

Analyzing the Workload of the South-East Federation of the EGEE Grid Infrastructure

Georges Da Costa, Marios D. Dikaiakos
Georgio.da.costa@gmail.com
mdd@cs.ucy.ac.cy
Computer Science Department
University of Cyprus
POBox 20537
CY1678 Nicosia, Cyprus

Salvatore Orlando,
orlando@orlando@dsi.unive.it
Information Science and Technologies Institute
Italian National Research Council
56100 Pisa, Italy



CoreGRID Technical Report
Number TR-0063
February 22, 2007

Institute on Knowledge and Data Management

CoreGRID - Network of Excellence
URL: <http://www.coregrid.net>

Analyzing the Workload of the South-East Federation of the EGEE Grid Infrastructure

Georges Da Costa, Marios D. Dikaiakos
Georgio.da.costa@gmail.com
mdd@cs.ucy.ac.cy
Computer Science Department
University of Cyprus
POBox 20537
CY1678 Nicosia, Cyprus

Salvatore Orlando,
orlando@orlando@dsi.unive.it
Information Science and Technologies Institute
Italian National Research Council
56100 Pisa, Italy

CoreGRID TR-0063
February 22, 2007

Abstract

Grids have emerged as wide-scale, distributed infrastructures providing enough resources for always more demanding scientific experiments. EGEE is one of the largest scientific Grids in production operation today, with over 220 sites and more than 30,000 CPU all over the world. A further evolution of EGEE needs to be based on knowledge of deficiencies and bottleneck of the current infrastructure and software. To provide this knowledge we analyzed nine months of job submissions on the South-East federation of EGEE. We provide information on how users submit their jobs:throughput, bursts, requirements, VO. We study the current behavior of EGEE middleware too, by evaluating its performance and the retry policy. We finally show that even if the middleware provides advanced functionality, most submissions are still embarrassingly parallel jobs.

1 Introduction

Grids are large, distributed computing infrastructures that seeks to support resource sharing and coordinated problem solving in dynamic, multi-institutional Virtual Organisations (VOs) [6]. Typical Grid infrastructures comprise large numbers of geographically distributed and heterogeneous resources (hardware and software), belonging to different administrative domains and interconnected through an open network. Grids are quickly gaining popularity, especially in the scientific sector, where projects like *EGEE (Enabling Grids for E-science)* [2], provide and operate the resources required to accommodate large computational experiments with thousands of scientists, tens of thousands of computers, trillions of instructions per second, and petabytes of storage [2]. At the time of this writing, the infrastructure operated by EGEE assembles over 220 sites around the world, thousands of job queues, more than 30,000 CPUs, about 5PB of storage, and supports over 200 Virtual Organizations.

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

So far, however, the configuration of EGEE resources and services has evolved empirically. Due to the complexity of EGEE's middleware, which consists of numerous cooperating software components operated and maintained by different institutions (middleware services, logical file systems, databases, end-user portals and gateways), it is difficult to have an *a priori* understanding of how the infrastructure is used by different user communities and what are its constraints and bottlenecks. However, as more users are attracted to EGEE, information about submitted jobs, resource usage, etc. can be collected and analyzed in order to provide valuable insights about emerging patterns of Grid use, and to guide future improvements in EGEE's middleware design and configuration.

To derive such insights, we investigate the usage of EGEE as represented by the characteristics of Grid jobs submitted for execution. We retrieve and analyze logs from the *Resource Brokers* (RB) operated by the South-Eastern Europe (SEE) federation of EGEE. In particular, we examine logs retrieved from the RB located in Cyprus¹ and operated by the University of Cyprus (RB-CY). These logs capture the RB activity during a nine-month period, extending from February 14, 2006 to November 24, 2006. During this period, the broker handled 41,124 jobs and gathered 1.3 GB of log-data. To validate our observations further, we analyzed a log-file retrieved from a second Resource Broker² of the SEE federation, which is hosted in Athens, Greece, and is operated by GRNET, the Greek Research and Education Network (RB-ATH). Although, the RB-ATH log corresponds to a time-frame shorter than that of the RB-CY (the RB-ATH log extends from November 7, 2006 to January 30, 2007) and maintains fewer details about the life-cycle of Grid jobs, it captures a significantly larger number of submitted jobs (103,265).

