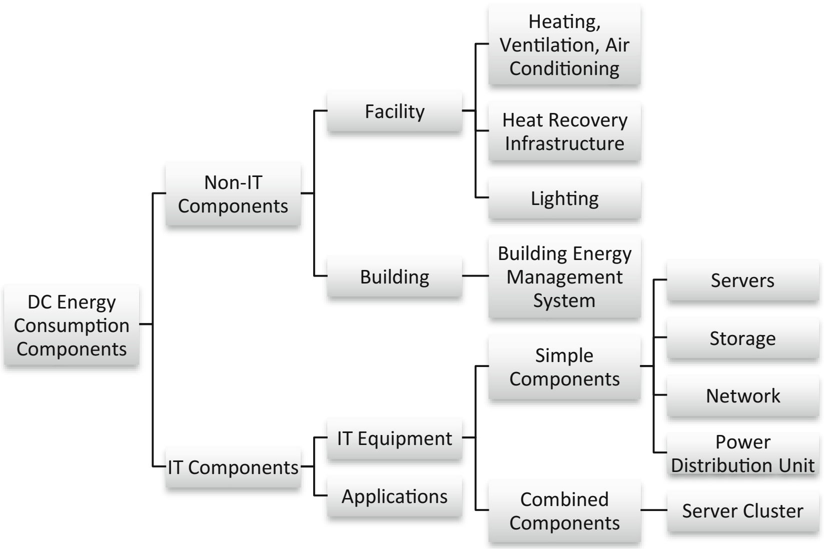


Fig. 3. Classification of DC energy consumption components

The ***DC Components that produce energy*** (associated or internal to the DC), part of the GEYSER semantic data model are classified as:

- Green Energy Sources – resources that produce renewable energy (for example photovoltaic, wind, and geothermal);
- Brown Energy Sources – resources that produce energy and generate pollution (for example, Diesel Generators);
- On-site Energy Storage Components - components which are used to store energy and provide it when requested (for example, UPS).

DC Energy Efficiency Optimization Actions describe the classes of actions that can be taken to improve the energy efficiency of the DCs. For each type of components that can be monitored and controlled, we have identified classes of optimization actions that can be used to improve their energy efficiency. Figure 4 shows the classification of DC energy efficiency optimization actions to be considered in GEYSER.

DC Energy Efficiency Indicators are representing metrics and their associated thresholds used for assessing the DC energy efficiency. Figure 5 presents a categorization of DC parameters that need to be measured and used to assess the energy efficiency of DCs. One of the challenges to be addressed in GEYSER is to find an optimal combination of state of the art metrics that covers all interesting parameters shown in Fig. 5 or to define new metrics for covering all relevant aspects.

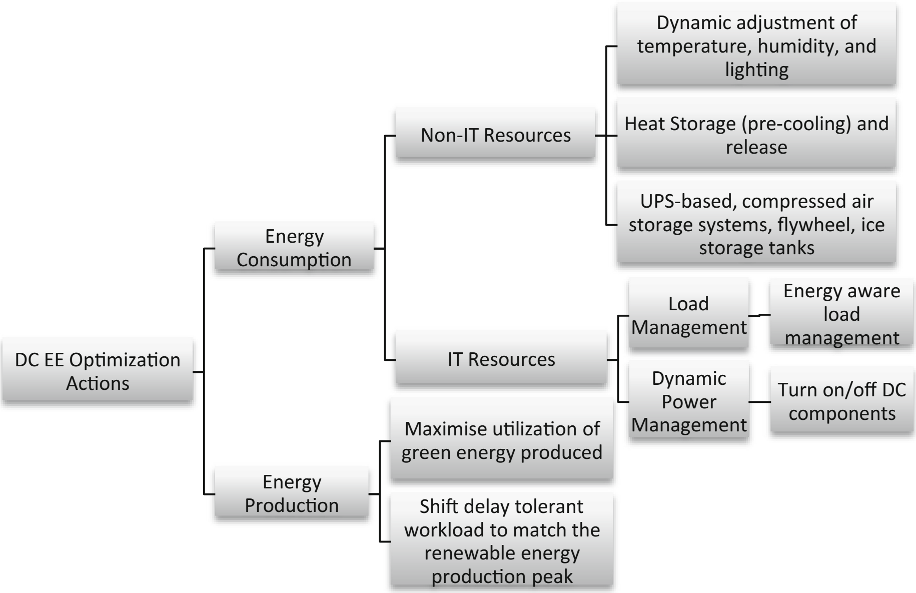

DC EE Optimization Actions
Energy Consumption
Energy Production
Non-IT Resources
IT Resources
Dynamic adjustment of temperature, humidity, and lighting
Heat Storage (pre-cooling) and release
UPS-based, compressed air storage systems, flywheel, ice storage tanks
Load Management
Energy aware load management
Dynamic Power Management
Turn on/off DC components
Maximise utilization of green energy produced
Shift delay tolerant workload to match the renewable energy production peak

Fig. 4. Classification of DCs energy efficiency optimization actions

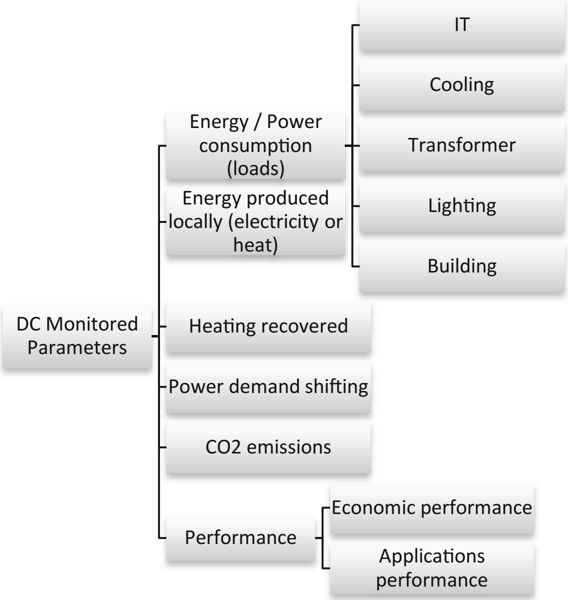


Fig. 5. DC monitored parameters for energy efficiency assessment

For each indicator the following properties are represented in the GEYSER semantic model: (i) the indicator formula or condition which is the subject of evaluation (if the condition is false, optimization actions must be taken), (ii) policy's evaluation value for the current data collected from the DC (true or false), (iii) the set of DC components on which the indicator imposes restrictions, and (iv) the set of DC components through which the indicator may be enforced.

Data Centre External Context. In this section we describe the relevant concepts for semantically modelling the DC external context, more specifically the interaction with the Smart Grid and Smart City. We consider the DC as an Energy Element of a smart neighbourhood and use the model developed within the EU FP7 COOPERATE project, NIM, for representing the DC integration within the smart grid and smart city. Figure 6 presents the integration of the GEYSER semantically enhanced model with the COOPERATE NIM and the association of its elements with the DC internal context information.

The dark boxes in Fig. 6 indicate the extensions to the COOPERATE NIM, while the light boxes indicate the elements of the NIM which are used to represent the integration and interaction between the DC and Smart Grid and Smart City. In our vision the NIM *EnergyData* element needs to be extended to represent the overall amount of the DC energy consumption and production provided by means of the DC internal components (see above). The *DataCenter* element is a model based

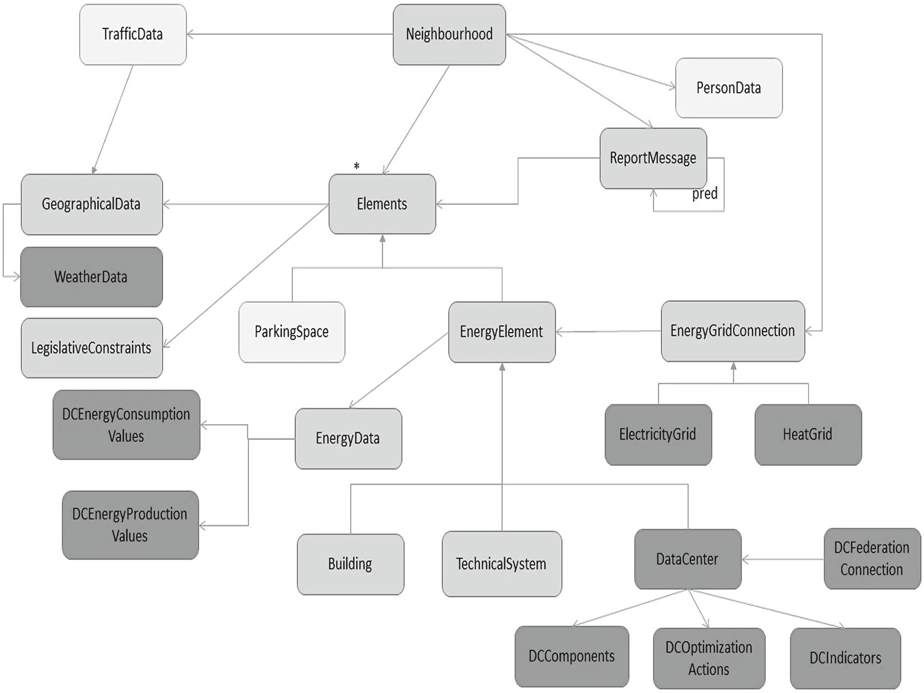


Fig. 6. Integration of the GEYSER semantic data model with COOPERATE Neighborhood Information Model

representation of the DC itself. It must refer to all components inside the DC relevant for assessing its energy efficiency. The *EnergyGridConnection* element will be extended to represent the two different types of energy flows that may exist, according to GEYSER vision, between a DC and smart neighborhood: heat grid connection and electricity grid connection. The *LegislativeConstraints* element will be extended to express constraints related to SLAs, while the *GeographicalData* will be extended to cover also the weather forecasting data which will be used in GEYSER for forecasting renewable energy sources availability and energy prices.

4 GEYSER Simulation Environment

In order to deliberate on various aspects of integrating DCs in smart energy networks, the various scenarios developed within GEYSER are to be tested. These scenarios are to be evaluated using simulations as well as Hardware In-the-Loop (HIL) test beds. This approach ensures the uniformity of the boundary conditions in each simulation, making the comparison and optimization procedures possible.

4.1 Modelling

All the components related to DCs as well as smart cities are to be represented as models. The appropriate modelling especially for electric components is a challenge, as the observed behaviour depends on the observation and later simulation resolution. Energy shifting actions have a time constant of several minutes until they show a beneficial value for the user. In contrast to that, power quality phenomena have a timescale of less than 1 ms. Since large scale simulations are necessary to study the behaviour of different systems, the models only need to reproduce the behaviour of each component at its boundaries. For example, a rack model in the DC should not include the detailed structure and components inside the rack but only how the rack and its surroundings interact dynamically.

All models are designed to be expandable. This ensures the scalability of the whole system with regards to the different scenarios making it possible to use the same platform for diverse applications. The models are also developed in such a way that real-time simulations are possible in order to run HIL simulations.

Since the whole electrical and thermal behaviour of the system is taken into account, the modelling language Modelica [15] has been chosen. Modelica is an object-oriented, equation-based language for modelling physical systems which allows multi-physic simulations including mechanical, thermal, electrical and hydraulic as well as control components [16].

An extensive library for different components is required for the whole system, containing but not limited to energy sources such as generators, CHPs and PVs, cooling equipment such as chillers and air conditioning units and IT components like different types of racks, lightning, and so on.

The modelling will not only consider internal to DCs components but also the interfaces of bidirectional energy exchange between a DC and the Smart City. An overview of the various model blocks is given in Fig. 7.

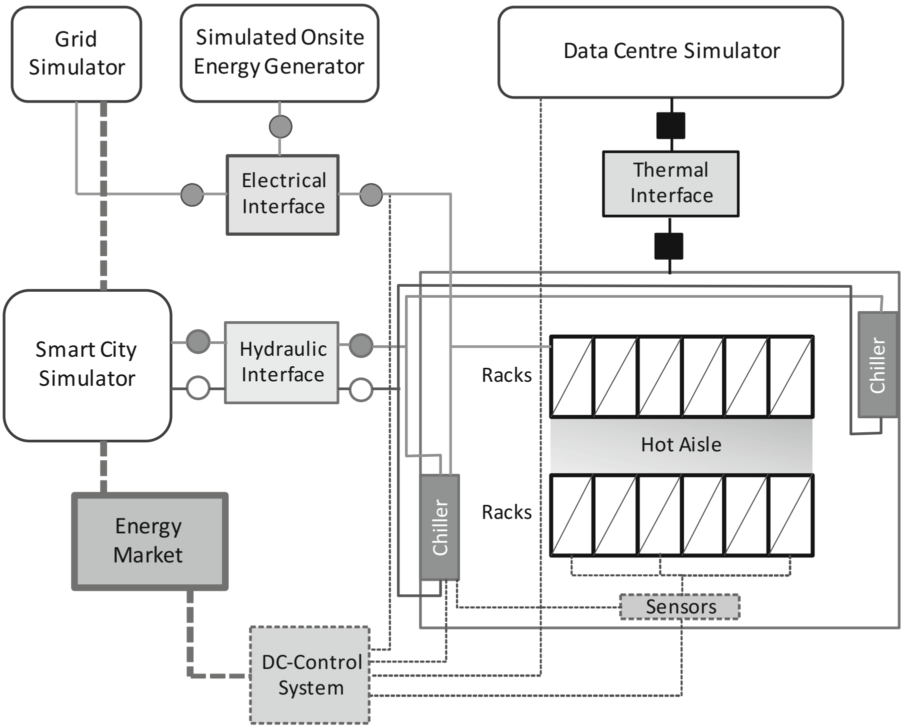


Fig. 7. Model blocks

As a consequence a DC-Smart Grid interaction is to cover not only pure energy transfer but also to model the provision of ancillary services such as voltage support from the DC towards the Smart City. Since the latter is categorized as a power quality topic the models used for the simulation should be designed so as to hold for this timescale as well. The provision of ancillary services from the DC to the Smart City can then improve the energy efficiency of the whole system, as this provision does not need to come from other sources.

4.2 Simulation Methods

Two different simulation parts are considered. First whole systems are simulated and optimized using different energy sources and sinks as well as different energy carriers and control strategies. This enables the incorporation of futuristic scenarios and optimizations in DCs.

Another part of the simulation includes the HIL simulation. Different components in the simulation can be replaced with real hardware. For example, a complete DC or part of it can be used as the hardware connected to simulation models of the rest of the DC as well as the smart city. Simulation models and hardware will be connected using the so called electrical, thermal and hydraulic interfaces [17]. Depending on the boundaries between the software and hardware, different interfaces should be built to emulate the simulation conditions for the hardware side.

As mentioned earlier, one of the challenges in simulating multi-physic systems is the different time constants for different components. Thermal and hydraulic components can use time steps in the order of seconds or minutes while electrical components need to be computed at least every 1 ms for sufficient accuracy. To overcome this problem, as suggested in [17] the complete system model should be split in different nodes according to their simulation time step. Each node can be executed on a separate platform which then can interact with other platforms using shared memory architecture.

The HIL simulation will be used to evaluate and validate the methodology of GEYSER solutions in a lab-based environment aiming at better understanding the energy profiles as well as energy related processes in DCs.

- This approach enables us to successfully verify the following scenarios and the scalability of the models: Real time testing of the complete DC and interaction with the Grid
- Detailed testing of the DC operation and its components with a limited model of the grid
- Power Hardware in the Loop of the efficiency of the cooling of one or more computational racks

In any of the cases, the simulation models will be able to communicate with all external sources such as the marketplace.

5 Conclusions

Although considerable efforts have been made in the last years to reduce the energy consumption of the DC industry and improve its efficiency, these are not enough in the dawn of the Internet of Things society. Especially, for DCs located within urban environments there is still room for improvement by removing the barriers between the DC sector and Smart Cities. This paper presented and discussed the first steps taken within the EU FP7 GEYSER project towards this direction. In particular, within the previous sections we have identified the GEYSER visionary scenarios, described the overall system architecture, and outlined its novel approach in optimization mechanisms to be considered and simulation methodology to be followed. These constitute the building blocks for GEYSER to design, implement and validate a fully innovative conceptual, business, and software framework for green energy-sustainable DCs acting as energy *prosumers* within a Smart City and Smart Grid paradigm, in which the DCs will be the key transformational nodes by also providing energy as a service.

References

1. Kumar, R.: How to measure Energy Consumption in Your Data Center, Gartner RAS Core Research Note G00205428, September 2010. http://www.gartner.com/resId=1433244
2. EU FP7 GAMES, Green Active Management of Energy in IT SErvice Center. http://www.green-datacenters.eu/
3. EU FP7 FIT4Green, Federated IT for a sustainable environment impact. http://www.fit4green.eu/
4. EU FP7 All4Green, Active collaboration in data centre ecosystem to reduce energy consumption and GHG emissions. http://www.all4green-project.eu/
5. EU FP7 CoolEmAll, Platform for optimising the design and operation of modular configurable IT infrastructures and facilities with resource-efficient cooling. http://www.coolemall.eu
6. Parolini, L., Sinopoli, B., Krogh, B.H.: Reducing data centre energy consumption via coordinated cooling and load management. In: HotPower 2008: Workshop on Power Aware Computing and Systems, December 2008
7. Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centres. In: IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), pp. 826–831, May 2010
8. Aikema, D., Kiddle, C., Simmonds, R.: Energy-cost-aware scheduling of HPC workloads. In: 1st International Workshop on Sustainable Internet and Internet for Sustainability, June 2011
9. Miller, R.: Google embraces thermal storage in Taiwan, April 2012. http://www.datacentreknowledge.com/archives/2012/04/03/google-embraces-thermal-storage-in-taiwan/
10. http://community.opennebula.org/ecosystem:green_cloud_scheduler
11. Yuce, B., Packianather, M.S., Mastrocinque, E., Pham, D.T., Lambiase, A.: Honey bees inspired optimization method: the bees algorithm. Insects **4**, 646–662 (2013). http://dx.doi.org/10.3390/insects4040646
12. Babu, D., Krishna, P.V.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl. Soft Comput. **13**(5), 2292–2303 (2013). http://dx.doi.org/10.1016/j.asoc.2013.01.025
13. EU FP7 Cooperate, Control and Optimization for Energy Positive Neighbourhoods. http://www.cooperate-fp7.eu/
14. Salomie, I., Cioara, T., Anghel, I., Moldovan, D., Copil, G., Plebani, P.: An energy aware context model for green it service centers. In: Maximilien, E., Rossi, G., Yuan, S.-T., Ludwig, H., Fantinato, M. (eds.) ICSOC 2010. LNCS, vol. 6568, pp. 169–180. Springer, Heidelberg (2011)
15. Modelica and the Modelica Association – Modelica Association. https://www.modelica.org/
16. Stoyanova, I., Matthes, P., Harb, H., Molitor, C., Marin, M., Streblow, R., Monti, A., Muller, D.: Challenges in modeling a multi-energy system at city quarter level Complexity in Engineering (COMPENG), pp. 1–5 (2012)
17. Molitor, C., et al.: Mutliphysics test bed for renewable energy systems in smart homes. IEEE Trans. Industr. Electron. **60**(3), 1235–1248 (2013)

Analysis of the Influence of Application Deployment on Energy Consumption

Marco Gribaudo, Thi Thao Nguyen Ho, Barbara Pernici(✉), and Giuseppe Serazzi

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano,
Piazza L. da Vinci, 32, 20133 Milan, Italy
{marco.gribaudo,thithao.ho,barbara.pernici,giuseppe.serazzi}@polimi.it

Abstract. Energy efficiency for data centers has been recently an active research field. Several efforts have been made at the infrastructure and application levels to achieve energy efficiency and reduction of CO_2 emissions. In this paper we approach the problem of application deployment to evaluate its impact on the energy consumption of applications at runtime. We use queuing networks to model different deployment configurations and to perform quantitative analysis to predict application performance and energy consumption. The results are validated against experimental data to confirm the correctness of the models when used for predictions. Comparisons between different configurations in terms of performance and energy consumption are made to suggest the optimal configuration to deploy applications on cloud environments.

Keywords: Green ICT · Energy efficiency · Performance evaluation · Cloud computing · Application deployment

1 Introduction

Cloud computing has enabled new paradigms for computing and providing services. On one side, it offers new business solutions that can reduce costs and improve business agility; on another side, researches have shown that cloud computing is the dominant contributor of carbon footprint in Information and Communication Technology (ICT) [1] due to the rapidly increasing power consumption of data centers. This triggers the need of investigating energy-efficiency solutions to design, implement and deploy applications in cloud environments.

Buyya et al. [2] have performed an analysis to understand the main sources of energy consumption in clouds and solutions to address them. According to the analysis, the application profile, including response time, CPU utilization and memory usage, has a significant impact on energy consumption. This motivates investigation on methods for design and implement applications to achieve energy efficiency. Other elements that contribute significantly to energy consumption are resources allocation and provisioning. Indeed, resources are often over provisioned to meet required Service Level Agreements (SLAs).

S. Klingert et al. (Eds.): E^2DC 2014, LNCS 8945, pp. 87–101, 2015.
DOI: 10.1007/978-3-319-15786-3_6

Network devices, used to carry out data communication over Internet, add non-negligible components to energy consumption. The last important element of this analysis is the data centers with their large scale cooling infrastructures and their impact on the environment.

Different solutions have been proposed and implemented to achieve energy efficiency based on identified dominant energy-consumed sources [2]. The key drivers for achieving this are (1) Dynamic Provisioning: resources are allocated dynamically according to runtime demand; (2) Multi-tenancy: the same infrastructure and software are used to serve multiple companies resulting in minimization of extra infrastructure; (3) Server Utilization: maximizes server usage in order to reduce numbers of active servers; (4) Data Center Efficiency: focuses on advanced power management and cooling systems to improve Power Usage Effectiveness (PUE).

In Data Centers, research efforts have been put at both hardware and software levels. According to [6], four main techniques have been employed in Data Centers: (1) Dynamic Voltage and Frequency Scaling (DVFS) to dynamically change electrical voltage and CPU frequency with respect to changing workload; (2) Resource throttling: to maximize resource utilization; (3) Dynamic Component Deactivation (DCD): to activate/deactivate resources on demand; (4) Workload consolidation: to dynamically allocate the workload, either incoming requests or virtual machines, to a minimal amount of physical resources while satisfying SLAs. Moreover, the main part of power consumed by a server is drawn by the CPU, followed by memory and by the losses due to the power supply inefficiency [6]. Therefore the goal of the energy-aware consolidation is to keep servers well utilized, while avoiding the performance degradation due to high utilization.

For the purpose of our study, here we briefly discuss the state of the art of energy efficiency at application level, in particular for Information Systems (IS) in cloud environments. Vitali and Pernici [4] have performed a survey on energy efficiency in IS. According to this survey, the majority of work about energy efficiency has been given to physical resource management, including provisioning virtual machines (VMs), dynamic workload placement, scheduling cloud instances. Some studies have focused on the design of processes to obtain greener solutions. To this regard, re-engineering process life cycle to obtain green solutions is required and composed by four relevant areas [7]: (1) Process Design, (2) Process Measuring, (3) Process Improvement and Change, (4) Process Implementation. During process design and implementation, environmental constraints have to be considered.

However, to the best of our knowledge, there is no work that systematically study the influence on energy efficiency of applications deployment. While a layered approach to consider energy efficiency has been proposed to manage resources associated to VMs according to application requirements [8], the deployment configuration of applications on VMs and its impact on energy consumption is still an open issue. According to the study of Mayo et al. [3], the infrastructure and configuration to deploy applications have a significant effect on energy consumption.

The goal of the present work is to investigate different ways to deploy an application in cloud environments and to analyze simultaneously the energy consumption together with the system performance. The results obtained can be used by services providers to choose the best deployment for the applications, considering their profiles and requirements in terms of response time and energy consumption.

Our approach uses queueing networks to model different deployments of an application in cloud infrastructures. Specifically, the models help to easily analyze several system performance indices and to estimate energy consumption for each deployment configuration. The results are validated against experimental data to check the correctness of the models. Comparisons between different deployment configurations are made to select the best one, considering both system performance and energy consumption. We focus on the problem of comparing different deployment configurations for the execution of an application, considering different deployments on VMs with homogeneous configurations, while the more general case of selecting the most appropriate configuration of resources in VMs for an application is not examined in this paper. In particular, we analyze the behavior of applications in High Performance Computing (HPC) domain, characterized by separate data loading and processing phases.

The rest of this paper is organized as follows: Sect. 2 introduces our case study, Sect. 3 presents model analysis, Sect. 4 describes the power model used in our work, Sect. 5 provides validation of the results obtained from the models, Sect. 6 provides an analysis of different configurations based on the models to support application deployment decisions; finally, Sect. 7 addresses future research directions.

2 The ECO$_2$Clouds Project and the Eels Application

In this section, we introduce the ECO$_2$Clouds project, which provides the experimental basis for our work, and the case study used in our experimental analysis.

2.1 ECO$_2$Clouds

ECO$_2$Clouds is a European project[1] studying ways of reducing the environmental impact of applications in a federated cloud infrastructure, incorporating ecological concerns (such as energy efficiency and CO$_2$ footprint) as key design parameters for cloud infrastructure and application deployment strategies. The aim of the ECO$_2$Clouds project is to develop an energy efficient solution for the deployment of workloads on cloud infrastructures. To achieve this goal, the project establishes a set of key metrics (eco-metrics) to expose energy consumption of applications as well as of cloud infrastructures. The project has developed a scheduler that places workloads on the Cloud with the aim to achieve optimal performance within agreed service level parameters, while keeping the energy usage and environmental impact as low as possible [12].

[1] http://eco2clouds.eu.

