Analysis of Overhead and Waiting Time in the EGEE Production Grid

M. Berger, T. Zangerl and T. Fahringer

Institute of Computer Science, University of Innsbruck

Abstract

In the paper we present an analysis of the EGEE production Grid infrastructure, to verify if it fulfills the promised claim of providing a scalable computing resource plattform. We measured scheduling latency and information service overhead.

In our measurements we discovered that the speed of the gLite middleware has significantly improved in the last two years. We have taken measurements for every step of the job lifecycle. We have also measured the actual time when a job started and finished its execution through a callback-mechanism. We present current results which show that the information service introduces an an additional overhead between the time a job actually finishes and the notification of the user.

We also analyzed the the delays over weekdays and hours of the day. Where did not find any significant results which would support the "weekend effect".

We discovered that the job latency and reliability are directly dependent on the site a job is actually scheduled to, and that some sites seem to be much faster and more reliable than others.

We conclude that a simple analysis of scheduling time in the EGEE network or on one single site does not provide sufficient results the factor site plays a large role in the actual scheduling latency. We outline some of the possible decision changes that could be made to improve scheduling in the EGEE infrastructure. We also set the overhead time in relation with job run-time, to estimate the types of jobs suitable for EGEE Grid execution.

1 Introduction

The Enabling Grids for E-sciencE (EGEE) [1] project provides the largest Grid infrastructure in the world, spanning over 140 institutions with the goal to create a reliable and scalable production Grid. It consists of about 300 sites in 50 countries, giving 10,000 users access to 80,000 CPU cores. It currently handles about 300,000 computational jobs per day. It provides a production infrastructure, which gives us the unique opportunity to investigate an environment that is actually used instead of pure simulations.

The EGEE Grid infrastructure is run by the gLite [5] middleware. The gLite distribution is an integrated set of components designed to enable resource sharing.

For this paper we tested the EGEE production Grid to investigate if it can provide the promised scalability for all types of Grid jobs. In particular, we were interested in the latencies of the EGEE Grid.

We define two types of latency: *scheduling latency* and *information service overhead*. The scheduling latency is the delay between the submission of a job and the start of its actual execution, measured in seconds. This includes the time for submission, scheduling decisions, and waiting in a site queue. The information service overhead is the delay between the actual occurrence of an event, such as the completion of computation, and the notification of the end-user about it. This information is transported through several layers of the information service (IS).

We have measured the scheduling latency and information service overhead in the EGEE production grid over several weeks in the summer and fall 2008. We used the virtual organization for central Europe (VOCE) and its associated sites. We have analyzed this data and present an overview of the current situation.

This paper is organized as follows: section 2 gives a short introduction to the architecture of the EGEE grid, and the roles of the workload management system (WMS) and the information service (IS); section 3 presents, analyses, and interprets the experimental data and its results for scheduling latency and IS overhead; section 4 provides some conclusions and future work.

2 Background

To support scalability, the EGEE Grid is organized in multiple Virtual Organizations (VO). Each VO collects information about its associated sites and provides one or more instances of the WMS. Each VO has access to a subset of Grid resources. Grid resources are not exclusively bound to a VO, they are shared among multiple VOs.

These tests where run in the context of VOCE. Whereas most VOs are bound to specific research (for example to a particular domain of high-energy physics) VOCE is bound to the geographic region of central Europe. It is supported on 18 Grid sites. VOCE considers itself an entry-point for Grid programming, and therefore has a considerably liberal usage policy: Users must be from the Central Europe region. Other than that they are allowed to do research related to any domain. This allowed us to do our tests without violating the usage policy.

A WMS is a meta-scheduler for a specific VO. When a Grid job is submitted, it is submitted to a WMS. The WMS keeps a list of all Grid sites belonging to a specific VO. It tries to find a site with free resources, and forwards the job to this site. At the site, the job is kept in a local job queue, until it is submitted to a specific worker node (WN), where it will be executed.

The IS collects and forwards information in the EGEE Grid. It is implemented in several layers: The top layer collects information about all sites in one VO. Each layer polls the underlying layer for new information on a regular basis: The top layer polls the site, the site the scheduler, the scheduler the individual worker nodes.

