On the Dynamic Resources Availability in Grids

Alexandru Iosup, Mathieu Jan, Ozan Sonmez, Dick Epema

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On the Dynamic Resource Availability in Grids

Alexandru Iosup — Mathieu Jan — Ozan Sonmez — Dick H.J. Epema

N° 6172

Avril 2007

Thème NUM

ISSN 0249-6399 ISRN INRIA/RR--6172--FR+ENG

On the Dynamic Resource Availability in Grids

Alexandru Iosup* , Mathieu Jan† , Ozan Sonmez* , Dick H.J. Epema*

Thème NUM — Systèmes numériques
Projet Grand-Large

Rapport de recherche n° 6172 — Avril 2007 — 22 pages

Abstract: Currently deployed grids gather together thousands of computational and storage resources for the benefit of a large community of scientists. However, the large scale, the wide geographical spread, and at times the decision of the rightful resource owners to commit the capacity elsewhere, raises serious resource availability issues. Little is known about the characteristics of the grid resource availability, and of the impact of resource unavailability on the performance of grids. In this work, we make first steps in addressing this twofold lack of information. First, we analyze a long-term availability trace and assess the resource availability characteristics of Grid'5000, an experimental grid environment of over 2,500 processors. Based on the results of the analysis, we further propose a model for grid resource availability. Our analysis and modeling results show that grid computational resources become unavailable at a high rate, negatively affecting the ability of grids to execute long jobs. Second, through trace-based simulation, we show evidence that resource availability can have a severe impact on the performance of the grid systems. The results of this step show evidence that the performance of a grid system can rise when availability is taken into consideration, and that human administration of availability change information results in 10-15 times more job failures than for an automated monitoring solution, even for a lowly utilized system.

Key-words: grid, resource availability, performance evaluation, trace-based simulation.

This paper has been submitted to the Grid'2007 conference.

* Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, The Netherlands, {Alexandru.Iosup,O.O.Sonmez,D.H.J.Epema}@tudelft.nl
† LRI/INRIA Futurs, University of Paris South, 91405 Orsay, France, Mathieu.Jan@lri.fr

Unité de recherche INRIA Futurs
Parc Club Orsay Université, ZAC des Vignes,
4, rue Jacques Monod, 91893 ORSAY Cedex (France)
Téléphone : +33 1 72 92 59 00 —Télécopie : +33 1 60 19 66 08

Sur la disponibilité des ressources dans les grilles

Résumé : Les grilles actuellement déployées offrent des centaines de ressources de calcul et de stockage à de larges communautés de scientifiques. Toutefois, la grande echelle, la distribution géographique des ressources et parfois le choix des propriétaires de ces ressources de les allouer pour d'autres besoins, soulève une problématique de disponibilité de ces ressources. Par ailleurs, il n'existe pas actuellement de connaissance précises de la disponibilité des ressources d'une grille. D'autre part, l'impact de l'indisponibilité de ressources sur les performances d'une grille n'a jusqu'à présent pas été étudié. Dans ce papier, nous réalisons un premier pas pour remédier à cette double absence d'informations. Premièrement, nous analysons les caractéristiques d'une trace de plusieurs mois de la disponibilité des ressources de Grid'5000, une grille de recherche de plus de 2 5000 processeurs. À partir de ces résultats, nous proposons un modèle pour la disponibilité des ressources d'une grille. Nos résultats d'analyse et de modélisation montrent que les ressources d'une grille deviennent indisponibles avec un fort taux, affectant de manière importante la capacité d'une grille à exécuter de longues tâches. Deuxièmement, à travers des simulations réutilisant cette trace de disponibilité de Grid'5000, nous montrons que la disponibilité des ressources a un impact fort sur les performances d'une grille. Nos résultats mettent notamment en valeurs le fait que : 1) les performances d'une grille augmentent lorsque la disponibilité des ressources est prise en compte et 2) que comparé à un outils automatique de surveillance de la disponibilité des ressources, une gestion humaine résulte en 10 à 15 fois plus d'échec d'exécution de tâches.

Mots-clés : grille, disponibilité des ressources, évaluation de performances, simulation par traces.

1 Introduction

Large-scale computing environments, such as the current grids CERN LCG [8], NorduGrid [7], TeraGrid [18] and Grid'5000 [4] gather (tens of) thousands of resources for the use of an ever-growing scientific community. At such scale, a significant part of the system resources may be at any time out of the users' reach due to distributed resource ownership, scheduled maintenance, or unpredicted failures. Many of today's grids comprise computing resources grouped in clusters, whose owners may share them only for limited periods of time. Often, many of a grid's resources are removed by their owner from the system, either individually or as complete clusters, to serve other tasks and projects. Furthermore, grids encompass the problems of any large-scale computing environment, with the additional problem that their middleware is relatively immature, which increases further the resource unavailability rate. However, resource availability, and, most importantly, its impact on the performance of large-scale computing environments have yet to be analyzed. To address this gap, in this work we answer two questions.

The first question we address is: **What are the characteristics of the resource (un)availability in large-scale environments?** In Section 2, we present detailed availability results at the resource, the cluster, and the system levels. Several other studies characterize or model the availability of environments such as super- and multi-computers [21], clusters of computers [1, 23], meta-computers (computers connected by a wide-area network, e.g., the Internet, also called desktop grids) [13, 17], and even peer-to-peer (file-sharing) systems [3, 11]. Our analysis is the first based on long-term availability traces from a multi-cluster grid environment. We further model four aspects of grid resource availability: the time when resource failures occur, the duration of a failure, the number of nodes affected by a failure, and the distribution of failures per cluster. The modeling results show that grid computational resources become unavailable at a high rate, and that nodes have rapidly increasing chances of failing with their uptime, negatively affecting the ability of grids to execute long jobs (even single-processor).

The second question we answer is: **What is the performance impact of dynamic resource availability?** We answer this question in Sections 3 and 4. First, we adapt traditional performance indicators, such as utilization for instance, to account for variable resource availability. Then, we show through trace-based simulation of a large-scale environment that the performance when taking into account availability is much better than when availability is not considered. In the first part of our simulations, we contrast the performance of a steady (available at all times) environment to that of a system in which resources fail. We continue our investigations by making various assumptions about the amount of availability information that is available to the resource manager, from perfect information to completely inaccurate. Our results also show that having more availability information is critical to achieve better system performance. During the experimental work for this part, we simulate Grid'5000 based on both workload and availability traces. To the best of our knowledge, ours is the first study to combine real information for both the jobs and the computational resources of a grid, for simulation purposes.

