Understanding and mastering dynamics in computing grids: processing moldable tasks with user-level overlay
Moscicki, J.T.

Citation for published version (APA): Mościcki, J. T. (2011). Understanding and mastering dynamics in computing grids: processing moldable tasks with user-level overlay
In this Chapter¹ we present a descriptive analysis of spatial and temporal dynamics in large, production-grade grid infrastructures. The analysis is focused on computing services. It is based on publically available data and various monitoring sources in the EGEE Grid, and is complemented by the data from our specific observations of the system behavior. The aim of our analysis is to gain a better understanding how large grids work, what are the forces driving their development and functioning, and how they impact the end users and application communities. We use the EGEE Grid – the largest grid in operation today – as a representative example of a distributed system for processing of scientific tasks which exposes complex and dynamic behavior.

2.1 EGEE – world’s largest computing and data Grid

The EGEE Grid² is a globally distributed system for large-scale processing and data storage. At present it consists of around 300 sites in 60 countries and supports more than $10^5$ jobs a day. It offers more than $10^5$ CPU cores and 20 PB of storage to $10^4$ users in nearly 200 Virtual Organizations (VOs)³. EGEE is a multidisciplinary Grid,

---


²Enabling Grid for E-sciencE (EGEE), http://www.eu-egee.org

supporting users in both academia and business, in many areas of physics, biomedical applications, theoretical fundamental research and earth sciences, as summarized in Tab. 2.1. The largest user communities come from the High Energy Physics, and in particular the experiments active at the Large Hadron Collider (LHC) at CERN\(^4\).

Development of a large-scale, production grid is a long-term process which involves funding entities (e.g. national research councils or international agencies), resource providers (e.g. computing centers), resource consumers (e.g. user communities), hosting entities (e.g. large scientific laboratories) and external entities (e.g. cross-domain scientific or infrastructure projects). The process is complex and based on negotiation due to the distributed funding model and complicated relationships between involved parties. The EGEE Grid development started in 2002 as R&D at the Data Grid project, and was carried through several implementation and deployment phases of EGEE project to the currently ongoing consolidation phase and transition to European Grid Initiative (EGI). EGEE Grid emerged as a federation of three large infrastructures: WLCG which uses the gLite [113] middleware, OSG [155] which uses the Globus [66] middleware and NDGF which uses the ARC [58] middleware.

<table>
<thead>
<tr>
<th>Application domain</th>
<th>Active VOs</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Energy Physics</td>
<td>43</td>
<td>5205</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>28</td>
<td>2535</td>
</tr>
<tr>
<td>Multidisciplinary VOs</td>
<td>32</td>
<td>1825</td>
</tr>
<tr>
<td>Life Sciences</td>
<td>14</td>
<td>571</td>
</tr>
<tr>
<td>Computational Chemistry</td>
<td>4</td>
<td>448</td>
</tr>
<tr>
<td>Astronomy, Astrophysics and Astro-Particle Physics</td>
<td>21</td>
<td>362</td>
</tr>
<tr>
<td>Earth Sciences</td>
<td>11</td>
<td>325</td>
</tr>
<tr>
<td>Computer Science and Mathematics</td>
<td>6</td>
<td>28</td>
</tr>
<tr>
<td>Fusion</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Others</td>
<td>36</td>
<td>1878</td>
</tr>
<tr>
<td>Total</td>
<td>197</td>
<td>13184</td>
</tr>
</tbody>
</table>

Table 2.1: Virtual Organizations and application domains in the EGEE Grid. Users may belong to multiple VOs and thus may be double counted. Data source: CIC-OP.

### 2.1.1 Middleware services

EGEE Grid is a distributed federation of computing, storage, information and monitoring services [145].

Information services aggregate static and dynamic information about the status of the Grid and provide it to other services. They allow discovering of types of available services and querying for the characteristics of each service. Information services also handle the authorization data which defines VOs allowed to use the services.

\(^4\)General updated information on the LHC programme is available on CERN web site at [http://www.cern.ch](http://www.cern.ch)
Computing services include *Computing Elements* (CEs) which represent resources within a single cluster or batch farm. CE is the smallest unit for resource brokering and selection, which is realized in one of two models. Central resource brokering model, which is a default mode in gLite, uses *Workload Management System* (WMS) services, which perform job-resource matchmaking and routing of jobs to appropriate CEs. Client-side resource brokering is more common with new gLite CREAM CE services and also in NDGF with ARC middleware and OSG with globus/GRAM protocol.

