Provisioning and scheduling resources for world-wide data-sharing services
Iosup, A.; Garbacki, P.; Epema, D.H.J.

Published in:
Proceedings of the 2nd IEEE International Conference on e-Science and Grid Computing (e-Science'06, Amsterdam, The Netherlands, December 4-6, 2006)

DOI:
10.1109/E-SCIENCE.2006.261168

Published: 01/01/2006

Citation for published version (APA):
Provisioning and Scheduling Resources for World-Wide Data-Sharing Services

Alexandru Iosup, Paweł Garbacki, D.H.J. Epema
Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology
Mekelweg 4, 2628 CD, Delft, The Netherlands
Email: {A.Iosup, P.Garbacki, D.H.J.Epema}@tudelft.nl
Members of the Virtual Institute on Grid Resource Management Scheduling of CoreGRID

Abstract

Grid computing is becoming the natural way to aggregate and share large and heterogeneous sets of resources. However, grid development and acceptance hinge on proving that grids reliably support large communities of users, and their real applications. In this paper we assess the ability of existing grid infrastructures to provision resources for a class of applications with numerous potential users, namely the class of world-wide data-sharing services. For this purpose, we first analyze the requirements of this class of applications, and match them against the existing spare capacity in three existing large-scale grid environments, namely OSG/Grid3, NorduGrid, and CERN LCG. Having shown that the existing capacity is insufficient, we devise and assess through trace-based simulation five domain-specific scheduling policies. Our findings give evidence that grid technology could be successfully leveraged for world-wide data-sharing services, without impacting the level of service for the currently existing load.

1 Introduction

Grid computing’s long term goal is to become the standard way to share heterogeneous resources, and to aggregate them into virtual platforms, used by multiple organizations and independent users [7]. With the grid infrastructure starting to meet the requirements of such an ambitious goal [1, 21], the current evolution of grids undoubtedly depends on proving that it can run real applications, that are used by large communities of users. In this work we focus on a class of such applications, namely the class of dynamic world-wide data-sharing applications.

World-wide data-sharing applications are currently encompassing millions of concurrent users worldwide1 [13]. Simple applications operating on top of the network layer ensure that popular data are being shared across large communities of people [18]. However, with the dynamic problem of users joining and leaving the collaborative network at relatively short intervals (churn) growing with the collaborative network size [2], it is becoming increasingly difficult to guarantee a reliable infrastructure. Moreover, the quality of service demanded by the collaborative applications rises yearly, e.g., the average network bandwidth has risen from 240 Kbps to 500 Kbps in just two years [20, 13].

Thus, an important research question gains weight at the same pace with the remarkable growth of these applications: “How can world-scale data-sharing services be reliably provided?”. In this paper we seek an answer to this question, and find it as a combination of two issues: the SuperPeer concept, and a mix of resource provisioning and scheduling mechanisms. Our main contributions are:

• We identify the resource provisioning requirements of a class of SuperPeer-based world-wide data-sharing applications, and assess the ability of currently deployed grid infrastructures to accommodate their provisioning needs, without adverse impacts on the level of service provided by the grids for their currently existing load;

• Having shown evidence that the current and future needs of world-wide data-sharing applications exceed the available infrastructure’s capacity, we devise five domain-specific resource allocation policies;

• We assess through trace-based simulations the capabilities of the five policies, if they were to be deployed in three major cluster-based grid infrastructures, namely OSG/Grid3, NorduGrid, and CERN’s LCG, or in a hypothetical grid meta-infrastructure.

2 World-Wide Data-Sharing Services

In this section we present a motivating scenario, the concept of SuperPeers (SP), and a world-wide data-sharing service model that will be used throughout this work.

2.1 Motivating Scenario: World-Scale P2P File-Sharing Environments

World-scale Peer-to-Peer (P2P) file-sharing environments comprise a large number of users (customers, if the access to data is paid) collaborating in data sharing [18]; the data is usually shared as system-independent files. A community is the complete group of customers involved in sharing a single file.