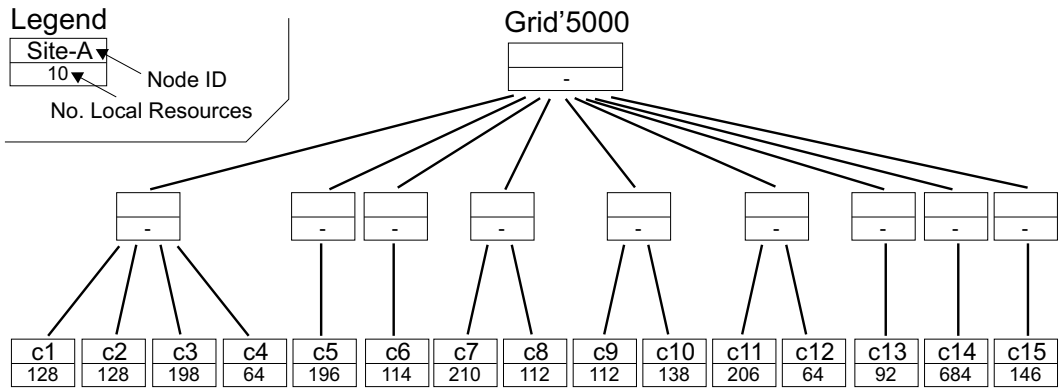

Figure 1: The structure of Grid'5000 (the number of processors per cluster are shown).

2 Resource Availability in Large-Scale Computing Environments

In this section, we present an analysis of resource availability in a large-scale multi-cluster grid: Grid'5000 [4]. Grid'5000 is an experimental grid platform consisting of 9 sites (grid VO) geographically distributed in France. Each site comprises one or several clusters, for a total number of 15 clusters and over 2,500 processors. Each cluster is made of set of bi-processors nodes. Figure 1 shows the structure of Grid'5000. The number of processors per cluster is valid for 12 December, 2006.

2.1 Workload data analysis

We have analyzed availability traces recorded by all batch schedulers handling Grid'5000 clusters (OAR [5]), from mid-may 2005 to mid-November 2006. Altogether, this trace is made of more than half million of individual events that occurs on nodes. Each event in the trace represents a change in the status of nodes: either a node becomes available or unavailable. Note that most clusters of Grid'5000 were made available during the first half of 2005. However, availability information were only activated across the grid platform after mid-may 2005. In addition, note that we filter out from the trace the impact of the reconfiguration system used in Grid'5000, which allows to reboot a set of nodes. In Table 1, we summarize the content of the considered availability trace in this work, and the corresponding workload trace for this period. We refer the reader to the Grid Workloads Archive [22] for more details about the workload trace of Grid'5000.

In the remainder of this section, we first perform an analysis at grid and clusters level, that is by considering nodes from the whole platform and restricted to a specific cluster, respectively. Then, we perform an analysis at nodes level, that is considering all nodes

Table 1: Summary of the Grid'5000 availability and workload traces.

		Observed				
Period	Clusters	No. Nodes	Avail. Events	No. Users	User Jobs	Work [CPUy]
05/'05-11/'06	15	1296	600k	445	750K	585

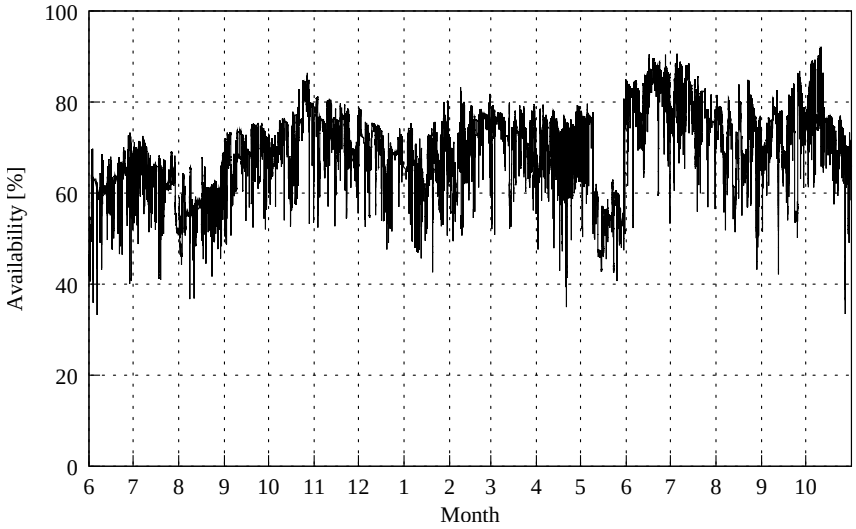

Figure 2: Availability of resources in Grid'5000 at a grid level over the time.

across platform. The difference being that an node level analysis shows values of metrics for individual nodes, whereas a grid level analysis show values for the platform considered as a single entity.

Figure 2 shows the availability of resources in Grid'5000, at a grid level[1]. In average, resource availability in Grid'5000 at this level is 69% (±11.4), with a maximum of 92% and a minimum of 35%. The mean time between failures (MTBF) of the environment is of 744±2631 seconds, that is around 12 minutes. Figure 3 shows the cumulative distribution function (CDF) of the different values of this MTBF for Grid'5000. At a cluster level, resource availability varies from 39% up to 98% across the 15 clusters. The average MTBF for all clusters is 18404±13848 seconds, so around 5 hours. As expected, this value is much higher than the MTBF at the grid level.

[1]May 2005 is not shown as availability information of clusters are starting to be recorded at different date during this month.

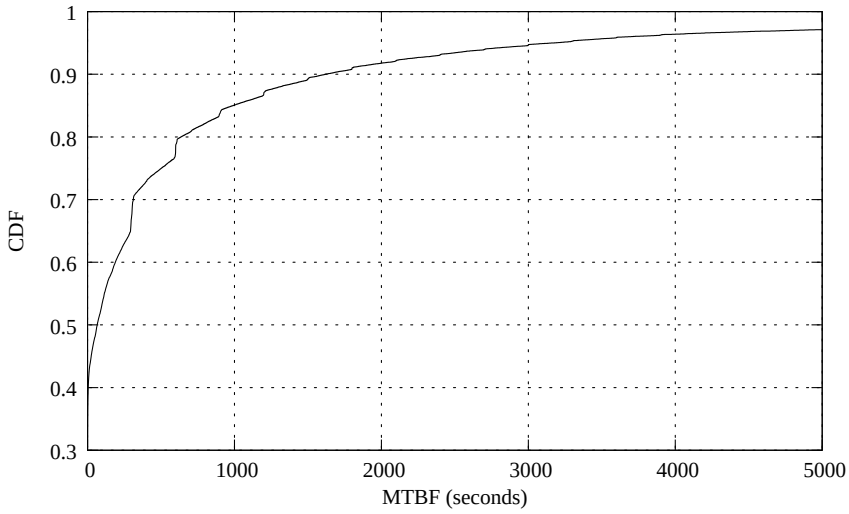

Figure 3: Cumulative distribution function (CDF) of the MTBF of Grid'5000.

At a node level, our analysis shows that in average a node fails 228 times (for a trace that spans over 548 days). However, some nodes fail only once or even never according to our results. Figure 4 shows the CDF of the number of unavailability (also called failures in the remainder of the paper) and availability events, per node, for all nodes of Grid'5000. We define the duration of a failure as the time elapsed between the occurrence of the failure, and the recovery of the resource affected by the failure. Note that our definition is similar to that of recovery time used in [6] or of time-to-repair used in [21, 23]. We define in a similar way the duration of an availability. The average availability duration of a node is 161315±113678 seconds (45 hours), whereas the average failure duration is of 51375±48267 seconds (14 hours). The latter value is quite low, while the former value is quite high. However in both case, the standard deviation is quite high. In addition, for the failure duration, values may include night hours during which administrators of sites of a grid are not available. Furthermore, some node failures may require, for instance, a processor or a memory slot to be replaced, adding delays to this failure duration. We define the inter-arrival time of availabilities/failures, at the node level, as the time between two consecutive availability/failure events on the same node. The average inter-arrival time of availabilities is 212848±121122 seconds (59 hours), whereas it is of 11497901±9801502 seconds for failures.