Storage services include Storage Elements (SEs) and File Catalogues (FCs) which allow file-based access to data, file transfer and handling of multiple file replicas.

Existing monitoring projects and services allow to analyze the EGEE system at various levels. In particular, the GridObservatory [119] project is dedicated to providing datasets of traces reported by gLite services for scientific research\(^5\). Tab. 2.2 provides a summary of all data sources used in this Chapter.

<table>
<thead>
<tr>
<th>Name</th>
<th>Full name and reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIC-OP</td>
<td>CIC Operations Portal</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://cic.gridops.org/index.php?section=home&amp;page=volist%5C">http://cic.gridops.org/index.php?section=home&amp;page=volist\</a>]</td>
</tr>
<tr>
<td>GO</td>
<td>Grid Observatory</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www.grid-observatory.org%5C">http://www.grid-observatory.org\</a>]</td>
</tr>
<tr>
<td>Gstat</td>
<td>GStat</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://gstat-prod.cern.ch/gstat/geo/openlayers%5C">http://gstat-prod.cern.ch/gstat/geo/openlayers\</a>]</td>
</tr>
<tr>
<td>GOCDB</td>
<td>Grid Operations Center Database</td>
</tr>
<tr>
<td></td>
<td>[<a href="https://goc.gridops.org%5C">https://goc.gridops.org\</a>]</td>
</tr>
<tr>
<td>CESGA</td>
<td>EGEE Accounting Portal</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://www3.egee.cesga.es/gridsite/accounting/CESGA/egee_view.php%5C">http://www3.egee.cesga.es/gridsite/accounting/CESGA/egee_view.php\</a>]</td>
</tr>
<tr>
<td>GridView</td>
<td>Monitoring and Visualization Tool for LCG</td>
</tr>
<tr>
<td></td>
<td>[<a href="http://gridview.cern.ch/GRIDVIEW/job_index.php%5C">http://gridview.cern.ch/GRIDVIEW/job_index.php\</a>]</td>
</tr>
</tbody>
</table>

Table 2.2: List of EGEE Monitoring Services used as data sources for analysis of EGEE Grid dynamics.

### 2.1.2 Conflicting forces: resource providers and consumers

Grid resources, including networks, are distributed among different administrative domains and controlled by different local policies. VOs bring subsets of resources into single virtual administration domains. This should allow setting of global access and resource sharing policies for the VO members, but in practice it is difficult to achieve due to incomplete and complex interfaces between grid services and local fabric layers.

The ultimate control of the resources is retained by the resource providers which define local policies to satisfy the demands of grid and local users – computing centers typically provide an important part of their resources to non-grid communities. For this reason system administrators in grid sites prefer to use well-established technologies, such as batch systems, to control the usage of their computing resources. Therefore,

\(^5\)The Grid Observatory is a part of the EGEE-III EU project INFSO-RI-222667.
achieving additional capabilities above the ones already provided by the local systems is difficult.

Trust relationships are a delicate issue and resource providers require the accountability and traceability of the user activities as a part of the grid-participation agreement. This allows to exclude the misbehaving users from accessing the site services. The traceability requirements may be waived in well-understood, exceptional cases for restricted access of few power users who are mandated by the VO to submit jobs on behalf of other users. This is a case, for example, in HEP data production systems where centrally managed jobs are typically mapped to few management accounts. For some purposes, such as the data management, large VOs are also capable of negotiating the setup of VO-specific services, which are placed under direct VO control and deployed on dedicated physical nodes in grid sites. However, for smaller communities or individual scientists running their applications, such special arrangements are often impossible.

2.2 Grid as an infrastructure

2.2.1 Grid structures

The EGEE Grid infrastructure is heterogeneous and it integrates sites of varying sizes, computing capacity and internal structure, as shown in Fig. 2.1. The number of computing nodes per site spans four orders of magnitude. Nearly 5% of sites consist of above 2000 nodes with few exceptionally large sites with more than 10,000 computing nodes. On the other hand nearly 10% of tiny sites consists of less than 10 computing nodes and 40% of sites are below 100 nodes. Majority of the sites use up-to-date, mainstream hardware technologies as shown by the clear peak in the distribution of average computing node capacity (estimation of processing capacity of installed hardware is based on standard benchmarks, such as SPECINT\(^6\)). However, tails on both ends of the distribution show the capacity variation of one order of magnitude for more than 10% of the sites. The absolute majority of sites expose one CE service, the remaining 10% of sites expose up to 8 CEs and, in extreme cases, even up to 25 CEs. A large number of CEs per site may be an indication that a site integrates several computing clusters for grid use, or that more CEs are needed to increase scalability of the service.