The P2P file-sharing applications (P2P-FS-Apps) offer three main mechanisms to their users: connecting to the network (from hereon used interchangeably with bootstrap), delivering data, and searching for data. Most P2P-FS-Apps are deployed such that they provide bootstrap services on some dedicated resources, with high availability. The customers (peers) connect to the network

---

1Slyck.com lists size information for many such applications.
through these services; with connections taking each a very short time, a few dedicated resources can handle very high demands. However, during failures of the bootstrapping services the P2P-FS-Apps display poor quality of service [20]. Following bootstrapping, peers spend most of their collaborative network time engaged in data transfers. Here, the free-riding phenomenon, that is, the fact that the majority of peers attempt to obtain data without further sharing it [22], greatly reduces the transfer performance. Searching for data can be time-consuming in networks where the content is not reliable: the phenomenon affects up to 50% [15] of the shared data in real environments, depending on the application-specific mechanisms.

2.2 The Role of SuperPeers

There exist deployed world-scale P2P-FS-Apps that guarantee simultaneously 0% pollution rate, low or no free-riding, and which have proven capable to serve millions of users daily [20, 13]. However, even in this case, individual node properties such as churn rate, resource sharing policy, uptime distribution, and connectivity directly affect the quality of service offered by the system [20].

A well known mechanism for improving the quality of service offered by a P2P-FS-App is based on assigning additional responsibilities to the rare high capacity, high availability, nodes, commonly called SuperPeers [25]. SuperPeer-based approaches have been successfully used to improve the search performance [16] and content availability [24] in content sharing networks, as well as to improve the network reliability [8].

As increasing the population of SuperPeers consequently improves the overall system performance, the main issue in providing reliable world-scale data-sharing services is to provision enough resources for the needed population of SuperPeers. Assigning SuperPeers roles to customers which are active in the network at a given moment is not desirable, as this would not give reliable guarantees on the level of service (there may not be enough customers with SuperPeer-like capabilities, or customers may be unwilling to contribute their resources freely for a payed service). In the rest of this paper, we argue that SuperPeers could be deployed on resources provisioned from existing grid infrastructures’ spare capacity, and assess the feasibility and implications of this approach.

3 Resource Provisioning and Scheduling

In this section, we present and discuss resource provisioning and scheduling mechanisms in grid infrastructures, and metrics for assessing the mechanisms’ performance.

3.1 Resource Provisioning

We model the SuperPeer services as multiclass M/G/1 queues, with two job classes: the collaborative environments class, and the flashcrowds class. The flashcrowd class corresponds to the high demands imposed by a large community appearing suddenly (interest peak). Each job is a SuperPeer application that consumes computational and bandwidth resources in an exclusive (space shared) or non-exclusive (time-shared) mode.

Table 1 summarizes the parameters of our service model, and their typical values, as observed in previous measurement studies of large-scale P2P networks [20, 10, 13] and large-scale multimedia streaming networks [19]. Both classes of jobs are characterized by the rate with which new environments need to be created, by the size of the community, by the duration of the community (limited span of the users interest), and by the capacity (in number of SuperPeers) required by a community. Here, there are 50-100 new collaborative communities (normal files), and 1-10 new flashcrowds (high-interest files) generated per day. The service dispatch constants ($\alpha_{CC}$ and $\alpha_{FC}$) model how the SuperPeer dispatches customer requests, and are much lower for collaborative communities (shared resources), than for flashcrowds (exclusive resource access).

In our service model, a collaborative environment of maximum size and duration is generated at the end of each flashcrowd, to emulate the long tail in which the flashcrowds’ number of users distribution typically degenerates. This approach is consistent with the mid- and end-phases of the analytical flashcrowd model by Massoulié and Vojnović [17], and with real scale observed in flashcrowd measurements [19, 20].

We can now answer the question “How many resources should be provisioned for a given world-wide data-sharing service?” The number of needed resources can be computed from a (predicted) community size, and a class to which this community belongs, e.g., normal, or flashcrowd. The required number of resources is limited by the number of free resources available in the system, such that the