In addition to this analysis at node level, we also performed an analysis of the potential patterns for (un)availability events. Figure 5 shows the daily and weekly patterns for these events. Our results show a similar results as for the patterns of jobs: a daily peak during day hours (from 8am to 8pm) and a weekly pattern, less events occurring during weekends.

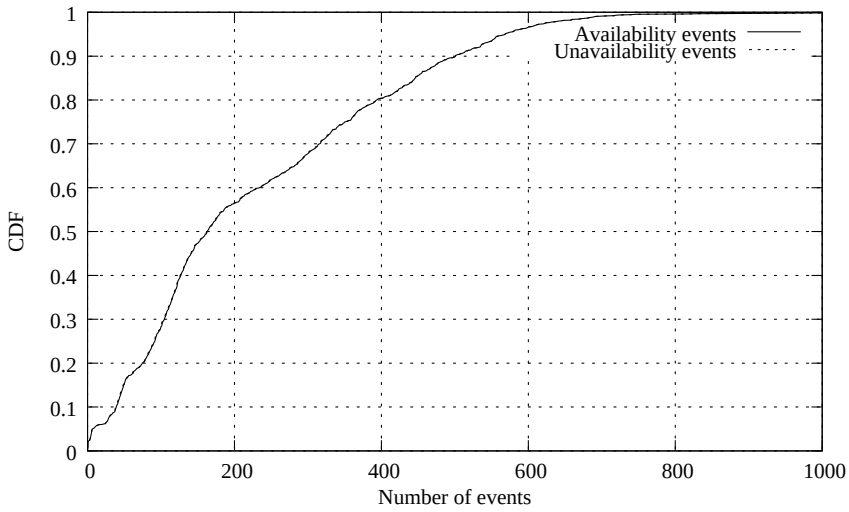

Figure 4: Cumulative distribution function (CDF) of the number of availability and failure events per node for all nodes of Grid'5000.

In addition, we can clearly see the impact of the increasing size of Grid'5000 between 2005 and 2006 and, more importantly, the increase in its utilization.

Finally, we have also investigated the notion of groups of unavailabilities, which we called correlated failures. We define $TS(\cdot)$ a function that returns the time stamp of an event. We therefore define correlated failures, with time parameter Δ, as a set of failures (ordered according to increasing event time), in which for any two successive failures E and F, $TS(F) \leq TS(E) + \Delta$. Note that we do not take into account the origin of the cluster for an individual failure to build a correlated failure. In our analysis, we vary Δ from 1 to 3600 seconds. However, we selected $\Delta = 60s$ as: 1) results for previous Δ (1, 10 and 30 seconds) show similar results and 2) this value is twice a commonly used value for timeout/delays in network operations (30 seconds). Figures 6 shows the CDF of the size of correlated failures for $\Delta = 60s$. Our analysis shows that in average the size of a correlated failure is 11.0±21.0, with a maximum of 339. This latter value is little less than the size of the largest cluster, which is made of 342 nodes. To confirm this value, we have analyzed the number of sites involved in a correlated failure. In average, this value is indeed of 1.06±0 with a maximum of 3 (for $\Delta = 60s$), that is to say that correlated failures generally do not expand beyond a site. To conclude about correlated failures, note that the number of correlated failures is 7473, for a total of around 85k failure events. Therefore, correlated failures represents less than 30% of the total number of failures events in the trace (around 300k).

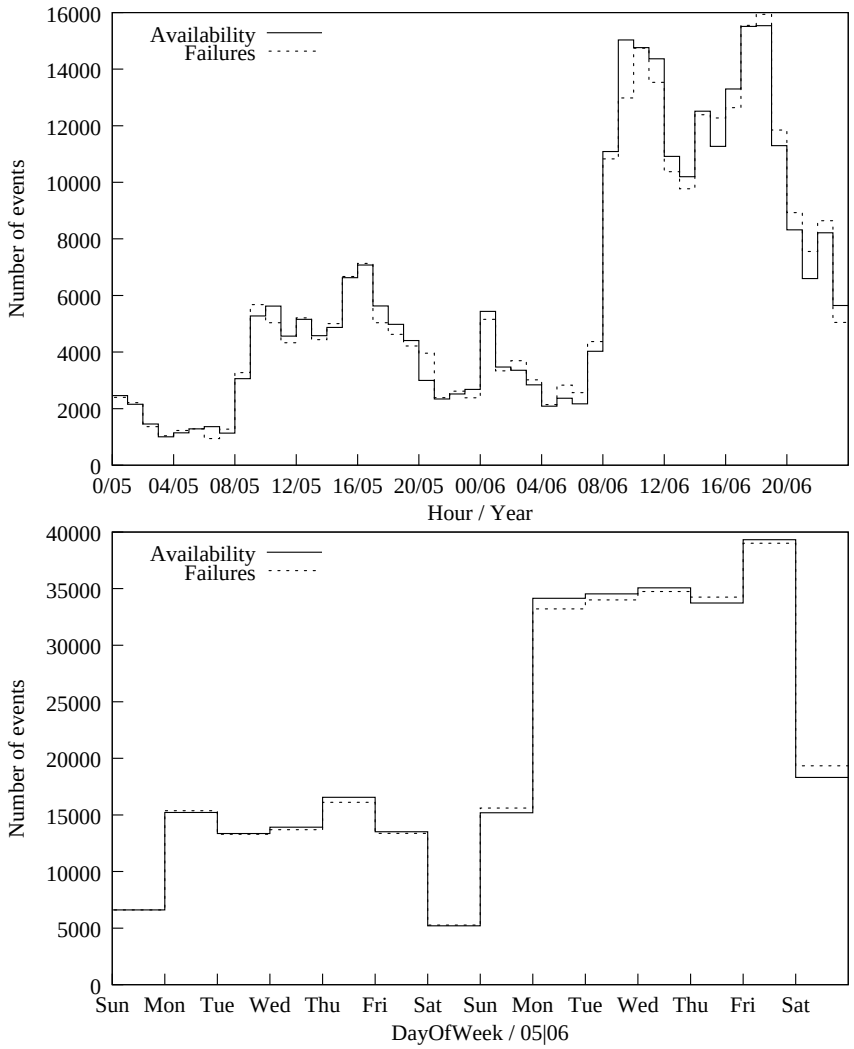

Figure 5: Daily and weekly patterns for the number of (un)availability events.

2.2 Availability Model

In this section, we build a model for resource availability in multi-cluster grids. Our model considers four aspects: 1) the time when resource failures occur, 2) the duration of a failure, 3) the number of nodes affected by a failure and 4) the distribution of failures per cluster.

