Dynamic scheduling in grid systems

Maria Chtepen

Supervisor(s): Prof. Dr. ir. Bart Dhoedt, Prof. Dr. ir. Peter Vanrolleghem

Abstract—**Grid computing is a relatively new distributed computing paradigm that is rapidly gaining importance. It offers a solution to the increasing demand of extensive computational and storage power, without requiring extraordinary investments in hardware infrastructure. However, in many cases the use of grids as such is insufficient without further optimizations, such as scheduling mechanisms for efficient assignment of tasks to available resources.**

This PhD research is dedicated to dynamic scheduling, which differs from static scheduling in that it allows for re-evaluation of already taken assignment decisions during task execution. Dynamically scheduled systems are more flexible, fault-tolerant and QoS-oriented, but also more complex and demanding than their static equivalents.

Generic network-aware scheduling algorithms will be developed for grid systems with dynamically changing resource and task properties. Moreover, calibration and evaluation of the developed methods will be performed in a simulation environment and in several existing grid systems.

Keywords— **Grid computing, optimization, dynamic scheduling, QoS (Quality of Service), simulation, software development**

I. Introduction

ALREADY for a long time man is on a quest for sources of extensive computing power that will enable him to look for extra-terrestrial radio signals, find prime numbers so large that they have more than ten million digits, and to find more effective drugs to fight AIDS. These projects are so elaborate, and require so much computing power, that they would be impossible for any one computer or person to solve in a reasonable amount of time. Recently man found a solution to this problem of increasing computational demand through grid computing [1]. Grid is a system for management and aggregation of autonomous, heterogeneous computational and storage resources across geographical and administrational boundaries. The key idea behind grid is to turn computation into an infrastructure service, available on demand such as water, gas or electricity thereby promoting resource sharing and thus avoiding extraordinary investments in hardware infrastructure. However, it turned out that in many cases the use of grids as such is insufficient without further optimizations, such as an introduction of a mechanism for efficient assignment of grid tasks (jobs) to available resources. Such a mechanism is known as scheduling and can be split into two broad categories: static and dynamic [2]. In static scheduling jobs are assigned to the appropriate resources before their execution begins. Once started, they keep running on the same resource without interruption. Opposed to static, dynamic algorithms allow for reevaluation of already taken assignment decisions during job execution. They can trigger job migration or interruption based on dynamic information about the status of the system and the workload.

In grids the status of resources can change rapidly as they often belong to different organizations and have hundreds or even thousands of users. Therefore situations where resources go offline unexpectedly, computational or network load increase drastically and jobs with high priority or short deadline arrive into the system, are very common. In all of the above-mentioned cases static algorithms will not provide a satisfactory performance as they assume that initial job assignments are still optimal. Thus it is obvious that dynamic algorithms are more flexible, fault-tolerant and QoS-oriented than their static equivalents, however they are also much more complex and demanding.

The main goal of this PhD research is development of generic dynamic scheduling algorithms for efficient, fault-tolerant distribution of jobs in grid systems. The developed algorithms will take into account specific job requirements and dependencies, properties of grid resources and QoS contracts. Moreover, calibration and evaluation of the new methods will be performed in a simulation environment and in several existing grid systems.

II. Dynamic scheduling algorithms

A. Initial problem

The project at hand can be considered as a continuation of the earlier research in static scheduling techniques for distributed systems performed at the research group IBCN (UGent). Although relevant knowledge and experience available inside the group can be directly utilized for the current project, the complexity of dynamic scheduling problems will still require gradual solution evolvement.

During the first stage of this project influence of dynamic changes in grid resource properties and user requirements on job performance will be studied. In this respect the following situations can occur:

- **Situations related to job execution:** job failure, unexpected arrival of high priority jobs
- **Situations related to dynamic grid behavior:** resource failure, activation of new resources, varying workload on resources, changing resource properties (number of users, cost, . . .)
- **Situations related to dynamic job behavior:** changing priority/deadline, changing dependencies

These situations are interesting because they can potentially lead to inefficient resource usage, job failure and break-down of QoS contracts. For these reasons they should be detected and solved by dynamic scheduling algorithms. However, integration of dynamic algorithms into grid should be done very carefully, as the algorithms themselves - if not properly constructed - can lead to system performance degradation and unnecessary job migrations. Thus, there is clearly a need for some extra software/hardware components such as an advanced performance monitoring system for jobs and resources (computational, storage, network), a checkpointing mechanism that allows for restarting jobs from their last execution status registration and an accurate job length estimation system.

