Grid Resource Management, Scheduling and Computational Economy

Rajkumar Buyya†, David Abramson†, and Jonathan Giddy‡

† School of Computer Science and Software Engineering
Monash University, Caulfield Campus
Melbourne, Australia

‡ CRC for Enterprise Distributed Systems Technology
General Purpose South Building,
University of Queensland, St. Lucia
Brisbane, Australia

Email: {rajkumar, davida, jon}@csse.monash.edu.au

Abstract

This paper identifies the issues in resource management and scheduling in the emerging grid computing context and briefly discusses techniques for scheduling using computational economy concept. In the context of Nimrod/G system, scheduling based on the deadline and cost for parametric computing over the grid is discussed. We also highlight the future work on grid architecture for computational economy (GRACE) infrastructure that the grid tools and application developers can use.

Emergence of the Grid: Introduction

The concept of grid computing is getting popular day by day with the emergence of the Internet as a ubiquitous media and the wide spread availability of powerful computers and networks as low-cost commodity components. The computing resources and special class of scientific devices or instruments are located across various organizations around the globe. These resources could be computational systems (such as traditional supercomputers, clusters, SMPs, or even powerful desktop machines), special class of devices (such as sensors) and even storage devices. A number of applications need more computing power than can be offered by a single resource in order to solve them within a feasible/reasonable time and cost. The LAN/switch connected clusters (of computers) platform has been employed to solve computationally intensive problems [2], however they alone cannot offer the computational power demanded by applications. All this means that these geographically distributed resources need to be logically coupled together to make them work as a unified resource. This led to the popularization of a field called *grid computing*. The global computational grid is expected to drive the economy of the 21st century similar to the electric power grid that drove the economy of the 20th century.

Although wide-area distributed supercomputing has been a popular application of grid computing, there are a number of other applications that get enabled or benefit from it [6]. However, our discussion focuses mostly on the use of the grid for solving supercomputing and high-throughput computing applications, in particular.

Grid Resource Management Issues

The resources that are coupled in grid computing environment are geographically distributed and different individuals or organizations own each one of them and they have their own access policy, cost, and mechanism. The resource owners manage and control resources using their favorite resource management and scheduling system (called local scheduler) and the grid users are expected to honor that and make sure they do not interfere with resource owners' policies. They may charge different prices for different grid users for their resource usage and it may vary from time to time. The global resource management and scheduling systems (e.g., Nimrod/G [1]), popularly called grid schedulers or meta-schedulers, coordinate the user access to remote resources (e.g. accessing ANL (Argonne National Laboratory) resources in the United States from Monash University in Australia) in cooperation with local schedulers (e.g., Condor [4] and LFS [7]) via grid middleware services (e.g., Globus [5]). Traditionally, most of the schedulers followed system centric approach (e.g., they just care about system performance) in resource selection and often

In *Proceedings of the 2nd International Workshop on Global and Cluster Computing (WGCC'2000)*
Tsukuba/Tokyo, Japan, March 15 - 17, 2000.

(completely) ignore the user requirements (e.g., resource access cost in terms of money). In order to overcome this problem, we propose economic-based approach for grid resource management and scheduling. When the user submits an application for execution, they expect that the application be executed within a given deadline and cost. They also need a means for trading off the cost and the deadline. These requirements appear complex, but under a computational economy they simplify the scheduling problem and reduce the complexity involved in the design and development of grid schedulers. There is no single perfect solution that meets all user requirements, hence this promotes the need for tailored grid schedulers for each class of applications.

Computational Economy: Nimrod/G in Context

The discussion is derived from our experience in the design and development of a global grid resource management and scheduling system called Nimrod/G tailored for task farming applications [1,3,8]. The key components of Nimrod/G system are Client/User Station, a Persistent Parametric/Task-farming Engine, Scheduler, Dispatcher, and Job-wrapper. It is built using the Globus toolkit services [5] and can be easily extended to operate with any other emerging grid middleware services. In the current implementation the user can specify the deadline by which the results are needed. The scheduler tries to use the cheapest resources in order to complete the assigned experiment with a minimum cost, yet meet the user deadline. However, the grid resources are shared and their availability and load varies from time to time. When scheduler notices that it cannot meet the deadline with the current resource set, it tries to select the next cheapest resources and continues to do this until the completion of task farm application meets the (soft) deadline.

In this talk we discuss Nimrod/G work with particular emphasis on the current implementation of a static computational economy and the dynamic scheduling heuristic/algorithm used. The other scheduling algorithms that we plan to explore are based on advance resource reservation, and a dynamic computational economy based on advertised costs, trading, and auction mechanisms for the grid resource management and scheduling will also be discussed.

Grid Architecture for Computational Economy (GRACE)

We are exploring the development of a generic framework/infrastructure for grid computational economy [3]. Named GRACE (Grid Architecture for Computational Economy), the components of the framework include: global scheduler (broker), bid-manager, directory server, and bid-server working closely with grid middleware and fabrics. The GRACE-infrastructure generic interfaces (APIs) can be used by the grid tools and applications programmers to develop software supporting the computational economy.

References

1. Abramson, D., Giddy, J., and Kotler, L., *High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?*, Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), IEEE Computer Society Press, 2000.
2. Buyya, R. (ed.), *High Performance Cluster Computing: Architectures and Systems*, Volume 1 and 2, 1/e, Prentice Hall PTR, NJ, USA, 1999.
3. Buyya, R., Abramson, D., and Giddy, J., *Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid*, Proceedings of the HPC ASIA'2000, the 4th International Conference on High Performance Computing in Asia-Pacific Region, IEEE Computer Society Press, USA, 2000
4. *Condor Home Page* - http://www.cs.wisc.edu/condor/
5. *Globus Project Home Page* - http://www.globus.org
6. Leinberger, W., and Kumar, V., *Information Power Grid: The new frontier in parallel computing*?, IEEE Concurrency, Vol. 7, No. 4, Oct.–Dec. 1999, IEEE Computer Society Press, USA.
7. *LSF Home Page* - http://www.platform.com
8. *Nimrod/G Project Home Page* - http://www.dgs.monash.edu.au/~davida/nimrod.html