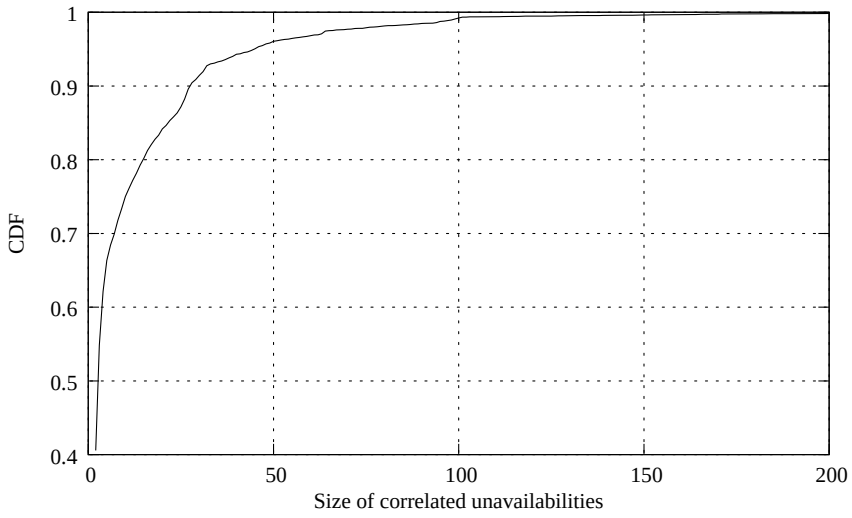

Figure 6: CDF of the size of correlated failures for $\Delta = 60s$.

Compared to traditional resource availability models [21, 10, 23], ours adds the necessary link between the failures and the clusters where they occur.

We begin by summarizing the basic statistical properties illustrated in the previous section. Table 2 shows the minimum, maximum, mean, and median values, and, for completion, the 1^{st} and the 3^{rd} quantiles of the availability Grid'5000 trace. The results for inter-arrival time and for size (rows A and C, respectively), show that the ratio between the mean and the median is relatively homogeneous across clusters. This indicates that a single distribution parameters (with the exception of scale) could be used across all clusters for a good fit with the inter-arrival time data and with the size data, respectively. Depending on the ratio between the mean and the median of the duration of failures, there are two main classes of clusters: class 1 with a ratio of about 1:1 (clusters c1 and c8), and class 2 with a ratio of about 1:6-9 (the remaining clusters). This may indicate the need for separate distributions for each of the classes, or for a distribution with more degrees of freedom, e.g., hyper-exponential or hyper-gamma. However, class 1 contains only clusters where only few jobs have been submitted over the duration considered for this study. Therefore, we choose to disregard this class, and use a single distribution to model the failure duration.

o We first perform a graphical analysis of fitting the availability data. This allows us to eliminate from the modeling process the distributions which clearly do not fit to the data. Figure 7 shows the graphical analysis of the fit between the CDF of the logarithm of inter-arrival time of failures, for one cluster. Clearly, the exponential distribution is not a good fit. The other distributions yield reasonably close results, with the Weibull distribution looking especially promising. Figure 2.2 leads to similar conclusions. Figure 8 shows the graphical

Table 2: Summary of the basic statistical properties of the logarithmic Grid'5000 availability data (log(item) for every item in data). Values higher than 10000 have been reported as ">10k". Omitted Max value rows that contain only ">10k" values.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
A. Inter-arrival time between consecutive failures [s]															
Min.	45	19	3	48	36	23	21	150	54	9	36	84	109	9	4
1st Qu.	3513	1165	1150	604	1357	1500	1640	3002	1709	1005	1509	533.5	601	2356	901
Median	7258	3388	1794	1207	3903	4764	4584	5213	4311	4225	4012	1883.5	1202	4660	1517
Mean	6527	4608	2841	2817	4927	5399	5202	5734	5239	4640	4864	3492.3	3178	5369	3041
3rd Qu.	>10k	8702	3475	3600	>10k	9824	9570	9783	9703	8156	8142	6432.8	4397	8582	3495
B. Failure duration [s]															
Min.	9	0	0	0	0	4	2	2	1	7	0	2	0	0	0
1st Qu.	971	122	195	151	97	159	126	358.3	247	116	152	86	125	106	175
Median	4475	263.5	225	303	100	163	157	1432	259	121	163	110	181	124	201
Mean	5022	1907.3	1047	2025	1655	553.7	772	1301.2	1103	418.3	560.6	1272	421	560.9	995
3rd Qu.	9631	1832.8	1080	2445	679	169	479	1762.8	294	134	179	1157	339	162	1304
C. Failure size [number of processors]															
Min.	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1st	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Median	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1
Mean	9.408	7.41	7.919	3.16	4.9	5.946	6.066	5.032	5.762	5.802	5.389	6.164	3.377	2.625	6.371
3rd	4	7	10	1	2	2	2	2	1	2	2	6	1	2	4
Max.	97	200	165	228	319	267	185	336	200	98	100	43	295	339	67

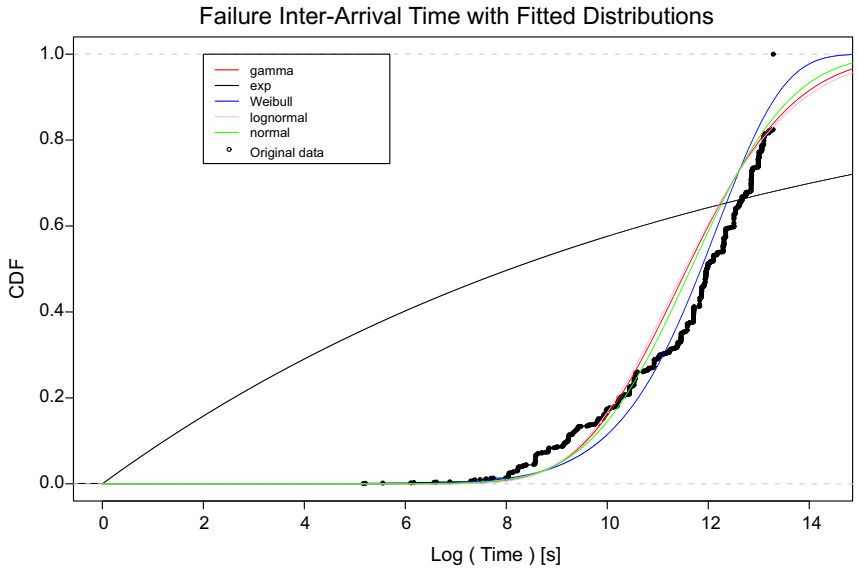

Figure 7: Sample fit between the CDF of the logarithm of inter-arrival time between consecutive failures for a cluster, and various statistical distributions.

analysis of the fit between the CDF of the logarithm of failure duration, for one cluster. The Weibull, log-normal and even normal distributions look promising. We therefore select for detailed model fitting the normal, the log-normal, the Weibull, and the gamma distributions.