The remainder of this paper is organized as follows: Section 2 describes briefly the process of job submission in EGEE. Section 3 presents our findings regarding the patterns of jobs submitted to EGEE through the RB-CY and RB-ATH brokers. Section 4 describes the error information captured in our logs and Section 5 examines the performance of EGEE's middleware. We conclude in Section 5.

To the best of our knowledge, this work is one of the first studies that characterize the workload of a large-scale Grid infrastructure at an extended time-frame. In contrast to previous studies[7, 8], which characterize the use of Grids in generic terms, our work focuses on the complete chain of one specific system, from job requirements to the middleware behavior and performance.

2 Job submission in EGEE

A job submitted for execution to EGEE usually comprises an executable and a set of input files. The executable is expected to process the given input using a set of selected EGEE resources. Resource selection is based upon a user-defined specification of the resource-characteristics required for the job to run successfully: for instance, computational capacity, available physical memory, network proximity with certain file systems, the availability of specific application software and/or software libraries. This specification is described in JDL, the *Job Description Language* [4], and stored in a file that accompanies the job submission.

During a job submission, the corresponding JDL file is sent to a Resource Broker (RB), a central (global) Grid service. The RB performs a *matchmaking* between the requirements of the submitted Grid job and the resources available on EGEE at submission time. The matchmaking is done based on data received by querying an Information Index, a middleware service that provides up-to-date information about the state of Grid resources, usually spanning several sites. If the matchmaking is successful, the job is sent from the RB to the matching Grid site for execution. Subsequently, the RB monitors the job's status and progress, and returns the results back to the end-user upon successful completion of the job.

For more efficient management, EGEE is divided into federations/regions, and into each such federation resides a Regional Operations Centre (ROC) that is responsible for supporting and monitoring a set of EGEE-participating grid sites, the Resource Centres (RCs). These divisions into federations are usually organised geographically. As an example, the South East Europe (SEE) federation has the Regional Operations Centre based in Greece and comprises production sites in Bulgaria, Cyprus, Israel, Romania, Serbia, and Turkey. Apart from ROCs and RCs, there is also the EGEE-wide Grid Operations Centre (GOC), responsible for coordinating and monitoring the operation of the Grid infrastructure, and a total of four Core Infrastructure Centres (CICs) which provide monitoring and operational troubleshooting services, acting as second-level support to ROCs.

Typically, a Grid user will submit his jobs through a Resource Broker that is available to his Virtual Organization or Regional Operation Center. The RB manages the whole life-cycle of the job and can keep track of all events that

¹rb101.grid.ucy.ac.cy

²rb01.egee-see.org, rb.isabella.grnet.gr and wms01.egee-see.org

Requirement type	Number of jobs	Percentage
BlackList	83102	58%
MaxCpuTime	80629	56%
Site	32299	22%
None	18287	13%
Library	11833	8%
WallClockTime	4457	3%
CpuNumber	1500	1%
MemorySize	1334	.9%
other	536	.4%

Table 1: Type of resources requirements found in the queries.

Virtual organization	Number of jobs
biomed	83065
see	29180
dteam+ops	19642
eumed	9281
others	3221

Table 2: Number of jobs for each Virtual Organization

occur during its life-cycle. Therefore, the analysis of RB traces can give us valuable insights about the characteristics of Grid jobs, their requirements, their performance, the overhead introduced to job execution by the middleware, job failure rates, etc.

3 Job characteristics

The two logs of our study (RB-CY and RB-ATH) capture a total of 144389 jobs. Nearly all the jobs found in these logs are of type “normal”. Besides the normal jobs, in our data-set we identified 24 interactive jobs. The next version of the middleware will also supports the execution of “interactive,” “collection-type,” “parametric”, and *dag-type* (directed acyclic graph) jobs.

In our data-set, 121518 out of 144389 jobs (84%) are composed of an executable and of input files. There are two ways to provide input files: the first one is to “stage-in” the input files together with the executable; the second, is to upload input files to a logical file system on the Grid, and have the running job accessing these files at run time through a logical file catalog. The second option was rarely taken in our data-sets, with only 295 jobs making use of the logical file catalog.

