A Resource Accounting and Charging System in Condor Environment

Csongor Somogyi, Zoltán László, and Imre Szeberényi

BUTE Dept. of Control Engineering and Information Technology, Budapest, Hungary
{csongor, laszlo, szebi}@iit.bme.hu

Abstract. The authors' aim is to create a resource accounting and charging system for the Grid environment that can be set up as part of the supercomputing Grid prototype, which is developed in an ongoing domestic project. Stiller, Gerke, Reichl and Flury [1,2] has introduced a distributed accounting and charging concept, in which they point out that it could be applied in Grid context as well. This model has been adopted and improved by the authors in the given context; they had created the specification of a proposed solution at algorithm level. The necessary data structures and its query interfaces have been defined. The metering, accounting and pricing processes have been implemented. The system has been adapted to the Condor workload management system. The resulting application has been deployed at the departmental cluster. The authors funnel is to continue developing the remaining modules of the system and bring out the completed version as the part of the professional services for the Grid.
This paper introduces the applied model, the specification that was built upon it, the results of the implementation and the operating test environment.

Keywords: Grid, Resource accounting and charging, Condor

1 Introduction and Motivation

1.1 Professional Services for Commercial Grids

It is necessary to consider a kind of infrastructure, which can give access to exploit many types of resources in mass without either interfering with the legacy environment of resource providers or constraining the resource consumers in the method of access, and it has to match the security and interoperability requirements, which are recently becoming more important. System scalability and high grade of diversity of utilization are also very important factors. This is the *Grid* initiative. The final result of this achievement is a proposal of standard.

The challenging requirements against the Grid resource management have been identified by the researchers of the Globus meta-computing system [3]. The collected experiences led to draw the initial concept of the Grid architecture, a four-layer "hourglass" model [4]. They have started standardizing the architecture through the *Open Grid Services Architecture* model [5] using Web Services.

H. Kosch, L. Böszörményi, H. Hellwagner (Eds.): Euro-Par 2003, LNCS 2790, pp. 417–420, 2003.

1.2 Research Context

Base services of the Grid may include the resource representation abstractions, resource management, informational services, data connectivity (data repositories, application repositories, file transfer, etc.) and the underlying communication protocols.

On the other hand, users or consumers can only exploit the benefits given by Grid providers if they have adequate access. We have to define what the most important accessibility features are:

- application development environment (like P-GRADE [7])
- refined user interface for the management of resource usage
- accounting and charging of resource usage and payment services
- user support (documentation, help, downloads, etc.)

The *Hungarian Supercomputing Grid* project [6] supported by the home Ministry of Education has been created to establish the first working prototype of an interior Grid. Their aim is to suit the above mentioned demands, set up against a professional Grid – except the user support, which is a very costly business.

The authors of this article have been developing the accounting and charging infrastructure to be applied in the Hungarian SuperGrid. On the other hand, their objective is to provide a flexible, sophisticated toolkit, which can be easily aligned with the requirements of the evolving *Grid architecture*.

The authors are also involved in the EASYCOMP[1] – IST Future and Emerging Technologies program – project. The goal of the project is to develop the foundation of such composition technology, which should enable users to compose (standard) components powerfully and easily.

2 System Specification

We have been applying in the field of Grid computing the charging and accounting model described by Stiller, Gerke, Reichl and Flury [1]. They started their work in order to elaborate the basis of the commerce of network (in the so-called CATI project — Charging and Accounting Technologies for the Internet) but they pointed out that this architecture is capable of being implemented in Grid context as well [2] because of its generality and modularity, which can be exploited in such a distributed environment like the *Grid*. The work of the authors of this article was to improve this model to adapt it to the Grid context.

The CATI researchers have identified and decomposed the terms related to accounting and charging. This decomposition contains the abstract definitions of the terms *metering, mediating, accounting, charging and billing* and the schema of their relations.

[1] "Work partially supported by the European Community under the IST programme – Future and Emerging Technologies, contract IST-1999-14191 – EASYCOMP. The authors are solely responsible for the content of this paper. It does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of data appearing therein."

The authors of this article examined the mentioned model in the context of Grid and proposed an escalation of the original model.