BonFIRE[2] is the cloud infrastructure used in ECO$_2$Clouds. The BonFIRE platform delivers a robust, reliable and sustainable facility for large scale experimentally-driven cloud research. The solutions resulted from ECO$_2$Clouds project are validated on BonFIRE, monitoring and collecting ecometrics and supporting ecofriendly scheduling of VMs and management of applications. The work in this paper, also in the scope of ECO$_2$Clouds project, studies in particular how the given ecometrics can be used to support an optimal deployment, taking into account application-level requirements specified in the execution profile of the application.

2.2 The Case Study

The subject of this case study is the oceanic migration of European eel larvae, aiming to understand the response of fish populations to anthropogenic pressures on marine ecosystems [5]. The model built to analyze eels trajectories, consisting of three main steps, i.e., Calibration, Simulation/Forecast, Data aggregation and analysis, requires significant computational effort in terms of CPU processing and management of large datasets.

The Eels application involves two main phases: Data Loading, the initial step of the execution, and Data Processing, that requires high CPU computation. At the beginning of the execution, an Eels instance will load the data specified by the input parameters to the machine that will perform the analysis. In our case study, the data loading usually takes 3 min to complete. After the data are ready, computational machine performs the processing phase which usually requires 30 min and it is computational intensive. The storage contains oceanographic data and, in our setup, is shared among multiple Eels application instances. Experiments are based on the request to execute multiple instances of the Eels application.

3 Different Deployment Configurations and Models

3.1 Application Deployment Configurations

Deploying applications on cloud environments is supported by means of virtualization of physical resources, providing users an abstract view of physical infrastructures. Thus, applications actually run on virtual machines created inside the physical servers. Users are provided a certain degree of flexibility to select a VM's configuration (e.g., number of CPUs, memory, size), and how they want to run their applications (e.g., concurrently, sequentially). Since the Eels application shares its execution pattern (i.e., data loading phase and CPU intensive computation phase) with many other HPC applications, we will analyze different possible deployment strategies for its execution. Given N application instances, we want to find the best configuration, in terms of number of VMs needed to execute the application instances, execution policies (e.g., parallel or

[2] bonfire-project.eu.

sequential execution), storage access strategies (e.g., synchronous or asynchronous), such that the best results concerning execution time and energy consumption are obtained. We will analyze the impact of different execution strategies with one or more virtual machines on response time, resources utilization, and energy consumption of the same configuration. In this specific scenario, we investigate five different deployment configurations (listed below) that can be used to execute the Eels application. Each one of them is associated with a specific scenario that the users might encounter. The method proposed in this paper can also be applied to other applications with a similar execution pattern.

Configuration 1 - Synchronous Parallel Execution. In this configuration, several users execute the application to analyze the Eels migration behaviors (Fig. 1). To minimize the users response time, several application instances, one for each user, are executed in parallel. A separated VM is assigned to each application instance; and all VMs are homogeneous and synchronized in accessing the storage. While this configuration promises a potential shortest response time, it might exhibit a risk of resource contention since the storage can become the bottleneck of the system.

Configuration 2 - Asynchronous Parallel Execution. This configuration is slightly different from the previous one with respect to the storage accesses (Fig. 2). In this case, to avoid potential resource contention in data loading, storage accesses have been scheduled with a mean delay of 3 min, the required time to complete a data loading phase. The remaining part of the setup is similar to Configuration 1. At the first sight, this configuration might lead to longer response time with respect to Configuration 1 due to the delay. However, by avoiding resource contention, it might result in shorter execution times when the number of VMs is large.

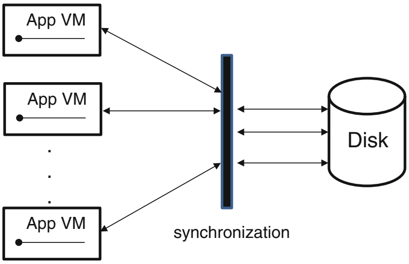


Fig. 1. Configuration 1: synchronous parallel execution

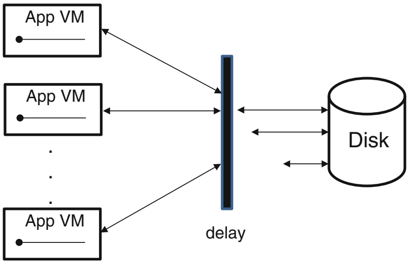


Fig. 2. Configuration 2: asynchronous parallel execution

Configuration 3 - Sequential Execution. This configuration describes the situation in which a user executes multiple instances of the application to analyze the same data set several times, possibly with different parameters (Fig. 3). The time required to analyze the data is not a critical constraint while the minimum

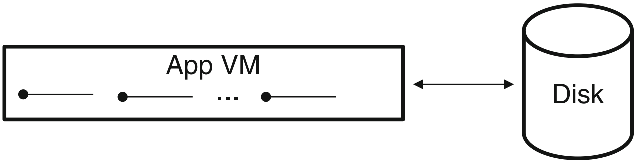


Fig. 3. Configuration 3 - sequential execution

amount of resources used it is. Therefore, multiple application instances will be executed sequentially on the same VM, the accesses to the storage are sequential.

Configuration 4 - Synchronous Parallel Execution with Minimal Resources. This configuration (Fig. 4) is another deployment alternative to Configuration 1 that considers only the minimum amount of computational resources, i.e., one VM. Multiple application instances are deployed on the same VM, executed in parallel and storage accesses are synchronized. This configuration might result in longer response time due to the higher workload assigned to computational resources (e.g., the application VMs). However, it will be interesting to compare its results with the ones of other configurations, analyzing the tradeoff of system response time and energy consumption, and evaluating the benefits in terms of energy consumption due to the limits of the computational resources.

Configuration 5 - Asynchronous Parallel Execution with Minimal Resources. This deployment is alternative to Configuration 4, where we consider a delay for each application instance when accessing the storage (Fig. 5).

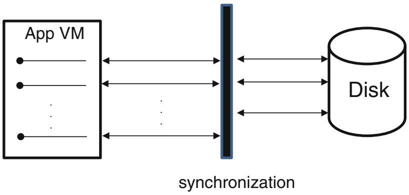


Fig. 4. Configuration 4 - synchronous parallel execution with minimum resources

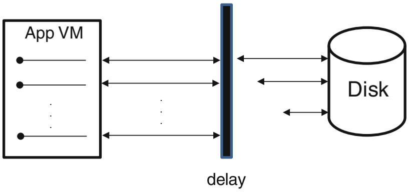


Fig. 5. Configuration 5 - asynchronous parallel execution with minimum resources

3.2 The Models Implemented

Given five possible configurations of the Eels application, in this section we present the models that capture their characteristics and we use them to analyze the performance indices and the energy consumption. We use JMT tools [11] to model the configurations.

Model 1: Synchronous Parallel Execution. Model 1, shown in Fig. 6, refers to Configuration 1 where the disk is modelled as a queue station, serving one request at a time. The requests arriving while the disk is busy will wait in queue. The VM hosting the application instance is modelled as a delay station (named Application in the figure) with no queue of requests since each VM is dedicated to the execution of one application instance. The synchronization is performed by the Fork and Join stations. The model captures the execution sequence: the data loading phase is performed first, followed by the processing phase that involves high CPU computation.

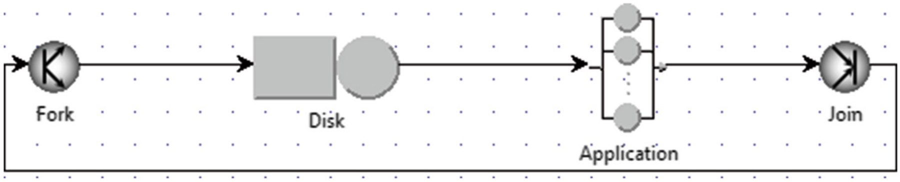


Fig. 6. Model 1: synchronous parallel execution

Model 2: Asynchronous Parallel Execution. Model 2, shown in Fig. 7, is associated to Configuration 2, with the only difference in the delay between the accesses to the disk. To model the delay, we add additional queue stations, each represents one delay. Hence, as a function of the application VMs in the system, the number of delay stations will be updated accordingly. Precisely, the number of delay stations is smaller than the number of application VMs of one unit.

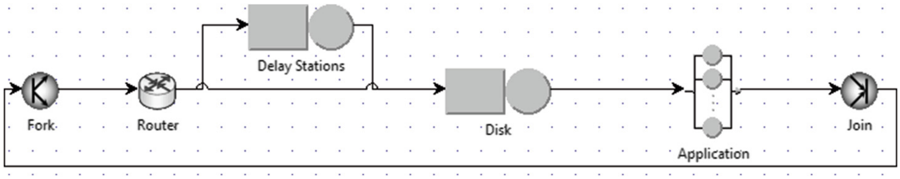


Fig. 7. Model 2: asynchronous parallel execution

Model 3: Sequential Execution. Model 3 refers to Configuration 3, simulating the sequential execution of multiple application instances. The VM hosting application instances is modelled by a queue station because multiple application instances are deployed on the same VM, therefore potentially leads to competing of resources usage. Moreover, due to the constraint of sequential execution, only one request can be present in the disk and in the application VM at a time. This constraint is satisfied by adding a finite capacity region, e.g. FCRegion0. Figure 8 shows this model.

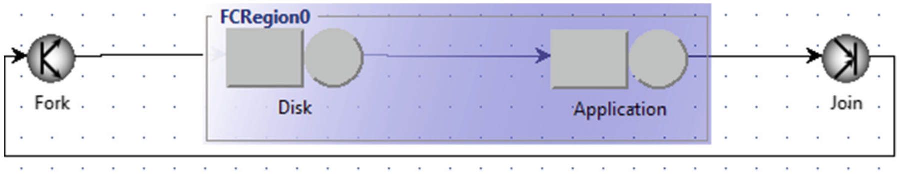


Fig. 8. Model 3: sequential execution

Model 4: Synchronous Parallel Execution with Minimal Resources. Model 4, shown in Fig. 9, is associated to Configuration 4. It simulates synchronous parallel execution of multiple application instances deployed on the same application VM. Application VM is modelled as a queue station, with no limitation in the number of requests in the disk and the application VM. The Fork and Join stations perform the synchronization.

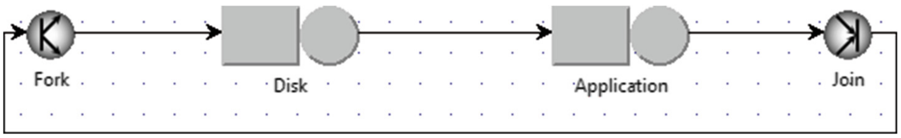


Fig. 9. Model 4: synchronous parallel execution with minimal resources

Model 5: Asynchronous Parallel Execution with Minimal Resources. Model 5, shown in Fig. 10, is associated to Configuration 5. It simulates asynchronous parallel execution of application instances hosted on the same application VM. Application VM is modelled as a queue station, while the delay is performed by introducing extra delay stations, depending on the number of delays needed.

4 Power Model

To compute the power consumption of the different configurations, it is necessary to adopt a power model able to predict the actual value of the consumption based on some runtime characteristics. Fan et al. [9] describe a linear

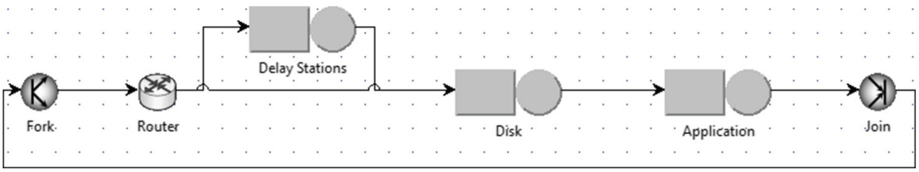


Fig. 10. Model 5: asynchronous parallel execution with minimum resources

relationship between the CPU utilization and the total power consumption of a server. According to their model, the power consumption of a server grows linearly with the CPU utilization. The initial value is P_{idle}, i.e., the power consumption in the idle state, and the final value is P_{busy}, i.e., the power consumed at 100 % of utilization. Equation 1 describes this relationship.

$$P(u) = P_{\text{idle}} + (P_{\text{busy}} - P_{\text{idle}}) \times U \tag{1}$$

where U is actual value of CPU utilization.

While Eq. 1 computes the power consumption of a VM considering only one physical host, in our work, we used Eq. 2 to estimate the power consumption when there are more than one VM involved and deployed on multiple physical hosts, assuming a single physical host allows to deploy up to a maximum number of VMs:

$$P(u) = P_{\text{idle}} \times ceil(N \div MaxVM) + (P_{\text{busy}} - P_{\text{idle}}) \times U \times N \tag{2}$$

where N is the number of VMs used in the experiment, *MaxVM* is the maximum number of VMs that can be deployed on a single physical host.

This model allows us to estimate the power consumption even in cases where multiple physical servers are used. With Eq. 2 we estimate the power consumption for each configuration, using the P_{idle} and P_{busy} measured from our BonFIRE infrastructure, and the value of U obtained from our models. To estimate the total energy consumption, we use Eq. 3:

$$E = P(u) \times R \tag{3}$$

where E is the estimated total energy consumption, P is the estimated power consumption computed using Eq. 2, and R is the system response time given by the models.

5 Validation

In this section, we validate the models, using experimental data provided by the BonFIRE and ECO_2Clouds platform. We explain how to obtain power measurement and the values of P_{idle} and P_{busy} in order to compute power consumption and derive energy consumption.

The ECO_2Clouds platform uses Zabbix to monitor its running environment from low-level infrastructure layer up to high-level application layer [10,12]. Physical hosts in BonFIRE are equipped with a power distribution unit (PDU), an external hardware power device that distributes electric power to the hosts, and monitors power consumption of each host. Given the data provided by the PDU, Zabbix monitoring system performs sampling each minute to sample power value at a time instant, and stores it in a monitoring database. Figure 11 shows the schematic representation of the monitoring environment in ECO_2Clouds.

To perform experiments on ECO_2Clouds, we use dedicated physical hosts having 2 × *QuadCore* Intel Xeon @ 2.83 GHz, 32 GB RAM as configuration.

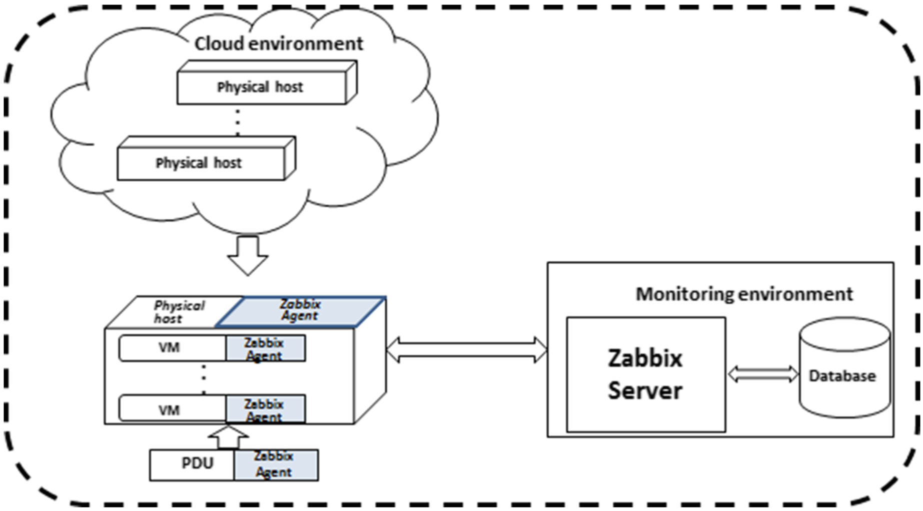


Fig. 11. Collecting data via monitoring environment

Each physical host is able to deploy up to a maximum of 6 VMs. Experiments involving more than 6 VMs will have to allocate more than one physical host with the same configuration. Since the PDU provides power consumption only at the host level, our hypothesis is that power measured from physical host is accounted for all the VMs running on the host. In specific, if there is one running VM, the measured power is accounted completely to that VM; in case of multiple VMs on the host, the measured power is accounted to all the VMs. In this work, we do not study how to distribute the measured power to specific VMs considering their performance and workload. Indeed, during the experiments only the VMs related to our application were running on the considered hosts, so the total measured power can be accounted globally to all of them.

To measure P_{idle} and P_{busy}, we created one VM on a host and deployed the Eels application on it. P_{idle} value is computed as the average of power samples over an idle period (e.g., 10 min) when the VM does not execute the application. P_{busy} is computed as the average of power samples over a peak period (e.g., CPU load is 100 %) when the application is being executed. The energy consumption of one experiment is computed by integrating power samples getting from the PDU over the experiment's lifetime.

Finally, we choose to validate Eq. 2 for two different configurations, Configuration 1 with synchronous parallel execution and Configuration 4 with synchronous parallel execution with minimal resources. These two configurations in fact are the representative of different strategies to deploy applications. Configuration 2 shares the same setup with Configuration 1 with the only difference in the delay (the same considerations applies also for Configurations 5 and 4). Configuration 3, instead, refers to a deployment in which one application instance runs on one VM and is then repeated several times.

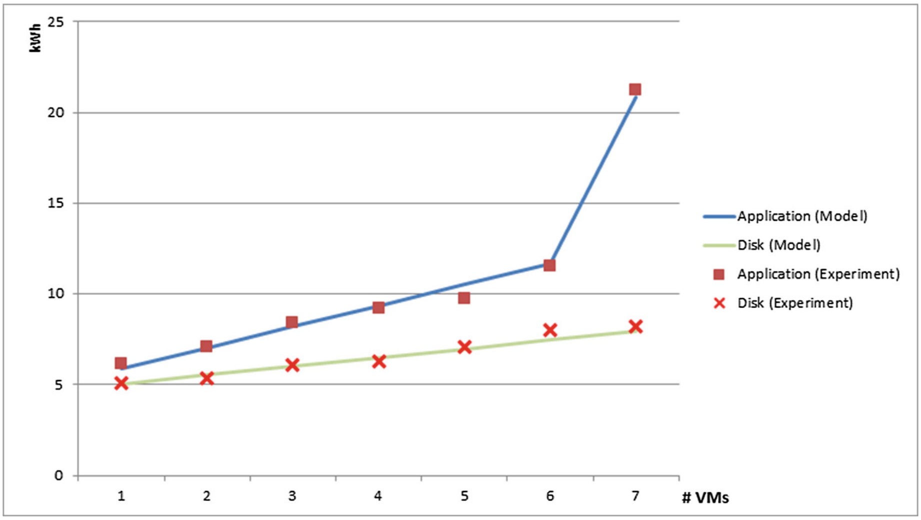


Fig. 12. Application and disk energy consumption for Configuration 1

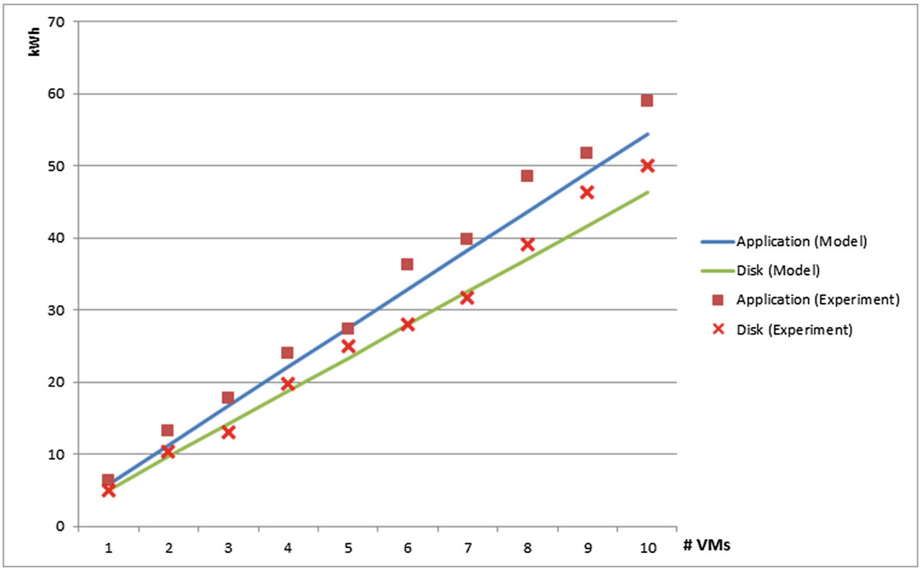


Fig. 13. Application and disk energy consumption for Configuration 4

We compare scenarios where the application instances vary from 1 up to 6, then we analyze particular values of application instances (e.g. 7, 8, etc.) where numbers of instances require to use multiple physical hosts to deploy the VMs. The comparison results, if then confirm the correctness of the models, will lead to their usage to predict unforseen cases.

Figures 12 and 13 show for Configuration 1 and Configuration 4 respectively the energy consumption of the application VM and the disk. The comparison between the model results and the experimental data is provided. As it can be seen, the total energy consumption of the application VM and the disk measured from real system are very close to the values predicted by the model. Moreover, it is shown that the application VM consumes more energy than the disk, and the increasing speed of energy consumed by application grows faster than the disk.

6 Deployment Configuration Analysis

In this section, we exploit the five different models to extract useful insights of system performance and energy consumption. We want to compare five configurations in terms of system response time and energy consumption, with a number of application VMs ranging from 1 to 30. This comparison will help us to find the dominant and dominated configurations with respect to energy consumption and system performance, i.e., the response time. Figures 14 and 15 show the comparison.

The figures unveil a linear increasing in energy consumption with respect to system response time in Configuration 3, 4 and 5; and a non-linear relationship of energy consumption in Configuration 1 and 2. The ladder step behavior in energy consumption of Configuration 1 and 2 is due to the ceil function ($ceil(N \div MaxVM)$) in Eq. 2, which is related to the number of physical hosts required to host N VMs.

The figures also show that Configuration 3 is slightly dominated by the others, considering both energy consumption and system response time. Moreover, the

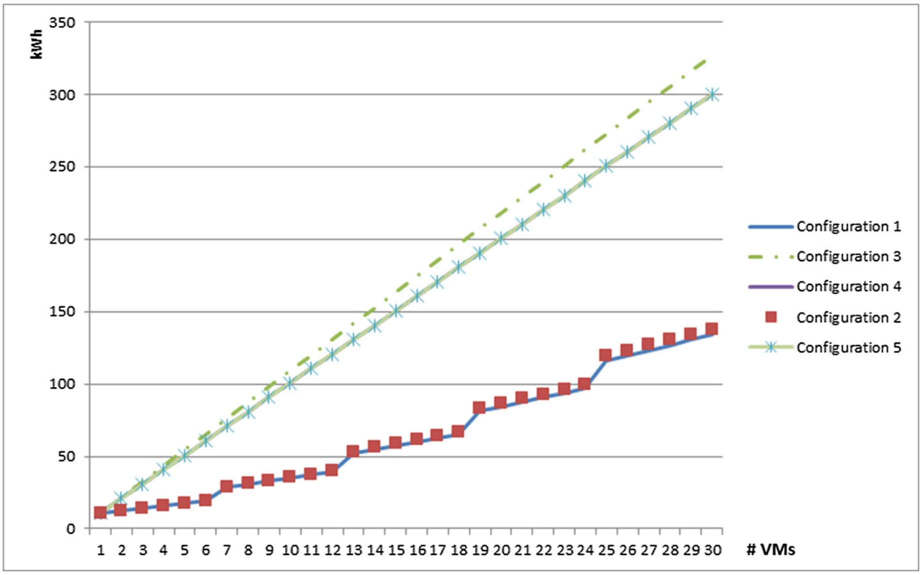


Fig. 14. Energy consumption comparison of different configurations

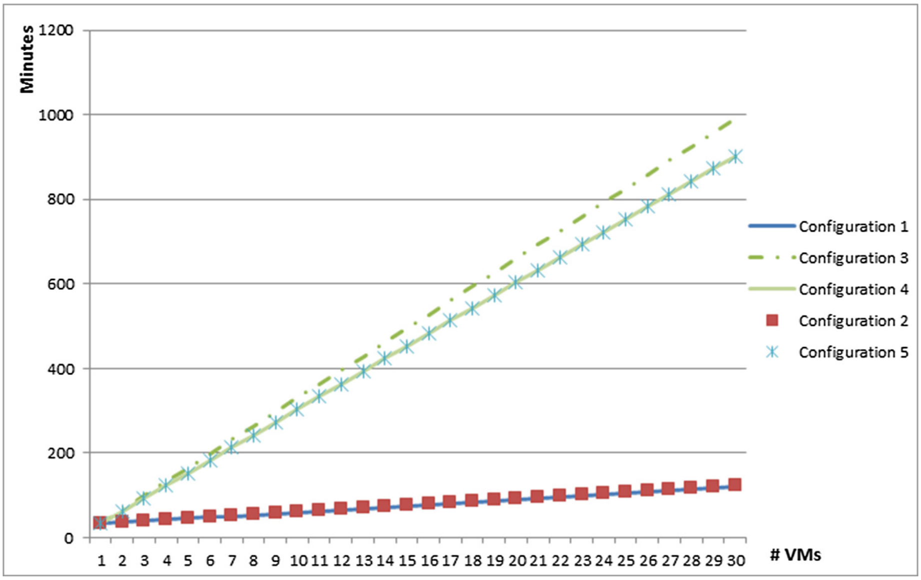


Fig. 15. System response time comparison of different configurations

energy consumption and system response time of Configuration 1 and 2 are identical, similarly for Configuration 4 and 5. This phenomenon can be explained by the way the delay is applied. In Configurations 1 and 4, the disk accesses are synchronized. These setups give the disk itself responsibility to schedule incoming requests; in fact, the delay is performed automatically by the disk. While in Configurations 2 and 5, the delay is performed manually by adding delay stations. So, for this type of problems, asynchronous disk accesses do not seem to provide a case for a better deployment configuration.