We now attempt to fit statistical distributions to our availability data. We use the following distributions: normal, log-normal, exponential, Weibull and gamma. For details regarding each of these distributions, e.g., their probability and their CDFs, we refer to [9]. We fit the above mentioned distributions using the Maximum Likelihood Estimation (MLE) method. Then, we perform goodness-of-fit tests to assess the quality of the fitting for each distribution, and to establish a best fit for each of the model parameters. For each distribution d, we formulate the hypothesis that the Grid'5000 availability data comes from the distribution d, whose parameters are found during the fitting process (*the null-hypothesis* of the goodness-of-fit test). We use the Kolmogorov-Smirnov test (KS-test [14]) for testing the null-hypothesis. The KS-test statistic, D, estimates the maximal distance between the CDF of the empirical distribution of the input data, and that of the theoretical distribution. The null-hypothesis is rejected if the D is greater than the critical value obtained from the KS-test table. The KS-test is robust in outcome (i.e., the value of the D statistic is not affected by scale changes, like using logarithmic values). The KS-test has the advantage over other traditional goodness-of-fit tests, like the t-test or the chi-square test, of making no

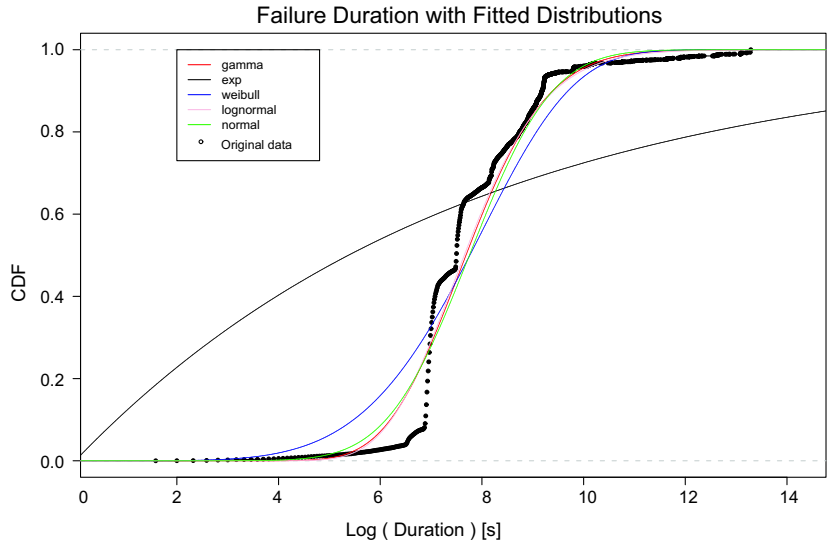

Figure 8: Sample fit between the CDF of the logarithm of failure duration for a cluster, and various statistical distributions.

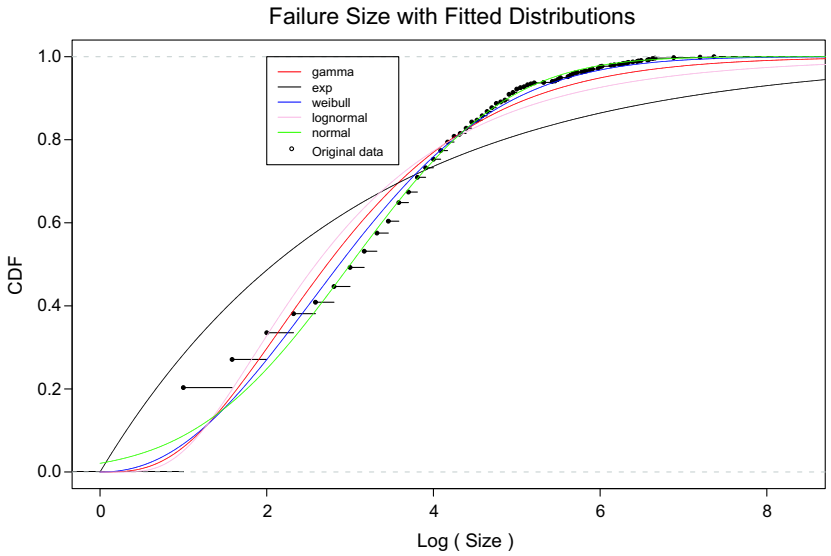

Table 3: Fitted model for the inter-arrival between consecutive failures (in seconds), for the Grid'5000 (noted G5K) availability data, per cluster and for the whole system.

	Weibull	
Cluster	α	β
C1	13.32417	12.76841
C2	8.82691	12.14668
C3	9.49738	11.50066
C4	7.19553	11.29538
C5	7.88283	12.18533
C6	9.64997	12.41524
C7	9.96752	12.37562
C8	12.94117	12.58577
C9	10.71829	12.41374
C10	7.79759	12.09732
C11	10.21044	12.29004
C12	6.60194	11.54027
C13	7.29009	11.47595
C14	11.88544	12.47498
C15	8.11074	11.49423
G5K	9.66772	12.23796

	Log-Normal	
Cluster	μ	σ
C1	2.40913	0.22558
C2	2.14771	0.27527
C3	2.15353	0.19601
C4	2.14365	0.34139
C5	2.03851	0.29664
C6	2.03350	0.13773
C7	2.07713	0.19244
C8	2.26752	0.17555
C9	2.13891	0.18302
C10	1.97464	0.14232
C11	2.03296	0.14878
C12	2.07510	0.28043
C13	2.03505	0.16220
C14	1.99036	0.17595
C15	2.13636	0.22205
G5K	2.33916	0.26363

assumption about the distribution of data[2]. The KS-test can disprove the null-hypothesis, but *cannot* prove it. However, a lower value of D indicates better similarity and a higher degree of similarity between the input data and data sampled from the theoretical distributions. We use this latter property to select the best fits. We find that the best fits for the inter-arrival time between failures, the duration of a failure, and the number of nodes affected by a failure, are the Weibull, Log-Normal, and Weibull, respectively. Tables 3 and 4 show the parameter values of the best fit of the best model for the Grid'5000 availability data per cluster and for the overall system, respectively. The results for inter-arrival time between consecutive failures are alarming: the shape parameter of the Weibull distribution is (high) above 1, which indicates an increasing hazard rate function (the frequency with which a system or component fails, provided that it has survived so far [6]). This indicates that the longer a computing node stays in the system, the higher the probability of the node's failure, preventing long jobs from finishing.

To complete the model, we need to decide where should a new failure occur. We consider for this the fraction f_s of failures occurring at site s, from the total number of failures occurring in the system. Table 5 shows the number of failures and the f_s value[3], per site.

[2]Pearson's chi-square test is applied to binned data (e.g., a data histogram). However, the value of the test depends on the how the data is binned.

[3]However, note that these numbers depend on the date when clusters were made available to users.

Table 4: Fitted model for the failure size (number of processors), for the Grid'5000 availability data, per cluster and for the whole system.

	Weibull	
Cluster	α	β
C1	1.71898	3.60045
C2	2.01629	3.40298
C3	2.17394	3.40015
C4	1.45947	2.45525
C5	1.75449	2.92801
C6	1.69720	3.08240
C7	1.58754	3.07541
C8	1.63994	2.87220
C9	1.59014	3.31293
C10	1.78423	3.38625
C11	1.63853	3.22535
C12	2.42596	3.28213
C13	1.72418	3.17537
C14	1.63197	1.80350
C15	1.81717	3.19713
Grid'5000	1.58526	2.61400

Table 5: Number of failures and the f_s value per site. Doubled separators ("||") group clusters administered by the same site.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Failures	13003	25432	25892	4749	8222	5750	34546	2876	5349	2161
f_s	0.044	0.086	0.088	0.016	0.027	0.019	0.117	0.009	0.018	0.007

	C11	C12	C13	C14	C15
Failures	12901	1123	21131	123353	7417
f_s	0.043	0.003	0.071	0.419	0.033

3 Performance Definition and Analysis

In this section, we define and analyze the performance of a large-scale system when the dynamic availability of resources is or is not considered. Our results show that there is a big difference in the performance of the two cases, which prompts the investigation in Section 4.