This work is not the first one which tries to analyze delays in the EGEE Grid. Glatard et. al [4][6][7] have tried to to create a statistical model from measured latency values of EGEE jobs. They found the time between submitting a job and its execution to average 393 seconds with a standard deviation of 792 seconds. Oikonomakos et. al [8] have analyzed the distribution and waiting-time of jobs on one particular Grid site. This work, however, left room for improvement: First, the measurements were taken two years ago; the gLite middleware has significantly improved in both speed and reliability. Second, the measurements were not related to the site they where run on, or where taken on only one site. And last, the measurements were taken on the reported time, and not the actual time; the EGEE software has significant delays in notifications due to the design of its information service.

3 Experiment

The setup for for this experiment was as follows: A test controller was run continuously on one machine. It submitted small jobs to the VOCE. Three jobs where sent every 30 minutes to gather enough data without overloading the Grid with these test jobs. We measured both the notified status changes and the actual status changes. Notified status changes where collected by repeated polling of the job status. Actual changes where collected using a simple callback from the executed job using a HTTP request. Jobs which didn't finish execution within 45 minutes were canceled. The test data was collected between August and October 2008.

The scheduling decision was made by the WMS. Ideally, it would find a site with free resources and submit the job there, where it would start running immediately.

3.1 Scheduling Latency

Scheduling latency is the time between submission of a job and its actual execution. We measured the actual time when a job was submitted, and the time the controller received the running notification through the callback. We then arranged the results by weekday and by hour for analysis.

The mean scheduling latency was 121 seconds, the median 91 seconds. In most cases, the scheduling time was very short (25% where below 119 sec., 95% below 231 sec.). However, the exceptional cases could take a very long time, up to the the 45 minutes (2700 seconds) limit, where they where cancelled.

The scheduling latency results are not stable enough to be predictable: Even though the time is short in most cases, it is much longer in the few exceptional cases. A practical approach to dealing with this imbalance is to cancel a job after a given time, and resubmit it [4].

Figure 1 shows the results for the measured scheduling latency in seconds normalized over the days of the week. A folk theorem claims that Grid scheduling times would be lower during the weekends and off-peak usage. Preliminary

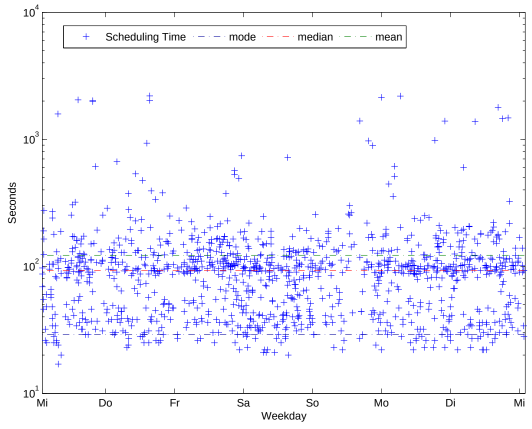

Fig. 1: Scheduling latency according to weekdays.

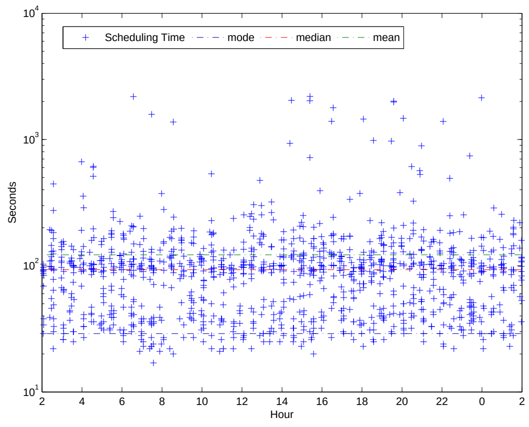

Fig. 2: Scheduling latency according to hours of the day.

results gave the impression, that the weekend-effect may exist, but on Mondays and Tuesdays instead of Saturdays and Sundays. However, during the long-term tests there where no significant changes in scheduling latency over different weekdays.

Figure 2 shows the measured scheduling latency normalized over the hours of the day. Here, we wanted to investigate if there is a difference in scheduling latency in different times of the day, such as mornings, afternoons, evenings, and at night. Again, there are no significant changes in the measurements.

There where no significant changes in the scheduling latency over different days and hours. From this we conclude that the Grid is in use all the time.

