Adaptive Scheduling in Grids

Maria Chtepen

Supervisor(s): Bart Dhoedt

I. Introduction

Grid computing is a technology that has significantly gained importance in recent years, replacing the stand-alone cluster approach. The idea behind grid is to combine the capacity of resources belonging to different organizations to execute strongly computationally or storage intensive applications. Think of applications in the fields of high energy physics, financial simulation and climate modeling, which can run for weeks on an average size local cluster. Probably the most well-known example of a grid application is the processing of data originating from the Large Hadron Collider (LHC). LHC is the world's largest particle accelerator, within which protons are moving around at a speed close to the speed of light, producing 600 million collisions per second, resulting in 700 megabytes of data each second. To process this huge amount of data, the LHC grid [1] infrastructure was put together that connects over 140 scientific centers in 33 countries around the globe. However, using grid computing is not only advantageous with regard to computational speed-ups, but also from an economic point of view. Instead of buying large and expensive infrastructure that is able to capture peak loads, organizations can now share resources, which obviously leads to lower hardware investments.

Management of complex grid environments raises several important issues, such as security, efficient distributed data management and advanced scheduling. The focus of our research lies on the latter topic. We take into account grid-specific properties, such as varying resource load and availability, complex application structure and a large number of competing users, to optimize the assignment of grid applications (jobs) to the available resources.

II. Adaptive Scheduling

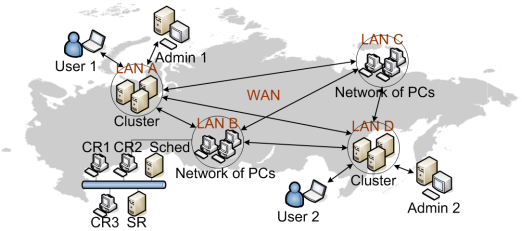

Figure 1. An example of a grid infrastructure: WAN (Wide Area Network), LAN (Local Area Network), CR (Computational Resource), SR (Storage Resource), Sched (Grid Scheduler).

When scheduling jobs within clusters, one often relies on static approaches: the system status is analyzed at a particular point in time, and based on this information the final scheduling decision is taken. While well-suited for relatively stable cluster environments, this approach leads to significant drawbacks within grids due to the following grid properties:

- *Grids are highly dynamic*: strong load variations on resources as well as network, and resource failures are not uncommon. Therefore, there is a need of flexible fault-tolerance and migration mechanisms.
- *Users from various organizations are competing for the same resource pool*: an efficient, priority-based scheduler should be provided.
- *Complex applications are often run on grids*: a grid scheduler should be able to deal with incomplete information on job parameters and with job dependencies.

Adaptive scheduling algorithms developed in scope of this PhD take into account the above-mentioned grid properties while deciding upon the best job-resource assignment. *Adaptive* means that the scheduling policy as well as job schedules can be modified based on information on grid status and jobs progress collected at run-time . In the next subsections a short discussion of the proposed methods and their performance evaluation are given.

A. Dynamic grids

To deal with unpredictable resource failures and load variations, two fault-tolerant approaches were considered: job replication and job checkpointing. Since both approaches can lead to significant system overhead, algorithms were developed and implemented that modify the number of job replicas and the frequency of checkpointing depending on resource load, failure frequency, and remaining job execution times. Figure 2 shows that the adaptive approaches significantly reduce the number of jobs lost due to failures, compared to unconditional replication approaches. At the same time they considerably reduce system overhead by omitting redundant replicas / checkpoints, which leads to higher grid throughput [2].

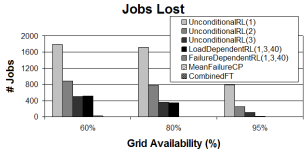

Figure 2. Number of jobs lost by unconditional replication algorithms (UnconditionalRL), adaptive replication algorithms (LoadDependentRL, FailureDependentRL) and adaptive checkpointing algorithms (MeanFailureCP, CombinedFT).

B. Competing users

In grids it is often desirable that particular jobs receive higher priority. Therefore, we developed an algorithm that differentiates between jobs based on their computational load. The algorithm defines an adaptive *load boundary*, which guarantees that as many prioritized jobs (jobs with load less than a predefined value) as possible get executed. Non-prioritized jobs are processed only if this does not penalize the execution of prioritized jobs.

C. Complex applications

Many existing scheduling algorithms for grids depart from the assumption that job execution times are known in advance. Unfortunately, for most real-world applications this parameter is extremely hard to determine *a priori*. In our research, we considered applications containing interdependent tasks for which execution times can be estimated based on periodic updates on task progress. We developed a scheduling approach that reassigns tasks dynamically, taking into account the estimates, in such a way that application execution times are reduced at the cost of eventually slowing down execution of incorporated tasks. Simulation results have shown that our approach achieves an improvement of 60% on average [3].

III. Conclusions

Grid computing is a new technology, which still features several unsolved issues. One of these issues is efficient job scheduling, which takes into account grid-specific properties, such as the dynamic nature of grid resources, large numbers of competing users and the complexity of grid jobs. Our research addresses these issues, introducing adaptive scheduling solutions.

References

[1] Worldwide LHC Computing Grid Website *http://lcg.web.cern.ch/LCG/*, CERN.

[2] Maria Chtepen et al., *Adaptive Task Checkpointing and Replication: Toward Efficient Fault-Tilerant Grids*, IEEE Transactions on Parallel and Distributed Systems, 20(2): 180-190, February 2009.

[3] Maria Chtepen et al., *Scheduling of Dependent Grid Jobs in Absence of Exact Job Length Information* In Proceedings of the 4th IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management, Samos Island, Greece, 22 - 26 September 2008.