The EGEE Grid is structured in few ways, including high-level structures derived from computing models of heavy VOs, such as the MONARC [8] model, which is a hierarchical computing model for data processing at the LHC. The implementation details of the model differ for each LHC experiment but the hierarchical organization, shown in Fig. 2.2, is in common. In this model, sites are classified into Tiers, according to their size, provided level of service and operational support [26]. Tier0 is the data source for LHC experiments. Sites in Tier1 are large computing and storage farms which perform heavy data reprocessing. Sites in Tier2, typically research labs and universities, are medium-sized and support simulation and further stages of data analysis. Tier3 sites are small clusters, primarily used for end-user data analysis. The hierarchical model also applies to networking. Sites in Tier1 may be connected with high-speed, private

---

\(^6\)Standard Performance Evaluation Corporation, \url{http://www.spec.org}.
Figure 2.1: Histograms of basic characteristics of computing sites in EGEE Grid: number of worker nodes per site, average computing capacity of worker nodes and number of CE services per site. Y-axis shows a fraction of sites with a given value of X. Data source: GStat (gridmap).

optical networks [29], whereas most of other sites are connected using a public Internet infrastructure. Currently EGEE infrastructure includes 13 Tier1 sites (around 3% of all sites) and 123 Tier2 sites (around 30% of all sites).

Figure 2.2: Tier model of the Grid is a basis of functioning for heavy user communities at the LHC at CERN.

Another structure is defined in the context of the development of the infrastructure itself. The EGEE sites are grouped into regions defined by Regional Operations Centers (ROCs) which provide operational support to the EGEE Grid sites. ROCs assist the sites in resolving operational problems, coordinate the infrastructure support, deploy middleware releases and provide testing and operational procedures in their region. Fig. 2.3 is a snapshot of the GridMap monitoring page which shows the structure of the regions, where the areas of regions and corresponding sites are proportional to the total computing capacity measured in SPECINT 2000.

2.2.2 Operational dynamics

Large-scale grids are living structures, where availability of resources changes dynamically due to infrastructure maintenance, hardware and software upgrades and random,
unscheduled events. In the EGEE Grid, site availability is tracked by the operation centers, where sites may declare scheduled (planned) and unscheduled (emergency) downtime. During 7 years of operation, nearly 15000 site interventions (64% of planned and 36% unplanned) were recorded. From the point of view of the users, site interventions look as system failures, or service interruptions, which lead to temporary degradation of system performance. Hence, site interventions are an important source of observable instabilities in computing grids.

The weekly number of site interventions has been consistently increasing in the last years. It reflects the growth of the infrastructure and it currently oscillates around 100, as shown in Fig. 2.4 (c). Typical duration of interventions is below 3 hours, but as show in Fig. 2.4 (a), may take up to a week per site. Sites also tend to declare downtime in the unit of full days, what is visible as spikes in the histogram. The weekly accumulated downtime across all sites is above 100 days (Fig. 2.4 (d)), what indicates
that availability of resources is dynamically changing at rather large scale. A clearly visible plateau in Fig. 2.4 (d) reflects the efficiency improvement of operating a large grid: the operational experience of the support teams grows in time but it is balanced by the increased complexity of supporting of a larger system.

How much in advance the users know about the resources becoming unavailable? The notice period, calculated as a difference between the declared start of the intervention and the time of registration of the intervention in the operations database, is shown in Fig. 2.4 (b). Large majority of interventions is registered very late (close to the event time) or even after the event actually happened (in this case the notice period is negative). A majority of these interventions correspond unscheduled site downtimes due to failures, urgent security patches etc. The tails of up to 1 month in both directions indicate that, on one hand some interventions are planned well in advance, and on the other hand, that quite frequently the operational information does not propagate very efficiently in a large grid collaboration.

