Grid Scheduling and Monitoring*

Miroslav Ruda[1] and Hana Rudová[2]

[1] Institute of Computer Science [2] Faculty of Informatics
Masaryk University
Botanická 68a, Brno 602 00, Czech Republic
ruda@ics.muni.cz, hanka@fi.muni.cz

Grids [4] are distributed heterogenous systems of computational, data, and information resources connected by high speed network. Their goal is to create an integrated virtual system of the resources available to users. Efficient use of Grids is the subject of many recent projects. However, there are still many areas which were not studied sufficiently. We will try to describe some problems where ideas from planning and scheduling research areas could be applied.

Our first goal was the study of the *Grid scheduling* problem [3]. Here we are generally looking for an optimal assignment of jobs to resources. The resources (e.g., CPUs, storage space, software licenses, network bandwidth) often have limited capacity and their characteristics change in time (for example due to machine breakdown or network disconnection). Jobs dynamically arrive from users together with their request for some resources. The idea is to optimize placement of jobs to resources with respect to criteria like minimal makespan, minimal cost of assigned resources, or maximal throughput. The problem can be also seen as a *matching* process [6] where we would like to match jobs with particular resources. Jobs and resources can be understood as some entities which *provide or require a service*. They advertise their characteristics and requirements in *classified advertisements (ClassAds)*. A designated matchmaking service matches ClassAds in a manner that satisfies the specified constraints. A simple example of the ClassAd describing a workstation and a job is described in Figure 1. A *bilateral matching* problem can be defined to

```
Machine = { Memory = 64,       Job = { Memory = 31,
            System = solaris,          Requirement = Machine.System == solaris &&
            Disk = 323496                            Machine.Disk >= 10000 &&
            Rank = 1/Job.Memory }                    Memory =< Machine.Memory }
```

Figure 1: ClassAds for a workstation and a job

find out a possible compatibility between both ClassAds. These ideas were extended in order to provide for example more information about matching results or to support a matching between multiple requests/jobs and resources [5]. A simple constraint solving mechanism was applied to solve variety of matching problems defined using ClassAds extensions. Node consistency, the basic local propagation, and hyper-arc consistency were applied to solve the problem. To our knowledge, this work is the only attempt trying to apply constraint methodology to scheduling of tasks over resources in the grid environment. So, we believe it is an interesting problem for application of the ideas from constrained-based scheduling. For example, we would like to present a CLP(*FD*) model [1] with cumulative resources for matching of multiple jobs to multiple resources.

Another problem is the *Grid monitoring* which allows to gather information about the state and performance of a large scale distributed system. It allows to monitor the behavior of the Grid using producers of information/data (what is the load of the system, what is the list of running jobs, what is the current performance for the given running job, etc.) and consumers who

*This work is supported by the Ministry of Education of the Czech Republic under the research intent No. 0021622419.

would like to be informed about certain behavior of the system. For description of producer's or consumer's capabilities, ClassAds mechanism [2] can be again applied and the idea of matching can help to solve the problem. Also data can introduce additional constraints to be satisfied. A simple example of ClassAd description for a matching problem is shown in Figure 2.

```
Producer = { SupportedProtocol={http,https}; }
Consumer = { AcceptedProtocol={https};
             SecurityLevel=5;
             Requirements = member(AcceptedProtocol,Producer.SupportedProtocol);}
Data =     { MinSecLevel=4;
             Requirements = (Consumer.SecurityLevel>=MinSecLevel); }
```

Figure 2: ClassAds for a producer, data, and consumer

While Grid scheduling problem clearly belongs to the planning and scheduling domain, Grid monitoring does not seem to be its representative from the first point of view. However, producers can be seen as resources and consumers as some activities which want to be assigned to producers. If some producer/resource is not needed (for example nobody needs to know load of some computer), it remains free without any penalty. If some consumer/activity does not have anything to do (for example consumer would like to listen about all jobs of some user and none such job is running), the activity can have zero duration. New producers or consumers may become available from the *factory* for some price which just corresponds to the setup cost. Data specification adds some other constraints to the constrained-based scheduling model. Grid monitoring also belongs to the class of dynamic problems (e.g., new data are needed, new producers or consumers may show up).

We would like to solve the Grid monitoring problem together with Grid scheduling problem as the common CLP(*FD*) problems using system like ECL^{i}PS^e [1]. We believe that the use of standard modeling and solving techniques and tools are crucial for a successful reuse of the results. We are not aware any work which would apply constrained-based reasoning or scheduling ideas for Grid monitoring problems. We plan to compare Grid scheduling and Grid monitoring problems. It seems to be very interesting because both problems come from similar application areas, they have the same description language, they can be defined as matching problems, and many properties needed by Grid monitoring can be defined as standard features of scheduling problems. Also matching can be extended in direction towards optimisation which is often needed for Grid scheduling.

References

[1] Andrew M. Cheadle et al. ECL^{i}PS^e: An introduction. Technical Report IC-Parc-05-1, 2005.

[2] Aleš Krenek et al. C-GMA – capability-based grid monitoring architecture. Technical Report 6/2005, CESNET, Czech Republic, 2005.

[3] Pavel Fibich, Luděk Matyska, and Hana Rudová. Model of grid scheduling problem. In *AAAI'05 Workshop on Exploring Planning and Scheduling for Web Services, Grid and Autonomic Computing*. AAAI Press Technical Reports, 2005.

[4] Ian Foster and Carl Kesselman, editors. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan-Kaufmann, 2004.

[5] Chuang Liu and Ian Foster. A constraint language approach to machmaking. In *Proceedings of the 14th International Workshop on Reserch Issues on Data Endineering: Web Services for E-Commerce and E-Government Applications (RIDE'04)*. IEEE, 2004.

[6] Rajesh Raman, Miron Livny, and Marvin Solomon. Matchmaking: Distributed resource management for high throughput computing. In *Proceedings of the Seventh IEEE International Symposium on High Perfomance and Distributed Computing*, 1998.