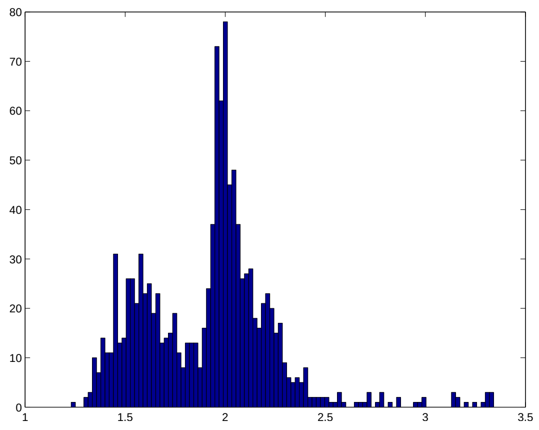

Fig. 3: Scheduling latency histogram.

The measured scheduling latency is significantly lower than the measured latency found in the related work (393 vs. 121 sec). Two factors may have influenced the values: The type of measurement and the improvements in the gLite middleware. The first difference lies in the method used to measure the scheduling latency: We use our own callback to measure the actual start of execution, thus leaving out any overhead which may be created by the IS. The measurements in the related work used the information from the gLite system. The second factor is the improvements of the IS performance in the EGEE middleware, and in particular the IS used in the VOCE system [3].

Figure 3 shows a histogram over the scheduling latencies. Two peaks are clearly visible: the first one at about 40 seconds, and a second one at about 100 seconds. This suggests that there is an additional variable which plays a major role for the scheduling latency.

Figure 4 shows a box-plot of the scheduling latency in relation to the site chosen by the WMS. This figure gives a clear indication that some sites seem to react much faster to a scheduled job than other site. We can classify the sites into three categories: Sites which react fast, and almost always result in a scheduling latency of 40 seconds, sites which are sometimes fast and sometimes slower, and sites which always take about 100 seconds for scheduling and execution. Most sites use the PBS queuing systems and the Maui scheduler, so the setup is similar. We are currently investigating if there are significant changes in the configuration settings.

We are currently cooperating with Cyfronet for finding an explanation for their sites performance. After presenting the figure 4, we shere approached by one of the administrators of the Cyfronet Grid site. We discussed possible setting changes which could improve the sites performance. This investigation is still ongoing.

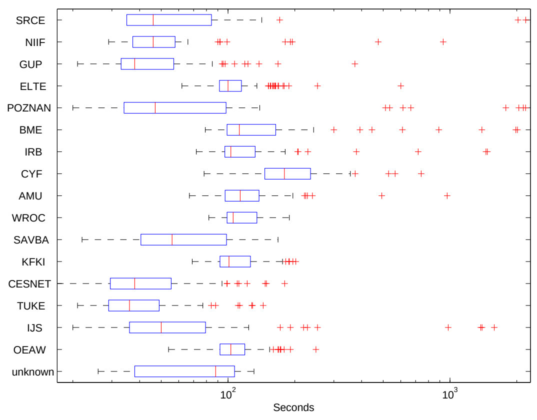

Fig. 4: Scheduling latency according to Grid site chosen by WMS.

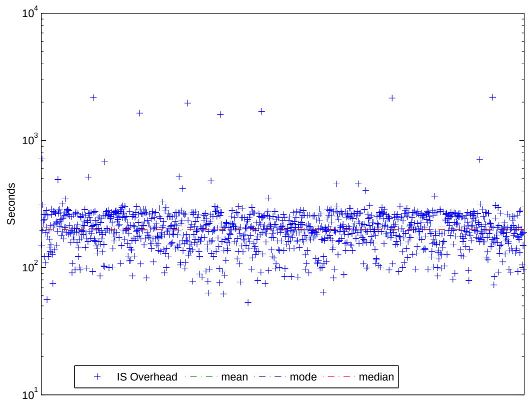

Fig. 5: Information Service Overhead

3.2 Information Service Overhead

To measure the overhead of the IS we measured the time between the occurrence of the actual event and its notification at the user level. The job did a callback after the execution of the main program part, and the time between this callback and the notification of the DONE event was recorded.

Figure 5 shows the results of the IS overhead measurements. The results where stable, with a mean of 208 seconds and a median of 198 seconds. 50% of all values are in the range of 161 seconds to 248 seconds, 95% of all values are in the range of 53 seconds to 364 seconds.