As a conclusion, assuming an unlimited number of resources is available, Configuration 1, with a single VM for each application instance, consumes less energy than using a single VM for all application instances, either executed in parallel or serial. Hence, Configuration 1 is the optimal deployment for this type of application profile. However, further analysis is needed to extend the results in cases in which congestion might occur at a certain number of application instances or in which data dependencies are present.

7 Conclusions and Future Work

In this paper, we analyze the influence of application deployment on energy consumption in cloud environments. We build various models to simulate different possible configurations for deploying the application, and we compare their results with experimental data to validate them. We perform comparisons among configurations considering total energy consumption and system response time, to suggest optimal configuration with our infrastructures and setup. The models can be applied, varying their parameters, to applications with a similar profile to the one which has been analyzed: the instances of the applications access the same storage unit, and are characterized by a data loading phase and a computing phase. The durations of the data loading phase and of the computing phase are stable and are the parameters for the analysis. The model also allows the application user to evaluate the effects of setting requirements on response time or on energy consumption of applications, therefore enabling the user to set the most appropriate constraints according to his needs.

In future work, we are also planning to extend the analysis to other application profile patterns, considering the behaviour of other types of applications, such as web services and providing models for their analysis. The result will be the basis for providing adaptive applications on cloud infrastructures, able to react to changing parameters, such as an increase of service time due to external factors, such as temporary shortage of resources.

Acknowledgements. This work has been partly funded by the European Commission's 7th Framework Program (contract numbers 318048) within the ECO$_2$Clouds project (http://eco2clouds.eu/). This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work. We thank University of Stuttgart (USTUTT-HLRS) for the support in our work.

References

1. Global e-Sustainability Initiative (GeSI). SMART 2020: Enabling the low carbon economy in the information age (2008)
2. Garg, S.K., Buyya, R.: Green cloud computing and environmental sustainability. In: Murugesan, S., Gangadharan, G. (eds.) Harnessing Green IT: Principles and Practices, pp. 315–340. Wiley Press, UK (2012)
3. Mayo, R.N., Ranganathan P.: Energy consumption in mobile devices: why future systems need requirements-aware energy scale-down. In: Proceedings of 3rd International Workshop on Power-Aware Computer Systems, San Diego, CA, USA (2005)
4. Vitali, M., Pernici, B.: A survey on energy efficiency in information systems. J. Coop. Inf. Syst. **23**, 38 pp. (2014). http://www.worldscientific.com/doi/abs/10.1142/S0218843014500014
5. Melià, P., Schiavina, M., Gatto, M., Bonaventura, L., Masina, S., Casagrande, R.: Integrating field data into individual-based models of the migration of European Eel Larvae. Mar. Ecol. Prog. Ser. **487**, 135–149 (2013)
6. Beloglazov, A., Buyya, R., Lee, Y.C., Zomaya, A.: Taxonomy and survey of energy-efficient data centers and cloud computing systems. In: Zelkowitz, M.V. (ed.) Advances in Computers, vol. 82, pp. 42–111. Elsevier, Amsterdam (2011)
7. Nowak, A., Leymann, F., Schleicher, D., Schumm, D., Wagner, S.: Green business process patterns. In: Proceedings of the 18th Conference on Pattern Languages of Programs, ACM (2011)
8. Song, Y., Sun, Y., Shi, W.: A two-tiered on-demand resource allocation mechanism for VM-based data centers. IEEE Trans. Serv. Comput. **6**(1), 116–129 (2013)

9. Fan, X., Weber, W.-D., Barroso, L.A.: Power provisioning for a warehouse-sized computer. In: Proceedings of the ACM International Symposium on Computer Architecture, San Diego, CA (2007)
10. Cappiello, C., Datre, S., Fugini, M.G., Melià, P., Pernici, B., Plebani, P., Gienger, M., Tenschert, A.: Monitoring and assessing energy consumption and CO2 Emissions in Cloud-based Systems. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC) (2013)
11. Bertoli, M., Casale, G., Serazzi, G.: JMT: performance engineering tools for system modeling. ACM SIGMETRICS Perform. Eval. Rev. **36**(4), 10–15 (2009)
12. Pernici, B., Wajid, U.: Assessment of the environmental impact of applications in federated clouds. In: SmartGreens 2014, Barcelona (2014)

Minimization of Costs and Energy Consumption in a Data Center by a Workload-Based Capacity Management

Georges Da Costa[1], Ariel Oleksiak[2,4](✉), Wojciech Piatek[2],
Jaume Salom[3], and Laura Sisó[3]

[1] IRIT, University of Toulouse, Toulouse, France
georges.da-costa@irit.fr
[2] Poznan Supercomputing and Networking Center, Poznań, Poland
{ariel,piatek}@man.poznan.pl
[3] IREC, Institut de Recerca En Energia de Catalunya, Barcelona, Spain
{jsalom,lsiso}@irec.cat
[4] Institute of Computing Science, Poznan University of Technology, Poznań, Poland

Abstract. In this paper we present an approach to improve power and cooling capacity management in a data center by taking into account knowledge about applications and workloads. We apply power capping techniques and proper cooling infrastructure configuration to achieve savings in energy and costs. To estimate values of a total energy consumption and costs we simulate both IT software/hardware and cooling infrastructure at once using the CoolEmAll SVD Toolkit. We also investigated the use of power capping to adjust data center operation to variable power supply and pricing. By better adjusting cooling infrastructure to specific types of workloads, we were able to find a configuration which resulted in energy, OPEX and CAPEX savings in the range of 4–25 %.

Keywords: Data centers · Energy efficiency · Simulations · Heat-aware · Metrics · OPEX · CAPEX

1 Introduction

The problem of capacity management in data centers is a well known issue, which data center planners and operators must deal with. The problem can be defined as finding such a data center configuration that its space, power and cooling capacity is maximized. In other words, the goal is to put maximal number of servers into a data center subject to its size, electrical infrastructure power limits, and heat dissipation constraints. Usually, this process is based on server power usage nameplates and by getting theoretical peak values from specifications. Unfortunately, these values are often the Power Supply Unit (PSU) maximum capacity so they substantially overestimate actual power loads. Therefore, vendors sometimes deliver calculators that help to obtain estimations closer to

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 102–119, 2015.
DOI: 10.1007/978-3-319-15786-3_7

real values. Still, most of these methods neither take into consideration characteristics of specific applications nor dynamic properties of workloads that are executed in data centers. Some attempts to! apply more advanced power capping to improve efficiency of the whole data center can be found in literature. An alternative method to power capping based on managing distributed UPS energy is presented in [8]. Interesting approach to combine IT workloads, power, cooling and renewable energy was studied in [17] but without use of power capping techniques. In [10] authors propose adaptive power capping for virtualized servers, however they investigate neither the cooling system nor variable power supply. Dynamic power capping to enable data center participation in power markets was proposed in [4] but without detailed cooling consideration, either. To address these issues, we propose modeling and analysis of data center workloads and hardware to identify real power limits that should be met. Based on these limits we present methods to save energy and optimize cooling capacity of a data center including adaptation of limits to power supply and pricing.

To meet this objective we have used the SVD Toolkit developed within the CoolEmAll project [5]. The toolkit enables data center designers and operators to reduce its energy impact by combining the optimization of IT, cooling and workload management. For this purpose, CoolEmAll project investigated in a holistic approach how cooling, heat transfer, IT infrastructure, and application-workloads influence overall cooling- and energy-efficiency of data centers, taking aspects into account that traditionally have been considered separately. SVD Toolkit was used to conduct experiments described in this paper. In particular, most simulations were done using one of the main tools of the SVD Toolkit - the Data Center Workload and Resource Management Simulator (DCworms) [9].

Using the CoolEmAll SVD Toolkit we demonstrate how to improve capacity management by taking into account knowledge about applications and workloads as well as by using power capping techniques and proper cooling infrastructure configuration. To obtain total energy consumption, we simulate both IT software/hardware and cooling infrastructure in parallel. In this way, by better adjusting cooling infrastructure to specific types of workloads, we were able to find a configuration which result in energy savings and even in improvement of CAPEX (Capital Expenditures) without significant workload performance deterioration. Decrease in CAPEX was achieved by the selection of smaller chiller which fits the foreseen workloads better. Energy savings were achieved by increase of server inlet temperature. This was possible by limiting power used by particular racks and by compliance to the latest ASHRAE recommendations. Finally, we applied power capping to adjust data center operation to! variable power supply and achieved additional OPEX (Operating Expenditures) savings. The structure of this paper is as follows. In Sect. 2 we present a model of a data center including models of IT hardware, cooling, workloads and applications. This section also contains definitions of metrics used for the assessment of data center configurations studied in this paper. We analyze workloads along with their impact of on energy-efficiency in Sect. 3. Based on this analysis we define power limits which allow reducing energy consumption and costs of a data center operation. Section 4 contains results of the data center optimization using power capping methods and decisions about cooling infrastructure deployment and configuration. Section 5 concludes the paper.

2 Data Center Model

2.1 Modeling Workloads

In terms of workload management, workload items are defined as jobs that are submitted by users [19]. Thus, modeling of workloads consists in providing information about structure, resource requirements, relationships and time intervals of jobs arriving to the management and scheduling system. Primary properties of a workload include:

- number of jobs to be scheduled
- jobs arrival rate, expressed as a time interval between successive jobs
- reference to an application profile describing behaviour of particular job on the hardware (resource requirements and execution times)

The last one is described in the next section in more detail.

Having these dependencies established, it is possible to express the impact of particular workload on the hardware layer. For now, one of the main and commonly used format that provides unitary description of workloads models and logs obtained from real systems is Standard Workload Format (SWF) [22].

As mentioned, workload profiles may be obtained by monitoring real systems or generated synthetically. The main aim of synthetic workloads is to reflect the behavior of real observed workloads and to characterize them at the desired level of detail. Moreover, they are also commonly adopted to evaluate the system performance for the modified or completely theoretical workload models. Usage of synthetic workloads and their comparison to the real ones have been the subject of research for many years [12].

2.2 Modeling Applications

Concerning application-led management a maximum feedback is needed from the applications from different point of view. The focus is on power-, energy- and thermal-impact of decisions on the system. Still it is impossible to put a wattmeter on an application. In order to obtain the same kind of information, we monitored applications to evaluate their resource consumption at each second. At each of these points, using system values and hardware performance counters, processor, memory and I/O resources are monitored. Using these information and models we produce for each of these timestamps an evaluation of the power consumption [6]. Each of those values are monitored, computed and stored in real-time in a database for future use.

In the system, an application is then described as the resources it uses on a particular hardware. Each application can be run on different hardware or configuration (frequency for example) and those data are associated with the same application. In case the data for a particular application on a particular hardware is not available, a translation tool is used to evaluate the behavior of the application using its behavior on a different hardware. First, it models the resource bottleneck of an application using the monitored resource consumption on a particular hardware. Using the target hardware specification, it evaluates the resource bottleneck and thus overall resource consumption on that hardware.

Using the monitored data, we create a description of applications based on their phases following the same methodology as in [11]. A phase is defined as a duration when resources consumption are stable. As an example, Fig. 1 show the profile of a Fast Fourier Transform algorithm with its phases. Using the XML files describing exact application behavior and resource consumption, SVD toolkit can evaluate precisely the impact of its decisions.

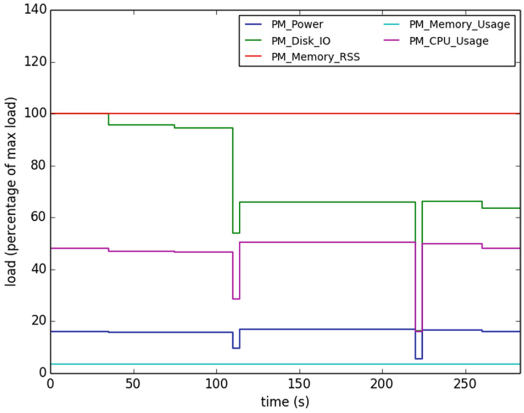


Fig. 1. Profile of the benchmark test3d: 3D real-to-complex FFT routine

2.3 Modeling Servers

In the scope of CoolEmAll, data center server room is composed by a number of racks. Each rack consists of a set of node groups, which are then responsible for hosting a collection of nodes. Node groups are defined by a means of chassis that models the placement of nodes within the node group as well as mounted fans. The main component of the node is a processor with assigned number of cores and computing capability (expressed by a clock speed). Moreover, each processor comes with its power and computing profile, described by the means of C-States and P-States defining operating states with corresponding power usage values for different utilization levels. Node definition is supplemented by a description of memory and network. Rack represents a standardized enclosure for carrying server and power supply modules. Power profiles of IT infrastructure are the basis for calculating the power consumption of particular resources.

The following equations show how the power usage for different resource levels is estimated.

$$P_{cpu}(P_x, load) = P_{cpu}(P_x, 0) + load * (P_{cpu}(P_x, 100) - P_{cpu}(P_x, 0))/100 \tag{1}$$

where $P_{cpu}(P_x, load)$ is a power consumed by a processor operating in a given P-State P_x and utilized in a level denoted by *load*. $P_{cpu}(P_x, 0)$ and $P_{cpu}(P_x, 100)$ expresses an idle and fully loaded processor working in a given P-State, respectively (these constant values are part of the processor power profile providing power consumptions levels for all available frequencies).

$$P_{node} = \sum_{i=1}^{n} P_{cpu_i} + P_{mem} + P_{net} \tag{2}$$

where P_{node} is a power consumed by a node, n is the number of processors assigned to a node, P_{mem} is a power drawn by a memory, while P_{net} by a network.

$$P_{node\_group} = \sum_{i=1}^{m} P_{node_i} + \sum_{j=1}^{k} P_{fan_j} \tag{3}$$

where P_{node_group} is a power consumed by a node group, m is the number of nodes placed in a node group, k is the number of fans mounted within it and P_{fan_j} is a power used by particular fan j.

$$P_{rack} = (\sum_{i=1}^{l} P_{node\_group_i})/\eta_{psu} \tag{4}$$

where P_{rack} is a power consumed by a rack, l defines the number of carried node groups and η_{psu} is efficiency of a power supply unit.

Finally, each component is accompanied with its carbon emissions and electricity costs. Apart from IT equipment, data center server room is composed by a cooling devices, which are the subject of next subsection.

2.4 Cooling Models

The SVD CoolEmAll toolkit integrates models to calculate the power associated to cooling equipment and other electric facilities required in data center to fulfill its mission related with IT services. The cooling model provided consists of a simple data center where central fan and air-water coil cools the IT equipment and other related loads (PDU, UPS and lighting). A chiller placed outside provides cooling water to the coil and dissipates the exhausted heat from the room to the atmosphere by a dry-cooler (Fig. 2 shows details). The power model adds the consumption of IT, fans, chiller, PDU and lighting. Other electric components of a data center as back-up generator or transformer are excluded from the present model.

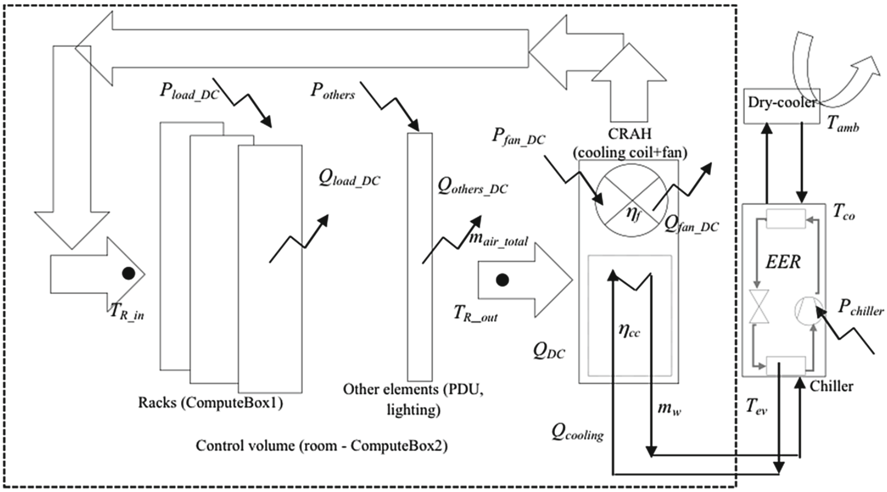


Fig. 2. Model of cooling and power facilities of a data center

The following model description is based on a single time-stamp where Q is referred to heat dissipated and P to power consumption. The time variability is indicated by (t). This model has been constructed based on basic thermodynamic equations of conservation of mass and energy. The total power consumption of a data center (P_{DC}) will be calculated with Eq. 5, where P_{load_DC} is the power used by IT components, $P_{chiller}$ is the consumption of the chiller, P_{fans_DC} is the consumption of fans in data center and P_{others} is the consumption associated to PDU and lighting:

$$P_{DC}(t) = P_{load\_DC}(t) + P_{chiller}(t) + P_{fans\_DC}(t) + P_{others}(t) \tag{5}$$

The total thermal load (Q_{DC}) is the sum of the heat associated to IT load (Q_{load_DC}), the heat from other loads, as PDU and lighting (Q_{others_DC}) and the heat from fans distributing air inside a data center room (Q_{fan_DC}).

$$Q_{DC}(t) = Q_{load\_DC}(t) + Q_{fan\_DC}(t) + Q_{others\_DC}(t) \tag{6}$$

The cooling demand that should be covered ($Q_{cooling}$) is the thermal load in data center including the inefficiencies in the air-water coil represented by η_{cc} according Eq. 7. That corresponds to the heat exchanger efficiency of a common CRAH, where heat of the room is transferred to the water flow ($Q_{cooling}$).

$$Q_{cooling}(t) = \frac{Q_{DC}(t)}{\eta_{cc}} \tag{7}$$

The chiller has been modelled with generic profiles based on condenser temperature (T_{co}), evaporator temperature (T_{ev}) and partial load ratio (PLR). Thereby, the model presented here should provide a general method to determine the power consumption of the chiller without knowing the specific characteristics of the chiller provided by a certain manufacturer. As a result, it has been used parametric curves implemented by the Building Certification Code in Spain [1] named $COOL(T_{ev}, T_{co})$ and $CoolPR(T_{ev}, T_{co}, PLR, EER_{rated})$ following certain relations depicted in Eqs. 8 and 9.

$$Q_{cooling\_nom} = Q_{cooling\_rated} \cdot COOL(T_{ev}, T_{co}) \tag{8}$$

$$CoolPR(t) = CoolPR(T_{ev}, T_{co}, PLR, EER_{rated}) = \frac{1}{EER(t)} \tag{9}$$

Partial Load Ratio is the relation between the cooling demand in a certain conditions and the cooling load in nominal conditions ($Q_{cooling\_nom}$) corresponding to the operation of the chiller at the chilled water temperature (T_{ev}) and condenser water temperature (T_{co}) set-up (Eq. 10). At the same time, $Q_{cooling\_nom}$ has relation with the cooling capacity rated ($Q_{cooling\_rated}$) which corresponds to load of the chiller in Standard Conditions (full load; temperature of chilled water leaving the chiller at 7 °C and temperature of condenser water entering the chiller at 30 °C) as stated in Eq. 8.

$$PLR(t) = \frac{Q_{cooling}(t)}{Q_{cooling\_nom}} \tag{10}$$

The relation between the cooling load and the power consumed in the chiller ($P_{chiller}$) is linked by the Energy Efficiency Ratio (EER), that quantifies the cooling provided by the chiller by each unit of power consumed, according Eq. 11. EER_{rated} corresponds to the value of the parameter measured at Standard Conditions defined above.

$$P_{chiller}(t) = \frac{Q_{cooling}(t)}{EER(t)} \tag{11}$$

2.5 Assessment of Data Center Efficiency, Performance, and Costs

Metrics CoolEmAll SVD Toolkit provides a set of metrics divided in the level of granularity of the analysis (node, node-group, rack and data center). The whole group of metrics assesses the resource usage, capacity, energy, heat-aware, green and financial concepts. The total selection of metrics of CoolEmAll are described in public report of the project [14] as well as in some articles [15,16].

Total Energy Consumed: this corresponds to the total energy consumed by the data center in a certain period of time.

Power Usage Effectiveness (PUE): defined by The Green Grid [3] this metric consist of dividing power used by the data center between power used by the IT equipment. The accuracy level of the metrics is related with the point of measurement of IT power, that can be the UPS (Uninterruptible Power Supply Unit), the PDU (Power Distribution Unit) or the IT itself, after PSU (Power Supply Unit). When the measurement is done after the PSU the metric is defined as PUE Level 3.

When the measurement is referred to IT properly, excluding PSU and fans, the metric is named PUE Level 4, according CoolEmAll project proposal [15].

Carbon emissions: this metric is calculated multiplying the total power consumed by carbon emissions factor (CEF). CEF depends on the country power generation mix and power system efficiency. For the approach of this study, 0.34 kg/kWh has been used as average value for the European Union according to [7].

OPEX: it is calculated multiplying the total power consumed by the price of electricity. The price of electricity has been considered as 0.0942 €/kWh for EU-28 as average of 2013 according to [21].

CAPEX: it is the amount of money used to acquire equipment or to improve the useful life of existing facilities.

3 Analysis of Workloads

As mentioned in Sect. 2.1, workloads are characterized by the number of jobs, their arrival rate, resource requirements and execution time of particular applications. The following section contain describes the results of workload simulations performed by the means of Data Center Workload and Resource Management Simulator, which is part of SVD Toolkit.

3.1 Simulation of Diverse Workloads Using DCworms

Resource Characteristics. In our experiments we used a configuration of the real server room. Each server was equipped with a processor belonging to Intel Xeon processors family. The following table (Table 1) summarizes overall characteristics of particular racks.

Table 1. Power characteristics of racks in the server room

Rack name	Number of nodes	Number of processors	Processor type	Min. power usage (idle) [W]	Max. power usage (100 % load) [W]
Rack 1	84	2	Xeon E5-2603	10292	27672
Rack 2	84	2	Xeon E5-2630	12030	30568
Rack 3	84	1	Xeon L5310	4499	11258
Rack 4	84	1	Xeon L5310	4499	11258
Rack 5	84	2	Xeon E5-2603	10292	27672
Rack 6	84	2	Xeon E5-2603	10292	27672
Rack 7	84	2	Xeon E5-2630	12030	30568
Rack 8	56	2	Xeon E5-2630	8020	20379
Sum	644	1120	-	71955	187046

Additionally, server room was equipped with the cooling facilities presented in Table 2.

Finally, the following input parameters were applied to the simulation environment (Table 3).

Table 2. Cooling facilities characteristics

Parameter	Symbol in the equations	Value
Cooling capacity rated	$Q_{cooling\_rated}$	240000 [W]
Energy efficiency ration rated	EER_{rated}	3
Efficiency of cooling coil	η_{cc}	0.95
Data center fans efficiency	η_f	0.6
Temperature difference between T_{ev} and T_{R_in}	ΔT_{hex}	10 °C

Table 3. Input parameters

Parameter	Symbol in the equations	Value
Relation between P_{loadDC} and P_{others}	α	0.2
Inlet temperature	T_{R_in}	18 °C
Outlet temperature	T_{R_out}	33 °C
Pressure drop	Δp	65 J/m^3

Workloads and Application Profiles. In our experiment we evaluated two workloads with different utilization levels what was achieved by the modification of arrival rate (all tasks arrive according to the Poisson distribution) and the number of submitted tasks. The former workload consists of 1280 tasks, while the latter consists of 1760 tasks.

A distribution of applications constituting both workloads is the same in both cases and looks as follows: 20 % - App1, 50 % - App2, 30 % - App3. Their general overview is shown in Table 4. The understanding of the cells content is as follows: number of requested processors, execution time, load level (in [%]).

Table 4. Application characteristics

Processor type	App1	App2	App3
Xeon E5-2630	1, 380, 84	4, 3200, 62.6	6, 3200, 94
Xeon E5-2603	1, 400, 86	4, 3600, 92	-
Xeon L5310	1, 1200, 92	-	-

3.2 Identifying Power Caps

Based on the simulation results obtained for execution of both workloads using Load Balancing policy, we observed two visible increases on utilization criteria, reaching almost 75 % and 95 % in the highest peak for Workload 1 and Workload 2 respectively. High utilization values have direct impact on the power consumption and thus might result in sudden power drawn peaks. Identification of such levels is crucial in terms of avoiding hot spots and decreasing data center costs. Taking into account power consumption ranges for the modeled server room, power consumption distribution obtained during the experiments and the utilization curves we decided to use the following approach to specify the values of power caps. As there occured temporary, but significant load rises and we were not considering the possibility of switching nodes on/off, we wanted to ensure constant computational capabilities for all the servers within particular racks. To this end the power cap level is determined by the total power consumption of the rack, with all the processors fully loaded and working in the highest P-State (with lowest frequency). The following formula can be used to calculate this value (PC) for the given rack j.

$$PC_j = \sum_{i=1}^{n} P_{CPU_i}(P_{h_i}, 100\,\%), \tag{12}$$

where n is the number of processors in a rack, P_{CPU} is the power consumed by the processor working under given utilization level and in the given P-State, P_h refers to the highest P-State (power consumption is lower at higher P-State).

On the other hand, in order not to observe the performance losses (due to frequency downgrading) another threshold is necessary. It aims at setting the power consumption level PU below which the current processor performance state will increase. It is defined by the following equation:

$$PU_j = PC_j \cdot \frac{\sum_{i=1}^{n} P_{CPU_i}(P_{h_i}, 100\,\%)}{\sum_{i=1}^{n} P_{CPU_i}(P_{h_i-1}, 100\,\%)}, \tag{13}$$

where n is the number of processors in a rack, P_{CPU} is the power consumed by the processor working under given utilization level and with the given P-State, P_h and P_{h-1} refer to two highest P-States. As power consumption is lower at higher P-States, thus, PU_j is lower than PC_j.

