Chapter 24

A Fault Tolerant Decentralized Scheduling in Large Scale Distributed Systems

Florin Pop
University “Politehnica” of Bucharest, Romania

ABSTRACT

This chapter presents a fault tolerant framework for the applications scheduling in large scale distributed systems (LSDS). Due to the specific characteristics and requirements of distributed systems, a good scheduling model should be dynamic. More specifically, it should adapt the scheduling decisions to resource state changes, which are commonly captured through monitoring. The scheduler and the monitor are two important middleware pieces that correlate their actions to ensure the high performance execution of distributed applications. The chapter presents and analyses agent based architecture for scheduling in large scale distributed systems. Then the user and resources management are presented. Optimization schemes for scheduling consider the near-optimal algorithm for distributed scheduling. The chapter presents the solution for scheduling optimization. The chapter covers and explains the fault tolerance cases for Grid environments and describes two possible scenarios for scheduling system.

INTRODUCTION

Grid computing became a very important model for resource sharing in Virtual Organizations (VOs). Grid systems allow the use of temporarily available resources, the execution of large tasks that require high computing power and large memory volumes. On the other side, resource sharing in grid systems (generally, in very large distributed systems) is more complex and asks for more complicated management policies and techniques. An important management function is task scheduling. In the case of Grid systems, task scheduling has two objectives. One objective targets the efficient use of resources, similar with schedulers found in traditional operating systems. The second objective, not less important, is related to the VO concept and aims to respond to the requirements stated by the users in a VO concerning the performance of tasks execution, such as the response time. This is
why the scheduling function has been distributed to two components: one which is closer to the resources (the local scheduler), and a second (the meta-scheduler) closer to the application. Scheduling in distributed systems has been significantly improved due to innovations proposed in Grid systems and VO management. The scheduling algorithms for large scale distributed systems (LSDS), such as the Grid systems, are subject to recent research.

The scheduling in Grid systems is very complicated. The resource heterogeneity, the size and number of tasks, the variety of policies, and the high number of constraints are some of the main characteristics that contribute to this complexity. The necessity of scheduling in Grid is sustained by the increasing of number of users and applications. The design of scheduling algorithms for a heterogeneous computing system interconnected with an arbitrary communication network is one of the actual concerns in distributed system research.

The optimization of scheduling process for Grid systems tries to provide better solutions for the selection and allocation of resources to current tasks. The scheduling optimization is very important because the scheduling is a main building block for making Grids more available to user communities. The optimization methods for Grid scheduling are the main subject of this thesis. The scheduling problem is NP-Complete. Consequently, approximation algorithms are considered, which are expected to quickly offer a solution, even if it is only near-to-optimal (Fangpeng, 2006).

QoS (quality of services) is a requirement for many Grid applications. QoS might refer to the response time, the necessary memory, etc. It might happen that these requirements are satisfied only by specific resources, so that they only these resources can be assigned for that application. Situations might become more complex when there are more tasks having QoS requirements, and several resources exist which satisfy them.

The resource allocation under QoS constrains is another subject for the optimization process.

The fault tolerance is also important in Grid. The fault tolerant solutions for Grid Scheduling are based on error recovery and re-scheduling. Two of the problems related to re-scheduling are the high cost and the lack of coping with dependent tasks. For computational intensive tasks, re-scheduling the original schedule can improve the performance. But, re-scheduling is usually costly, especially in Directed Acyclic Graphs (DAGs) where there are extra data dependencies among tasks. Current research on DAG rescheduling leaves a wide open area on optimization for the scheduling algorithms.

According with all these presented aspects, many research activities are being conducted to develop a good scheduling approach for distributed nodes. The activities vary widely in a number of characteristics, e.g. support for heterogeneous resources, objective function(s), scalability, co-scheduling methods, and assumptions about system characteristics.

The current research directions are focused on multi-criteria optimization of Grid scheduling; approaching complex task dependencies, new scheduling algorithms for real-time scenarios, backup and recovery from service failures, and optimization of data transfers (provide an optimal solution to the problem of co-scheduling).

In compliance with the new techniques in application development, it is more natural to consider schedulers closer to Grid applications. They are responsible for the management of tasks, such as allocating resources, managing the tasks for parallel execution, managing of data transfers, and correlating the events. To provide their functions, a scheduler needs information coming from monitoring services available in the platform.