Figure 6 shows a histogram for the overhead measurements. This histogram shows several peaks, which are approximately 60 seconds apart. This is an

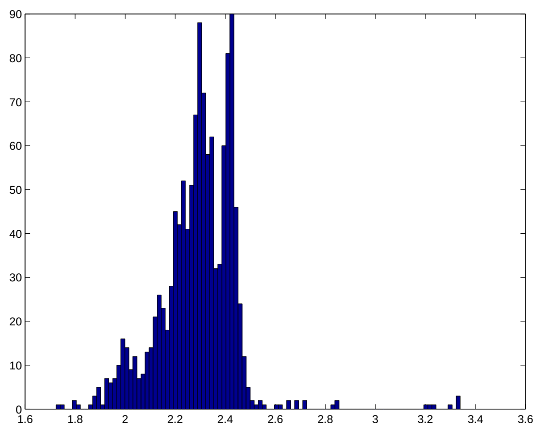

Fig. 6: Information Service Overhead - Histogram

artifact related to the layered structure of the information service: As each layer polls the underlaying layer, the information is propagated in waves instead of continuously. The polling intervals dominate the time needed for the propagation of information.

4 Conclusions

We have analyzed the scheduling latency and IS overhead in a real production Grid. The resulting measurements where notably higher than in research grids, where they are usually below 5 seconds. This difference in overhead must be kept in mind when porting applications to the Grid. An application will run much slower on production Grid infrastructure than under test circumstances in the research Grid.

We found that the scheduling latency averages around 2 minutes in most cases, but can be exceptionally long in other cases. This corresponds to the observation that the job execution time in the EGEE grid is heavy-tailed [4]. Furthermore, we observed that latency varies greatly with the chosen site - possible workarounds would be canceling jobs, and blacklisting sites according to the observed measurements. In timeout and resubmission strategies, different timeout values would have to be applied with respect to execution sites.

We also found that there is a significant overhead in the IS. In particular, the average scheduling latency is now even smaller than the IS latency, meaning that starting a Grid job takes less time than the notification about its termination. This means that the design of the IS should be reconsidered.

The collection of data presented in this paper is relevant for making informed decisions on how to port applications to the Grid. In particular, it is useful to decide about the size of the workload to be submitted in one job. Just two years ago there was the general view, that any job taking less than 30 minutes would not be worth submitting to the EGEE Grid. Although the recent improvements in the infrastructure and services have shortened that time, it is still only feasible to submit jobs to the Grid which take at least a few minutes of execution time.

When scheduling workflows consisting of smaller Grid activities, we have to investigate different methods to deal with these delays: One approach is the mentioned job canceling, another approach is circumvention of the job queues using the worker model.

The EGEE production Grid infrastructure shows us that a production Grid can exist and will exhibit new and challenging properties which were not present in research Grids. It shows again that only tests in an actual environment can prove if a concept really works.

Acknowledgements. This work was partially supported by EU project EGEE-III, INFSO-RI-222667. We would also like to thank Patryk Lason from the Cyfronet Grid site for sharing ideas on the configuration issues and experimenting with different setups.

References

1. Enabling grids for E-sciencE (EGEE). `http://www.eu-egee.org/`.
2. *Seventh IEEE International Symposium on Cluster Computing and the Grid (CC-Grid 2007), 14-17 May 2007, Rio de Janeiro, Brazil.* IEEE Computer Society, 2007.
3. Jan Astalos, Radecki L, and M. Ziajka. W. performance improvements to BDII - grid information service in EGEE. In *Cracow07 Grid Workshop : proceedings. - Krakow : Academic Computer Centre CYFRONET AGH*, pages 398–404, 2008.
4. Tristan Glatard and Xavier Pennec. Optimizing jobs timeouts on clusters and production grids. In *CCGRID* [2], pages 100–107.
5. Claudio Grandi. The gLite middleware. In *EGEE'07*, 2007.
6. D. Lingrand, J. Montagnat, and T. Glatard. Estimation of latency on production grid over several weeks. 2008.
7. Diane Lingrand, Johan Montagnat, and Tristan Glatard. Modeling the latency on production grids with respect to the execution context. In *CCGRID*, pages 753–758. IEEE Computer Society, 2008.
8. Michael Oikonomakos, Kostas Christodoulopoulos, and Emmanouel A. Varvarigos. Profiling computation jobs in grid systems. In *CCGRID* [2], pages 197–204.