<p>| Table 1. The service model parameters. CC, FC, and SP stand for collaborative community, flashcrowd, and super-peer, resp. |
|---------------------------------|---------------------------------|-----------------|</p>
<table>
<thead>
<tr>
<th>Param.</th>
<th>Description</th>
<th>Typical Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{CC}$</td>
<td>CC creation rate</td>
<td>50-100 CCs/day</td>
</tr>
<tr>
<td>$\sigma_{CC}$</td>
<td>Average CC size</td>
<td>0.5-2 KPeers</td>
</tr>
<tr>
<td>$\delta_{CC}$</td>
<td>CC duration</td>
<td>15-30 days</td>
</tr>
<tr>
<td>$\Pi_{CC}$</td>
<td>Capacity required by a CC</td>
<td>$\sigma_{CC}/\alpha_{CC}$ SP</td>
</tr>
<tr>
<td></td>
<td>(depends on $\sigma_{CC}$; $\alpha_{CC}$ is a service dispatch rate constant)</td>
<td>$\alpha_{CC} = 1000$</td>
</tr>
<tr>
<td>$\lambda_{FC}$</td>
<td>FC creation rate</td>
<td>1-10 FCs/day</td>
</tr>
<tr>
<td>$\sigma_{FC}$</td>
<td>Average FC size</td>
<td>10-100 KPeers</td>
</tr>
<tr>
<td>$\delta_{FC}$</td>
<td>FC duration</td>
<td>3-7 days</td>
</tr>
<tr>
<td>$\Pi_{FC}$</td>
<td>Capacity required by an FC</td>
<td>$\sigma_{FC}/\alpha_{FC}$ SP</td>
</tr>
<tr>
<td></td>
<td>(depends on $\sigma_{FC}$; $\alpha_{FC}$ is a service dispatch rate constant)</td>
<td>$\alpha_{FC} = 20,000$</td>
</tr>
</tbody>
</table>
level of service offered by the grid systems to their already existing users is not affected.

### 3.2 Scheduling Policies

Consider the motivating scenario (see Section 2): many collaborative communities compete to obtain the needed resources, on which to deploy SuperPeers. When the number of available resources is insufficient, a scheduling policy is used to select the communities that will receive service. Below we describe five scheduling policies, addressing the domain-specific issue of balancing the number of communities and of peers that do not receive service. We also discuss the relative merits of a random scheduling policy.

**Random Assignment Policy (Random-Pol)** This policy randomly assigns resources to communities. The expected outcome is that the resources are fairly allocated: no community obtains more resources than another on the basis of its properties. An expected advantage is the relative ease of enforcement: the complexity of the assignment is $O(1)$. The expected downside of this policy is that highly populated communities may be left without resources.

**Communities First Policy (CommF-Pol)** This policy attempts to maximize the number of communities that receive the resources needed to operate (the *number of serviced communities*). This policy operates in $O(C)$ time, with $C$ the number of communities, which, given realistic assumptions (see Section 4.2), means that the policy is scalable. The main downside is that highly populated communities, which require many resources to operate, will remain unserviced in case of resource shortages.

**Peers First Policy (PeerF-Pol)** This policy attempts to maximize the *number of serviced peers*, by provisioning first larger communities. The PerfF-Pol policy operates in $O(C)$ time, with $C$ the number of communities; similarly to CommF-Pol, the PerfF-Pol policy is scalable. The main downside here is that un-populated communities become *unattractive*, and may remain unserviced.

**Oldest Community First Policy (OldF-Pol)** This policy assigns resources such that the oldest communities receive the resources needed to operate first. This policy rewards customer fidelity; the complement of the OldF-Pol, the *Newest Community First Policy* (NewF-Pol) can be adopted to attract new users; from this work’s perspective, it has identical statistical properties to the OldF-Pol, and will not be independently investigated. The OldF-Pol policy operates in $O(C)$ time, with $C$ the number of communities; similarly to CommF-Pol, the OldF-Pol policy is scalable. The main downside is that new customers may remain unserviced.

**Non-Exclusive Resources First Policy (NonExclF-Pol)** This policy attempts to allocate resources first to communities that are going to share them, which may lead to a higher number of serviced communities. The policy operates in $O(\max(C, R))$ time, with $C$ the number of communities, and $R$ the number of aggregated resources.

### 3.3 Service Metrics

For the performance evaluation of various systems addressing the world-wide data-sharing services we define the following domain-specific service metrics.

The **Community Coverage** metric ($\text{CommCov}$) is defined as $\text{CommCov} = \frac{|C_s|}{|C|} \times 100$, where $C$ is the overall set of collaborative communities, and $C_s$ is the set of successfully serviced collaborative communities. The CommCov metric evaluates the coverage of the service provided to the collaborative communities.

The **Peer Coverage** metric ($\text{PeerCov}$) is defined as $\text{PeerCov} = \frac{|P_s|}{|P|} \times 100$, where $P = \{p, p \in C\}$ is the overall set of peers from all collaborative communities, and $P_s$ is its counterpart for $C_s$. The PeerCov metric evaluates the coverage of the service provided to the members of the collaborative communities (peers).