3.1 Jobs requirements

As mentioned earlier, the JDL description of submitted job represents the requirements of this job in terms of Grid resources. For example, the JDL specification for a job may require that $WallClockTime \geq 72h$, i.e., the job should be able to execute without interruption for at least 72 hours, or $Mpich \in Library$, i.e., the job should be scheduled on a site with an installation of the Mpich library. The names of requirement-attributes found in JDL files are taken from the Glue schema specification, which is adopted by the Information indexes of EGEE [3]. Below, we present an example of a composite resource requirement found in a JDL job specification:

```
GlueCEInfoHostName == "ce101.grid.ucy.ac.cy"
                    && Member("MPICH", SoftwareRunTimeEnvironment)
                    && GlueCEInfoTotalCPUs >= 4;
```

It is worth noting that in our data-sets we found a very small set of distinct requirements: out of the 144389 jobs, there were only 1056 jobs with distinct requirements specified; identical requirements share exactly the same resource requirements and associated values. Table 1 shows the distribution of requirements found in our data-sets. A JDL specification defines a *site* in order for the job to run on a particular site, and *blacklist* in order to prevent the job from running on a particular site. *Library* is used when a particular library is needed. This table shows that a large amount of jobs (22%) do not really use the requirements system as they already specify a site name. Some of those (8.5% of the jobs requiring a particular site) still use requirements in order to run on a subset of the site by requesting other characteristics.

The majority of jobs (77%) define requirements that focus on four particular requests: i) The blacklisting of certain sites. This is used primarily by jobs submitted by the *biomed* Virtual Organization, with the number of blacklisted sites growing over time. During the large-scale computational experiments of the biomed VO, failing sites were identified

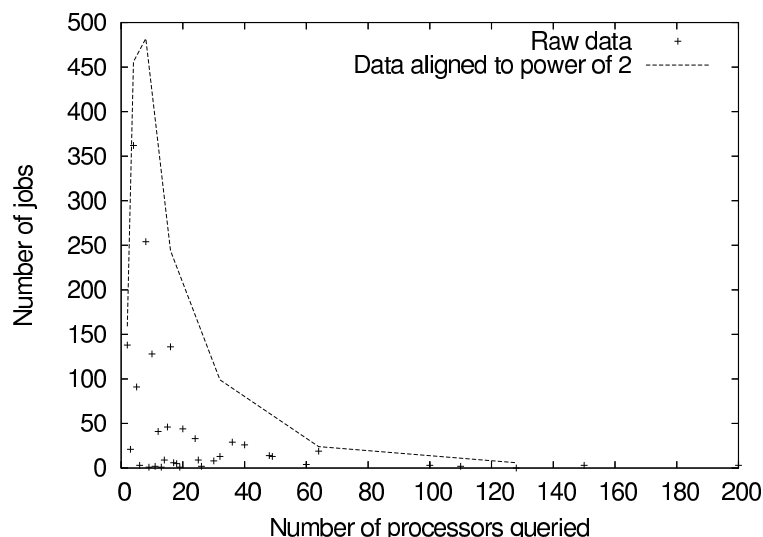


Figure 1: Repartition of jobs according to the number of processors queried. The curve shows raw data reduced to power of two.

and blacklisted in order to enhance the efficiency of the experiments. (ii) To find sites that accept long-running jobs (specified through the *MaxCpuTime* or *WallClockTime* attribute). (iii) To run on a specific site, and (iv) to guarantee the availability of a particular software library.

It is worth noting that the requirements defined correspond primarily to static characteristics of resource configuration, such as the *WallClockTime* policy or the installation of specific libraries. Also, that the requirements that request a certain performance capacity from the Grid resources, mainly specify the number of requested processors. Figure 1 presents a diagram of the number of jobs that request a particular number of processors. This characteristic seems to follow a long tailed distribution. 61% of these request are for a number of processor that is equal to a power of two. The curve on Figure 1 shows that when the raw values are reduced to powers of two only, they follow the *lognormal* distribution.

3.2 Virtual organizations

As mentioned earlier, EGEE support over 200 Virtual Organizations. Table 2 presents the VOs that are actually using EGEE through the two Resource Brokers of our study. The active Virtual Organizations are:

Biomed : focuses on biomedical applications

See : this is a generic VO introduced to support scientists from South-east Europe that do not yet belong to another established VO.