PU_j allows increasing the current processors performance states at least by one without exceeding the power cap limit (PC_j). Below table introduces boundary values according to the aforementioned approach;

Table 5. Power caps values for the racks in the server room

Rack name	Rack 1	Rack 2	Rack 3	Rack 4	Rack 5	Rack 6	Rack 7	Rack 8	Sum
PC level [W]	20333	21878	8940	8940	20333	20333	21878	14585	137220
PU level [W]	18854	20223	8449	8449	18854	18854	20223	13482	127388

Adjusting Power Limits to Workloads. Having information about historical or predicted workloads it is possible to adjust power caps. For instance, there may exist specific patterns of incoming tasks related to peak hours, time of a day, etc. This knowledge can be applied to identification of optimal power caps.

There are two main requirements that should be taken into consideration while setting the values of power caps. First of all, the use of power capping shouldn't cause significant increase of IT energy consumption for a given workload. Second, the mean completion time of tasks should not go below certain required threshold.

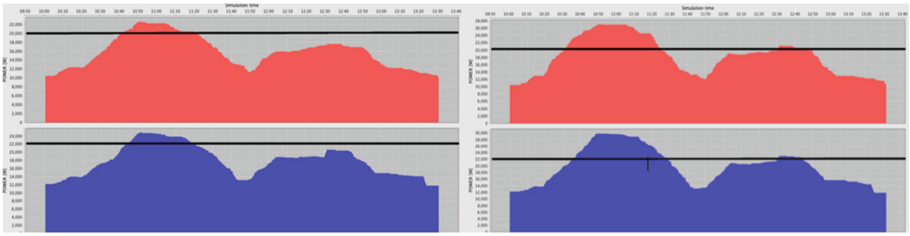


Fig. 3. Power distribution for Workload 1 and Workload 2 for two racks

The first requirement can be defined as follows. Let assume that energy decrease caused by power capping in rack j is denoted as $E_j^{eexcess}$ and given in Eq. 14. This amount of energy can be illustrated by the field above the power cap line in Fig. 3. On the other hand, let denote by $E_j^{reserve}$ the amount of additional energy that can be used in a rack without exceeding the set power cap. This can be seen as a free space below the power cap in Fig. 3 and defined by Eq. 15.

$$E_j^{eexcess} = \int_{t_1}^{t_2} max(0, P_j^{IT}(t) - PC_j)dt \tag{14}$$

$$E_j^{reserve} = \int_{t_1}^{t_2} max(0, PC_j - P_j^{IT}(t))dt \tag{15}$$

Then, the condition $E_j^{eexcess} < E_j^{reserve}$ must be met. Otherwise, tasks whose execution times are increased by decreased performance states of CPUs could cause additional delays of additional tasks. Of course, this method is approximation as the actual results depend on sizes of tasks, their distribution, and exact relation between CPU performance states and execution time. However, as results in Sect. 4.3 show this approach is helpful to avoid increase of energy consumption caused by power capping.

The second requirement is meant to limit the power but without visible performance lost. The mean completion time increase caused by power capping can be estimated as a product of the CPU frequency change (we assume proportional relation to execution time) and a percentage of time for which power capping was used. We empirically set a 5 % as a threshold for mean completion time increase to limit overall delays of the workload completion time and this condition was met (see results in Table 6). This parameter was used to limit CPU frequency decrease according to a model presented in Sect. 2.3 and can be based on specific Service Level Agreements with end users.

4 Optimizing Capacity Using Power Capping Methods

4.1 Power Capping Methods

Generally, power capping solutions can be divided into: software-based (coarse-grained and slower) and hardware-based solutions (fine-grained and faster).

Software-based solutions can be introduced independently from the vendor and regardless of whether hardware power capping is available. It can be applied on higher levels, e.g. managing tasks in a queue and balancing the load (with respect to power) among racks. The drawback of the software-based approach is longer time of reaction and more coarse-grained granularity.

Hardware-based power capping addresses this issue by the means of two main technologies available at processor level that enable the use of power capping. The first one is related to processor P-States and consists in lowering the processor core frequency and voltage. That provides a good power reduction for a relatively small loss in performance. However, using P-States can lower power consumption only to a certain point. Reducing consumption below that point requires the use of second technology, namely clock throttling. In this case, depending on the processor model, the system BIOS can either reprogram the processor to run at a lower frequency or modulate the processor between running periods and stopped periods.

In this paper we focus on the hardware-based approach benefiting from the processors P-States, as in the real data centers it ensures more reliable and faster effects. Moreover, it is often supported by hardware vendors and can be easily applied on the resource management level without affecting existing queueing system configuration (comparing to software-based approach). Its pseudo code for a rack is depicted by Algorithm 1.

Algorithm 1. Pseudo code of power capping algorithm

```
Require: $P$                    ▷ description of current power consumption of a rack
Require: $PC$                   ▷ power cap level for a rack
Require: $PU$                   ▷ power threshold for a rack
Ensure: $P_{x_i}$               ▷ final P-state of a processor i
  if $P > PC$ then
      repeat
          $P_y \leftarrow$ lowest P-State of all processors      ▷ lower P-State=higher frequency
          for each processor $i$ in a rack with $P_{x_i} = P_y$ do $P_{x_i} = P_{x_i+1}$
              if $P <= PC$ then break
              end if
          end for
      until $P <= PC$
  else if $P < PU$ then
      repeat
          $P_y \leftarrow$ highest P-State of all processors     ▷ higher P-State=lower frequency
          for each processor $i$ in a rack with $P_{x_i} = P_y$ do $P_{x_i} = P_{x_i-1}$
              if $P >= PC$ then break
              end if
          end for
      until $P >= PC$ or $P_y$= lowest available P-State of all processors
  end if
```

4.2 Simulation Experiments

To study the impact of power capping approach we performed another two simulations each time applying the power caps levels introduced in Sect. 3.2. Moreover, in the first simulation run we increased the inlet temperature (temperature of air entering the room) to 27 °C, while in the latter one we additionally modified, according to Table 5, the cooling capacity rated factor to $180[kW]$. Below we introduce the nomenclature used to compare the simulation results.

- Experiment A: Load Balancing strategy, $T_{R\_in} = 18\,^{\circ}\mathrm{C}$, reference case.
- Experiment B: Load Balancing with Power Capping approach, $T_{R\_in} = 27\,^{\circ}\mathrm{C}$
- Experiment C: Load Balancing with Power Capping approach, $T_{R\_in} = 27\,^{\circ}\mathrm{C}$, $Q_{cooling\_rated} = 180[kW]$.

4.3 Simulation Results

This section shows the simulation results for three types of experiments performed. Due to the paper constraints, only the results for Workload 1 are presented (Table 6).

Table 6. Simulation results for Workload 1

Metrics	A	B	C
Total IT energy consumption [kWh]	308.9	313.2	313.2
Total rack energy consumption [kWh]	370.8	376.2	376.2
Total cooling device energy consumption [kWh]	77.4	48.38	64.18
Total energy consumption [kWh]	525.22	502.66	518.47
Mean rack power [kW]	105.13	106.31	106.31
Mean power [kW]	148.916	142.04	146.51
Max rack power [kW]	144.82	130.38	130.38
Max power [kW]	214.45	176.87	183.49
PUE	1.416	1.336	1.378
PUE Level 4	1.7	1.605	1.655
Mean completion time [s]	6919	7262	7262
Mean task execution time [s]	2906	3249	3249
System load [%]	24.65	27.65	27.65

Figure 4 depicts the power distribution before and after applying a power capping technique.

Total Savings Achieved. The following section shows the savings achieved for the simulations carried out. The reference case, with Load Balancing policy, is named “A”. Optimized cases are named “B” and “C” respectively. First, when

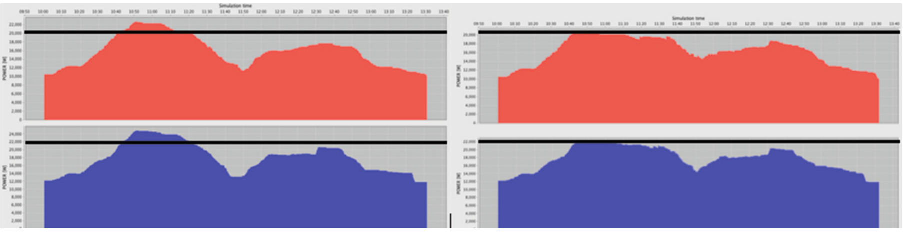


Fig. 4. Example power distribution on two racks for Workload 1 before (left) and after (right) applying a power capping technique

Table 7. Savings on particular metrics

Metrics	Savings (A-B)/A*100	Savings (A-C)/A*100
Execution time	−0.33 %	−0.33 %
Maximum rack power	9.98 %	9.98 %
Maximum power	17.53 %	14.44 %
Average power	4.61 %	1.61 %
Total energy consumed	4.30 %	1.29 %
PUE3	5.65 %	1.29 %
PUE4	5.59 %	2.68 %
Carbon Emissions	4.30 %	1.29 %
OPEX	4.30 %	1.29 %

the strategy of power capping is applied, main savings can be observed in the chiller consumption (reaching 37,50 % and 17.09 % respectively) due to the efficiency of the chiller (EER) improves with higher inlet temperatures. This leads to savings in terms of total energy consumed that are equal to 4.19 % in case "B" and 1.20 % in case "C".

In the simulation carried out, the strategy consisted of cutting the maximum power of racks keeping same cooling facilities (in case "B") or changing the chiller capabilities (in case "C"). The result obtained in these cases is a reduction in OPEX associated to power saved mainly in chiller and in CAPEX due to reduction of IT infrastructure. The metrics calculated from the results of those simulations are shown in Table 7.

Proposed approaches provide small benefits on PUE - the savings are obtained due to the lower power consumption of the chiller. With a power capping of 10 % the savings obtained in total energy consumed, carbon emissions and electricity costs (OPEX) are 4.30 % for case (B) and 1.29 % for case (C). The corresponding values obtained in savings extrapolated to a whole year considering a 24 × 7 operation time are 60 MWh/year, 20 tones CO_2/year, 5666 Euros/year and 21 MWh/year, 7 tones CO_2/year, 1982 Euros/year, respectively. Also, the CAPEX costs associated to less equipment required are calculated based on the following approach. Total building cost of traditional data center is estimated as

Table 8. Cost of data center placed in a room for three cases (thousand Euro)

Costs by sub-system	A	B	C
Project management	156	156	156
Power equipment	562	506	506
Cooling equipment	187	187	141
Engineering & installation	562	562	562
Racks	62	56	56
System monitoring	31	31	31
TOTAL	1562	1500	1453

15 million-US$ per MW of IT load according market survey developed by 451 Research company, referred as [18]. Converting this value to Euros with average annual ratios determined by the European Central Bank [20] referred to 2012, the corresponding value is 10784 Euros/kW(IT). On the other hand, the following distribution of cost between subsystems is considered according the study done by Schneider Electric [13].

The 10 % capping on maximum power of racks will affect directly the cost of those IT equipment but also on the sub-system of power equipment. Table 8 shows the distribution of costs of the three cases simulated. The costs of case (B) and (C) have been calculated estimating a reduction of 10 % in racks and power equipment. Finally, with this assumption, the savings obtained in CAPEX over the total cost of the data center is a 4 % or 62 thousands of Euros and 7 % or 109 thousands of Euros, respectively.

4.4 Application to Demand-Response Management

Nowadays, power grids face significant transformations. More open energy market, increased contribution of renewable energy sources, and rising energy prices stimulate changes of power grids to cope with new challenges such as adaptation to changing demand and supply, i.e. demand-response management. The approach to apply demand-response management to data centers was also already studied, e.g. proposed in [2]. We show that our approach to analysis of workloads and power capping mechanism can be applied to reduce costs in data centers.

Table 9. Comparison of approaches with and without power capping to deal with high demand periods

Approaches	Total energy cost [€]	Average energy price [€]	Mean completion time [s]
No power capping	128.24	0.12	6919
Mix	96.8	0.0942	7090

Let's assume that for a period assumed in previous Sections (3 h 20 min) there is a regular price for energy: 0.0942 /kWh. Now, let's also assume that period of the same size is a peak period in which energy provider is struggling with a demand that exceeds provider's supply. The provider to cope with this demand proposes the following contract to its customers: a regular price for this period will stay on the same level provided that a customer guarantees that it will not exceed 200 kW of power at anytime. Otherwise, the cost of 1 kWh will rise up to 0.15 /kWh. To reduce costs in this case we applied power capping to the peak period. The comparison of approaches without and with power capping are presented in Table 9.

In the first case power capping was not used in any period. In the second case power capping was applied to the second (peak) period. As it can be easily seen, the total cost savings reached almost 25 %. Extrapolating these numbers to the whole year would give around 45000€ of savings.

5 Conclusions

In this paper we demonstrated the use of the CoolEmAll SVD Toolkit to improve power and cooling capacity management in a data center by taking into account knowledge about applications and workloads. We applied power capping techniques and proper cooling infrastructure configuration to achieve savings in energy and costs. To obtain estimated values of a total energy consumption we simulated both IT software/hardware and cooling infrastructure using our tools. In this way, by better adjusting cooling infrastructure to specific types of workloads, we were able to find a configuration which resulted in energy savings by around 5 % and corresponding OPEX decrease. We have also found improvements of CAPEX without significant workload performance deterioration. Decrease in CAPEX was achieved by the selection of smaller chiller which is sufficient for the foreseen types of workloads. Savings in CAPEX reached 7 % for the case in which a smaller chiller was used according to the work! load analysis results and power capping strategies. Replacing only electrical equipment brought 4 % of savings in CAPEX. Energy savings were achieved by increase of the server inlet temperature. This was possible by limiting power used by particular racks and by compliance to the latest ASHRAE recommendations. Finally, we applied power capping to adjust data center operation to variable power supply and pricing. We achieved additional OPEX savings in order of 25 % (45000€ per year in the studied case).

Future work will include further improvements and tuning of cooling models. It will also include closer integration of CFD simulations into this analysis in order to identify hot spots and other consequences of modifications in a data center configuration. This approach will be used for various types of data centers. Finally, we plan to study more dynamic power capping strategies by adjusting power caps to the situation in a data center such as level and priority of load, energy supply and prices.

Acknowledgements. The results presented in this paper are partially funded by the European Commission under contract 288701 through the project CoolEmAll and by a grant from Polish National Science Center under award number 2013/08/A/ST6/00296.

References

1. AICIA Grupo de Termotecnia de la Escuela Superior de Ingenieros Industriales de la Universidad de Sevilla. Calificación de Eficiencia Energética de Edificios. Condiciones de aceptación de procedimientos alternativos a LIDER y CALENER. Gobierno de España. Ministerio de vivienda. Ministerio de Industria, Turismo y Comercio. Instituto para la diversificación y ahorro de energía (2009)
2. The All4Green project website. http://www.all4green-project.eu
3. Avelar, V., Azevedo, D., French, A.: The Green Grid. White paper # 49. PUE ^{TM}: A comprehensive examination of the metric (2012)
4. Chen, H., Hankendi, C., Caramanis, M.C., Coskun, A.K.: Dynamic server power capping for enabling data center participation in power markets. In: Proceedings of the International Conference on Computer-Aided Design, ICCAD 2013, pp. 122–129. IEEE Press, Piscataway (2013)
5. vor dem Berge, M., Da Costa, G., Kopecki, A., Oleksiak, A., Pierson, J.-M., Piontek, T., Volk, E., Wesner, S.: Modeling and simulation of data center energy-efficiency in CoolEmAll. In: Huusko, J., de Meer, H., Klingert, S., Somov, A. (eds.) E^{2}DC 2012. LNCS, vol. 7396, pp. 25–36. Springer, Heidelberg (2012)
6. Da Costa, G., Hlavacs, H., Hummel, K., Pierson, J.-M.: Modeling the energy consumption of distributed applications. In: Ahmad, I., Ranka, S. (eds.) Handbook of Energy-Aware and Green Computing. Chapman & Hall, CRC Press, Baco Raton (2012)
7. Kemma, R., Park, D.: Methodology Study Eco-design of Energy-using Products MEEUP. Final report. VHK. Delft, The Netherlands (2005). http://ec.europa.eu/enterprise/policies/sustainable-business/ecodesign/methodology/index_en.htm. Accessed 10 Jan 2014
8. Kontorinis, V., Zhang, L.E., Aksanli, B., Sampson, J., Homayoun, H., Pettis, E., Tullsen, D.M., Simunic Rosing, T.: Managing distributed UPS energy for effective power capping in data centers. In: 2012 39th Annual International Symposium on Computer Architecture (ISCA), pp. 488–499, 9–13 June 2012. doi:10.1109/ISCA.2012.6237042
9. Kurowski, K., Oleksiak, A., Piatek, W., Piontek, T., Przybyszewski, A., Weglarz, J.: DCworms - a tool for simulation of energy efficiency in distributed computing infrastructures. Simul. Model. Pract. Theory **39**, 135–151 (2013). ISSN 1569–190X, http://dx.doi.org/10.1016/j.simpat.2013.08.007
10. Hankendi, C., Reda, S., Coskun, A.K.: vCap: adaptive power capping for virtualized servers. In: 2013 IEEE International Symposium on Low Power Electronics and Design (ISLPED), pp. 415–420, 4–6 September 2013. doi:10.1109/ISLPED.2013.6629334
11. Chetsa, G.L.T., Lefevre, L., Pierson, J.-M., Stolf, P., Da Costa, G.: DNA-inspired scheme for building the energy profile of HPC systems. In: Huusko, J., de Meer, H., Klingert, S., Somov, A. (eds.) E^{2}DC 2012. LNCS, vol. 7396, pp. 141–152. Springer, Heidelberg (2012)
12. Lo, V., Mache, J., Windisch, K.: A comparative study of real workload traces and synthetic workload models for parallel job scheduling. In: Feitelson, D.G., Rudolph, L. (eds.) JSSPP 1998. LNCS, vol. 1459, pp. 25–46. Springer, Heidelberg (1998)

13. Rassmussen, N.: Determining Total Cost of ownership for data center and network room infrastructure. WP 6 APC. Schneider Electrics Data Center Science Center (2011)
14. Sisó, L., Forns, R.B., Napolitano, A., Salom, J., Da Costa, G., Volk, E., Donoghue, A.: D5.1 White paper on Energy- and Heat-aware metrics for computing modules - CoolEmAll Deliverable (2012). http://coolemall.eu
15. Sisó, L., Salom, J., Jarus, M., Oleksiak, A., Zilio, T.: Energy and heat-aware metrics for data centers: metrics analysis in the framework of CoolEmAll project. In: Third International Conference on Cloud and Green Computing (CGC), pp. 428–434 (2013)
16. Volk, E., Tenschert, A., Gienger, M., Oleksiak, A., Sisó, L., Salom, J.: Improving energy efficiency in data centers and federated cloud environments: comparison of CoolEmAll and Eco2Clouds approaches and metrics. In: Third International Conference on Cloud and Green Computing (CGC), pp. 443–450 (2013)
17. Liu, Z., Chen, Y., Bash, C., Wierman, A., Gmach, D., Wang, Z., Marwah, M., Hyser, C.: Renewable and cooling aware workload management for sustainable data centers. SIGMETRICS Perform. Eval. Rev. **40**(1), 175–186 (2012)
18. Research: The economics of prefabricated modular datacenters (2012)
19. Feitelson, D.: Workload modeling for computer systems performance evaluation. http://www.cs.huji.ac.il/feit/wlmod/. Accessed 30 Dec 2012
20. European Central Bank. http://www.ecb.europa.eu/home/html/index.en.html
21. Eurostat. European Commission. http://epp.eurostat.ec.europa.eu/tgm/table.do?tab=table&init=1&plugin=1&language=en&pcode=ten00114. Accessed 06 March 2014
22. ParallelWorkload Archive. http://www.cs.huji.ac.il/labs/parallel/workload/

Building Application Profiles to Allow a Better Usage of the Renewable Energies in Data Centres

Corentin Dupont(✉)

University of Trento, Via Sommarive, 5, 38123 Trento, Italy
cdupont@create-net.org

Abstract. Data centres are powerful and power-hungry facilities which aim at hosting ICT services. The current trend is to, on the one hand, try to reduce the overall consumption of a data centre, and on the other hand to prioritize the utilization of renewable energies over brown energies. Renewable energies tend to be very variable in time (e.g. solar energy), and thus renewable energy aware algorithms tries to schedule the applications running in the data centres accordingly. However, one of the main problems is that most of the time very little information is known about the applications running in data centres. More specifically, we need to have more information about the current and planned workload of an application, and the tolerance of that application to have its workload rescheduled. In this paper, we present a work in progress on Plug4Green, a flexible VM manager able to reduce energy consumption in data centres. We extend Plug4Green with the second goal of increasing the usage of renewable energy in data centres. This includes the development of specific application profiles, and a new optimization technique.

1 Introduction

Data centres are large facilities which purpose is to host information processing and telecommunication services for scientific and/or business applications. Due to the rise in service demands together with energy costs, the energy efficiency has now been added as a new key metric for data centres. Energy-aware strategies are beginning to be integrated inside the data centre resource manager. In practice, a Virtual Machine (VM) placement algorithm considers the data centre and the workload characteristics to place the VMs among the servers in the most efficient way, considering performance and energy consumption. This placement must be done respecting the requirements of the Service Level Agreement (SLA) existing between the data centre and its clients.

In parallel to reducing the overall energy consumption, the current trend is to foster the use of renewable energies. Renewable energies have the problem to be very variable and time-dependent: for example solar power is available only during the day, and is subject to variations due to the meteorological conditions. Thus, data centre operators must try to shift the workload of running

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 120–131, 2015.
DOI: 10.1007/978-3-319-15786-3_8

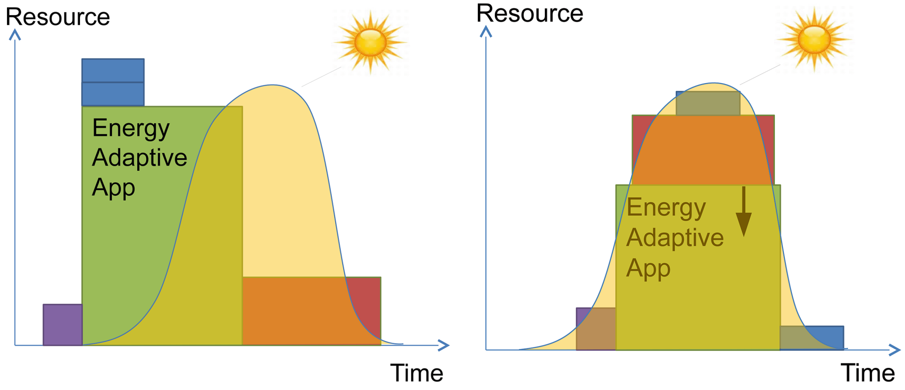


Fig. 1. Adapting applications for a better usage of renewable energies

applications in time, to match it with the availability (or forecast availability) of renewable energy, as it is depicted in Fig. 1.

As preliminary work, we present Plug4Green [1], an energy aware VM manager based on Constraint Programming (CP) [2]. The use of CP allows to attain a relatively good flexibility and extensibility: indeed data centres are evolving permanently and new use cases are added regularly. We already proposed and implemented 23 VM placement constraints to address common concerns such as hardware compatibilities, performance, security issues, and workload instability. The usage of CP makes placement constraints, objectives, and algorithms independent from each other: new concerns can be added in the VM manager without changing the existing implementation.

The goal of this paper is to present a work in progress on the extension of Plug4Green, so as to extend its objectives to not only reduce the overall energy consumption of a data centre, but also allow a better usage of the renewable energies. A great challenge of efficiently using the renewable energies in a data centre is to be able to schedule correctly the workload of the applications. This shows the importance of being able to know the workload an application will have to run at a certain point of time, to understand under what conditions it can be shifted or delayed, and *in fine* to schedule it correctly.

Yet, currently most of the applications running in data centres are unaware of their self workload: they are unable to predict how much computing power they will require and when. In data centres, the knowledge of the requirements of an application in terms of resources is still "meta-knowledge", i.e. the knowledge of the data centre operators. For example, in data centres, database indexing maintenance operations are usually performed at night, to minimize the impact on the overall performance. However, in a data centre using primarily solar power, it would be interesting to shift this task during the lunch break, when the sun is shining. The knowledge that this particular task, "database indexing", can cope with a 12 h shift, and that it takes approximately half an hour, belongs to the operator's knowledge. In this paper, we show how this knowledge can be encoded and used by Plug4Green to schedule the application workload correctly. We propose a design for the extended version of Plug4Green and discuss the possible optimization techniques.

The remainder of this paper is structured as follow: we will first perform a survey of the related works in Sect. 2. We then present the extended design of Plug4Green in Sect. 3 and preliminary implementation in Sect. 4. We conclude in Sect. 5.

2 Related Work

A few flexible and extensible frameworks for VM allocation have been proposed recently. For example, BtrPlace [3] is a CP-based flexible consolidation manager. Plug4Green leverages on Btrplace [4,5]. BtrPlace does not take into consideration energy related problems and does not provide an operator with the opportunity of setting optimization objectives. In contrast to BtrPlace, Plug4Green directly addresses energy consumption problem. This required numerous extensions: the development of a power model and different model extensions, two objectives with their associated heuristics, 7 energy-related constraints, and a domain-specific language to directly exhibit energy concerns and metrics such as PUE, CUE[1] and Watts, to the end-users.

Similar modular consolidation manager adopting CP paradigm is presented in [6]. The authors ensure high availability for VM placement by guaranteeing at any time a certain number of vacant servers to allocate VMs with regards to placement constraints. The manager scalability is effective for 32 servers and 128 VMs.

A hybrid system proposed in [7] solves a resource reallocation problem. This system includes Business Rules Management System (BRMS) and CP. A user can customize both business rules and constraints. The BRMS monitors and analyses the servers' state at a period of time to detect overloaded servers and bottlenecks. Once a problem is identified the BRMS models its instance and sends it to the CP solver which resolves it within seconds. In contrast to our manager, both the systems presented in [6,7] are not addressing energy-efficiency problems.