This part of the project will be dedicated to the development of dynamic algorithms, which make use of the above-mentioned components for scheduling of jobs with varying requirements on grids that change.

B. Job dependencies

Many applications generate jobs that are interconnected in one way or another. An interesting type of job dependencies are input dependencies, whereby a job can be executed only after its inputs have been generated by another job. These dependencies are of particular interest for the current project because they offer an opportunity for further performance optimization.

Consider an example known as optimization through the Simplex-method (see Figure 1), which optimizes a value of variable J that is a function of α and β. Initially the algorithm chooses three points on the surface α x β where the function ϕ is evaluated. In each following iteration the algorithm takes a mirror image of the worst of the three evaluated points as a new point. The procedure is repeated until the optimal solution is found. If a grid system possesses some idle computational power, the performance of the algorithm can be improved by starting image computations of all three points in advance, before the actual worst point in the initial triangle is known. In that case some redundant computations will take place but they will not negatively affect the system performance because a number of computational resources were idle anyway.

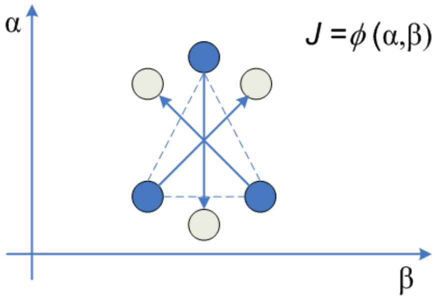

Fig. 1. Optimization with Simplex-method.

Given a set of job dependencies, this project will extend the algorithms from the previous section with dependency-aware scheduling to guarantee better resource utilization and reduction of job execution times. However, to be able to construct such algorithms there is a need for a general job interconnection representation mechanism. Here the definition of such a general representation will also be considered.

C. Network-aware scheduling

A network is an important component of every distributed system and has a great influence on its performance. There however exists no general grid network structure, and the way in which different resources are interconnected depends on the type of grid. For example, resources of grid systems belonging to a single organization are mostly located within a single LAN (Local Area Network) while world-wide grids include different LANs interconnected by WANs (Wide Area Network). When designing general scheduling algorithms, all the possibilities should be taken into account. Therefore the research at hand will be based on the tree network structure represented in Figure 2.

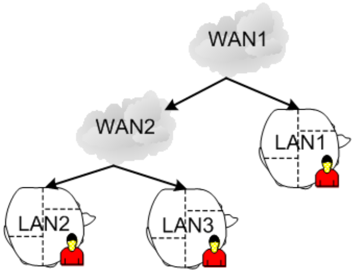

Fig. 2. Tree network structure.

This part of the PhD project will consider hierarchical scheduling whereby scheduling decisions can be taken independently on each level of the network hierarchy. Furthermore, facets such as local/global resources, bandwidth sharing and dynamic network behavior will be taken into consideration.

III. Validation

Tests and calibration are important phases in the development of efficient scheduling algorithms. In these phases use is often made of simulation. Currently, next to diverse well-known general purpose simulators (GPSS, SLX, ...), there exists a large number of grid-specific simulation environments (GridSim, SimGrid, NSGrid, ...). However, the previous study has shown that reproduction of totally dynamic grid and user behavior is rather complicated with most of them. The necessity to model specific components required for dynamic scheduling and large simulation times form other important shortcomings.

In the scope of the current research project a new discrete-event simulator is being developed that is based on already existing components of NSGrid (IBCN, UGent). The design of the new simulator takes into account the dynamics of jobs, grid components and network. The simulator will be used for performance validation of the developed scheduling algorithms.

As the final stage of this PhD research the developed scheduling algorithms will be tested in several existing grids using real jobs generated by the Tornado modeling and simulation application for water quality management (BIOMATH, UGent).

IV. Conclusions

During the last decennia grid systems have been gaining importance fast due to their ability to deal with continuously growing computational and storage demands. However, to provide an efficient use of the available resources further optimizations of the existing systems are required. The current research is dedicated to the development of efficient and fault-tolerant dynamic scheduling algorithms for grids, which take into account dynamics of grid jobs and resources. An appropriate simulation environment will be constructed to testing and calibration of the developed algorithms under particular conditions. Later on, the surplus value and usefulness of the algorithms will be tested in real grid systems.

References

[1] Grid Computing Info Centre Website, http://www.gridcomputing.com/.
[2] G.E. Vieira et al., Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods, *Journal of Scheduling*, 6(1):39–62, January 2003.