Eumed : this is a VO established to support euro-mediterranean collaboration in Grid infrastructures.

dteam, ops : these are VOs established to support administrators who are responsible for deploying and testing middleware, and for operating the EGEE infrastructure.

others : Other scientific projects using EGEE, such as atlas, Lhcb, alice.

As we can see from Table 2, the south-east federation of EGEE mainly supported production jobs coming from *biomed* and *see* Virtual Organizations.

3.3 Job throughput

Figure 2 shows the number of submitted jobs per day in the 9-month long RB-CY dataset. Two types of behaviors alternate: chaotic with activity bursts, and calm with constant throughput. Grid middleware has to be able to handle

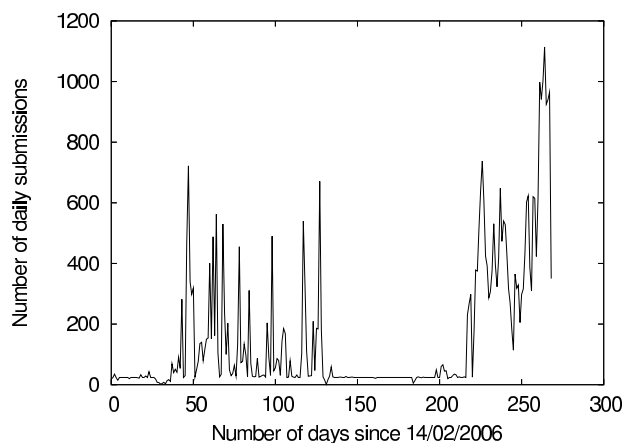


Figure 2: Number of daily submitted jobs during the 9 months

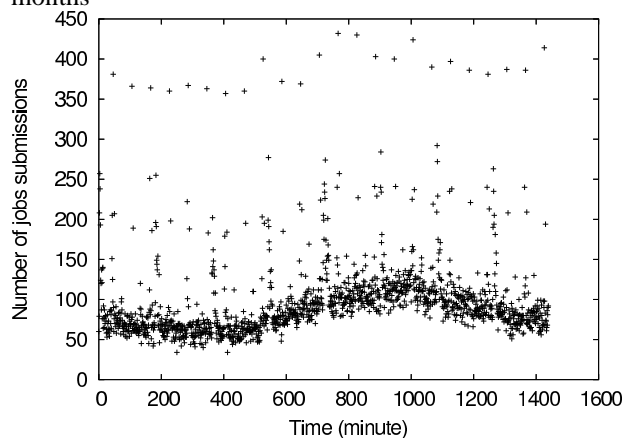


Figure 4: Number of submission according to the minute of the day using RB-CY and RB-ATH

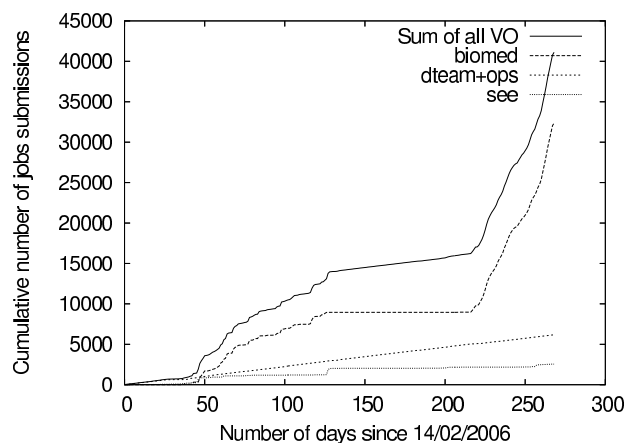


Figure 3: Cumulative number of submitted jobs during the 9 months for the main VO