In [8], the authors proposes GreenSwitch, a model-based approach for dynamically scheduling the workload and selecting the source of energy to use. In this work, the authors focuses on the trade-offs involved in powering data centres with solar and/or wind energy, and propose an implementation of their solar powered mini data centre called Parasol. With contrast to this approach, we propose the possibility to schedule the workload at a finer grain, which is the application level.

[1] PUE and CUE are defined by The Green Grid Consortium: http://www.thegreengrid.org/.

3 Design

We present the design selected for the Plug4Green prototype in Sect. 3.1. We then present our advancement in defining the application management engine in Sect. 3.2. We finally discuss the technology choice made for the optimization engine, and compare it especially to SMT, a technology that we envisage to use in the future development of Plug4Green in Sect. 3.3.

3.1 Plug4Green

Plug4Green is an extensible VM manager. The architecture chosen allows to easily extend the engine by adding new concerns, without modifying the underlying algorithms. In particular, new constraints can be added straightforwardly, as we showed by implementing 23 constraints commonly encountered in data centres, including energy-oriented ones. As can be seen in Fig. 2, Plug4Green has the following inputs:

- The *SLAs*
- The *data centre configuration*
- A *Single Allocation* request
- Or a *Global Optimisation* request

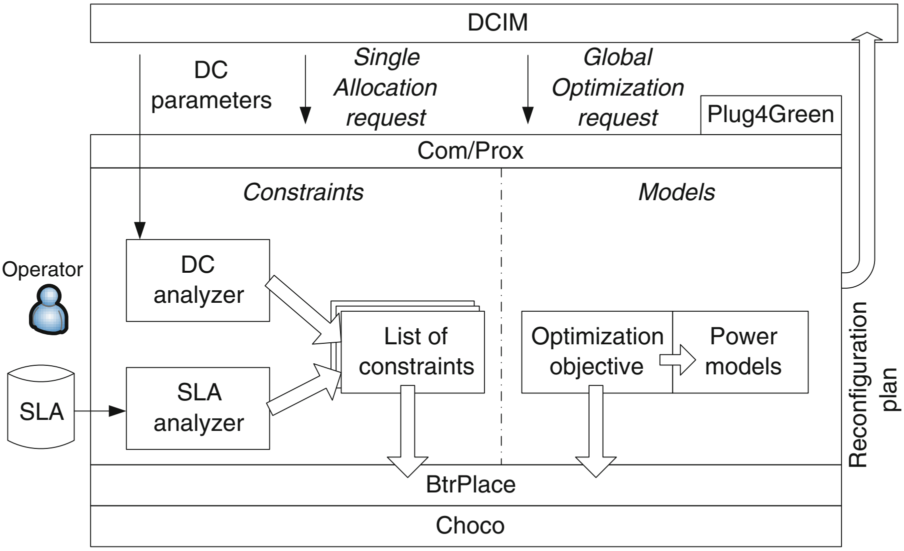


Fig. 2. Plug4Green architecture

Plug4Green considers a set of *SLA* constraints along with the *data centre configuration* to compute a *reconfiguration plan* as an output. The *data centre* *configuration* captures all the relevant ICT resources of a data centre with their energy-related attributes and interconnections, in an XML format. The reconfiguration plan consists of a set of actions such as *powering on*, *powering off*, *waking up* and *putting in idle mode* a server, and *migrating* a VM, that satisfies all the constraints and minimizes the current objective. The objective can be to minimize either the power consumption of a federation of data centres, or the CO_2 emissions. The diagram shows the clear separation between the *Constraints* part ("what" we want to do) and the *Models* part ("how" to solve the problem), which is fundamental for extensibility.

Plug4Green is called by the Data Centre Infrastructure Management (DCIM) for two different events: *Single Allocation* or *Global Optimisation*. The *Single Allocation* event is triggered when a new VM have to be allocated. Plug4Green will compute and return the best server to allocate the VM on, taking into account the characteristics of the VM, the current state of the data centre, the SLAs and the current objective. The *Global Optimisation* event is itself triggered regularly (every ten minutes in our experimentation) and Plug4Green will return a reconfiguration plan. In manual mode, the data centre operator has the possibility to accept or reject this reconfiguration plan, while in automatic mode, it is enacted automatically. Plug4Green will then execute the reconfiguration plan in order to reduce the overall consumption of the data centre (either power consumption or gas emission) while also respecting the SLAs. The *Com/Prox* layer ensures that Plug4Green can be plugged easily to different existing DCIM: its the only part that must be updated when adapting the software for a new DCIM. Currently, Plug4Green can be integrated into VMWare[2], Eucalyptus[3], and HP Matrix Operating Environment[4] infrastructures. Plug4Green is based on the flexible consolidation manager BtrPlace [3].

We evaluated Plug4Green in an industrial test bed, to show that it is both efficient and scalable:

- Using our framework in a realistic cloud data centre environment allowed to reduce the overall energy consumption up to 33 % and the gas emission up to 34 %. These savings are achieved by considering the servers hardware heterogeneity, their different energy-efficiency and different compositions of SLAs.
- We showed by simulation how such an approach can be scalable. In particular, we were able to compute the improved placement of 7,500 VMs on 1,500 servers, while respecting their SLA.

3.2 Energy Aware Software Controller

In order to allow Plug4Green to optimize the usage of renewable energies, we extend the design of Plug4Green presented previously: we define the Energy

[2] http://www.vmware.com.

[3] http://eucalyptus.com.

[4] http://h18004.www1.hp.com/products/solutions/insightdynamics/overview.html.

Aware Software Controller (EASC), as depicted in Fig. 3. For each application, the EASC is in charge of:

- building an energetic profile of that application,
- defining the tasks and working modes,
- building the list of constraints,
- executing the activity plan as computed by Plug4Green.

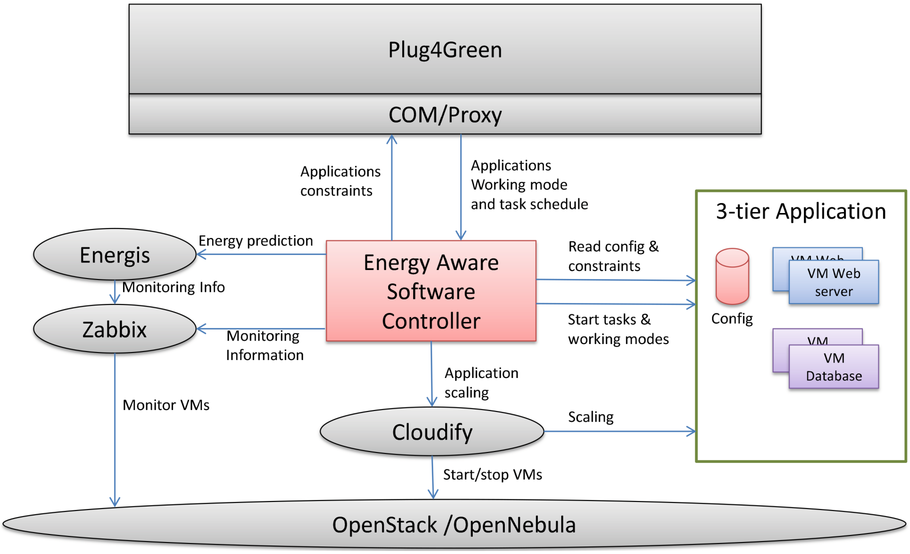


Fig. 3. Energy aware software controllers for aPaaS

A working mode, specifically, is a particular way for an application to perform a task, according to its SLA. For example, a typical 3-tier application can have several VMs containing its web server, and be allowed to scale up or down the number of VMs according to the number of requests. Each possible combination of VMs is called a working mode. In order to build the energetic profile of the working modes, we use Zabbix to collect monitoring data for the VMs used by the application. We then use Energis[5] to compute the energy necessary for each working modes and tasks of the applications. Energis is a tool using predictive algorithms based on historical measurement data in order to predict the energy consumption of a particular VM.

The energetic profiles together with the defined working modes and tasks are then transmitted to Plug4Green, that will compute an optimized scheduling, called the activity plan. This activity plan is transmitted back to the EASC to be performed. In practice, the activity plan consists in spawning more or less

[5] Energis: http://www.freemind-group.com/index.php/products/energis.html.

VMs to execute the tasks of the application, such as front-end web servers or back-end databases. A PaaS management tool such as Cloudify[6] can provide such a scalability service, together with OpenStack[7].

The EASC is instantiated into three flavours: The EASC aPaaS (Application Platform as a Service), showed in the picture, is in charge of controlling the Cloud applications inside a data centre. It will scale up and down 3-tier applications according to the availability of renewable energies. The EASC IaaS (Infrastructure as a Service) is in charge of collaborating with the Cloud management system to manage the data centre infrastructure. In practice, it we will tune the VM consolidation factor to allow more or less energy saving and thus follow the renewable energy availability. Finally, the EASC TM (Tasks Management) will shift in time the maintenance tasks that are performed by the data centre, such as virus scan or server decommissioning tasks. Those tasks will be scheduled when the renewable energy is available.

3.3 Optimization

Plug4Green is based on Constraint Programming, which is a programming paradigm devoted to solve Constraint Satisfaction Problems (CSP). In a CSP, relations between variables are stated in the form of constraints. Each constraint restricts the combination of values that a set of variables may take simultaneously. While CP was a very good choice and fulfilled most of the requirements, we discovered some practical drawbacks. Especially, defining new constraints easily is one of the main design goal of Plug4Green: as data centres evolves, new use cases arrives regularly, and a qualified operator should be able to insert new constraints into the engine. However, defining a new constraint takes a lot of lines of code and is also very error prone. The debugging period for each new constraint is also quite long. This diminishes the flexibility of the tool, which should imply the easy creation of new constraints to fit the new requirements arriving in a data centre.

To tackle this problem of flexibility, we started exploring alternatives to the couple Constraint Programming/Java. As an alternative to CP, we propose Satisfiability Modulo Theories [9] (SMT). A SMT problem is to determine the satisfiability of ground logical formulas with respect to background theories expressed in classical first-order logic with equality. Modern SMT solvers integrate a Boolean satisfiability (SAT) solver with specialized solvers for a set of literals belonging to each theory. The problem consists in finding an assignment to the variables that satisfy all constraints. We also surveyed the feasibility of using Pure Functional languages such as Haskell[8] as the base language for the constraint engine of Plug4Green. Programs in Haskell tend to be much less verbose than in Java (in the order of ten time less lines). It is also a declarative language, like Constraint

[6] Cloudify: http://getcloudify.org/.

[7] OpenStack: https://www.openstack.org/.

[8] http://haskell.org.

Programming is, so the expression of constraints is more clear and natural. Furthermore, Haskell is pure, which combined with its strong type system allows to reduce drastically the number of bugs.

4 Implementation

To show the usability of both SMT and pure functional languages to tackle energy efficiency problems in a flexible way, we implemented the classical problem of packing VMs on servers using the library SBV[9], with only one dimension for the sake of simplicity. In the example[10] showed in Listing 1.1, each VM has a demand in term of CPU, and each server has a certain CPU capacity to offer. The objective is to find the placement of the VMs on the servers that minimizes the number of servers needed. The only constraint applied is that the total CPU consumption of the VMs that will be running on a server must not exceed the capacity of that server.

```

--concrete IDs for VMs and servers
type VMID = Integer
type SID = Integer

--symbolic IDs of the servers
type SSID = SBV SID

--A VM is just a name and a cpuDemand
data VM = VM { vmName :: String,
               cpuDemand :: Integer}

--a server has got a name and a certain amount of free CPU
data Server = Server { serverName :: String,
                       cpuCapacity :: Integer}

--list of VMs
vms :: Map VMID VM
vms = fromList $ zip [0..] [VM"VM1" 100, VM "VM2" 50, VM "VM3" 15]

--list of servers
servers :: Map SID Server
servers = fromList $ zip [0..] [Server "Server1" 100, Server "Server2" 100, Server "Server3" 200]

--number of servers ON (which we'll try to minimize)
numberServersOn :: Map VMID SSID -> SInteger
numberServersOn = count . elems . M.map (./= 0) . vmCounts

--computes the number of VMs on each servers
vmCounts :: Map VMID SSID -> Map SID SInteger
vmCounts vmls = M.mapWithKey count servers where
   count sid _ = sum [ite (mysid .== literal sid) 1 0 | mysid <- elems vmls]

--All the CPU constraints
cpuConstraints :: Map VMID SSID -> SBool
cpuConstraints vmls = bAnd $ elems $ M.mapWithKey criteria (serverCPUHeights vmls) where
   criteria :: SID -> SInteger -> SBool
   criteria sid height = (literal $ cpuCapacity $ fromJust $ M.lookup sid servers) .> height

```

[9] http://leventerkok.github.io/sbv/.

[10] The full implementation can be seen at https://github.com/cdupont/Plug4Green-design.

```
--computes the CPU consummed by the VMs on each servers
serverCPUHeights :: Map VMID SSID -> Map SID SInteger
serverCPUHeights vmls = M.mapWithKey sumVMsHeights servers where
   sumVMsHeights :: SID -> Server -> SInteger
   sumVMsHeights sid _ = sum [ite (sid' .== literal sid) (literal $ cpuDemand $ fromJust $ M.lookup
         vmid vms) 0 | (vmid, sid') <- M.assocs vmls]

--solves the VM placement problem
vmPlacementProblem :: IO (Maybe (Map VMID SID))
vmPlacementProblem = minimize' numberServersOn cpuConstraints

main = do
   s <- vmPlacementProblem
   putStrLn $ show s
```

Listing 1.1. Example of VM placement problem solved using SMT

When run, this program returns the placement for the VMs that minimizes the number of necessary servers. In this case, it will place all three VMs on the third server. While it is difficult to compare, it is anyway striking that this program is shorter than its equivalent in Java/Choco[11]. The definition of a constraint takes only a few lines (for example *numberServersOn* takes 2 lines) and flows with the program definition. Furthermore, as it is usually the case in Haskell, the type signature of the functions are carrying a lot of information that can be used both by the programmer to understand and reason about the program, and by the compiler to prove its correctness. For example, the type signature *numberServersOn :: Map VMID SSID -> SInteger* makes it clear that the function *numberServersOn* is a constraint that takes the positions of all the VMs on the servers (denoted as a mapping between the VM ids and the server symbolic ids) and returns a symbolic integer representing the necessary number of servers.

Furthermore, programming at the symbolic level, as it is required when designing a CSP, is not very different than programming in concrete Haskell. This is because a lot of the Haskell standard functions, like the function *sum* in our example program, can be reused in a constraint programming program. The definition of *sum* in the standard library of Haskell is generic enough to be able to be used also at the symbolic level. On the other hand, programming in Choco is completely different than programming in concrete Java: all the operators are necessarily different, due to the low genericity of Java. Therefore, the intuition of the Java programmer cannot be completely reused.

SBV is also a theorem prover, and that can be used to prove properties of the constraints expressed. For example, we might want to prove some properties about our constraint *vmCounts*. This function counts the number of VMs present on each servers. We want to prove the property that the count of VMs on a server has for absolute maximum the total numbers of VMs present in the data centre.

[11] For example this implementation of bin packing: http://www.dcs.gla.ac.uk/~pat/cpM/jchoco/binPack/CPBinPack.java.

```

*Main> prove $ \x y -> bAll (.<= 2) $ vmCounts' [x, y]
Q.E.D.
```

Listing 1.2. Example of proof about a constraint

The Listing 1.2 show how we can ask SBV to prove that the number of VMs per server computed by the constraint *vmCounts* cannot exceed the total number of VMs (in this simplified example with only 2 VMs and a version of vmCounts defined for lists instead of maps). SBV simply replies with *Q.E.D*, showing that it found a proof of our property (this proof can be exhibited if needed).

In order to give to an optimization engine complementary informations about the profile of an application, we use a configuration file. An example is given in Listing 1.3 (written in Yaml).

```
Name: PetClinic
Tasks:
  - Name: T1
    Duration: 1h
    StartTask: ./startT1
    Dependencies:
      - finishBefore T2
      - finishBefore 00:00
    WorkingModes:
    - Name: W1D1
      SwitchMode: ./switchMode.sh W1D1
      Resources: m1.small
      TimeFrame: WeekEnds
    - Name: W1D2
      SwitchMode: ./switchMode.sh W1D2
      Resources: m1.small
      TimeFrame: WeekDays, WeekEnds
  - Name: T2
    Duration: 2h
    StartTask: ./startT2
    Dependencies:
      - finishBefore 00:00
Constraints:
    - MutuallyExclusive: W1D1, W1D2
    - AtLeastOne: W1D1, W1D2
```

Listing 1.3. Example of profile configuration file

The above listing describes a 3-Tier application, PetClinic, that has an Apache front-end, a Java back-end and a database. The front-end as well as the back-end can be scaled up and down using Cloudify: for example in the case too much web pages are requested, a new VM will be spawned and the Apache server will be installed in it using Chef[12]. The example file defines two tasks T1 and T2. Tasks define a punctual activity with a duration. We define a script able to start the task, and some absolute and relative dependencies. We also define working modes, W1D1 and W1D2, with the way we can switch from one mode to another (switchMode shell script), the needs in term of resources, and the possibly repetitive time frames during which those working modes are authorized. Finally we define

[12] http://www.getchef.com/.

overall constraints, such as “MutuallyExclusive”, which describe a relation between two or more working modes that should not be active at the same time. Another example “AtLeastOne” describe the fact that there should be at least one working mode active at all time.

5 Conclusion

In this paper we presented our plan to enhance Plug4Green, an energy-aware VM manager based on Constraint Programming, in particular to allow it to increase the usage of renewable energies in data centres. We introduced the Energy Aware Software Controller, a new component communicating with Plug4 Green and able to build energy profiles for Cloud applications, and to control the application following the activity plans computed by Plug4Green. To compute the activity plans, we surveyed the SAT Modulo Theory technique in order to integrate it inside Plug4Green.

Acknowledgments and Availability. The author would like to thanks the University of Trento, the EU FP7 projects FIT4Green and DC4Cities, and the Create-Net research centre. Evaluations presented in this paper were carried out by HP Innovation Centre Milan and INRIA.

Plug4Green is licensed under the terms of the Apache 2.0 License. The current prototype is available for download at https://github.com/fit4green/Plug4Green.

References

1. Dupont, C., Schulze, T., Giuliani, G., Somov, A., Hermenier, F.: An energy aware framework for virtual machine placement in cloud federated data centres. In: Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, e-Energy 2012, pp. 4:1–4:10. ACM (2012)
2. Rossi, F., van Beek, P., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York (2006)
3. Hermenier, F., Lawall, J., Muller, G.: Btrplace: a flexible consolidation manager for highly available applications. IEEE Trans. Dependable Secure Comput. **10**(5), 273–286 (2013)
4. Hermenier, F., Demassey, S., Lorca, X.: Bin repacking scheduling in virtualized datacenters. In: Lee, J. (ed.) CP 2011. LNCS, vol. 6876, pp. 27–41. Springer, Heidelberg (2011)
5. Hermenier, F., Lorca, X., Menaud, J.-M., Muller, G., Lawall, J.: Entropy: a consolidation manager for clusters. In: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, VEE 2009, pp. 41–50. ACM (2009)
6. Bin, E., Biran, O., Boni, O., Hadad, E., Kolodner, E.K., Moatti, Y., Lorenz, D.H.: Guaranteeing high availability goals for virtual machine placement. In: 2011 31st International Conference on Distributed Computing Systems (ICDCS), pp. 700–709 (2011)

7. van der Krogt, R., Feldman, J., Little, J., Stynes, D.: An integrated business rules and constraints approach to data centre capacity management. In: Cohen, D. (ed.) CP 2010. LNCS, vol. 6308, pp. 568–582. Springer, Heidelberg (2010)
8. Goiri, Í., Katsak, W., Le, K., Nguyen, T.D., Bianchini, R.: Parasol and greenswitch: managing datacenters powered by renewable energy. SIGARCH Comput. Archit. News **41**(1), 51–64 (2013)
9. Franch, J.S.: Satisfiability modulo theories approach to constraint programming (2013)

Energy Efficiency Metrics for Data Centres

Review on Performance Metrics for Energy Efficiency in Data Center: The Role of Thermal Management

Alfonso Capozzoli[1], Marta Chinnici[2](✉), Marco Perino[1],
and Gianluca Serale[1]

[1] Politecnico Di Torino, Corso Duca Degli Abruzzi, 24 – 10129 Turin, Italy
{alfonso.capozzoli,marco.perino,
gianluca.serale}@polito.it
[2] ENEA, C.R. Casaccia, Via Anguillarese, 301 – 00123 Rome, Italy
marta.chinnici@enea.it

Abstract. Energy consumption and thermal performance are the two most important tasks in data centers (DCs) facility management. In recent years, to monitor and control their variation several performance metrics were introduced. In this paper an overview on the main important energy and thermal metrics is provided. A critical analysis to investigate mutual relations among metrics was performed, with the aim to clarify some physical aspects regarding the assessment of DC global energy performance.

Indeed, although these metrics are commonly used to assess the energy efficiency of DCs, their usefulness for encouraging lower energy consumption was poorly investigated. Moreover, an analysis on the effect of the DC thermal performance on metrics was done. The thermal management assume a key role for achieving energy saving during the operation of a DC and for the improvement of the IT equipment reliability.

Keywords: Data centers management · Energy efficiency · Metrics · Temperature in data center · IT equipment · Data center thermal performance

1 Introduction

Data centers (DCs) are indispensable elements of information systems. They have been upgraded by information systems organizations continuously to offer institutional and private customers a constantly growing variety of IT services. As a consequence, inventory, power density and energy use of DCs are rising steadily. Recently, to provide a set of green strategies for energy consumption of DCs, the attention is addressed above all, on servers ability to properly function and not lose data – commonly called reliability. This goal can be reached using redundant systems and connections, backup power supplies and an appropriate environmental control by HVAC systems (servers are less susceptible to failure and faults when operating at certain environmental conditions [1]).

In recent years, energy efficiency became an important issue in DC design due to the increase of energy prices and policy pressures [2]. Besides, expansion of demands

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 135–151, 2015.
DOI: 10.1007/978-3-319-15786-3_9

has increased the power consumption. Indeed electricity usage became the most expensive portion of a DC's costs [3]. In general, 80 % of capital cost is related to IT power supply and cooling, while constructing a DC costs cover remaining 20 % [4]. In the last decade, the growth trend of electricity usage contribution of DCs is +11 %/year, compared to the sum of all sectors equal to +3 %/year [5].

In light of this trend, companies operating in DCs have launched research efforts to improve the energy efficiency of several components. Subsequently, different international initiatives and energy efficiency criteria, benchmarks, best practice measures and efficient product technologies were proposed supporting efficiency both at the IT hardware and the infrastructure level [6]. These strategies allow to slightly reduce the growth trend of electricity usage. Based on several data collected from IT equipment manufacturers, the Uptime Institute predicted an annual increase of product heat density (W/m2) from 7 % to 28 %, for tape storages and communication equipment respectively [7]. However, energy consumption in DCs of past five year has only increased by about 50 % of this predicted scenario [6]).

Although much research effort was made in the field of energy efficiency best practices on DCs so far, there is only little step forward on DC green performance measurement system. Moreover, several different metrics were proposed during the last few years [8] which allow to outlay a trend over time by means the evaluation of their variation in different periods. In details, the paper addresses two major research questions: real time thermal diagnostic analysis and the energy assessment of a DC. These tasks become easier where a metering system monitoring environmental conditions and power/energy usage of different loads is in place [9].

The paper provides academic and practitioners with the body of Knowledge on DC green performance measurement, and moreover formulates open research challenges. In details, this paper focuses on the most important energy performance metrics currently used. The importance of each metric analysed is evaluated by its matching with the thermal and energy efficiency requirements, its popularity/diffusion and its recent introduction. The variables and physical models on which they are based were discussed in the first part of the paper. Afterwards, mutual relations among performance metrics were underlined and reciprocal physical influences were evaluated, taking into account that this aspect was poorly investigated in literature. Moreover, the impact of local thermal phenomena on the behavior of the global energy consumption and on the reliability of the IT equipment was carried out. At last, the strategies for achieving energy saving through a thermal awareness approach were discussed. The key role of thermal management during the operation of a DC in order to achieve energy saving is demonstrated. A proposal of a conceptual framework to follow is carried out. Finally, the conclusion summarizes the findings.

2 A Review of Data Center Energy Efficiency Metrics

A DC is an integration of complex systems and this complexity creates serious difficulties in pinpointing a methodology in terms of energy efficiency. Indeed, within DCs many variables are to be taken into account. In recent years, to measure these variables, various metrics were proposed. However, in Europe there is a lack of a complete plan which provides standard metrics and methodologies for DCs. Evaluation of DCs should be based on globally accepted assessment systems in terms of common metrics that promote the improvement of energy saving, renovation and improvement of infrastructures, management methods and so on.

In order to provide a systematically approach - in terms of metrics for energy efficiency for DC - a classification of metrics into four macro-categories are proposed by the authors: waste and emission metrics; component efficiency metrics; power and energy consumption metrics; thermal/energy metrics.

Waste and Emission metrics allow to measure the amount of natural sources wasted or the quantity of pollution generated by building and managing a DC. Examples of these metrics are the WUE (Water Usage Effectiveness) and the CUE (Carbon Usage Effectiveness) proposed by The Green Grid [10, 11]. The first one measures the water usage and the second one the carbon emissions associated with DCs energy consumption. However, this paper focuses only on the effect of thermal management on the other three categories and the Waste and Emission metrics are not discussed in-depth.