3.1 Performance Metrics

The evaluation of grid performances depends on many factors, amongst which the system's architecture, the workload, and also the system's and the user's objectives. For instance, resource providers may have as objective to maximize the number of jobs completed for a specific user. Another possibility is to maximize the utilization of the whole system. Similarly, users may have as objective completing as many jobs during a fixed time interval, or seeing the jobs being started with as little waiting time as possible. Several metrics have been traditionally used as a *de-facto* performance indicator of a grid, as they have often contrary impact on the performance of a system. However, in lack of availability-aware performance metrics, the performance results of systems with highly dynamic availability, e.g., grids, cannot be compared with those for other systems. This is especially true with the results of the cluster computing and of parallel production environments communities. We propose in the remainder of this section five availability-aware performance metrics, each adapted from a traditional metric.

First, we consider *utilization*, which is defined as the percentage of resources consumed by the system users, from the total resources present in the system, over a period of time. The ideal utilization value is 100%. However, due to resource fragmentation and other reasons, a utilization of 60-70% is considered high for systems that run parallel jobs [12]. For large-scale systems in which resources are not always available, computing utilization raises major practical difficulties, as the resource availability is usually not rigorously and accurately recorded.

We also consider the traditional metrics of *wait time* and *response time*. Note that in multi-cluster environments, jobs may spend time in several levels of queues, and computing the actual wait time of a job becomes a non-trivial task.

Finally, we consider the *normalized throughput* and the *normalized goodput* metrics. The throughput traditionally characterizes the number of jobs finished during a time interval, e.g., one day. Higher throughput values are considered better. The goodput characterizes the amount of resources consumed by all jobs towards their completion (this excludes the amount of time spent waiting in queues or for data to arrive). In both cases, in order to be able to compare grids of different sizes, we normalize these metrics, that is, we divide them by the number of processors in the system.

3.2 Models of Availability Information

The performance of a grid resource manager depends on the availability of the resources it manages. However, it is not the actual availability of computing nodes, but the information regarding it that the resource managers have to use. We therefore introduce below four models of grid availability information, from complete lack of to perfect:

1. Systems with Steady Availability (SA).
This model assumes that all resources are online at all time. Many resource management results are readily available for these steady systems [20, 2].

2. **Systems with Known Availability (KA).**
   This model assumes a system with dynamic resource availability. However, the information regarding availability is perfect (complete and on-time). We are interested to understand what is the impact of perfect availability information on grid performance.

3. **Systems Automated Monitoring of Availability (AMA).**
   This model assumes a system with dynamic resource availability. It also assumes that the most recent resource availability information is available from a monitoring system, which samples periodically the grid for individual computing nodes' availability. If the monitoring period is high, the monitoring information can be stale; if it is low, the monitoring overhead is unbearable for the grid. We are interested to understand what is the impact of the information staleness.

4. **Systems with Human Monitoring of Availability (HMA).**
   This model is similar to the AMA model, but assumes that the availability information is provided by the (human) system administrator at fixed, but relatively large intervals: 1 week or 1 month for instance. We are interested to understand what is the impact of human intervention.

4 Performance evaluation

In this section, we first present our experimental setup for our simulation. Then, we present our results for the previously introduced metrics (see section 3.1) with our different models of availability information (see section 3.2).

4.1 Experimental Setup

We have developed a custom trace-driven discrete event simulator which operates under the assumptions of identical processors for all grid nodes, and of FCFS policy for each cluster. We have simulated the Grid'5000 [4] platform, based on its availability trace (see section 2.1) as well as the associated job trace during this period[4].

In our simulations, jobs may fail due to two reasons. First, a job fails when the scheduler has inaccurate information about the number of available (i.e., alive) processors in the system. Therefore, the scheduler wrongly considers that there are enough number of idle processors for the job. We call this situation a job submission failure. Second, a job fails when at least one processor used by this job crashes. We call this situation a job execution failure. We do not consider jobs that can cope with this situation. The time that the failed job has spent on used processors is taken into consideration for our the performance analysis.

[4]The workload trace is taken from Grid Workload Archive [22].

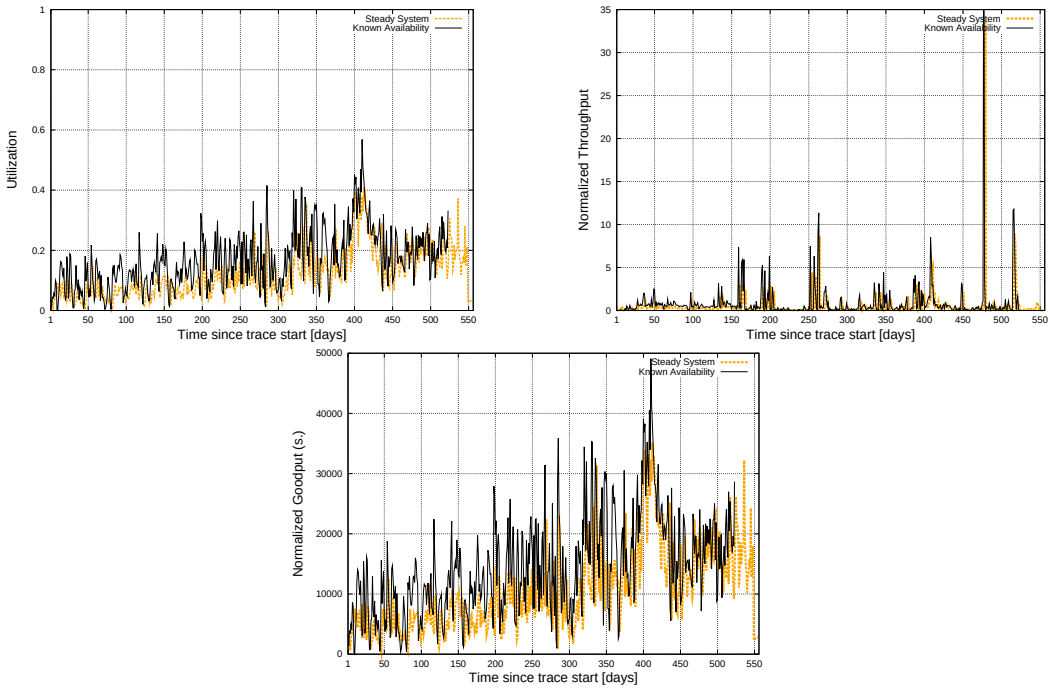

Figure 9: Performance of systems with PA and KA, over $1\frac{1}{2}$ years: utilization (top left), normalized throughput (top right) and goodput-cputime (bottom).

4.2 Results

Figure 9 presents the comparison of utilization (top left), throughput (top right) and goodput-cpu time (bottom), respectively, in a system with perfect availability (PA) and in a system with known availability (KA). As one may expect, the performance of all metrics, when taking into account availability information, is much better compared to the case where it is not (see Table 6 for average values). The reason behind this is obviously that a more precise number of resources are taking into account by schedulers, leading to less job submission failures.