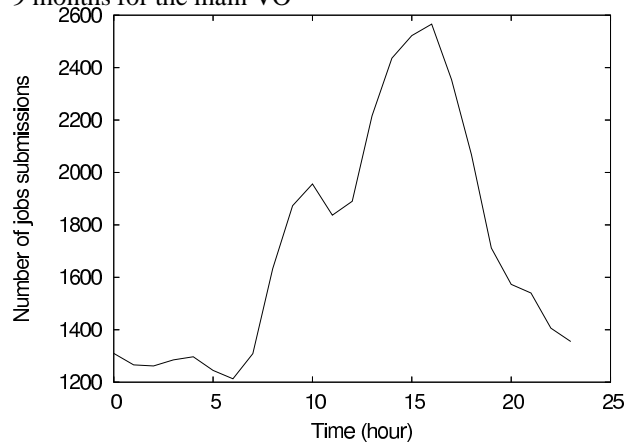


Figure 5: Number of submission according to the hour of the day using RB-CY

large number of jobs submissions in a small amount of time. During peaks, 10 jobs were submitted during one second on one RB, and at other scales, 20 jobs were submitted during ten seconds. Those peaks are of low duration as the maximum number of jobs submitted during one minute is 24, 95 for one hour, and 1114 during one day.

Figure 3 shows the cumulative throughput per Virtual Organization for the same dataset. The two previous behavior are dependent of the Virtual Organization type. Production jobs, like the ones coming from the *biomed* VO, are more localized in time. Operational jobs coming from the *ops* and *dteam* VOs have a stable throughput. More precisely, *biomed* used EGEE to undertake two “data challenges” during the period captured inside our data-sets: the first one, on the Bird Flu during April and May, and the second one finishing at the end of January 2007 on the malaria began in October. As stated in [7], large Grid like EGEE are project driven.

3.4 Daily load repartition

We were not able to identify a submission pattern on day-to-day basis. Yet, user behavior during the day are structured. Figure 4 shows the number of jobs according to the minute of its submission for the two data-sets. Every hour, several points go out of the mean curve. These points are linked to the use of WMProxy[1] like services: to reduce peak of workload on the RB, the WMProxy can be used as a buffer when a large number of jobs are to be sent to EGEE. Instead of sending jobs directly to the RB, users can choose to send them to this service. It wakes up every hour and, if enough jobs are finished, submits new ones. This leads to the hourly aggregation shown in Figure 4. Currently WMProxy is not deployed on SEE and this technique is applied by hand.

Final state	Number of jobs	Percentage
OK	19334	48%
CANCEL	16658	42%
FAIL	3858	10%

Table 3: Final state of the jobs in RB-CY. 1274 jobs marked as NONE (i.e. not enough information in the database to conclude) are excluded.

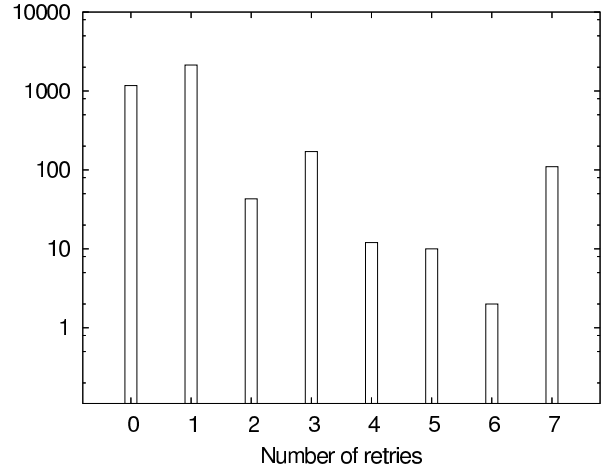
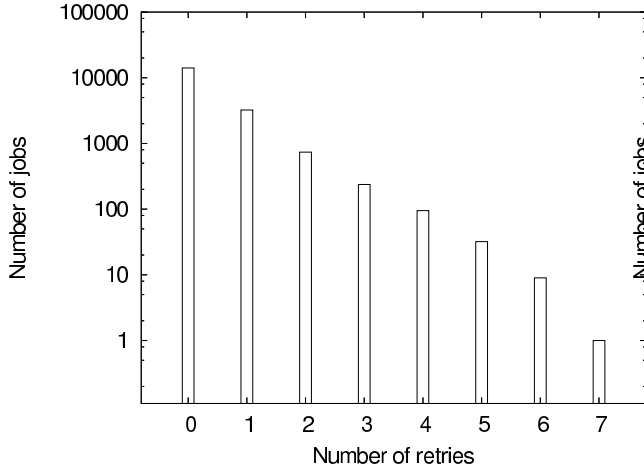


Figure 6: Number of retries needed for jobs to eventually success Figure 7: Number of retries taken for jobs before failing

Figure 5 shows an aggregated view of RB-CY with a broader granularity of one hour. This dataset is used to reduce the dispersion due to timezones. It appears that some job submissions are done during the morning, and most of them are done in the afternoon at around 3pm. During the workload peak, submissions are more than doubled compared to laid-back times. This curve follows a distribution similar to the one observed in the context of Web and Peer-to-Peer usage [5].