Figure 2.4: Operational parameters of the EGEE Grid: (a) frequency histogram of site unavailability duration, (b) histogram of notification delays, (c) evolution of weekly number of interventions in time and (d) weekly accumulated downtime for all sites. Plots (c) and (d) smoothed using a moving window with Gaussian smoothing. Data source: GStat and operations database.
2.2.3 Middleware and application deployment

Certification, quality assurance and testing of new middleware releases is performed centrally [69]. However, maintenance and deployment of the middleware in the EGEE Grid is not centralized because resource providers have local schedules and obligations towards their local (non-grid) communities. This affects the maintenance of existing grid protocols and services as well as the introduction of new ones. The middleware changes are propagated very slowly in a large grid and often take several months or years to complete. Fig. 2.5 shows how gLite middleware installations in the EGEE Grid sites evolve in time. Different curves correspond to different versions of the middleware and show the number of sites with a particular version installed. Multiple middleware versions exist in the Grid at any moment. The lifetime of a particular middleware version spans several years as many grid sites are free to arbitrarily delay the adoption of changes. Therefore, the infrastructure (thus the middleware) evolves quasi-independently of the evolution of the application functionalities. Moreover, efficient evolution of grid protocols in the current EGEE Grid is very difficult and mostly restricted by backwards compatibility. This makes it more difficult to improve existing functionalities in the middleware, especially if they are heavily used.

Figure 2.5: Evolution of deployment of gLite middleware in the EGEE Grid. The number of sites with a given version of gLite middleware installed is shown for all major middleware releases. Plots are semi-transparent and overlaid. Gaussian smoothing was applied with a 5-day moving window. Data source: GOCDB

The installation of application software is decoupled from middleware deployment. A standard approach is based on static distribution of software by the VO managers. The EGEE Grid Information Service may be queried for so-called software tags, which are published at the time of software installation and may be subsequently used for resource brokering and matchmaking of jobs. This centralized approach is not flexible
when users require new or locally modified software to be deployed in grid sites. It is also quite difficult and time consuming to achieve consistency of installed application versions across many sites. Therefore, sometimes users distribute application software dynamically, using storage elements (part of data management system) or using job file sandboxes. The drawback of this approach is that application distribution knowledge is not integrated in the Grid Information Service. Also, the use of network resources may be suboptimal if multiple, redundant transfers of software packages are needed, especially for large software packages. Sometimes software is also distributed out-of-the-grid, using HTTP servers (curl/wget) or even version control systems such as SVN/CVS to checkout latest software updates at runtime (this is the case of the ATLAS PANDA pilot-job framework).

2.3 Grid as a task processing system

2.3.1 Scheduling model

Similarly to classical batch systems, the bulk of EGEE Grid infrastructure is designed for optimizing of the throughput of long, non-interactive jobs. Access to computational resources is realized by the Workload Management System (WMS) using distributed queues in a multi-layer, multi-point, early-binding scheduling architecture presented in Fig. 2.6.

Figure 2.6: Workload Management System and job scheduling model in EGEE Grid. Storage services are deliberately omitted.

User jobs pass through three layers of computing services and associated job queues. The jobs are first transferred to Resource Brokers (RBs) which select suitable Computing Elements (CEs) based on the data acquired from the Information Service. CE

---

In this work we use the terms RB and WMS interchangeably to refer to the job brokering service in EGEE Grid. Historically, RB was used as a name for the first-generation of Grid middleware, while in more recent versions of gLite, WMS was introduced as a name for new service implementation. For
represent computing clusters within sites, and act as a common interface to local batch systems which queue jobs to be run on Worker Nodes (WN). The Information Service keeps a periodically updated status of grid resources and services. The scheduling is multi-point as different jobs of a single user may be routed to the same CE passing through different RBs, and conversely, may be routed to different CEs passing through the same RB. Both CEs and RBs receive load from multiple sources.

The model is based on early binding: users typically define and split the work before submitting the jobs and retrieve the results for successfully completed jobs by polling the WMS. Thus, one job carries one task which is specified when the job is submitted. One distinctive feature of this scheduling architecture is the presence of resubmission loops. Shallow resubmission of jobs is done automatically by the system if jobs may not reach the batch system queue. Shallow resubmission occurs before the job execution on the worker node and may happen due to a number of reasons, such as configuration or operational problems of the computing elements, matchmaking problems due to outdated information on available CE resources, or authentication and authorization problems. Deep resubmission of jobs is done automatically by the system in case of problems with job execution and it is typically due to application-specific errors, such as data access problems, bugs in the application code, or runtime faults. Both types of resubmissions are configurable, and if the retry count is reached, the job is reported to the user as failed.