Further, we compare the performance results of two systems with automated monitoring of availability (AMA), with sampling rates of 60 seconds and 1 hour respectively. The first sampling rate reflects a real grid monitoring setting (e.g. Ganglia [16] for instance), whereas the second one represents a reasonably long sampling rate. Table 6 shows that different monitoring intervals do not lead to any relative performance degradation on the considered metrics. This can be interpreted as, in a system with considered resource availability characteristics and under low utilization, submission failures do not have a large impact of the performance of a grid. Moreover, we can also claim that resource failures do not cause that

Table 6: Summary of results for performance metrics

Availability model	Avg. utilization %	Avg. normalized throughput	Avg. normalized goodput-cpu [s]	Avg. wait time [s]	Avg. response time [s]
PA	12.1	0.48	10535	12913	15489
KA	16.6	0.86	14320	12494	14911
AMA (60 s)	16.6	0.86	14320	12494	14911
AMA (1 h)	16.6	0.86	14320	12494	14911
HMA (1 week)	16.0	0.81	13833	10214	12832
HMA (1 month)	15.9	0.81	13808	10173	12713
HMA (fixed)	14.4	0.79	12491	7229	9793

Table 7: Results for number of job completions and failures

Availability model	Number of jobs submitted	Number of jobs completed	Number of job submission failures	Number of job execution failures
PA	739164	739164	0	0
KA	739164	734588	0	4576
AMA (60 s)	739164	734588	0	4576
AMA (1 h)	739164	734588	0	4576
HMA (1 week)	739164	687956	46917	4291
HMA (1 month)	739164	683722	51208	4234
HMA (fixed)	739164	671925	63180	4059

many job failures when the utilization of the system is low (see Table 7). As the comparison results are similar with KA (see Table 6), we do not present the related graphs.

Figure 10 presents the comparison of utilization (top left), throughput (top right) and goodput-cpu time (bottom) in a system with human monitoring of availability (HMA). Intervals are set to 1-week and 1-month. Note, that we only plot the two months period where the differences are the more visible. Figures show that for the considered metrics, 1-week and 1-month intervals give similar results. However, note that considering fixed values for resource availability degrades the performance, compared to results obtained using other models.

Table 7 presents the number of job completions and failures. The results imply that the HMA model leads to more job submission failures compared to the AMA model (see the job submission failure differences between these 2 models in Table 7). Of course, in a real system monitoring has the drawback of network overhead. However, our results also imply that with relatively long monitoring intervals, which would pose relatively low overhead on the network, same performance values can be attained. In addition, Table 7 shows that the number of job submission failures is 10 to 15 times higher than the number of job execution failures. Thus, more work is required to overcome this limitation of current schedulers.

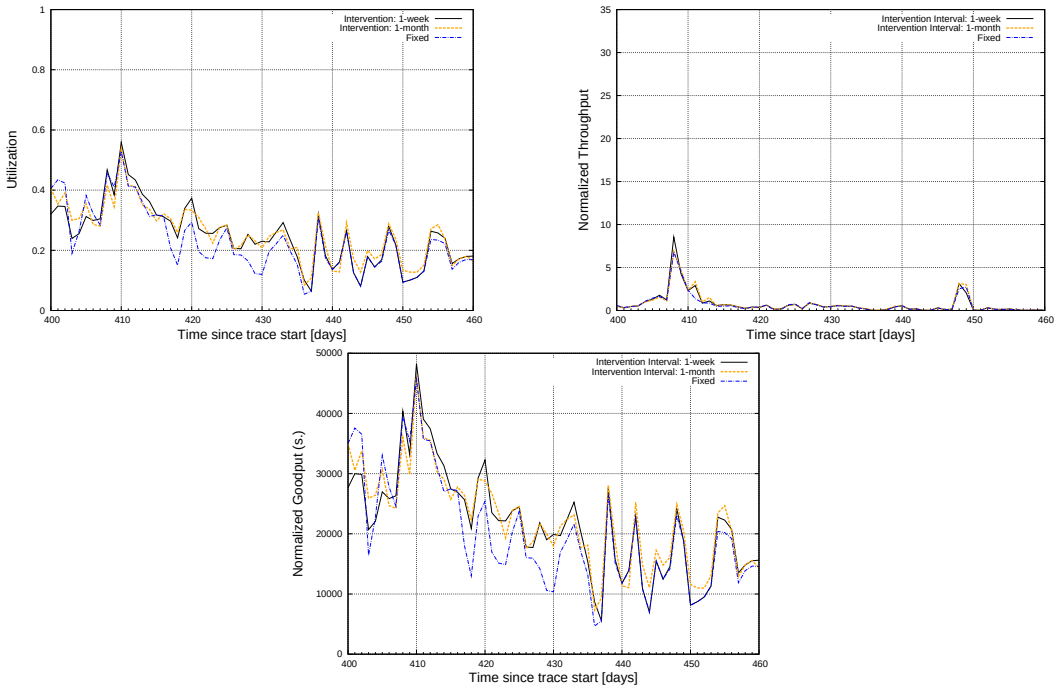

Figure 10: Performance of the system with HMA, over a sample period of 2 months: utilization (top left); normalized throughput (top right); Goodput-cputime (bottom).

5 Related Work

There exists a large number of studies that have considered the characteristics of system and workload (component) failures ([21, 3, 11, 23, 13, 17, 19]). From these, many refer to the systems of up to early 1990s ([10, 21]), are based on data sets spanning at most a few months ([13, 17]) or do not attempt to investigate the impact of these failures on the performance of their originating systems ([10, 3, 17, 19]).

Similarly to this work, the studies in [15, 3, 11, 13, 17, 19] consider uncorrelated failures. Other studies have shown that for some systems there exist bursts of failures ([10, 21, 23]). Our work combines these approaches by analyzing errors at different levels of resource aggregation, e.g., from individual resource to complete grid. Only a few analyze systems of size ([19]) and purpose ([1, 23, 13, 19]) similar to the ones presented in this study.

The study most closely related to ours is [23], which analyze the node (un)availability through CPU failure, and its implication on the performance of large-scale clusters. Through simulation, and using a parallel production environment workload, they assess that the

most important factor affecting performance is the failures arrival rate, which increases dramatically the job response time, and the work overhead.

Also closely related, [1] and [13] analyze the availability of desktop grids. They also give evidence that the performance of such a kind of system is around 70% of that of a cluster composed from equivalent resources, when the workload comprises parallel and sequential jobs, respectively.

6 Conclusion

Currently deployed grids gather together thousands of computational and storage resources for the benefit of a large community of scientists. However, the large scale, the middleware immaturity, and at times the decision of the rightful resource owners to commit the capacity elsewhere, raise important resource availability issues. In this work, we have make first steps in analysis the scale and the characteristics of resource availability in grids.

First, we have analyzed a long-term resource availability trace from a multi-cluster grid, Grid'5000. Our analysis shows that the resource availability in grids varies greatly. We find that the MTBF is high: around 12 minutes at grid level, 5 hours at cluster level, and around 2 days per computing node. The duration of the computing nodes failures is 14 hours. We further find that when a failure occurs, it affects on average 10 or more computing nodes.