Component efficiency metrics measure the productivity and the energy requested by individual components or sub-assemblies. The metrics measuring efficiencies of an individual component are straightly related to the global power/energy consumption metrics. Indeed, the global energy consumption of the DC is directly connected to the efficiency and consumption of its single sub-systems. Generally, the major causes of energy consumption in a DC are the IT equipment and the cooling system of the infrastructure. Thus, the most important component efficiency metrics belong to these two categories.

Power and Energy Consumption metrics are proposed in order to capture the "greenness" concept of the DC. Indeed, the concept of green DC recently became an important issue with the aim to reduce the energy consumption through an improvement of both infrastructure and IT productivity management [8]. Typically these metrics are defined as an efficiency ratio of obtained useful work on power/energy consumed for its production, or vice versa.

Thermal/Energy metrics are generally indices related to airflow performance and thermal management. These metrics are used to label the thermal efficiency in DC and to detect local anomalous behaviors, which have influence on temperature fields and, consequently, on reliability of IT equipment.

Hence, the energy efficiency can only be represented by a vector which captures the effect of energy consumed by a metrics suite (as above mentioned). In the next sections, a critical description of the most important metrics for DCs is carried out.

2.1 Thermal Metrics

The most common design structure in a DC is characterized by a raised floor with the racks arranged in Hot/Cold Aisle layout. In the Computer Room Air Conditioning (CRACs) the hot exhaust air from racks is cooled and supplied chilled in the floor plenum.

Figure 1 shows this typical air distribution layout in a DC. In the hot/cold aisle layout, some air management problems were observed by several researchers. The hot exhaust air could flow back into the cold aisle to mix with the supply cold air. The mixing of hot exhaust and cold supply air can raise the inlet IT equipment temperature resulting in the hot spot phenomenon. In [12] the authors found that, on average, one over ten racks works at a temperature higher than the standards recommendations and that hot spots occur especially in DC with light load.

As far as a correct air management is concerned, two problems have been identified: by pass air and recirculation air. In the first case a fraction of inlet cold air does not contribute to the cooling of the IT equipment, usually because the flow rate is too high or it leaks through cable cutouts. Moreover, a higher amount of cold air results in excess fan energy and reduced return air temperatures. In [13] the author assess that on average only the 40 % of cold air supplied by CRAC, directly enters in the IT equipment cooling it down. In the second case the cold air intake into IT equipment is not sufficient (for by pass problem or incorrect air distribution system design) and as a consequence a fraction of the hot air is recirculated, resulting in higher equipment inlet temperature. The same effect of recirculation can be related to the incomplete rack slot occupation of IT equipment. Finally a negative pressure phenomena (hot air drawn into the under-floor plenum) could occur. The problems of air management in reality lowers the cooling efficiency and generates a vicious cycle of rising local temperature [14]. In this context became very important to investigate the supply air flow efficiency in the DC environment. The aim of a correct air management system is to minimize the recirculation of hot air and minimize by pass of cool air.

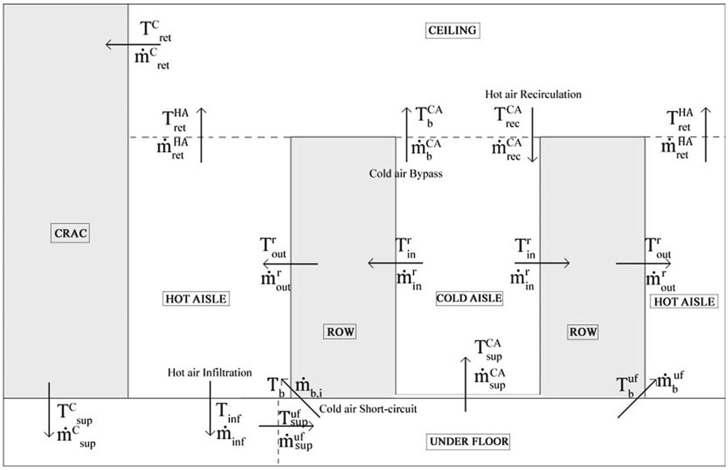


Fig. 1. By pass and recirculation air in a typical DC distribution layout

Several thermal performance metrics were introduced in literature to enable real time feedback and control of DC thermal architecture. Due to standardized vent tiles and rack, composed by Unity Rack multiples, these indices can be applied regardless on different DCs. These kind of metrics can be useful for a real time feedback regarding the air management particularly because the normalized non dimensional nature enables scalability across the rack, row and room levels. However, even if some metrics can be related to a single rack, they are not always able to predict in a comprehensive way local hot spots. For this reason some thermal metrics are devoted to give information on local phenomena while through others only average thermal condition can be derived. In Table 1 the formulas useful to calculate the main thermal metrics following described are summarised.

SHI (Supply Heat Index) [15] is defined as the ratio between the enthalpy rise in cold aisle due to infiltration of warm air and the total enthalpy rise at rack exhaust. RHI (Return Heat Index) is defined as the ratio between the total enthalpy rise in DC airspace and total enthalpy rise at rack exhaust. A value of 0 for SHI and 1 for RHI means that there is no recirculation of hot air, and the air inlet temperature of the racks is equal to the air supply temperature from the CRAC unit. As benchmark, it is considered good performance a value of SHI < 0.2 and RHI > 0.8.

However some authors [14, 16] assess that SHI/RHI are not always effective because a DC with a favorable global profiles can be still affected by local hotspots. A single rack could have a rapid increase in temperatures, that could comport a local hot spot that these indices could not identify. Several researchers claim that these indices give a macro evaluation of the recirculation phenomena, but they fail especially when the rack intake temperatures have to be evaluated in respect of IT equipment reliability even if they are calculated at rack level. Schmidt has tried to overcome this problem proposing the β Index [17]. This index gives information about local temperatures in a rack and it is obtained from the ratio of a punctual temperature difference across a rack on the average difference across the rack. The common values of this index range between 0, which represent the condition of absence of recirculation, and 1. A value above 1 indicates the presence of self-heating phenomenon.

The metric RCI (Rack Cooling Index) [18] is a measure of how effectively the IT equipment is cooled and maintained within a given intake temperature specifications. It was based on the comparison between air inlet temperature and benchmark values [1]. Due to the two-fold values of temperatures, cold and hot, the index is divided into $\mathrm{RCI}_{\mathrm{Lo}}$ and $\mathrm{RCI}_{\mathrm{Hi}}$, for low and high threshold values respectively. In general, a value equal to 1 means that the overall rack intake air temperature is in the range recommended and a value of benchmark of RCI > 96 % is considered a good cooling condition.

Some other indices were introduced to characterize how the air flow rate supply is distributed in the environment [19]. In particular, starting from the mass and heat balance equation, NP (Negative Pressure) index measures the ambient air infiltration into the under-floor, BP (bypass) measures the rate of air which doesn't get inside IT equipment, R (Recirculation) indicates the rate of hot air that goes into the IT equipment and BAL (balance) measures the difference between cold air produced by the cooling plant and the server request. The ideal condition of airflow performance and thermal management is NP, BP and R equal to 0 and BAL equal to 1. The flow performance, equal to 1-BP is defined as the ratio between the air mass flow rate from cooling units supplied to IT equipment and air mass flow rate through cooling units. The thermal performance, also equal to 1-R defines how much of the air used by IT equipment actually comes from the CRAC. The ratio of thermal performance on flow performance represents the flow availability.

The index RTI (Return Temperature Index) [20] gives information about recirculation or bypass of the airflow: the value is obtained as a ratio of the temperature difference in the CRAC unit on temperature difference across the IT equipment, expressed in percentage. RTI equal to 100 % is the ideal condition, RTI > 100 % indicates recirculation, RTI < 100 %, instead, indicate a bypass. This index is considered as an opportunity to improve energy efficiency: a bypass result means that the cooling plant is producing more airflow rate than which requested by the IT equipment. This index has to be measured at room level.

Another index, based only on the mass flow rate, is CI (capture index) [21]. It may be calculated both in the cold and hot aisle. If related to the cold aisle, CI evaluates the mass rate ingested from a rack that comes directly from the CRAC unit. In the hot aisle, instead, CI measures the fraction air captured by a local vent or local cooler. The ideal

Table 1. Most important thermal metrics for DCs

Metric	Formula	Introduced by
Supply Heat Index	$SHI = \frac{\sum_i \sum_j \left(T^r_{in_{i,j}} - T^C_{sup}\right)}{\sum_i \sum_j \left(T^r_{out_{i,j}} - T^C_{sup}\right)}$	[15]
Return Heat Index	$RHI = 1 - SHI$	[15]
β index	$\beta = \frac{T^r_{in}(z) - T^C_{sup}}{T^r_{out} - T^r_{in}}$	[17]
Rack Cooling Index Low	$RCI_{Lo} = \left[1 - \frac{\sum \left(T_{minRec} - T^r_{in_{i,j}}\right)_{T^r_{in_{i,j}} < T_{minRec}}}{(T_{minRec} - T_{minAll}) \cdot n} \right] \cdot 100$	[18]
Rack Cooling Index High	$RCI_{Hi} = \left[1 - \frac{\sum \left(T^r_{in_{i,j}} - T_{maxRec}\right)_{T^r_{in_{i,j}} > T_{maxRec}}}{(T_{maxAll} - T_{maxRec}) \cdot n} \right] \cdot 100$	[18]
Negative Pressure	$NP = \frac{T^{uf}_{sup} - T^C_{sup}}{T^c_{ret} - T^{uf}_{sup}}$	[19]
Bypass Ratio	$BP = \frac{T^S_{out} - T^C_{ret}}{T^S_{out} - T^{uf}_{sup}}$	[19]
Recirculation	$R = \frac{T^S_{in} - T^{uf}_{sup}}{T^S_{out} - T^{uf}_{sup}}$	[19]
Balance	$BAL = \frac{T^S_{out} - T^S_{in}}{T^C_{ret} - T^C_{sup}} = \frac{1 - R}{(1 - BP) \cdot (1 + NP)}$	[19]
Return Temperature Index	$RTI = \frac{T^C_{ret} - T^C_{\mathrm{sup}}}{T^r_{out} - T^r_{in}} \cdot 100$	[20]
Capture Index (cold aisle)	$CI = \frac{\dot{m}^C_{in_i}}{\dot{m}^C_{sup_i}}$	[21]
Capture Index (hot aisle)	$CI = \sum_{J=1}^{N} \frac{C^C_{ret_j} . \dot{m}^C_{ret_j}}{\dot{m}^r_{out_j}}$	[21]

condition is represented by a value of CI equal to 1. This index and the similar ones (e.g. contaminant index k [22] and R recirculation) mean that the total heat flow produced by an IT equipment is not fully dissipated from the airflow supplied by the cooling plant but only by a fraction of this, on the base of the index magnitude.

2.2 Component Efficiency Metrics

Component efficiency metrics measure the performance of individual components or sub-assemblies. Improving the efficiency of a single apparatus allows to improve the efficiency of the overall structure. For this reason the energy consumption of a DC is straightly related to the productivity and efficiency of its single components. In a DC the energy consumption is mainly due to the IT equipment and the cooling system. For these reason in literature many indices related to the components of these two categories were introduced.

Main component efficiency metrics for IT equipment

Although the connection between performance and energy consumption of IT equipment is becoming necessary information for data center efficiency, there isn't an universally accepted metric for this sector [23]. However, to capture a more accurate picture of the energy efficiency (performance) of IT equipment, are proposed different benchmarks in order to stress different components, such as the processor, memory, and disk (Standard Performance Evaluation Corporation and LINPACK benchmark).

Main component efficiency metrics for cooling system

The component efficiency metrics for cooling systems listed in Table 2 are in the following described.

The index CSE (Data Center Cooling System Efficiency) characterizes the overall efficiency of the cooling system (including chillers, pumps, and cooling towers) in terms of energy input per unit of cooling output [24]. This metric is defined by the ratio of the average system power used on the average cooling load required. Considering the only chiller without auxiliary (e.g. cooling tower) and the inverse of CSE it is possible to find the COP (Coefficient Of Performance), an index widely diffused in refrigeration systems.

The index CSS (Cooling System Sizing factor) is defined as the ratio of the installed cooling capacity on the peak cooling load. It is an indicator useful to measure the percentage of working hours of the cooling system at part load condition. Usually the cooling system has different efficiencies between partial and full loads (generally the highest efficiency is around 80 % of the load [24]). An high CSS value suggest a good potential and scalability of the cooling system [8].

The AEU (air economizer utilization) is a metric related to the airflow economizer system. The airflow economizer is a component of the cooling system which allows to work in "free cooling" condition, controlling the environment temperature directly with colder outdoor air. A consistent energy saving is due to "free cooling" especially in cold climates, where the outdoor air is colder than indoor condition for a long period of time. AEU is defined as the percentage of hours in a year during which the economizer system work in either full or complete operation [8, 24]. The external conditions strongly influence the AEU and hence the opportunity to increase the DC efficiency.

The index AE (airflow efficiency) measures the overall airflow efficiency. It is defined in terms of the total fan power required per unit of airflow [24]. This metric provides an overall measure of how efficiently air is moved through the DC, from the supply to the return [8]. An high value of AE suggest that the fan system is inefficient. However only few data on this index are available in literature.

Table 2. Most important component efficiency metrics for cooling system.

Metric	Formula
CSE	$CSE = \frac{Average\ cooling\ system\ power\ usage}{Average\ cooling\ load}$
CSS	$CSS = \frac{Installed\ chiller\ capacity}{Peak\ chiller\ load}$
AEU	$AEU = \frac{Air\ economizer\ hours}{24 \cdot 365}$
AE	$AE = \frac{Total\ fan\ power}{Total\ fan\ airflow} \cdot 100$

2.3 Power and Energy Consumption Metrics

Due to the diversity and complexity of the infrastructure, understanding the total data center power consumption is not easy [22]. In order to clarify the problem, a division into categories it is necessary.

Pelley et al. [22] identified five distinct sub-systems that account for most of a data center's power draws: servers and storage systems, power conditioning equipment, cooling and humidification systems, networking and connecting equipment, light-ing/ physical security. According to Wang et al. [8] it is possible to differentiate the power consumption in servers and in storage system. The second one could be merged with the power draws in networking and connecting equipment, thus forming a category of power draws of all delivery components external to the IT equipment (i.e. UPS, switch gear, generators, PDUs, batteries, networking and connecting equipment). The total power consumed by a typical data center is broken down into previous categories as follow [22, 26, 27]: IT equipment (30 ÷ 60 %); cooling plant: (25 ÷ 40 %); power conditioning system (8 ÷ 15 %); lighting/security (1 ÷ 3 %); delivery components and others (5 ÷ 15 %).

Generally the useful work of a data center is represented by the activity of the computing activities coming from IT, while all other categories are only auxiliaries to this purpose. Several power and energy consumption metrics for data centers were suggested in recent year following this guideline. A brief overview of these main indices is proposed in this section, before analysing and discussing the interaction of these metrics with thermal and component efficiency indices. In Table 3 the formula of energy consumption metrics for DCs below described are shown.

Data center infrastructure efficiency (DCIE) is an accepted metric used to determine the energy efficiency of a data center. The metric, which is expressed as a percentage, is calculated by dividing IT equipment power by total facility power. DCIE was developed by members of the Green Grid, an industry group focused on data center energy efficiency [28, 29]. The inverse of the DCiE, is another index called PUE (Power Usage Effectiveness) [29]. PUE is the most popular metric for energy consumption in data centers, due to its simplicity of interpretation: the lower the better [6]. Many companies uses the PUE as an indicator to show how green their data centers are. Similar to the PUE is the KPITE (Key Performance Indicator of Task Efficiency) introduced by ETSI [30].

The metric, Power Usage Effectiveness (PUE), was successfully used for improving the energy efficiency of DCs [12, 31]. However, several problems and shortcomings were found. The main of these issues, is the fact that PUE only measure the efficiency of the whole infrastructure, defined as a "black box". Moreover, IT equipment inefficiencies, due to rising environmental temperature, are not taken into account by the PUE [32]. This metric is strongly dependent on the climatic environment of the site and on the IT load. These variables may change over time and the DC operate for considerable periods at part load conditions [33].

In order to overcome previous problems, other indices were recently introduced. In details, as PUE does not account for the power distribution and energy losses inside IT equipment, two new metrics are proposed: ITUE (IT-power usage effectiveness), similar to PUE but "inside" the IT and TUE (total-power usage effectiveness), which combines PUE and ITUE for a total efficiency picture [32]. In details, TUE provides a ratio of total energy (internal and external support energy uses) into the DC divided by the total energy to the computational components inside the IT equipment. Moreover, The Green Grid proposed the indices CPE (Compute Power Efficiency) and DCeP (Data Center Energy Productivity) for characterizing the energy request to produce useful computational work in a data center [34].

To measure how "green" a data center is, it is also necessary to take into account the presence of renewable energy sources and the recovery of some energy wastes. Therefore, The Green Grid introduced the coefficients ERF (Energy Reuse Factor) and GEC (Green Energy Coefficient), which take into account these phenomena and, combined with the PUE, produce the index ERE (Energy Reuse Effectiveness) [35]. Likewise, ETSI proposed the metrics $\mathrm{KPI}_{\mathrm{REUSE}}$(Key Performance Indicator of Energy Reuse) and $\mathrm{KPI}_{\mathrm{REN}}$ (Key Performance Indicator of Renewable Energy) which, combined with the $\mathrm{KPI}_{\mathrm{TE}}$, produce the DC index DC_{P} [30].

Table 3. Most important energy consumption metrics for DCs

Metric	Formula	Introduced by
DC infrastructure Efficiency	$\mathrm{DCiE} = \frac{\mathit{IT\ equipment\ power}}{\mathit{Total\ facility\ power}}$	[28, 29]
Power Usage Effectiveness	$PUE = \frac{\mathit{Total\ facility\ power}}{\mathit{IT\ equipment\ power}} = \frac{1}{\mathrm{DCiE}}$	[29]
Key Performance Indicator of Task Efficiency	$KPI_{TE} = \frac{\mathit{Total\ facility\ energy\ consumption}}{\mathit{IT\ equipment\ energy\ consumption}}$	[30]
IT-power Usage Effectiveness	$ITUE = \frac{\mathit{Total\ energy\ into\ IT\ equipment}}{\mathit{Total\ energy\ into\ compute\ components}}$	[32]

(*Continued*)

Table 3. (*Continued*)

Metric	Formula	Introduced by
Total-power Usage Effectiveness	$TUE = ITUE \cdot PUE$	[32]
Compute Power Efficiency	$CPE = \frac{IT\ equipment\ utilisation}{PUE}$	[34]
DC energy Productivity	$CPE = \frac{Useful\ work\ produced}{Total\ facility\ energy\ consumption}$	[34]
Energy Reuse Factor	$ERF = \frac{Reused\ energy\ consumption}{Total\ facility\ energy\ consumption}$	[35]
Green Energy Coefficient	$GEC = \frac{Renewable\ energy\ consumption}{Total\ facility\ energy\ consumption}$	[35]
Energy Reuse Effectiveness	$ERE = (1 - ERF) \cdot (1 - GEC) \cdot PUE$	[35]
Key Performance Indicator of Energy Reuse	$KPI_{REU} = \frac{Reused\ energy\ consumption}{Total\ facility\ energy\ consumption}$	[30]
Key Performance Indicator of Renewable Energy	$KPI_{REN} = \frac{Renewable\ energy\ consumption}{Total\ facility\ energy\ consumption}$	[30]
DC Performance	$DC_P = KPI_{TE} \cdot (1 - W_{Reu} \cdot KPI_{reu}) \cdot (1 - W_{Ren} \cdot KPI_{ren})$	[30]

3 Relations Among Thermal, Component Efficiency and Power/Energy Metrics

In this section an analysis of the effect of temperature variation on the DC metrics was performed. Moreover the mutual relations between thermal and energy/power indices were critically discussed. This aspect was poorly investigated in literature. The aim is to clarify the physical phenomena affecting the energy consumption considering both thermal and energy aspects in order to show the capability of different metrics in capturing in a comprehensive way the energy savings opportunities in DCs.

3.1 The Effect of Thermal Performance on Data Center Energy Consumption

For most critical DCs ASHRAE guidelines [1] recommends that the temperature at the inlet of the IT equipment T_{in}^{r} should be maintained between 18 and 27 °C, with an allowable range of 15–32 °C. The moisture level should be kept within a minimum 5.5 °C and maximum 15 °C dew point [1, 12]. To accomplish performance and reliability of IT equipment, a colder T_{in}^{r} is desirable. Even if the recommended temperature are referred to T_{in}^{r}, the control of the cooling system is generally based on the setting of air supplied temperature in the cold aisle T_{sup}^{CA}. This temperature is always colder than T_{in}^{r}. The difference between T_{sup}^{CA} and T_{in}^{r} is due to bad air management phenomena - such as recirculation and bypass - which are usually sources of coldspots and hotspots.

To evaluate the effect of temperature on thermal metrics is not an easy task. Cho et al. [36] applied a CFD method to study the variation of overall thermal metrics by changing different geometric and physical parameters. It was evinced that the temperature of the air supplied in the cold aisle T_{sup}^{CA} is the most important parameter for air flow efficiency. However, certain thermal metrics - like SHI, RHI and RTI - didn't change sensibly modifying this value. Indeed, thermal metrics are mainly related to temperature differences and not to a local and single value of temperature, as previously described. For example the difference between T_{sup}^{CA} and T_{in}^{r} is considered in SHI [15]. Durand-Estebe et al. [25] demonstrated that, varying T_{sup}^{CA} from 17 °C to 30 °C, the temperature difference between T_{sup}^{CA} and T_{in}^{r} remained quite stable. On the other hand, other metrics are related to difference between a monitored temperature and a benchmark value, such as RCI [18]. For this reason in both [25] and [36] RCI is sensitive to the temperature variation of T_{sup}^{CA}. On the other hand the impact that the temperature variation has on the efficiency of single components was widely studied [1, 33, 38]. Metrics related to this efficiencies and global power/energy consumption are straightly related. Indeed, the global energy consumption of the DC is directly connected to the efficiency and consumption of its single sub-systems.

The efficiency of the cooling system growth with the increasing of air temperature. Cooling system efficiency indices, such as CSE and COP, get to better values with higher temperatures, because the chiller efficiency is directly proportional to the supply temperature T_{sup}^{CA} [37]. Moreover, also the index AEU increase with ambient temperature rising. This fact is due to the increasing of the percentage of hours in a year in which the economizer system can work because outside air temperature is colder than inside temperature. Also at the air fan level (AE index) the temperature increasing reduce thepower required to move the air, due to lower density of hotter air. However, at higher temperatures, the specific heat c_p of air is lower. This means a lower cooling capacity and higher flow rates, which remove any improvement for the lower density [33].

Despite these improvements, rising the temperature is not necessary a source of energy saving, because it could cause problems for the IT equipment. First of all a worst reliability, but also a decreasing of energy efficiency of IT equipment by rising the T_{in}^{r} temperature occur. In recent year these effects were studied theoretically [1, 33] and experimentally [38]. It appears that, rising the T_{in}^{r} also the power consumption of the CPU growth, due to higher chip current leakage. At the same time the flow rate of the fans increase, on the basis of an algorithm which ensure that the chips operate under their upper temperature limit. As a consequence an increasing in fan energy consumption results. In general, it can be observed that the two phenomena (increasing of CPU leakage and of fan energy consumption) can occur not concurrently. Recent studies [33, 39] demonstrate that an optimal CRAC temperature set-point exists, that would be the ideal tradeoff between CRAC and IT energy consumption. This temperature mainly depends on the server and the CRAC characteristics [25].

The analysis of previous relations shows how the increasing of temperature causes an improvement of the efficiency of the cooling system components (infrastructure side), but it causes negative effects on IT equipment. This fact produces a better PUE or DCiE, since these metrics are very sensitive to the infrastructure improvement and not to the IT equipment. Patterson in [33] calculated the variation of PUE by varying the temperature in an ideal DC. The temperature was set from 20 °C to 30 °C and the COP of the cooling system was calculated as a function of temperature (higher the temperature the better the COP). For this reason PUE resulted better at the higher temperature: 1.91 at 30 °C compared to 2.0 at 20 °C. However, considering the effect of temperature on IT equipment the total power request by DC passes from 32 kW at 20 °C to 32.2–32.8 kW at 30 °C. In this way Patterson demonstrated that the PUE is not a good indicator for the assessment of energy saving because it neglects the IT side. Generally the TUE and DCeP should not be affected by the error due to this opposite trends because they consider also the effects on IT side. However, no examples are referred in literature on this aspect, because this metrics were introduced to take into account the improvements - and not the worsening – for IT equipment.

3.2 Mutual Relation Among Thermal and Energy Consumption Metrics

Previously described inefficiencies related to single components could be caused by local phenomena, such as hotspots, which are generally caused by bad air management. Hotspots are local rising temperature effects which have a bad influence on the IT equipment efficiency and reliability. However, in order to offset hotspots the CRAC unit are often controlled to supply air at a lower T_{sup}^{C}, with an higher energy consumption [40]. This fact required an oversized design of cooling system (bad CSS index) to deal with these instantaneous overcooling request phenomena. The effects of poor air management are identified through thermal metrics.

In general, global energy indices are not necessary capable to detect these phenomena. Indeed, hotspots are local phenomena whose influence on the global power/energy efficiency may be negligible. This is due to the fact that few rack in a DC could be interested on local air management problems and as a consequence global energy indicators may not vary significantly. Furthermore, the energy metrics are referred to a long period (e.g. a year or a season) while hotspots are phenomena which depends on short term variation of boundary conditions (e.g. IT data loads or temporary malfunctioning of the cooling system). Therefore it is necessary to apply a continuous commissioning to detect the occurrence of local phenomena through thermal metrics. An overall air management efficiency should be evaluated taking into account at the same time different air management metrics proposed in literature. In particular the thermal metrics at local level (e.g. RCI and β) should be assessed primarily. After that an analysis on the other global thermal indices (e.g. RTI, SHI and RHI), still important to characterize airflow behavior and energy saving opportunities, should be carried out.