### 2.3.2 Spatial workload distribution

We expect that members of the same user community, thus of the same VO, generate similar workload patterns as they run similar applications and use similar task submission and processing tools. This should be true for spatial distribution of workload, for example, groups of scientists analyzing a single dataset which is stored in few grid sites. This should also be true for temporal use patterns, for example, when the task processing activities increase before an important community event such as a conference.

Majority of VOs comprise between 10 and 100 users as shown in Fig. 2.7(a). Few individual VOs, mainly LHC experiments, comprise more than 1000 users. The distribution of VO population differs from the distribution of VO activity, shown as an average, daily number of submitted jobs in Fig. 2.7(b). Again, few LHC VOs stand out with several tens, or even several hundreds of thousand jobs a day. The activity of a large fraction of VOs is below 100 jobs a day. LHC VOs tend to generate constant background of job traffic due to heavy, high-throughput reprocessing of data while for many smaller VOs job traffic is more erratic and irregular.

Spatial distribution of workload in the EGEE Grid may represented as a graph, where vertices correspond to CEs and RBs, and edges represent job transfers from RBs to CEs. Job traffic is defined as a number of jobs flowing through a vertex or an edge. To construct workload distribution graphs we use a monitoring sample which covers a week of EGEE Grid activity in all VOs in April 2010\(^8\). This sample is representative for the purposes of this analysis we may safely assume that the goals and scope of RB and WMS services are the same.

\(^8\)RTM data provided by Grid Observatory.
for the typical workloads generated by all EGEE VOs: the distribution of the number jobs per VO in the sample is similar to the one recorded for a full year (shown in Fig. 2.7(b)). The monitoring sample contains some $2 \times 10^6$ job records covering 82 RBs and 614 CEs. The graph is sparse (density=0.035), however, it is difficult to visualize due to a large number of edges (8450). On the other hand, the distribution of edge traffic, shown in Fig. 2.8(a), indicates that the job traffic is small in a large fraction of edges. Therefore, it is possible to simplify the graph with little information loss about the job traffic. Simplification with a cutoff value $v$ consists of removing all edges with job traffic smaller than $v$. For example, for $v = 1$ the simplified graph contains only 75% of edges but it represents 99.5% jobs, and for $v = 15$ graph contains only 35% of edges representing 95% jobs. The number of pruned edges, shown in Fig. 2.8(b), depends on the distribution of vertex degrees and the distribution of edge traffic.

Workload distribution graph\textsuperscript{9} for non-LHC VOs is presented in Fig. 2.9. The job flow reveals small clusters of RBs and CEs with high internal job traffic (A,B) and even dedicated RB-CE pairs (C). Opposite pattern (D) is also visible, where one high-degree RB generates low job traffic to many CEs. Another structure is visible as a shaded convex hull (E) which includes all RBs and CEs which are placed in one country domain.

Certain VOs may generate workload in very particular patterns. For example, in the OPS VO which performs infrastructure testing, one central RB service sends small number of jobs to more than 500 CEs (Fig. 2.10). Smaller “satellite” RBs send jobs to selected CE subsets and contribute to less than 50% of the total job traffic.

### 2.3.3 Information Service

Reliability and efficiency of task processing in the EGEE Grid strongly depends on quality of data provided by the Information Service, which is assumed to be accurate

---

\textsuperscript{9}Graph analysis and visualization was performed using Guess [7].
Figure 2.8: The workload distribution histogram (top) shows the edge traffic (number of jobs flowing in the edges). The edge-wise simplification results are shown in the bottom plot, as a fraction of the number of edges and total job traffic in the graph after pruning with a given cutoff value. For small cutoff values a significant number of the edges is pruned, however, the remaining, simplified graph still represents large majority of jobs observed in the sample.