4 Faults

Complex systems as Grids are prone to errors and failures of various kinds [9]. Table 3 summarizes the possible final states of jobs in the RB-CY data-set. A job is denoted as *OK* when it finishes successfully and sends back its results. It is denoted as *CANCEL* when the user decided to stop the job at some point. The state *FAIL* means that the job failed for some reason. Finally the state *NONE* describes jobs for which our data-set does not contain enough information. *CANCEL* are mostly used when a job waits during a long time in a queue in order to run on an other site. Results in the following sections are extracted from RB-CY.

As Grids are large, there is a high probability that at some point a problem arises and prevents the job from achieving success. To reduce the number of failing jobs, Grid middleware retries to schedule failing jobs. Figures 6 and 7 show the number of retries used for jobs that respectively success and fail.

Figure 6 represents on a logarithmic scale the number of jobs that need n retries to success. The number of successful jobs decreases exponentially with the number of retries. But as there are usually several retries, 83% of jobs that are executed (i.e. not in *CANCEL* or *NONE* state) succeed.

Figure 7 shows the number of retries used by the jobs that eventually fail. A value of *1 retry* means the job was retried once: there was two failed attempts and the maximum authorized number of retries specified in the JDL was two.

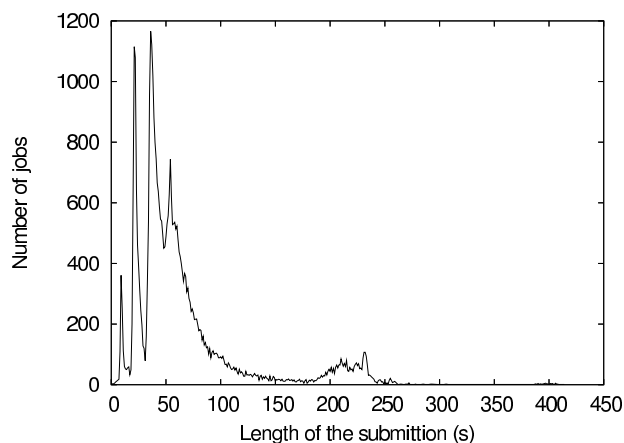


Figure 8: Time between the job submission and when the job is sent to the local queue

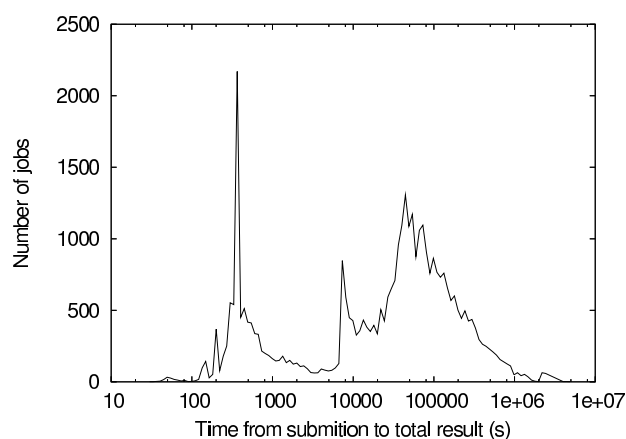


Figure 9: Time between the job submission and the final completion

Ratio Managing time/Total time	$\leq 1\%$	$\leq 5\%$	$\leq 10\%$	$\leq 20\%$	$\leq 50\%$
Percentage of jobs	74%	84%	95%	98%	99.9%

Table 4: Ratio between the managing time and the total time until completion. The managing time is between the job submission and when the job is sent to a queue.

5 Middleware performance

A preliminary study of EGEE middleware was done in [10], using a simulated workload. In the following, the performance evaluation is based on the real workload of production jobs captured in the RB-CY trace.

The complete execution of a job is split in three phases. First the job is sent by the user to the RB which chooses the site where it will be run. The second phase is when the site is given the job. The site puts it in one of its queue until the necessary resources are available. The last phase is the actual running of the job in the site.

