Chapter 4
Resource Management

INTRODUCTION

The resource management is an important component in LSDS implemented for a variety of architectures and services. This chapter considers the management of distributed resources, virtual resources and provides the requirements for resource management in large scale distributed system. A resource management system is defined as a service that is provided by a distributed network component system that manages a pool of named resources that is available such that a system-centric or job-centric performance metric is optimized. Due to issues such as extensibility, adaptability, site autonomy, QoS, and co-allocation, resource management systems is more challenging in large scale distributed computing environments. The taxonomy of resource management systems (RMS) for very large-scale network computing systems presents the variety of requirements for this tool. The taxonomy could be used to identify architectural approaches and issues that have not been fully explored in the research.

The resource management system could support different users constrains, so the multiple policies is provided. In general, requiring the RMS to support multiple policies can compel the scheduling mechanisms to solve a multi-criteria optimization problem. An important subject presented in this chapter is Agents Frameworks for resource management that offer a mechanism for distributes resources management. The chapter ends with presentation of WSRF (Web Services Resource Framework) that is the new solution for resources management based on SOA (OGSA – Open Grid Service Architecture). Resource management in Grid implies a quite large number of functionalities, from resource discovery to scheduling, execution management, status monitoring and accounting. In this section, we shall focus on scheduling systems, and we shall present the monitoring functionalities and the Grid information systems in a further section. We shall introduce here some general issues, and then we shall present taxonomy of the scheduling systems and some details regarding the scheduling mechanisms used in the most important current Grid projects.

DOI: 10.4018/978-1-61520-703-9.ch004
Wieder et al. (2006) distinguishes between two cases of Grid systems with respect to their requirements on resource management capabilities.

Case 1 is Specialized Grids for dedicated purposes, which are centered on a single or limited application domain and require high efficiency in execution. The Resource Management System (RMS) is adapted to the special application, its workflow and the available resource. Thus, the interfaces to the resources and the middleware are built according to the given requirements caused by the application scenario. While the Grid RMS is highly specialized, the handling for the user is often easier as the know-how of the application domain has been built into the system.

Case 2 is a Generic Grid Middleware, which has to cope with the complete set of the requirements to support applicability. The Grid RMS is open for many different application scenarios. In comparison to the specialized Grids, generic interfaces are required that can be adapted to many front- and backend. However, the generic nature of this approach comes at the price of additionally overhead for providing information about the application. For instance, more information about a particular job has to be provided to the middleware, such as a workflow description, scheduling objectives, policies and constraints.

The application-specific knowledge cannot be built into the middleware, and therefore must be provided at the frontend level. In this case, the consideration of security requirements is an integral aspect, which is more difficult to solve. It is possible to hide the additional RMS complexity of generic Grid infrastructures from the users or their applications by specialized components, which might be built on top of a generic middleware. Nevertheless, it can be concluded that in general a generic Grid middleware will carry additional overhead with less efficiency at the expense of broader applicability.

**Requirements for Resource Management in LSDS**

Current research is mostly focusing on presented Case 1 in which solutions are built for a dedicated Grid scenario in mind. As mentioned before, these systems are usually more efficient and will therefore remain the favorite solution for many application domains. That is, Case 1 will not become obsolete if corresponding requirements and conditions exist. However, for creating future generation Grids suitable solutions are required for Case 2.

One of the most important components of a RMS is the scheduler, which distributes the applications on the Grid resources and usually also handles the execution management. We shall present as follows a brief taxonomy for scheduling systems.

A criterion by which we can classify the schedulers is their organization. We can distinguish three categories of schedulers: centralized, hierarchical and decentralized. Most of the scheduling systems developed so far follow the centralized approach, in which the decisions about the place where a job should be executed are taken by a single entity. The decentralized model usually involves a multi-agent system, in which both the resources and the users are represented by agents, which negotiate the conditions of jobs’ execution.

Another criterion used to classify the schedulers is the way they perform the state estimation of the system. The state estimation could be done in predictive way (estimation based on heuristics, price model or/and machine learning) or in non-predictive way (considering heuristics or/and probability distribution). Some of the schedulers attempt to predict the load on the resources in the future or the execution time of the jobs, others take into account only the present information. However, the available information is always partial or stale, due to the propagation delay in large distributed systems (Darbha & Agrawal, 1998).
Related Content

A Study on the Effect of Application and Resource Characteristics on the QoS in Service Provisioning Environments
www.irma-international.org/article/study-effect-application-resource-characteristics/39042/

A Predictive Map Task Scheduler for Optimizing Data Locality in MapReduce Clusters
Mohamed Merabet, Sidi mohamed Benslimane, Mahmoud Barhamgi and Christine Bonnet (2018).
*International Journal of Grid and High Performance Computing* (pp. 1-14).
www.irma-international.org/article/a-predictive-map-task-scheduler-for-optimizing-data-locality-in-mapreduce-clusters/210172/

The Red Storm Architecture and Early Experiences with Multi-Core Processors
www.irma-international.org/article/red-storm-architecture-early-experiences/42977/

A General Framework for the Modeling and Simulation of Grid and P2P Systems
www.irma-international.org/chapter/general-framework-modeling-simulation-grid/40822/

Analysis and Evaluation of Novel Privacy Preserving Techniques for Collaborative Temporal Association Rule Mining Using Secret Sharing