In general power/energy metrics provide no information about bypass/recirculation phenomena and corresponding impacts on the thermal manageability of DCs. Vice versa, thermal metrics are still of limited use because few information is gained regarding the energy efficiency of the system [41]. Thermal metrics are used to enable real-time feedback and control of DC thermal architecture, while power/energy metrics to outline the global energy consumption. Usually these metrics are used in parallel. For example Lu et al. [12, 42] calculated SHI, RTI and PUE for a DC case of study. SHI and RTI were used for the detection of local phenomena of recirculation and bypass, PUE for attesting the global energy consumption. The fact that the analysis require three different indicators indicate a poor relationship among metrics.

Other researchers deepen how the energy performance is influenced by the variation of thermal metrics. For example Pelley et al. [22] considered the effect of a thermal metric in their DC power flow model, by means the index κ (containment index). This metric is based on previously described CI and its better benchmark value is 1. κ is influenced by the power and the supply temperature of CRAC and IT equipment. Authors showed that preventing air recirculation causes an increase of κ and drastically improve cooling efficiency.

However, Shah et al. [41] found that in general thermal metrics fail to provide adequate information regarding the energy consumption of a DC, due to the use of the first law of thermodynamic for their formulation. They propose a method to addresses this limitation through an exergy analysis of the DC thermal management system. The exergy of a system is a representation of the amount of useful work that can be obtained from a given quantity of energy. This approach recognizes that the mixing of hot and cold streams in the DC airspace is an irreversible process and must therefore lead to a loss of exergy. The proposed exergy-based approach can provide a foundation upon which the DC cooling system can be simultaneously evaluated for thermal manageability and energy efficiency [41]. This work has recently been taken over by Qian et al. [43] that introduce new indices using entransy, a novel physical quantity recently presented [44].

The discussion presented in this section demonstrate that the energy and thermal metrics accomplish to different tasks: energy assessment and real time diagnostic. These tasks always should be coupled for a comprehensive DC performance analysis. Moreover, although it is not yet widely considered, the local thermal performance management assume a key role for achieving energy saving during the operation of a DC. In the next section some examples supporting this assumption are reported. Recent studies focused on the energy saving strategies through a thermal awareness approach are analysed.

3.3 Energy Efficiency Management Trough Thermal Performance Awareness

Many authors proposed strategies to reduce the DC energy consumption and to increase the reliability the IT equipment. Tang et al. [37] identified two major steps for improving the DC performance. The first is a good DC design and planning perspective. For example Flucker and Tozer [9] established an order of priorities in strategies and measures to be taken in this step. The second step is to improve and optimise the performance during the operation of a DC. Generally this aim is achieved through the optimization of DC thermal performance. Lowering operation costs and extending the life of the IT equipment are key design objectives that can be achieved through the improvement of thermal management. A recent challenge is to schedule the task assignment to server with a thermal awareness approach.

Although several researches tried to consider the thermal management for task scheduling [45, 46]. In this approach the management of computing workload is controlled considering the minimization of recirculation heat [37]. This effect can be considered by means Cross Interference Coefficients α, which characterize the thermal interference among the computing nodes within a DC. Starting from the DC thermal behavior some algorithms were developed to schedule the workload on servers with the aim of minimize heat recirculation and to maximize CRAC air supply temperatures. In this way computing and cooling costs can be improved.

DCs very often work with an utilization rate lower than 100 %. For example, an High Performance Computing DC utilization rate is generally between 60 % and 80 % [47]. One of the most representative algorithm is Xint-GA, a genetic algorithm that follow the condition to Minimize the Peak Inlet Temperature through Task Assignment. Considering the total computing and cooling cost as a function of power computing, the tool dispatches tasks following the concept of consolidation.

In order to consider the thermal environment, for a set task to assign to computing devices, airflow IT inlet temperatures are obtained through Cross Interference Coefficients. Air supply temperature is optimized from the difference between the maximum air inlet temperature and a threshold value. The optimal CRAC supply temperature is set choosing the task assignment that minimize the peak air IT inlet value. In this manner the supply temperature is the highest allowable taking into account the recirculation effect caused by heat interference among the computing nodes.

From comparison between different scheduling algorithms (as Uniform Output Profile, Minimal Computing Energy and Uniform Task [37]), Minimizing Heat Recirculation [46] and XInt-GA assure a lower SHI value for each utilization rate, validating the approach of heat recirculation minimization. At the same time, with thermal awareness approach the air supply temperature is highest, allowing the most efficient way to manage operational DC with optimization of cooling and computing power, on the base of heat recirculation minimization. This results seems to be the most efficient because it take into account three different aspects that are often analysed without correlations. A correct thermal management represents an efficient criteria for the optimization of energy request both for cooling and computing and to obtain energy saving during the operation of a DC.

4 Conclusions

In this paper, a critical analysis on the most important energy performance metrics currently used for DCs was presented. The variables and physical models on which they are based as well as their mutual relations were discussed. The impact of temperature on metrics, on behavior of the global energy consumption and on reliability of the IT equipment was carried out.

In order to achieve energy saving the key role of thermal management during the operation of DCs was demonstrated. Energy assessment and real time thermal environment diagnostic should be not considered as separate tasks for a comprehensive DC performance analysis. These two aspects should be always coupled. In details, the thermal management should be achieved through the calculation of local thermal metrics primarily, and then by other average thermal metrics referred to the whole DC environment. On the other hand the energy assessment should be performed through power/energy metrics capable to capture in a correct way the effect of energy consumption variation for both cooling and computing. The improvement and optimisation of DC performance through a thermal awareness approach represents an effective way to obtain energy savings. To this purpose thermal management through the detection of local temperature faults, cooling efficiency, IT reliability and computing energy request are take into account for DC energy assessment.

Acknowledgments. Authors would like to gratefully thank Lucia Liuzzo.

References

1. ASHRAE: 2011 Thermal Guidelines for Data Processing Environments – Expanded Data Center Classes and Usage Guidance. Technical report, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc (2011)
2. The Green Grid: The Green Grid Energy Policy Research for Data Centres. Technical report, White Paper #25, The Green Grid (2009)
3. Tozer, R., Flucker, S., Romano, A.: Scalable data centre efficiency. In: CIBSE Technical Symposium, Liverpool John Moores University, Liverpool, UK, 11–12 April 2013
4. Barroso, L.A., Hölzle, U.: The datacenter as a computer: an introduction to the design of warehouse-scale machines. Synth. Lect. Comput. Archit. **4**(1), 1–108 (2009)
5. Koomey, J.G.: Worldwide electricity used in data centers. Environ. Res. Lett. **3**(3), 034008 (2008)
6. Schaeppi, B., Bogner, T., Schloesser, A., Stobbe, L., De Asuncao, M.D.: Metrics for energy efficiency assessment in data centers and server rooms. In: Electronics Goes Green 2012+ (EGG), pp. 1–6. IEEE (2012)
7. The uptime institute: heat density trends in data processing, computer systems and telecommunication equipment. Technical report, The Uptime Institute (2000)
8. Wang, L., Khan, S.U.: Review of performance metrics for green data centers: a taxonomy study. J. Supercomput. **63**(3), 639–656 (2011). doi:10.1007/s11227-011-0704-3(2011)
9. Flucker, S., Tozer, R.: Data centre energy efficiency analysis to minimize total cost of ownership. Building Serv. Eng. Res. Technol. **34**(1), 103–117 (2013)
10. The Green Grid: Water Usage Effectiveness (WUE): A Green Grid Data Center Sustainability Metric. Technical report, White Paper #35, The Green Grid (2011)
11. The Green Grid: Carbon Usage Effectiveness (CUE): A Green Grid Data Center Sustainability Metric. Technical report, White Paper #32, The Green Grid (2010)
12. Lu, T., Lü, X., Remes, M., Viljanen, M.: Investigation of air management and energy performance in a data center in Finland: case study. Energy Build. **43**(12), 3360–3372 (2011). doi:10.1016/j.enbuild.2011.08.034
13. Sullivan, R.F.: Reducing bypass airflow is essential for eliminating computer room hot spots. Technical report, The Uptime Institute (2004)
14. Cho, J., Kim, B.S.: Evaluation of air management system's thermal performance for superior cooling efficiency in high-density data centers. Energy Build. **43**(9), 2145–2155 (2011)

15. Sharma, R.K., Bash, C.E., Patel, C.D.: Dimensionless parameters for evaluation of thermal design and performance of large-scale data centers. In: 8th ASME/AIAA Joint Thermophysics and Heat Transfer Conference, pp. 1–1 (2002)
16. Tang, Q., Mukherjee, T., Gupta, S.K., Cayton, P.: Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. In: Fourth International Conference on Intelligent Sensing and Information Processing, ICISIP 2006, pp. 203–208. IEEE (2006)
17. Schmidt, R.R., Cruz, E.E., Iyengar, M.: Challenges of data center thermal management. IBM J. Res. Dev. **49**(4.5), 709–723 (2005)
18. Herrlin, M.K.: Rack cooling effectiveness in data centers and telecom central offices: the rack cooling index (RCI). ASHRAE Trans. **111**(2), 725–731 (2005)
19. Tozer, R., Kurkjian, C., Salim, M.: Air management metrics in data centers. ASHRAE Trans. **115**(1), 63–70 (2009)
20. Herrlin, M.K.: Improved data center energy efficiency and thermal performance by advanced airflow analysis. In: Digital Power Forum, pp. 10–12, San Francisco (2007)
21. Vangilder, J.W., Shrivastava, S.K.: Capture index: an airflow-based rack cooling performance metric. ASHRAE Trans. **113**(1), 126–136 (2007)
22. Pelley, S., Meisner, D., Wenisch, T.F., VanGilder, J.W.: Understanding and abstracting total data center power. In: Workshop on Energy-Efficient Design, June 2009
23. Chinnici, M., Quintiliani, A.: An example of methodology to assess energy efficiency improvements in datacenters. In: 2013 Third International Conference on Cloud and Green Computing (CGC), pp. 459–463. IEEE, Karlsruhe (2013)
24. Mathew, P., Ganguly, S., Greenberg, S., Sartor, D.: Self-benchmarking guide for data centers: metrics, benchmarks, actions. Technical report, Lawrence Berkeley National Laboratory, Berkeley, California, June 2010
25. Durand-Estebe, B., Le Bot, C., Mancos, J.N., Arquis, E.: Data center optimization using PID regulation in CFD simulations. Energy Build. **66**, 154–164 (2013)
26. Dumitru, I., Fagarasan, I., Iliescu, S., Said, Y.H., Ploix, S.: Increasing energy efficiency in data centers using energy management. In: 2011 IEEE/ACM International Conference on Green Computing and Communications (GreenCom), pp. 159–165. IEEE (2011)
27. Yuventi, J., Mehdizadeh, R.: A critical analysis of power usage effectiveness and its use in communicating data center energy consumption. Energy Build. **64**, 90–94 (2013)
28. The green grid: the green grid metrics: data center infrastructure efficiency (DCiE) detailed analysis. Technical report, White Paper #14, The Green Grid (2008)
29. The green grid: the green grid data center efficiency metrics: PUE and DCIE. Technical report, White Paper #6, The Green Grid (2007)
30. ETSI: Operational Energy Efficiency for User (OEU): Global KPI for Data Centers. Technical report, ETSI (2013)
31. Brady, G.A., Kapur, N., Summers, J.L., Thompson, H.M.: A case study and critical assessment in calculating power usage effectiveness for a data centre. Energy Convers. Manage. **76**, 155–161 (2013). doi:10.1016/j.enconman.2013.07.035
32. Patterson, M.K., Poole, S.W., Hsu, C.-H., Maxwell, D., Tschudi, W., Coles, H., Martinez, D.J., Bates, N.: TUE, a new energy-efficiency metric applied at ORNL's Jaguar. In: Kunkel, J.M., Ludwig, T., Meuer, H.W. (eds.) ISC 2013. LNCS, vol. 7905, pp. 372–382. Springer, Heidelberg (2013)
33. Patterson, M.K.: The effect of data center temperature on energy efficiency. In: 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, ITHERM 2008. pp. 1167–1174. IEEE (2008)
34. The Green Grid: The Green Grid Productivity Indicator. Technical report, White Paper #15, The Green Grid (2008)

35. The Green Grid: ERE: a Metric for Measuring the Benefit of Reuse Energy from a Data Center. Technical report, White Paper #29, The Green Grid (2010)
36. Cho, J., Yang, J., Park, W.: Evaluation of air distribution system's airflow performance for cooling energy savings in high-density data centers. Energy Build. **68**, 270–279 (2014)
37. Tang, Q., Gupta, S.K.S., Varsamopoulos, G.: Thermal-aware task scheduling for data centers through minimizing heat recirculation. In: 2007 IEEE International Conference on Cluster Computing, pp. 129–138, (2007). doi:10.1109/CLUSTR.2007.4629225
38. Moss, D., Bean, J.H.: Energy impact of increased server inlet temperature. Technical report, APC White Paper, 138 (2009)
39. Biswas, S., Tiwari, M., Sherwood, T., Theogarajan, L., Chong, F.T.: Fighting fire with fire: modeling the datacenter-scale effects of targeted superlattice thermal management. In: 2011 38th Annual International Symposium on Computer Architecture (ISCA), pp. 331–340. IEEE (2011)
40. Favoino, F., Capozzoli, A., Perino, M.: Temperature field real-time diagnosis by means of infrared imaging in data elaboration center. In: Abramowicz, W., Llorente, I.M., Surridge, M., Zisman, A., Vayssière, J. (eds.) Proceedings of the 8th International Symposium on Heating, Ventilation and Air Conditioning. LNEE, vol. 263. Springer, Heidelberg (2014)
41. Shah, A.J., Carey, V.P., Bash, C.E., Patel, C.D.: Exergy analysis of data center thermal management systems. J. Heat Transf. **130**(2), 021401 (2008). doi:10.1115/1.2787024
42. Lu, X., Lu, T., Remes, M., Viljanen, M.: Preliminary analysis of energy efficiency in data center: case study. World Acad. Sci. Eng. Technol. **5**(4), 04–26 (2011)
43. Qian, X., Li, Z., Li, Z.: A thermal environmental analysis method for data centers. Int. J. Heat Mass Transf. **62**, 579–585 (2013)
44. Guo, Z.Y., Zhu, H.Y., Liang, X.G.: Entransy - a physical quantity describing heat transfer ability. Int. J. Heat Mass Transf. **50**(13), 2545–2556 (2007)
45. Sharma, R.K., Bash, C.E., Patel, C.D., Friedrich, R.J., Chase, J.S.: Balance of power: dynamic thermal management for internet data centers. Internet Comput. IEEE **9**(1), 42–49 (2005)
46. Moore, J., Chase, J., Ranganathan, P., Sharma, R.: Making scheduling "cool": temperature aware workload placement in data centers. Technical report, in USENIX Annual Technical Conference (2005)
47. Mukherjee, T., Banerjee, A., Varsamopoulos, G., Gupta, S.K., Rungta, S.: Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. Comput. Netw. **53**(17), 2888–2904 (2009)

Gain More from PUE: Assessing Data Center Infrastructure Power Adaptability

Daniel Schlitt[1(✉)], Gunnar Schomaker[1], and Wolfgang Nebel[2]

[1] OFFIS – Institute for Information Technology, Oldenburg, Germany
{schlitt,schomaker}@offis.de
[2] C.v.O. University of Oldenburg, Oldenburg, Germany
nebel@informatik.uni-oldenburg.de

Abstract. The power usage effectiveness (PUE) for data centers is used by operators as KPI to measure the absolute infrastructure power overhead. However, this only draws conclusions on static or average operation conditions during an usual annual time period. For analyzing the aspect of dynamics in the IT to infrastructure power relation, we propose two new metrics. First, the power variability (PVar). It simply indicates the relative rates and heights of power variations. Second, the infrastructure power adaptability (IPA). It relates the power variabilities and relative average deviations of IT and infrastructure power in order to represent the scalability and adaptability of the infrastructure to the IT demands. Both metrics use the same input data also needed for a continuous PUE calculation. Thus, the applicability in a data center running a PUE-metering can be ensured. In an evaluation, we applied the IPA on power traces of a container data center (in the following denoted as CDC) and compared the results with PUE scalability, a metric with the same scope. The comparison showed, that IPA covers more operating states and is therefore more robust and reliable than its counterpart.

1 Introduction

Data centers are identified as system relevant for future evolutions of IT services of any kind [1]. Their growing demands, induced by society needs and IT evolution, have defined an increasing electric power consuming industry [2]. An ongoing challenge is the procurement of energy in the order of multiple mega watts covered by renewable energy whilst ensuring the availability of IT services. Because of the dominance of the availability requirement, some data centers and their operators are blamed as reckless energy squanderers [3]. This slightly distorted perspective on energy procurement neglects the continuous effort made to operate data centers as energy efficient as possible. One main objective in this domain is the reduction of operational costs. Currently, the energy demand covers a significant fraction of the operational costs. Thus, the reduced usage and economic procurement is highly desired! Especially, technical achievements like server virtualization, interoperability of cloud services, and a high inter connect bandwidth will possibly increase the competition in this market [4].

S. Klingert et al. (Eds.): E^{2}DC 2014, LNCS 8945, pp. 152–166, 2015.
DOI: 10.1007/978-3-319-15786-3_10

Such a growing market competition and the public importance of green IT possibly argues the intention to establish a simple performance indicator of efficiency. Such indicator has to emphasize the practical benefit in relation to the resources needed to attain it. It could then be used to reflect the effort made being efficient and not wasting money or natural resources. One established metric used to compare data centers with each other is the PUE. It is defined by the ratio of total facility energy demand to energy demand of IT hardware systems. The metric's purpose is to reflect the power overhead needed to run the IT under desired conditions. The main problem is the superficial and isolated consideration of PUE values, which misleads to wrong assumptions. As PUE does not consider the useful work done by the data center nor its performance, two data centers sharing the same PUE value may not have the same overall efficiency. Thus, PUE can be used as infrastructure efficiency indicator for a single data center site but it cannot be used to compare different sites.

The construction and the resulting power overhead of a data center depends on several parameters and constraints, where location is a dominating one. The recent evolution and resulting variety of air conditioning systems exemplifies the needs to adapt to these operational differences. An example is given by the capability to use direct or indirect free cooling systems. Simplified, the influence of location on power overhead depends on two parameters. One is the implementation of temperature recommendations to operate the IT within a data center [5]. The second, which cannot be influenced after a data center is built, is the course of the outdoor temperature and the expected environmental maximum. Both define the constraints how a backup refrigeration system needs to be included and sized. This second system has to support the free cooling system to ensure the operating conditions of IT. Concluding, even if two data centers are built equally and are running the same services synchronously on equal IT systems, the power overhead can differ. This occurs, if changing temperatures cause the need for the backup refrigeration system.

The previously described scenario considers the adaptation to location only and does not consider the adaptation to internal causes. The implementation of temperature recommendations defines the target temperature range of IT systems and the IT load defines the amount of cooling power needed. Assuming that the IT load differs, it will have an impact on cooling power demands [6]. So, the internal effort for providing cooling power should differ, too. Thus, it would be helpful to know whether the infrastructure can adapt to these changes or not. Further, if the infrastructure adapts, the latency and power demand needed will decide if these actions are economic or not. Finally, avoiding over-provisioning of infrastructure services by fast and economic adaptation is relevant to obtain operational efficiency.

To obtain a full view on data center infrastructure efficiency, we believe beside existing metrics, it is important to emphasize and express the capability of the infrastructure to adapt to real IT demands at run-time. Further, its application should base on existing measuring points and data to ease its application. Therefore, we propose the infrastructure power adaptability (IPA) metric in combination with the power variability (PVar) as an addition to PUE. While PUE constitutes the average absolute infrastructure power overhead, IPA represents the adaptability of infrastructure to IT power whereas PVar is an general indicator for dynamics in power behavior. By combining all three metrics, the infrastructure power behavior can be suitably described.

2 Related Work

There exists a broad spectrum of different energy efficiency metrics for data centers. The most common ones are the power usage effectiveness (PUE) and the data center infrastructure efficiency (DCiE). Both metrics are developed by The Green Grid [7]. The PUE and the DCiE relate IT and complete data center power demand to reflect the infrastructure power overhead. A drawback, results have a limited significance, since dynamics caused by varying workloads and power demands are not considered. Even the analysis of a time series of PUE values will not allow to identify the dynamic behavior sufficiently, as a constant PUE e.g. may stand for a perfectly adaptive infrastructure but it may also mean that nothing changes at all.

The PUE scalability metric was introduced as an addition to the PUE [8]. By relating the slopes of two linear functions – for the actual and the proportional PUE scalability – the metric determines the percentaged PUE scalability. It assumes that the infrastructure scales linearly with IT power. Furthermore, if infrastructure power demand reacts with delay to changes in IT power, PUE scalability will provide delusive results. The metric only correlates values at the same time point. Thus, due to a possible temporal gap it is not able to recognize the real correlation.

The infrastructure focused facility fixed/proportional overhead multiplier metric was proposed by bcs Data Centre Specialist Group [9]. It splits the infrastructure overhead into fixed and proportional portions. Hence, a theoretical dynamic power range from zero to full IT load is known, but actual dynamic is not surveyed by this. Even the successor, the fixed to variable energy ratio (FVER) [10], which relates idle to full IT productivity, does not consider typical workload and power variations in a time span.

Energy efficiency metrics with focus on computing assess power/energy demands in relation to the useful work done. Summarized, they only differ in their approach how to assess useful work. However, all of them possess a subjective component, as productive outcome (e.g. processed orders per time) of data center applications must be defined by humans. Thus, an application of such a metric is complex and unique for each data center. This finally avoids a fair comparison between different data centers. Example metrics are given by the data center performance per watt (DCPpW) [11] by Dell and the data center energy productivity (DCeP) [12] by The Green Grid. Due to mentioned definition problems, there are also eight proxy measures given, which can be used instead of useful work. These proxies reduce the useful work basically to productivity, performance, or utilization.

Besides the mentioned high-level metrics there are also known examples of system-level metrics and benchmarks for single components. They focus on energy efficiency of single server systems or storage systems connected to network and other server components. Examples are SUN Microsystems' space, watts, and performance (SWaP) metric [13], SPECpower_ssj2008 [14] by the Standard Performance Evaluation Corporation (SPEC), TPC-Energy [15] by the Transaction Processing Performance Council (TPC), and SPC-1/E [15] by the Storage Performance Council (SPC). In contrast to high-level metrics, they deliver detailed information for single systems, but they have to be combined to get a view on the whole IT surroundings. The load dependent energy efficiency (LDEE) metric [16] takes this approach to model the data center performance.

3 Conception of IPA @ PVar

We propose two new metrics, the infrastructure power adaptability (IPA) and the power variability (PVar), which are intended as an enrichment of the PUE. While the PUE indicates the relative average overhead by infrastructure components in a data center, it does not represent the dynamics of IT and infrastructure power. If for instance a data center operates a load and power management, which adjusts the number of active servers to the current resource demand, the IT power will possess a certain dynamic. With PUE alone it is not comprehensible, how dynamics in IT power influences infrastructure power, and therefore energy optimization potential is lost. With our proposed metrics, it is possible to additionally rate the variability of IT and the adaptability of infrastructure. The combined view of all three metrics will provide a detailed insight into the overall efficiency of data center infrastructures.

3.1 Power Variability

In addition to a data center's PUE also the characteristics of changes in IT and infrastructure power are important to know. This would allow that the PUE can be related to the dynamics of the data center in operation. Such power variability can be extracted from the IT and infrastructure power traces, which are measured to determine the annual average PUE.

The variability is measured in a statistical manner. First, the power trace has to be flattened to filter the irrelevant peaks according to cooling reaction times. Averaging the data in intervals of one minute is suitable, because random noise can be reduced, whereas greater changes in power demand (e.g. high workloads, de-/activation of servers, fan speed changes) are still fully distinguishable. If the input data is only available in a less detailed time resolution, it is still possible to compute variability. However, a time resolution of more than 30 min is not recommended, because there exist optimization techniques, which influences dynamic in a 30 min scale. One example would be the load and power management of servers presented by Hoyer et al [17]. With time resolutions above 30 min, variability induced by such techniques could not be covered.

After the input data preparation, the second step is to determine the power variation. To accommodate to usable optimization intervals regarding IT and infrastructure power dynamics, the power trace is divided into suitable segments. Only variability regarding those segments will be analyzed, as it corresponds to the power variation utilizable for optimization. The segments cover a time span of $t + 1$ time points and overlap with the neighboring segments by one time point. By this, a power trace with n time points will consist of n/t segments. An example is depicted in Fig. 1.

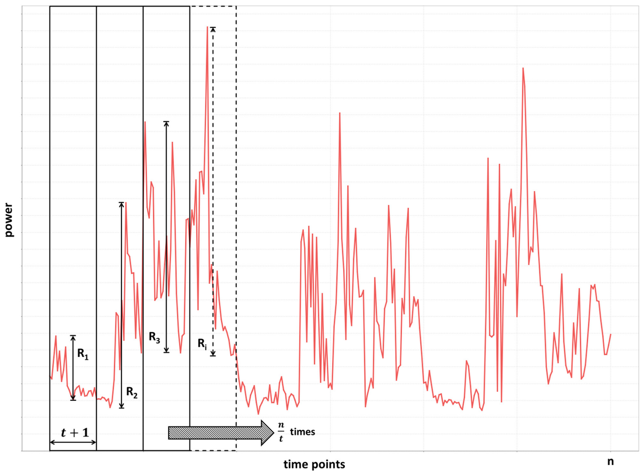


Fig. 1. Power variability estimation by averaging relative segment ranges. The relative range for segment i is computed by dividing the range R_i by maximum of segment i.