The data processed in this article are obtained in the RB. They provide only timing information of the first phase, plus the ending time of the job. It is sufficient to evaluate the cost induced by the middleware.

Figure 8 shows the time needed by the middleware to choose a queue and to send the job to this queue. There are four high peaks: *9s* *21s* *36s* *54s*, the last one is farther and smaller, around *230s*. This behavior is still under investigation as the reason for those precises peak is still unknown. At least, this figure shows how efficient the current middleware is. Most jobs are processed under half a minute, and a large majority are under one minute.

Figure 9 shows the total time the jobs are in the system, ie. the middleware time, the waiting time in the queue and the running time. There are several order of magnitude between the times. Peaks are: *6min*, *2h*, *12h*, *72h*. It shows that most jobs run at least one order of magnitude over the middleware time. Moreover, in EGEE *12h* and *72h* are quite common *WallClockTimes*. Each site has a policy and kills jobs which are not finished when the running time attains the *WallClockTime*. Dispersion around those values should be caused by the waiting time in queues before jobs effectively run.

Table 4 shows the ratio for each job between the time passed in the middleware and the total time. For most jobs, the middleware time is negligible compared to the total time. Jobs for which this ratio is relatively high are very fast jobs. EGEE seems to be more efficient to manages large jobs compared to really small ones.

A final remark. There is no correlation between the submission date and the length of the job or between the submission date and the middleware time. Those time are not project-dependant.

6 Conclusion

In this paper we analyze nine months worth traces of South-East of EGEE. Our goals were to provide insight on how users of EGEE are using the Grid, thus leading to being able to produce realistic workload and showing the lack

of local pattern of job submission. The second goal was to evaluate the current performance of EGEE middleware. We propose a characterization of real requirements and of job throughput. We evaluate the current retry mechanism used by EGEE middleware. We showed the efficiency of this middleware and identified a bottleneck: many jobs are manually canceled due to being put too deep in site queues. Finally we evaluate the middleware performance and compared it to the jobs running time. We demonstrated that submission of jobs are fast, but not enough for interactive submission. The middleware mostly showed its limitation in the poor usage of requirements or other high-level functionality.

Acknowledgement: This work was funded in part by the European Commission under the the Sixth Framework Programme through the Network of Excellence CoreGRID (Contract IST-2002-004265) and the Enabling Grids for E-sciencE project (Contract number INFSO-RI-031688). The authors wish to acknowledge Kyriacos Neokleous and Kostas Koumantaros for providing access to the RB-CY and RB-ATH databases respectively, and Nicolas Jacq for providing information about *biomed*.

References

- [1] Egee jra-1: Job description language attributes specification (submission through wmpoxy). <https://edms.cern.ch/document/590869/>.
- [2] Enabling Grids for E-Science project. <http://www.eu-egee.org/>.
- [3] Glue Schema. <http://glueschema.forge.cnaif.infn.it/> (accessed Jan. 2007).
- [4] Job Description Language: Attributes Specification. <http://edms.cern.ch/document/590869/>, May 2006.
- [5] Georges Da Costa, Corine Marchand, Olivier Richard, and Jean Marc Vincent. Resources availability for peer to peer systems. The IEEE 20th International Conference on Advanced Information Networking and Applications (AINA), 2006.
- [6] I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3):200–222, 2001.
- [7] Alexandru Iosup, C. Dumitrescu, Dick Epema, Hui Li, and Lex Wolters. How are real grids used? the analysis of four grid traces and its implications. In *The 7th IEEE/ACM International Conference on Grid Computing (Grid2006)*, pages 262–269, 2006.
- [8] Emmanuel Medernach. Workload analysis of a cluster in a grid environment. In *Job Scheduling Strategies for Parallel Processing, 11th International Workshop, JSSPP 2005*, pages 36–61, 2005.
- [9] Kyriacos Neocleous, Marios Dikaiakos, Paraskevi Fragopoulou, and Evangelos Markatos. Failure management in grids: The case of the egee infrastructure. Technical Report TR-0055, Institute on System Architecture, CoreGRID - Network of Excellence, July 2006.
- [10] N Svraka, A Balzs, A Belic, and A Bogojevic. glite workload management system performance measurements. In *IV INDEL*, pages 294–297, Banjaluka, 2006.