The **Overall Service Coverage** metric ($\text{AllCov}$) is defined as $\text{AllCov} = \min\{\text{CommCov}, \text{PeerCov}\}$, that is, it evaluates the percentage of time that all service requests were fully satisfied.

The **Extra Service Cost** metric ($\text{XCost}$) is defined as $\text{XCost} = XRes + \alpha_{xc} \times |\text{SysQRes}|$, where $XRes$ is the number of extra resources needed to fulfill all pending service requests, $\text{SysQRes}$ is the number of resources already reserved by other users of the grid system (not worldwide P2P services), over the existing number of resources, and $\alpha_{xc}$ models the penalty for requesting already reserved resources. The XCost quantifies the cost required to provision more resources than normally available.

### 4 Experimental Setup

In this section we present our experimental setup.

#### 4.1 Simulated environment

We have built a custom cycle-based simulator, with the simulation step of 1 day. We consider that resources are provisioned from already deployed grid systems, without interfering with the existing user’s load. Taking into account that the main resource consumption for SuperPeers concerns bandwidth, and thread and socket descriptors, rather than computational power, in our simulation a grid equivalent to a pool of identical resources.

During each cycle, we simulate the creation (arrival) of several world-wide data-sharing services, and assess whether the simulated grid system can provide the needed resources; if the system cannot provide sufficient resources, different scheduling policies yield different performance metrics values. We have adapted tools from the GRENCH-MARK framework [12] to ensure that the *same workload is generated for the evaluation of each scheduling policy*. 
Figure 1. Number of available resource over time for the grid environments.

The reason behind this rather coarse granularity is many-fold. First, this granularity is suitable for our service model (see Section 3.1); for communities with shorter lifetime there is no need for external resource provisioning [20]. Second, this allows us to keep our service model simple, by not taking into account service migration costs. Third, a day-size granularity may help with the practical issues of resource availability prediction.

4.2 Experimental data

Grid traces To extract the experimental data needed for describing the deployed grids’ sizes and their existing user’s load, we use three existing grid traces, and a fourth fabricated trace. We eliminate cases where resources are available for only a couple of minutes up to several hours, and consider only the resources free over a whole day. To the best of the authors’ knowledge, the type of grid data required for this article has not been analyzed elsewhere.

The first grid trace comes from the OSG/Grid3 [11]. We have analyzed the grid-level scheduler traces corresponding to one of the three main Virtual Organizations (from hereon, VO), namely the USATLAS, for over one year (i.e., from 06/2004 to 01/2006). To account for the other grid users, we have conservatively inferred the total utilization as $3.5 \times U$, where $U$ is the system utilization by the USATLAS VO.

The second trace originates from NorduGrid [4]. We have analyzed the full traces of NorduGrid for a period spanning from 2003 to 2006. Since the NorduGrid’s size (number of resources) has varied greatly over time, we use an approximated value of 3100 processors as the average system size, and process the trace accordingly.

The third trace is CERN’s LCG [11]. The official LCG web site publishes activity statistics daily; by building a data crawler, we have obtained and analyzed data spanning over one year (i.e., from 02/2005 to 04/2006).

For the fourth trace, we consider a hypothetic meta-infrastructure encompassing the three infrastructures OSG/Grid3, NorduGrid, and CERN LCG, which we name the MetaGrid. We assume that the MetaGrid will first provision resources to its constituent grids’ users, and only then offer spare capacity to external users; we refer from hereon to this assumption as the internal-use-first assumption.

Figure 1 shows the number of available resource over time. It can be observed that the larger the grid, the higher the number of available resources, e.g., the average number of free resources is over 800, 2000, and 6500 in Grid3, NorduGrid, and LCG, respectively.

Since most of the jobs that currently run in the considered grid environments are single-CPU jobs [11], we can conservatively consider the free resources as available for the first single-CPU job that arrives, including a SuperPeer job; with parallel jobs in the system, queued jobs might have been waiting for resources to be freed, effectively preventing newly-arrived jobs from starting.

World-wide data-sharing service parameters We take the values of the world-wide collaboration model’s parameters (see Section 3.1) from our previous work [20, 13], and from related large-scale models and measurements [19, 17].

For our simulations, we define five workload types: Low Load, Normal Load, High Load, Very High Load, and Extreme Load. Table 2 shows the values of the service model parameters, for the five types of workloads. The data for the Normal Load were extracted directly from our previous data [13]; the values for the High Load and Very High Load workloads were inferred from the trends recently reported [20, 13].

5 The Results

In this section we discuss our simulation results.