The three main subsystems (*accounting* – involving metering and mediation processes –, *charging* and *billing*) have been decomposed into their atomic elements, which have been matched with software components. The Grid related processes, which pass information between the Grid and the accounting and charging process, have been identified. The necessary data abstractions have been defined. These include the representations of Grid related entities (e.g. *resource*, *resource user*), the parameters that are exchanged between the atomic processes of the system and the data constructs that are stored or used during the entire process (*like accounting record*, *pricing rule*, *charging record* and *receipt entries*). For each of these abstractions, the corresponding data structures have been declared.

The process flow has been summarized in a data-flow model, while the data abstractions are represented as an entity-relationship model. Finally, a textual description of the dynamic behavior of the accounting and charging processes has been created.

Both the *logical*, which was briefly introduced above, and the *physical* specification of the system have been completed. The most important requirement, distributiveness, is fulfilled. The metering and mediation modules should only be placed locally at the resource, while the rest of the components can be placed independently anywhere. Surely, one component can serve an arbitrary set of resources or resource users, it does not matter how the information at various levels (accounting, charging or billing) are grouped together. Actually, this approach provides us balanced information processing and storage.

3 Implementation and Results

The implementation of the fundamental components (metering, accounting and pricing) has been completed. CPU time consumption has been chosen as sample measured parameter. SQL and HTTP/XML are used for interfacing the different components. Later, these can be easily changed into Web Services. Development work is done in UNIX environment using mostly Perl and partially C languages, Perl DBI and GTop libraries.

Condor system [8] is the selected local job manager infrastructure. Condor can run a job wrapper each time a job is started on a given host. This wrapper was developed in order to establish binding between the job manager and the accounting subsystem.

The developed parts of the system are deployed at the departmental Condor cluster. Test runs are performed demonstrating the operation of the accounting and charging system.

While the structure and the behavior of the most accounting and charging components are clarified, metering still faces with different challenges. The various platforms treat resource consumption parameters, especially *process accounting* parameters, in different ways. On the other hand, there are no stan-

dard(ized) units defined. Since our measuring method takes resource usage snapshots at equal time-intervals, sampling error can be perceived. Since we need to know the resource consumption at a given moment as well as the total amount of consumption at the resource usage completion, we have to examine what accuracy is necessary and may have to combine the usage of different operating system provided ways of resource usage accounting in order to achieve accuracy precisely. We are investigating these questions in the near future.

4 Summary and Conclusions

The initial accounting and charging model has been extended in order to fit the requirements against the Grid context. The authors have completed the specification of the proposed solution. Partial development of the system, the accounting and the pricing modules have been completed. The application has been merged with the Condor workload management environment and the test system has been set up at the departmental cluster. Test runs have been performed in order to demonstrate the operation of the system. Our funnel is to continue the development of the remaining parts of the system, integrate the solution with the mentioned domestic supercomputing Grid and investigate the metering related issues.

References

1. B. Stiller, J. Gerke, P. Reichl and P. Flury: A Generic and Modular Internet Charging System for the Cumulus Pricing Scheme. Journal of Network and Systems Management, **3(9)** (September 2001) 293–325
2. B. Stiller, J. Gerke, P. Reichl, P. Flury and Hasan: Charging Distributed Services of a Computational Grid Architecture. IEEE International Symposium on Cluster Computing (CCGrid 2001), Workshop on Internet QoS for the Global Computing (IQ 2001), (May 2001) 596–601
3. K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke.: A Resource Management Architecture for Metacomputing Systems. Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing (1998)
4. I. Foster, C. Kesselman and S. Tuecke.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, **15(3)** (2001)
5. I. Foster, C. Kesselman, J. Nick and S. Tuecke: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum (June 22, 2002)
6. IKTA-00075/2001, Hungarian Supercomputing Grid. Tender for Information and Communication Technology Applications by the Hungarian Ministry of Education, Budapest (November 2002)
7. P. Kacsuk: Parallel Program Development and Execution in the Grid. Proc. of PARELEC 2002, Warsaw (2002)
8. Douglas Thain and Miron Livny: Condor and the Grid. in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, Grid Computing: Making the Global Infrastructure a Reality, John Wiley (2003)