Second, we have created a grid resource availability model, which considers the time when resource failures occur, the duration of a failure, the number of nodes affected by a failure, and the distribution of failures per grid cluster. The results for the inter-arrival time between failures are alarming: the shape parameter of the Weibull distribution, our best fit, indicates an increasing hazard rate with rapid effects on the ability of grids to execute long jobs (even single-processor).

Third, we have analyzed the performance impact of dynamic resource availability in grids. We have considered four resource managers with different levels of resource availability information, and we have simulated their use in Grid'5000, based on real traces for both the resource availability and the workload. Our simulations show that: considering resource availability is important when assessing the performance of a grid, and that human monitoring and intervention of the system leads to 10 times more job failures than that of an automated alternative.

As future work, we would like first to validate our resource unavailability model using other traces[5]. We also plan to investigate the effect of varying resource availability characteristics in the model, e.g., the interarrival time between consecutive failure for instance, on the system performance. Finally, to extend our contribution, we plan to study how our results can be as much as possible applied to other large-scale computing environments, and in particular for parallel production environments.

[5]Obtaining availability traces is difficult: resource owners prefer to show that your system works (workload traces), than that it does not (availability traces). We urge potential contributors to consider also the benefits that they will get from resource managers that react properly to resource unavailability.

Acknowledgments

This work was 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 gratefully acknowledge and thank Dr. Franck Cappello and the Grid'5000 team, for providing us with the grid traces used in this work. We also thank Shanny Anoep for his help with the experiments.

References

[1] Anurag Acharya, Guy Edjlali, and Joel Saltz. The utility of exploiting idle workstations for parallel computation. In *Proceedings of the 1997 ACM SIGMETRICS international conference on Measurement and modeling of computer systems (SIGMETRICS '97)*, pages 225–234, Seattle, Washington, USA, 1997. ACM Press.

[2] Susanne Albers and Stefano Leonardi. On-line algorithms. *ACM Comput. Surv.*, 31(3es):4, 1999.

[3] Ranjita Bhagwan, Stefan Savage, and Geoffrey M. Voelker. Understanding availability. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, volume 2735 of *Lecture Notes in Computer Science*, pages 256–267, Berkeley, CA, USA, 2003.

[4] Raphaël Bolze, Franck Cappello, Eddy Caron, Michel Daydé, Frédéric Desprez, Emmanuel Jeannot, Yvon Jégou, Stéphane Lanteri, Julien Leduc, Noredine Melab, Guillaume Mornet, Raymond Namyst, Pascale Primet, Benjamin Quetier, Olivier Richard, El-Ghazali Talbi, and Touché Irena. Grid'5000: a large scale and highly reconfigurable experimental Grid testbed. *International Journal of High Performance Computing Applications*, 20(4):481–494, November 2006.

[5] Nicolas Capit, Georges Da Costa, Yiannis Georgiou, Guillaume Huard, Cyrille Martin, Grgory Mouni, Pierre Neyron, and Olivier Richard. A batch scheduler with high level components. In *Proceedings of the 5th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid '05)*, pages 776–783, Cardiff, UK, May 2005. IEEE Computer Society.

[6] Charles Ebeling. *An Introduction to Reliability and Maintainability Engineering*. McGraw-Hill, Boston, MA, 1997.

[7] Paula Eerola, Balázs Kónya, Oxana Smirnova, Tord Ekelöf, Mattias Ellert, John Renner Hansen, Jakob Langgaard Nielsen, Anders Wäänänen, Aleksandr Konstantinov, Juha Herrala, Miika Tuisku, T. Myklebust, Farid Ould-Saada, and Brian Vinter. The NorduGrid production Grid infrastructure, status and plans. In *4th International Workshop on Grid Computing*, pages 158–165, Phoenix, AZ, USA, November 2003.

[8] EGEE Team. LCG, 2004.

[9] M. Evans, N. Hastings, and B. Peacock. *Statistical Distributions*. Wiley, 3rd edition, 2000.

[10] Jim Gray. A Census of Tandem System Availability Between 1985 and 1990. In *IEEE Trans. on Reliability*, volume 39, pages 409–418, October 1990.

[11] P. Krishna Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *19th ACM Symposium on Operating Systems Principles (SOSP)*, pages 314–329, Bolton Landing, NY, USA, October 2003.

[12] James Patton Jones and Bill Nitzberg. Scheduling for parallel supercomputing: A historical perspective of achievable utilization. In *Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP'99)*, volume 1659 of *Lecture Notes in Computer Science*, pages 1–16, Puerto Rico, April 1999. Springer.

[13] Derrick Kondo, Michela Taufer, Charles L. Brooks III, Henri Casanova, and Andrew A. Chien. Characterizing and evaluating desktop grids: An empirical study. In *Proc. of the 18th International Parallel and Distributed Processing Symposium (IPDPS)*, Santa Fe, NM, USA, April 2004. IEEE Computer Society.

[14] H. W. Lilliefors. On the kolmogorov-smirnov test for the exponential distribution with mean unknown. *Journal of the American Statistical Association*, 64:387–389, 1969.

[15] Darrell D. E. Long, Andrew Muir, and Richard A. Golding. A longitudinal survey of internet host reliability. In *14th Symposium on Reliable Distributed Systems (SRDS)*, pages 2–9, Bad Neuenahr, Germany, September 1995.

[16] Matthew L. Massie, Brent N. Chun, and David E. Culler. The Ganglia Distributed Monitoring System: Design, Implementation And Experience. *Parallel Computing*, 30(7), July 2004.

[17] Daniel Nurmi, John Brevik, and Richard Wolski. Modeling machine availability in enterprise and wide-area distributed computing environments. In *11th International Euro-Par Conference (Euro-Par 2005)*, volume 3648 of *Lecture Notes in Computer Science*, pages 432–441, Lisbon, Portugal, August 2005. Springer.

[18] The TeraGrid Project. NPACI, March 2006.

[19] Bianca Schroeder and Garth A. Gibson. A large-scale study of failures in high-performance computing systems. In *International Conference on Dependable Systems and Networks (DSN 2006)*, pages 249–258, Philadelphia, PA, USA, June 2006. IEEE Computer Society.

[20] Jiri Sgall. On-line scheduling. In Amos Fiat and Gerhard J. Woeginger, editors, *Online Algorithms*, volume 1442 of *Lecture Notes in Computer Science*, pages 196–231. Springer, 1996.

[21] Dong Tang and Ravishankar K. Iyer. Dependability measurement and modeling of a multicomputer system. *IEEE Trans. Computers*, 42(1):62–75, 1993.

[22] The Grid Workloads Archive Team. The Grid Workloads Archive, 2006.

[23] Yanyong Zhang, Mark S. Squillante, Anand Sivasubramaniam, and Ramendra K. Sahoo. Performance implications of failures in large-scale cluster scheduling. In *10th International Workshop Job Scheduling Strategies for Parallel Processing (JSSPP)*, number 3277 in Lecture Notes in Computer Science, pages 233–252, New York, NY, USA, June 2004.

Unité de recherche INRIA Futurs
Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399