5.1 Resource Provisioning

We first focus on the question: "Can currently existing grids provide reliably enough resources for the load of world-wide data-sharing services, without disturbing their existing users?".

Figure 2 shows the evolution of the CommCov and PeerCov metrics over time for the OSG/Grid3, NorduGrid,
Table 2. The used values for the service model parameters.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Low Load</th>
<th>Normal Load</th>
<th>High Load</th>
<th>Very High Load</th>
<th>Extreme Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_{CC}$ [CCs/day]</td>
<td>1,5,10</td>
<td>10,20,50</td>
<td>20,50,100</td>
<td>100,200,500</td>
<td>100,500,1000</td>
</tr>
<tr>
<td>$\sigma_{CC}$ [peers]</td>
<td>20,50,100</td>
<td>20,50,100</td>
<td>50,100,200</td>
<td>100,200,500</td>
<td>200,500,1000</td>
</tr>
<tr>
<td>$\delta_{CC}$ [days]</td>
<td>7,15,30</td>
<td>7,15,30</td>
<td>15,30,90</td>
<td>15,30,90</td>
<td>15,30,90</td>
</tr>
<tr>
<td>$\lambda_{FC}$ [FSs/day]</td>
<td>0,1</td>
<td>0,1,2</td>
<td>1,2,5</td>
<td>2,5,10</td>
<td>2,5,10</td>
</tr>
<tr>
<td>$\sigma_{FC}$ [Kpeers]</td>
<td>5,10</td>
<td>5,10,20</td>
<td>20,50,100</td>
<td>100,500,1000</td>
<td>10,500,1000</td>
</tr>
<tr>
<td>$\delta_{FC}$ [days]</td>
<td>2,3</td>
<td>3,5</td>
<td>3,5,7</td>
<td>3,5,7</td>
<td>5,7,15</td>
</tr>
</tbody>
</table>

![Graphs showing communities and peers coverage over time for Grid3, NorduGrid, and LCG under normal load.](image)

Figure 2. Communities and peers coverage over time for the grid environments, under normal load.

CERN LCG, and MetaGrid environments, under normal load. We observe that even for normal load, only the largest current grid, namely CERN LCG, can ensure continuous coverage both for communities and peers, but that other grid systems may also provide more than 50% coverage. Note that the MetaGrid environment has lower service capabilities (the dip around 04/05) than one of its composing environments, the CERN LCG (!), due to the internal-use-first assumption: it must execute the jobs from overloaded clusters on its otherwise available clusters.

Table 3 shows the best performance per system, given the load class. Each value represents the best achieved performance of any of the scheduling policies, for each metric considered separately. As expected, the observed values for the real environments – OSG/Grid3, NorduGrid, and CERN LCG – show a decrease in performance with the increase in load. By design, the AllCov metric is much more susceptible to system capacity drops than the CommCov and the PeerCov metrics. Notably, for very high load on NorduGrid the AllCov value drops to 0% while the values of CommCov and PeerCov remain over 75% each.

By comparing the values presented in Table 3 with the
Table 3. The best performance per system, given the load class.

<table>
<thead>
<tr>
<th>System</th>
<th>OSG/Grid3</th>
<th>NorduGrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load</td>
<td>AllCov</td>
<td>CommCov</td>
</tr>
<tr>
<td>Low</td>
<td>78%</td>
<td>78%</td>
</tr>
<tr>
<td>Normal</td>
<td>76%</td>
<td>76%</td>
</tr>
<tr>
<td>High</td>
<td>71%</td>
<td>75%</td>
</tr>
<tr>
<td>Very High</td>
<td>0%</td>
<td>5%</td>
</tr>
<tr>
<td>Extreme</td>
<td>0%</td>
<td>1%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>CERN LCG</th>
<th>MetaGrid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load</td>
<td>AllCov</td>
<td>CommCov</td>
</tr>
<tr>
<td>Low</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Normal</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>High</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Very High</td>
<td>95%</td>
<td>100%</td>
</tr>
<tr>
<td>Extreme</td>
<td>0%</td>
<td>7%</td>
</tr>
</tbody>
</table>

Figure 3. The optimal system size for the first six months, under all load sizes.

optimal system size, as depicted in Figure 3, we observe that the general system size of the considered systems is at most half the optimum system size, for the extreme load (the potential high load of tomorrow). Assuming that the current grid load will grow at the same pace as the resource provisioning requirements, we infer that the optimal grid size to accustom such combined load is of around 100,000 resources (double the maximum size of today’s CERN LCG grid). However, we raise a question mark on the feasibility of enlarging grids to such sizes based solely on existing infrastructure. Besides the logistical problem, potentially solved by future research, federated resources may bring with them pending load, which will be dispatched on the other resources, lowering therefore the perceived global system throughput (see Figure 2 for anecdotal evidence !). We caution against the application of the Larger the better! syntagma, without adequate usage policies.