In practice the Information Service is known to have significant delays in notifications due to its architecture [25] and it suffers from delays in information gathering and publishing and inaccurate data due to poor quality of implementation of middleware components. Inaccurate or false data in the Information Service may create “strange” effects such as “black holes”. This may happen when a large number of jobs is sent to a CE in an instant of time in-between the CE status update in the Information Service, thus creating a sudden workload imbalance and increasing jobs waiting time. A misconfigured site, which is not properly black-listed, may be a reason for a consistent failure of all jobs sent to such a CE, which are then either reported as failed or resubmitted to other sites (thus increasing the job waiting time). Some resource selection parameters, such as runtime usage limits typical in batch systems (e.g. short and long queues) remain poorly implemented in the EGEE Grid because the mapping between the CE and the batch system is not consistently enforced, despite existing resource specification standards such as the GLUE [17] schema and JDL [113] attributes. Therefore, the traditional batch user logic – *send short jobs to fast queues and long jobs to slow queues* – is not readily available in the EGEE Grid.

Medium-term trends in the evolution of available resources, as published in the Information Service, are presented in Fig. 2.11 for three selected VOs. The data was
collected by querying the Information Service using the lcg-infosites utility at regular intervals (every 10 minutes) for a period of seven months. Correlations of the number of available CPUs and CEs for different VOs are visible in both plots and may be due to a significant fraction of physical resources being shared between the VOs. Another interesting feature is visible for Geant4 VO around day 100: a relatively small perturbation in the number of available sites resulted in a large variation of the number of available CPUs. This may be explained by a CE with a large number of CPUs disappearing from the Information Service.

The EGEE Information Service exhibits large variations in stability of provided data. This is clearly visible in Fig. 2.12 which shows the histogram of daily and hourly
Figure 2.10: Workload distribution graph showing centralized job handling in OPS VO. RBs are indicated as squares. CEs are indicated as circles. The main RB is responsible for more than 50% of job transfers to more than 500 CEs. Vertex size and edge width are scaled according to the number of jobs flowing through a given graph element.

Variations, calculated as a difference between maximum and minimum observed number of CEs in a given period. The hourly variations for a small Geant4 VO are much less pronounced than for a large Atlas VO. This may be related to poor scalability of the Information Service services. This effect is even more pronounced for daily variations: a large number of small perturbations in the Geant4 VO contrasts with a small number of large dips and spikes in Atlas VO.

Clearly, these short-term variations may not be fully accounted for the Grid maintenance and operations. They indicate limitations or bugs in the middleware itself. As an example, one particular problem of the Information Service was identified as an erroneous heartbeat implementation: a site which missed to send a heartbeat to the Information Service was automatically removed from the list of published sites. The system was later fixed to allow several heartbeats to be missed before the removal.
A natural question is, if the problems with stability of provided information are intrinsic to grid architectures, or if they are only a “feature” of a particular implementation of the EGEE middleware. The answer is complex, but definitely the architecture of gLite middleware, and Information Service in particular, is prone to provide incorrect information, independently of the brokering model (centralized or client-based). In the case of centralized brokering, many RBs may be setup to handle jobs of a single VO for scalability and operational reasons, however, there are no clear mappings between a VO and a number, and location, of these RB services. In the case of client-side brokering, querying the Information Service for every submitted job would be very inefficient and would increase the job submission time. Therefore, the brokering information is typically cached locally to speed-up job submission, and the clients are responsible for refreshing of their caches. However, the local caches tend to be out-of-date and there are no consistent policies on how they are refreshed by the clients.
2.3.4 Complexity of middleware architecture

The implementation of gLite middleware is based on a large number of middleware components from various providers, integrated into a complex architecture shown in Fig. 2.13. The detailed analysis and understanding of all components is not in the scope of our work, however, several features of the gLite architecture are critical for reliability and efficiency:

- large number of complex interactions between services implies less reliable operation of the system;

- alternative implementations of the CE service (LCG CE and CREAM CE) co-exist in the same system and increase the number of potential failure paths;

- Condor and Globus packages are integrated in several subsystems without a clear definition of component interfaces and responsibilities;

Figure 2.12: Histogram of the difference between highest and lowest number of CEs published in the EGEE Information Service in daily and hourly windows. The histogram bars are semi-transparent and overlaid. Data sample is the same as in Fig. 2.11.
- multiple credentials management services at the user level (VOMS and MyProxy services) must interact with a complex proxy forwarding mechanisms in WMS, including identity switching at the CE level (with glexec).