For every segment i the relative range RR_i will be estimated as shown in (1), where x^i_{max} and x^i_{min} denote the maximum and minimum value for segment i, respectively. The relative range is chosen over absolute differences, as the variabilities of different power traces have to be comparable, which is not the case when absolute values are used.

$$RR_i = \frac{x^i_{max} - x^i_{min}}{x^i_{max}} \tag{1}$$

A single representative for power variability $PVar$ is given by the average of the relative ranges of every power trace segment, cf. (2). The value is normalized between 0 and 1, where 0 represents a trace with constant power and a value of 1 means the trace varies extremely in every segment. Values in between represent the average relative variation range throughout the trace.

$$PVar = \frac{1}{n} \sum_{i=1}^{n} RR_i \tag{2}$$

3.2 Infrastructure Power Adaptability

The infrastructure power adaptability is a metric, which evaluates how good power demand of data center infrastructure scales with variations in IT power demand. These variations may be induced for instance by using workload dependent power management techniques, which switch servers off at times of low utilization and turn them on again at increasing demands. In times of lowered IT power load also the cooling demand decreases. Depending on the adaptability of the cooling components, they may adjust to the actual cooling demand and also use less power or they may still operate on a constant level. Our proposed metric represents this degree of adaptability.

In contrast to the power variability metric, not only the power load change rate and its magnitude are relevant for the rating but also the duration of altered load levels. Thus, the relation of power variability of IT and infrastructure power alone is not sufficient to indicate adaptability. In addition to the relative ranges the adaptability metric makes use of deviations to the power trace's baseline, which represents the most frequent power state.

To determine the power trace's baseline, we use the mode – the most frequent power value. As power values are real numbers, the power trace will be discretized into classes with a range of about 5 % of the maximum power value, resulting in 20 classes. After that, the mode for grouped data, which lies within the modal class (the class with the highest frequency of occurrences), will be ascertained. It can be computed with (3), where L is the lower class limit, f the class frequency, and h the class interval for the modal class m, its predecessor $m-1$, or its successor $m+1$.

$$mode = L_m + h_m \frac{f_m - f_{m-1}}{2f_m - f_{m-1} - f_{m+1}} \tag{3}$$

The variability as well as the extent of variations in power load can be determined by the relative average deviation RAD about the mode, which is shown in (4) with x_i as the power values and n as the trace length.

$$RAD = \frac{\sum_{i=1}^{n} |x_i - mode|}{n \cdot mode} \tag{4}$$

The infrastructure power adaptability metric IPA combines the relative average deviations RAD of IT and infrastructure power with the corresponding power variabilities PVar, respectively. The partial results for both power traces will then be related to each other. As for IT and infrastructure two (relative) values have been multiplied, respectively, the square root of the resulting value will be used to get to a normalized final result. The final adaptability metric is shown in (5), where IT denotes the IT power and inf the infrastructure power.

$$IPA = \sqrt{\frac{PVar_{inf} \cdot RAD_{inf}}{PVar_{IT} \cdot RAD_{IT}}} \cdot 100 \tag{5}$$

By relating both infrastructure and IT power variability, the metric indicates how well infrastructure power adapts to IT power variations. The result is a percentage and may even rise above 100 % as the metric bases solely on relative values. Target would be to have a rating of 100 %, which would suggest to have a perfect adaptability of infrastructure to IT inside the data center. Values less than 100 % mean the infrastructure power only adapts to a certain percentage to IT power. If the rating is above 100 %, the infrastructure adapts well but may operate inefficiently at certain times.

As the IPA metric delivers information solely about adaptability, the IT power variability should always be appended in the way **IPA**[%]@**PVar**, e.g. **63** %@**0.24**. If power variability is not known, the IPA rating may not be distinct. For instance, there is a difference in the cases that a very good adaptability is measured either at a high or at a low power variability. In the first case the infrastructure is definitely adaptive as variability was measured and rated within the metric. In the second case there were no information about how infrastructure power behaves at varying IT power, as the IT power only marginally changed. Naturally, the adaptability is optimal if both IT power and infrastructure power are constant. Thus, the IT power variability has to be included in the final result to get an idea of the measured power data upon which the result bases.

4 Integration in LDEE Metric

The main purpose of IPA and PVar is to get an overview over dynamic power behavior inside a data center alongside to the general power relation between IT and infrastructure given by the PUE. However, IPA and PVar in conjunction with PUE may also be embedded in the load dependent energy efficiency (LDEE) metric for data centers, where it will serve as a proxy for missing infrastructure power models.

4.1 Load Dependent Energy Efficiency

The LDEE metric for data centers introduced in [16] bases on performance and power models for data center IT and infrastructure components. This approach has several advantages: (1) By using predefined models no measurements are necessary and efficiency can be ascertained for arbitrary data center workload. (2) Inefficiencies can not only be recognized but also the sources can be identified by analyzing the component models. (3) In contrast to common metrics like PUE it enables a fair comparison of data center energy efficiency through abstraction from real measurements. (4) Possible changes in data center configurations can be explored by substituting/adding the appropriate component models.

The LDEE metric, shown in (6), is designed as a function on the data center hardware configuration. It needs data center workload and outside temperature as inputs to estimate the corresponding energy efficiency. The inputs are propagated to the combined data center performance /power models, which are also either represented as functions. The models then return data center performance and power, whose relation represents data center energy efficiency.

$$LDEE(load, T) = \frac{perf_{DC}(load)}{power_{DC}(load, T)} \tag{6}$$

The combined performance and power models are organized similarly. At first, a load and power management abstracting behavioral model (basing on Schroeder et al. [18]) estimates the workload distribution in the data center. The diverse load levels then are input to the performance and power component models. In the combined data center performance model, the maximum performance of IT components will be scaled with the load. After that, the normalized performance of server, storage, and network is aggregated in a weighted manner. In the combined power model, depicted in Fig. 2, the load has a direct influence on the IT component power, which will be estimated with the specific power models. The UPS power model then computes the complete IT power, which is the input for the climate control model as IT power is nearly completely converted to thermal power, which has to be handled by climate control. IT and climate power then is added to data center power.

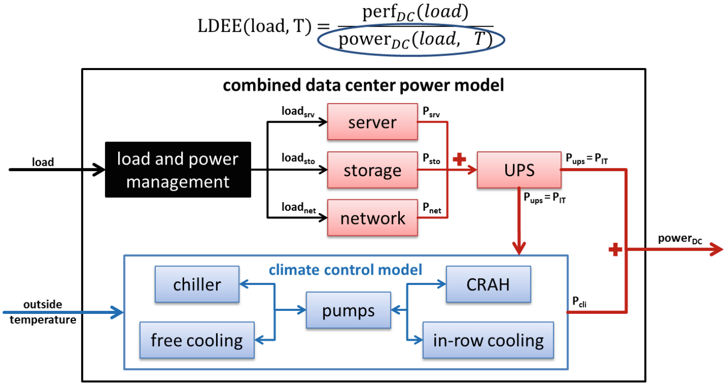


Fig. 2. Load dependent DC power model

4.2 Proxy for Infrastructure Power Models

LDEE bases strongly on power models for all relevant data center components. However, particularly for cooling components the availability of suitable power models is still low. Also, many power models have to be characterized for the particular hardware which means input and output data (cooling capacity and electrical power) has to be measured. Dependencies between cooling components complicate the building of accurate models. In the end, if there are no pre-characterized power models for the specific cooling hardware, it is easier and faster to use metrics as a proxy for power models.

A suitable choice for proxy metrics is the combination of PUE, PVar, and IPA. A big advantage is the fact, that the only data traces needed are the complete IT and infrastructure power. As the PUE is very common by now, a lot of data centers already measure the necessary data, so no further measurement equipment would be needed.

The LDEE metric with proxy infrastructure metrics utilizes an alternate computation of data center power. Instead of the function $power_{DC}(load, T)$ both the IT $p_{IT}()$ and infrastructure power $p_{inf}()$ are estimated separately. The resulting LDEE metric for load $l \in \mathbb{R} \wedge 0 \leq l \leq 1$ and l_{PUE} is shown in following (7) till (10).

$$LDEE(l, l_{PUE}) = \frac{perf_{DC}(l)}{p_{IT}(l) + p_{inf}(l, l_{PUE})} \tag{7}$$

$$p_{IT}(l) = P_{srv} + P_{sto} + P_{net} \tag{8}$$

$$p_{inf}(l, l_{PUE}) = p_{IT}(l_{PUE}) \cdot (PUE - 1) \cdot f(l, l_{PUE}) \tag{9}$$

$$f(l, l_{PUE}) = 1 - \frac{IPA}{100}\left(1 - \frac{p_{IT}(l)}{p_{IT}(l_{PUE})}\right) \tag{10}$$

While l is the data center load for which the energy efficiency is computed, l_{PUE} represents the (average) load level at which the PUE was determined. The IT power p_{IT} consists of the load dependent server, storage, and network power relating to the IT power model outputs in Fig. 2. The infrastructure power p_{inf} is composed of a constant and a dynamic part. The constant power is computed with help of the PUE and IT power at the corresponding load level. The dynamic load dependent part results from multiplication with a factor f, which is the quotient of current IT power to reference (PUE) IT power adjusted by the IPA.

5 Evaluation

In this section the IPA @ PVar metric will be applied on different data center power sets. These sets are assigned to two categories. (1) Manually created data sets representing extremes of IT and infrastructure power variability and adaptability. (2) Generated realistic power traces by utilizing real work loads on a container data center (CDC) power model. As The Green Grid's proposed PUE scalability metric has the same target as the IPA, results of both metrics will be compared and analyzed.

5.1 PUE Scalability

The Green Grid's PUE scalability is seen as an addition to PUE and has the aim to inform data center operators about the infrastructure's ability to scale the total facility power to accommodate IT power changes [8]. Similarly to the IPA metric it makes use of the power traces measured for PUE calculation and analyzes them in a statistical manner.

According to Azevedo et al. [8] PUE scalability is defined by the relation of the slope of actual PUE scalability ($\mathrm{m}_{\mathrm{Actual}}$) to the slope of proportional PUE scalability (mean PUE/$\mathrm{m}_{\mathrm{PUE}}$), cf. (11).

$$\mathrm{PUEscalability} = \frac{\mathrm{m}_{\mathrm{Actual}}}{\mathrm{m}_{\mathrm{PUE}}} 100\,\% \tag{11}$$

m_{Actual} is determined by a linear approximation of the IT (P_{IT}) to total facility power (P_{DC}) relation in the form $\mathrm{P_{DC}} = \mathrm{m_{Actual}P_{IT}} + \mathrm{b}$ using the least squares method. m_{PUE} is total facility energy usage divided by IT energy usage for a measuring period.

5.2 Extremes of IT/Infrastructure Dependency

The first data sets represent three cases: (a) A fully adaptive data center infrastructure, (b) a constant infrastructure power, and (c) a disproportional infrastructure power behaviour. The data is manually created to verify IPA @ PVar and PUE scalability results for these extreme cases. The corresponding results are summarized in Table 1. For the PVar metric a segment size (cf. Sect. 3.1) of five time points has been chosen.

The infrastructure power in data set A ($\mathrm{DS_A}$), shown in Fig. 3, is half the IT power for every time point, resulting in a constant PUE. Thus, infrastructure is fully adaptive. The metrics behave as expected as both come to an adaptability rating of 100 %.

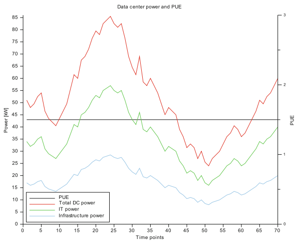


Fig. 3. $\mathbf{DS_A}$: Fully adaptive infrastructure power behavior

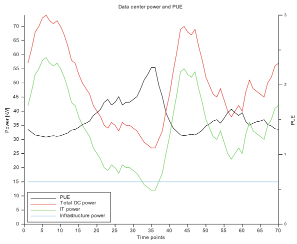


Fig. 4. $\mathbf{DS_B}$: Constant infrastructure power (no adaptability)

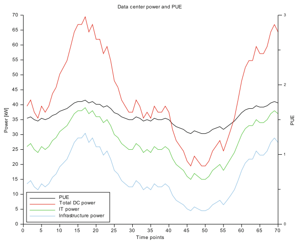


Fig. 5. $\mathbf{DS_C}$: Disproportionate infrastructure power behavior

The data set B ($\mathrm{DS_B}$, Fig. 4) shows a constant infrastructure power, which implies that the infrastructure is not adaptive at all. According to this, the metrics' results should be 0 %. However, only the IPA represents this case correctly. The PUE scalability returns still a result of 70 %. The reason for this can be found in the definition, where IT power is related to total facility power instead of infrastructure power. And due to the fact that DC power always varies with IT power, PUE scalability may even at constant infrastructure power be high. The result is not influenced by infrastructure power scalability, but it is by the absolute values.

Data set C ($\mathrm{DS_C}$, Fig. 5) represents the case of a disproportionate infrastructure power behavior, i.e. infrastructure power scales (relatively) better than IT power. A possible reason for such behavior could be the usage of plenty fans, whose power demands rise quadratically with fan speed (cooling demand).

The IPA metric covers this case with ratings above 100 %. This means, infrastructure is highly adaptive. However, there are also operational states in which the infrastructure runs inefficient, otherwise infrastructure power could not scale this well. The higher the result, the higher these inefficiencies are. In contrast, PUE scalability is limited to 100 % by definition. Although the metric would represent disproportionality correctly (applying the metric for $\mathrm{DS_C}$ results in 131.6 %), values above 100 % are according to [8] defined as invalid as they would indicate an inadequate number of samples.

5.3 Realistic Data Center Operation

The second evaluation will analyze the results of IPA @ PVar and PUE scalability for data generated by a data center simulation tool [6]. This simulation tool consists of two major components: A load and power management of virtual machines (VMs) and servers and a combined data center power model. The load and power management bases on the work by Hoyer et al. [17] and has been implemented and extended in the research project AC4DC [6]. The load and power management is connected to a data center power model, which is composed of power models for IT and infrastructure components, as shown in Fig. 2.

For this evaluation a CDC with six 19" racks has been modeled, configured, and characterized. Each rack was assembled with 21 Hewlett-Packard ProLiant DL380 G6 servers with two Intel Xeon X5670 6-core CPUs and 96GB RAM (126 servers in total). For the power supply a UPS with two modules of each 20kW power capacity was used and the cooling chain with a cooling capacity of 60kW consisted of six electronically commutated (EC) fans in the raised floor, a free cooling system, and pumps. The single evaluation runs have been performed with different sets of virtual machines, whose utilization profiles have been measured in productive operation. These utilization profiles were the input data to the load and power management, which determined the dynamic allocation of VMs to servers and managed actions like VM migrations and server power downs. With help of the power models the resulting IT, infrastructure, and total facility power could be estimated. The simulation time was one week (10080 min/time points).

By using the simulation tool, four data sets with different VM sets and therefore different characteristics have been generated. Then, the IPA and PUE scalability metrics have been applied to the generated power traces. The results are presented in Table 1.

The first simulation was performed with 900 VMs with partially matching utilization profiles (low/high utilization at same time points) generating the data set D ($\mathrm{DS_D}$, Fig. 6). The corresponding ratings of IPA and PUE scalability are quite divergent. While PUE scalability hypothesizes that the data center infrastructure power scales very well at this work loads (86 %), it is much less adaptive according to IPA (27 %). As already seen with $\mathrm{DS_B}$, the mismatch is caused by the PUE scalability, which focuses rather on absolute infrastructure power than its variability or scalability.

Table 1. Metric results

Metric	DS_A	DS_B	DS_C	DS_D	DS_E	DS_F	DS_G
meanPUE	1.50	1.44	1.58	1.22	1.12	1.19	1.10
PUEscalability [%]	100	69.5	100	86.4	94.5	88.6	98.6
IPA [%]	100	0	196.3	27.0	56.0	29.1	85.6
PVar_IT	0.17	0.24	0.14	0.34	0.35	0.13	0.19
PVar_Inf	0.17	0	0.26	0.11	0.22	0.03	0.17

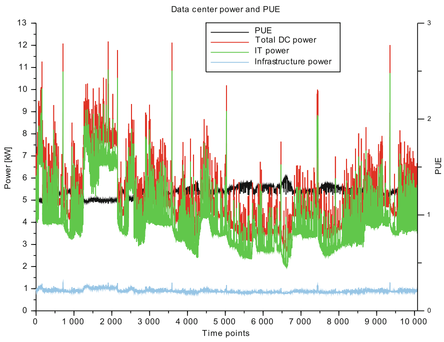


Fig. 6. $\textbf{DS}_\textbf{D}$: CDC, 126 servers, 900 VMs

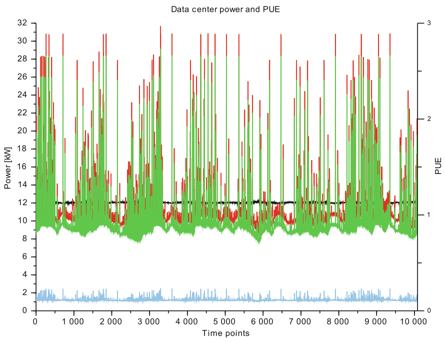


Fig. 7. $\textbf{DS}_\textbf{E}$: CDC, 126 servers, 1300 VMs

Data set E (DS_E, Fig. 7) bases on a simulation with 1300 VMs with similar utilization profiles, so that the servers may be fully utilized at times. This results for both metrics in a higher scalability rating: IPA with 56 % and PUE scalability with 95 %. Although the same data center components and the same configuration has been used, the infrastructure behaved more adaptive. Thus, the influence of the work loads in a data center on (energy) efficiency becomes evident. With higher utilization and therefore IT power load, the static part of infrastructure power decreases in relation to the dynamic part. This is a good example for the necessity of the LDEE metric (cf. Sect. 4), which can rate a data center for all work load levels or operational states, respectively.

In data set F (DS_F, Fig. 8) the power traces have been generated by simulating the load and power management with 1000 VMs with highly diverse utilization profiles. Consequently, periodical peaks are averaged out so that the IT power load has a small range with low variability (PVar of 0.13). Similar to data set E the divergence between IPA and PUE scalability is very high. However, due to the low IT power variability, results are not significant, as there was no data for any conclusions on adaptability.

For the generation of data set G (DS_G, Fig. 9) the CDC configuration in the simulation has been changed to fully utilize the cooling chain, i.e. to generate a cooling demand of up to 60kW. To achieve this, the server type has been changed to six NEC Corporation Express5800/A1080a-E per rack (36 servers in total), each containing eight Intel Xeon X7560 8-core CPUs and 768GB RAM. Due to

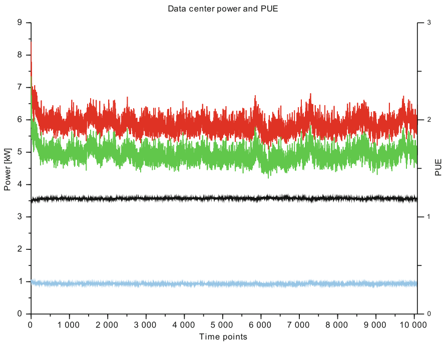


Fig. 8. DS$_F$: CDC, 126 serv., 1000 VMs

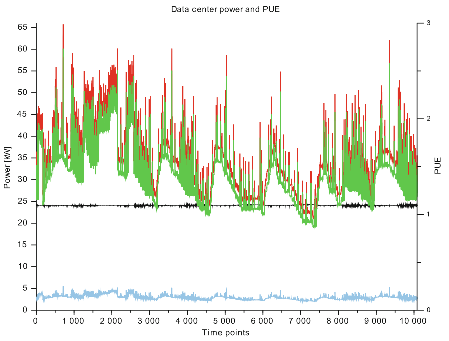


Fig. 9. DS$_G$: CDC, 36 serv., 1100 VMs

the higher maximum power demand of this server type, another UPS with 40kW capacity has been added. By utilizing the servers with 1100 VMs with matching utilization profiles, IT power ranges from 20kW up to 60kW at times. Compared to the other data sets, in this case IPA delivers a relatively high infrastructure adaptability rating (86 %). The reason is analog to data set E: The relative static overhead of infrastructure power decreases with higher absolute power values.

5.4 Proxy for Power Models

In another evaluation we assessed the practicability of IPA as proxy for infrastructure power models in the LDEE metric, cf. Sect. 4. Therefore, we used (9) and (10) to estimate the infrastructure power p_{inf} of a CDC and compared it to the power traces generated by the infrastructure power models. Input data were the generated IT power traces of the CDC model (DS$_D$ to DS$_G$) and the corresponding IT load traces as well as IPA ratings. Additionally, the average load in the time span of PUE computation (l_{PUE}) for each data set was used.

With given IT load and power the corresponding infrastructure power for every time point has been computed and subsequently compared to the generated infrastructure power. The error of computed to generated IT power for each data set is shown in Table 2. With a mean squared relative error of 1.1 % or less for every data set, the proxy is a good option for estimating dynamic infrastructure power in data centers without a complete infrastructure power model.

Table 2. Errors using IPA as proxy for infrastructure power models in LDEE

Error	DS$_D$	DS$_E$	DS$_F$	DS$_G$
Mean squared absolute error [kW]	0.007	0.016	0.000	0.029
Mean squared relative error [%]	0.8	1.1	0.04	1.1
Maximum absolute error [kW]	0.023	0.079	0.001	0.250
Maximum relative error [%]	1.9	4.1	0.1	4.5

However, if infrastructure models are available, they should always be preferred. By using proxies all advantages of LDEE, listed in Sect. 4.1, will be lost.

6 Conclusion

Power usage effectiveness, the most common energy efficiency metric for data centers, only considers absolute (static) power overhead by infrastructure components. In the absence of viable alternatives for rating the dynamic power behavior, we proposed two new metrics with focus on the dynamics in data center power which can be seen as an addition to the PUE, as they base on the same input data. The power variability (PVar) represents frequency and relative height of changes in power demand (IT, infrastructure, facility). The infrastructure power adaptability (IPA) relates IT to infrastructure power variations to ascertain the ability of infrastructure components to adapt to changes in IT operation regarding power demand.

Compared to the PUE scalability metric, which has the same target as IPA @ PVar, the combination of our proposed metrics is more reliable and robust. The evaluation showed that PUE scalability cannot represent certain cases like constant infrastructure power or disproportionate behavior correctly. On the contrary, IPA @ PVar provides proper results on a percentage scale from 0 % for constant infrastructure power to 100 % for perfect adaptability and values above 100 % for disproportionate scaling. Another analysis applying the metrics on power traces of a CDC confirmed these findings.

As a further use case IPA @ PVar can be used as a proxy for missing infrastructure power models in the LDEE metric. Compared to power model data the proxy has a mean squared relative error of about 1 % (4.5 % max), which makes it a suitable alternative. The only requirements are continuous PUE measurements and logging of data center utilization.

References

1. Gartner, Eight Critical Forces That Will Shape Enterprise Data Center Strategies for the Next Five Years. Gartner Inc (2013)
2. The Climate Group, SMART 2020: enabling the low carbon economy in the information age. Techical Report (2008). http://www.smart2020.org/_assets/files/02_Smart2020Report.pdf
3. Greenpeace Report, How Clean is Your Cloud? (2012). http://www.greenpeace.org/international/Global/international/publications/climate/2012/iCoal/HowCleanisYourCloud.pdf
4. Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., Ghalsasi, A.: Cloud computing - the business perspective. Decis. Support Syst. **51**(1), 176–189 (2011)
5. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Thermal Guidelines for Data Processing Environments, ASHRAE Technical Committee 9.9 (2012)
6. Schomaker, G., Schlitt, D., Schröder, K., Hoyer, M.: AC4DC - Adaptive Computing for green Data Centers, the European Union through the Future and Emerging Technologies programme (2011). http://www.ac4dc.com

7. Belady, C., Rawson, A., Pflueger, J., Cader, T.: The Green Grid Data Center Power Efficiency Metrics: PUE and DCiE (2008)
8. Azevedo, D., French, D.A., Power, E.N.: Pue: a comprehensive examination of the metric (2012)
9. Newcombe, L.: Data centre energy efficiency metrics. BCS Data Centre Specialist Group (2011)
10. Newcombe, L., Limbuwala, Z., Latham, P., Smith, V.: Data centre Fixed to Variable Energy Ratio metric DC-FVER (2012)
11. Pflueger, J.: Re-defining the 'green' data center (2008)
12. Haas, J., Monroe, M., Pflueger, J., Pouchet, J., Snelling, P., Rawson, A., Rawson, F.: Proxy proposals for measuring data center productivity. The Green Grid, Beaverton (2009)
13. Rivoire, S., Shah, M., Ranganatban, P., Kozyrakis, C., Meza, J.: Models and metrics to enable energy-efficiency optimizations. Computer **40**(12), 39–48 (2007)
14. Lange, K.: Identifying shades of green: the specpower benchmarks. Computer **42**(3), 95–97 (2009)
15. Poess, M., Nambiar, R., Vaid, K., Stephens Jr., J., Huppler, K., Haines, E.: Energy benchmarks: a detailed analysis. In: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, pp. 131–140. ACM (2010)
16. Schlitt, D., Nebel, W.: Load dependent data center energy efficiency metric based on component models. In: Proceedings of 2012 International Conference on Energy Aware Computing (ICEAC 2012), pp. 20–25. IEEE (2012)
17. Hoyer, M., Schröder, K., Schlitt, D., Nebel, W.: Proactive dynamic resource management in virtualized data centers. In: Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking, pp. 11–20. ACM (2011)
18. Schröder, K., Nebel, W.: Behavioral model for cloud aware load and power management. In: Proceedings of the 2013 International Workshop on Hot Topics in Cloud Services (HotTopiCS '13), pp. 19–26. ACM (2013)

Author Index