We conclude that current grid systems can at best handle reliably up to the normal load of world-wide collaborative services, but should double their capacity to accommodate the extreme loads that we can expect in the near future.

5.2 Scheduling Policies

We now turn our attention to the validation and comparison of the resource allocation mechanisms proposed in Section 3.2. We present in this section only the results obtained for very high load; we have obtained similar results for the other considered loads.

Figure 4 depicts and the percentages of served communities and peers for the five resource allocation mechanisms operating in the Grid3, NorduGrid, CERN LCG, and Meta-Grid, under very high load. Only the first three months of service are shown for each system-policy combination; the policies performed similarly throughout the whole duration of the traces. The Random-Pol achieves a remarkably good balance between peer and community service. As expected, the CommF-Pol and the PeerF-Pol maximize the percentage of served communities and peers, respectively. According to their goals, the OldF-Pol and the NonExclF-Pol both act as trade-offs between maximizing the percentage of served communities, and maximizing the percentage of served peers. We conclude that the proposed resource allocation policies perform according to their design.

To compare the policies, we evaluate for each load size the performance of each policy in every experimental environment. We consider a policy to be the best for a certain load if it performs on average at least 5% better than the next best policy, on every experimental environment. We observe that the CommF-Pol and the PeerF-Pol are the best in the CommCov and PeerCov categories, respectively, and only for the very high and the extreme loads. For
all other cases, there are at least two very close contenders to the title of best policy. We conclude that the policy design space was efficiently covered by the five selected policies.

6 Related Work

To the best of our knowledge, ours is the first approach to combine on-demand resource provisioning from dynamic environments (grids) with world-wide data-sharing services. Thus, we relate our work to research in world-wide data-sharing services, and to two separated (until now) areas of research: generic on-demand resource provisioning, and large-scale collaborative environments.

There are several approaches for resource provisioning for world-wide data-sharing services from static environments [9, 14] (but none for the dynamic case, e.g., grids). Closest in goals to our work is the Media-On-Demand architecture [9]. The authors take a hierarchical approach to this problem, and work in the context of a privately owned, albeit large, infrastructure, with static load.

Alternatively, there are many approaches for generic on-demand resource provisioning with multiple resource owners [5, 1, 6, 21]. In Epema et al. [5], applications may use the free resources situated in different clusters situated in different geographical locations, under various usage restrictions. Several economic approaches address the issue of generic resource provisioning in multi-cluster grids [6, 21]. Finally, a number of approaches deal with the underlying mechanisms of collaborative environments [19, 23], with the focus being the application control and service, as opposed to our system-oriented view.
7 Conclusions and Ongoing Work

This work has addressed the problem of resource provisioning for world-wide data-sharing services using grid infrastructures.

We have analyzed the requirements of these applications, and have identified the concept of SuperPeers as a potential solution. We have further assessed the capacity of already deployed grids to provision resources for SuperPeer-based services. Our trace-based simulations indicate that three currently existing production grid infrastructures, namely Grid3, LCG, and NorduGrid, can provide the necessary services for normal loads, but that a doubling of grid capacity is required to satisfy the requirements of extreme loads. However, our experiments have also shown that simply federating more resources to a common world-wide grid infrastructure could lead to a degradation of performance, and that new federation mechanisms are required. To cope with the case of resource shortage during high load, we have devised, assessed, and compared five domain-specific scheduling policies.

For the future, we plan to show how to apply in practice our results, in the context of our SuperPeer-based P2P file-sharing applications.

Acknowledgements

This research work is carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265). Part of this work was also carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl), which is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W), and which is part of the ICT innovation program of the Dutch Ministry of Economic Affairs (EZ).

We are thankful to the USATLAs/Grid3 experiment, and to the NorduGrid team (in particular to Dr. Balázs Kónya), which provided their respective Grid traces for our analysis. A.I. wishes to thank Ana Lucia Varbanescu and Catalin Dumitrescu for their helpful discussions.

References