Figure 2.13: The actual gLite task processing architecture, showing the dependency and complexity of internal service interactions at WMS and CE levels. Diagram courtesy of M.Litmaath. Source: [http://cern.ch/twiki/bin/view/EGEE/EGEEgLiteJobSubmissionSchema](http://cern.ch/twiki/bin/view/EGEE/EGEEgLiteJobSubmissionSchema)
Most of the complexity shown in Fig. 2.13 is not visible to the end users, however, it impacts the reliability and efficiency of processing of user tasks and is responsible for a perception of inadequate Quality of Service provided in grids. Common computing wisdom says that the central enemy of reliability is complexity.

2.3.5 Efficiency and reliability

Efficiency of task processing, defined as the ratio between the CPU time and the wall-clock time, strongly depends on applications. However, the submission and scheduling process before the job starts executing on a worker node is application-independent. Its efficiency may be studied as an intrinsic feature of a grid processing system. The time before the job reaches the worker node includes two main components: the middleware overhead $s$, which is the time for the job to pass from the user, through RB, to the CE, and the on-site queuing delay $q$, which is the time the job spends in a local batch queue (see Fig. 2.6). The turnaround time

$$ m = s + q + t $$

(2.1)

is the total time from submission to notification that the job has completed, and it also includes $t$, which is the job wall-clock execution time.

Fig. 2.14 shows the distribution of execution times for $50 \times 10^4$ successful production jobs from 66 users in the Biomed VO, covering one year (October 2004 to October 2005). Due to limited monitoring capabilities at the time, the analysis is limited to one particular RB\textsuperscript{10}.

The striking feature is the importance of short jobs: the 80% quantile is at 20 s. The second important point is the dispersion of $t$; the mean is 2 s, but the standard deviation is of the order of $10^4$ s. The very large fraction of extremely short jobs is partially due to the high usage of this particular broker by the EGEE Biomed VO. However, it was

\textsuperscript{10}Monitoring traces for grid09.lal.in2p3.fr.
verified with other sources, that for more than 50\% of the overall EGEE jobs in the same period, the execution time was less than 3 minutes.

Fig. 2.15 shows the distribution of the dimensionless overhead factor

$$\alpha_r = (m - t)/t,$$

which is the overhead normalized by the execution time. Left histogram shows the distribution of the full sample: only 26\% of the 53000 jobs are in the first bin, meaning than 74\% of the jobs suffer an overhead factor larger than 25. Right histogram shows a close-up for small overheads and indicates that only 13\% of the jobs experience an overhead factor lower than 2. It is clear that EGEE processing system is inefficient for shorter jobs.

The impact of the middleware and the queuing time on the global overhead is shown in Fig. 2.16, where distribution of $q/s$ indicates that the queuing time is a significant component of the overhead. This behavior was exhibited at an early stage of EGEE usage, where the pressure on the resource was only starting to increase. Finally, the median queuing time is 91 seconds, and the median middleware overhead is 221 seconds.

Reliability of task processing experienced by the grid users is variable in time and is impacted by many factors, as the failures in the grid environment occur at application, system and network levels. A comprehensive overview of reliability and types of failures in the EGEE Grid is provided in [143] and probabilistic modelling of job failures and resubmission strategy is presented in [118]. The study on Grid reliability from end user’s perspective [92] point at the site autonomy as one of the important sources of outages in resource and network elements in the Grid. At the same time job success rate strongly depends on the applications themselves. Despite application differences it is common for the user communities to experience varying job success rates as shown in Fig. 2.17.
Figure 2.16: Distribution of a job queing time as a fraction of middleware overhead, \( q/s \).

Figure 2.17: Distribution of job success rate from July 2009 to May 2010 and monthly reliability in this period for selected VOs. Data: GridView
2.4 Summary

Grids integrate computing resources on a large scale what provides a unique opportunity for many scientific communities. The large scale, however, comes with a cost: grids are structured in complex ways and are inherently dynamic in nature: resources are constantly reconfigured, added and removed. Long- and short-range changes in dynamics apply to use patterns, distribution of workload and processing reliability and efficiency. The delays in job execution, and the amount, quality and processing capacity of available computing resources at a given time, depend on the characteristics intrinsic to grids, such as scale and architecture of middleware services, and temporal activity of a global community of users.