How are real grids used? The analysis of four grid traces and its implications

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

How are Real Grids Used?
The Analysis of Four Grid Traces and Its Implications

Alexandru Iosup [‡], Catalin Dumitrescu [‡], and Dick Epema [‡]
Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology
Mekelweg 4, 2628 CD, Delft, The Netherlands
email: {A.Iosup,C.Dumitrescu,D.H.J.Epema}@tudelft.nl

Hui Li and Lex Wolters
Computer Systems Group, LIACS, Leiden University
Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands
email: {hli,llexx}@liacs.nl

[‡] Members of the Virtual Institute on Resource Management and Scheduling of CoreGRID

Abstract—**The Grid computing vision promises to provide the needed platform for a new and more demanding range of applications. For this promise to become true, a number of hurdles, including the design and deployment of adequate resource management and information services, need to be overcome. In this context, understanding the characteristics of real Grid workloads is a crucial step for improving the quality of existing Grid services, and in guiding the design of new solutions. Towards this goal, in this work we present the characteristics of traces of four real Grid environments, namely LCG, Grid3, and TeraGrid, which are among the largest production Grids currently deployed, and the DAS, which is a research Grid. We focus our analysis on virtual organizations, on users, and on individual jobs characteristics. We further attempt to quantify the evolution and the performance of the Grid systems from which our traces originate. Finally, given the scarcity of the information available for analysis purposes, we discuss the requirements of a new format for Grid traces, and we propose the establishment of a virtual center for workload-based Grid benchmarking data: The Grid Workloads Archive.**

I. INTRODUCTION

Grid technologies provide a means for harnessing the computational and storage power of widely distributed collections of computers. Grid computing has emerged as an important new field, distinguished by its focus on large-scale resource sharing, innovative applications, and coherent user communities usually consisting of scientists in a certain field. There are many ongoing efforts in improving current Grid middleware, in extending current Grids, and in designing and deploying new Grid infrastructures. Running applications in Grids in these circumstances is often a challenging problem due to their scale, their heterogeneity, and in particular, their evolving structure [1].

New Grid services and infrastructure components cannot be designed without a good understanding of how today's Grids are used, and of their performance. Also, existing services cannot be evaluated without understanding the characteristics of real Grid workloads [2], [3], [4], [5]. In order to achieve this goal, Grid traces from various types of environments must be collected and analyzed. In this paper, we present an analyis of several Grid traces taken from four real Grid environments,

namely LCG, Grid3 [6], and TeraGrid [7], which are among the largest production Grids currently deployed, and DAS-2 [8], which is an academic Grid.

The contribution of this paper consists, first, of a thorough analysis of representative Grid traces, which is to our knowledge the first such analysis [3], [9]. Here, we present many statistics and performance metrics of the workloads, including utilization, arrival rate, detailed job characteristics, throughput, average weighted wait/run/response time, and average wait time deviation, as defined by Shan and Oliker [10]. An important finding is that while much Grid research focuses on complex application types (parallel applications, workflows), most of the applications we find in the traces are in fact embarrassingly parallel. Other key findings are: (a) a small number of VOs and users dominate each workload, (b) system evolution can manifest itself at the system, the VO, and the user level, and should be taken into account when provisioning resources, and (c) most of the performance metrics confirm the evaluations of simulation research, with the notable exception of expected wait time. As a second contribution, we discuss, based on the results of our analysis, the shortcomings of the current logging formats, and the implications of creating a community-organized Grid Workloads Archive for Grid benchmarking.

II. TRACE DESCRIPTION

In this section we present the four environments considered in this work, including a summarized description of their workloads and resource management particulars.

The first testbed is the LHC Computing Grid (LCG) [11]. The LCG testbed currently has around 180 active sites with a total number of 25,000 CPUs and 3 PetaBytes of storage. A majority of jobs come from multiple large-scale high energy physics experiments, such as LHCB, CMS, ATLAS and Alice. At the Grid-level, jobs are managed and routed to resources via Resource Brokers (RBs), which conduct the job matchmaking and try to balance the workloads at the global level. In this work we have analyzed traces recorded by the resource manager of one of the largest LCG sites, with 880 CPUs.

System	Period	Number of observed				
		Sites	CPUs	Jobs	GRP	USR
LCG	05/05-01/06	1⋆	880	1.1M	25	206
Grid3	06/04-01/06	29	2208	1.3M	1⋆	19
TeraGrid	08/05-03/06	1⋆	96	1.1M	26	121
DAS-2	02/05-03/06	5	400	602K	12	251
DAS-2, Grid	02/05-03/06	5	400	302K	9	68

The second testbed is Grid3, which represents a multi-virtual organization environment that sustains production level services required by various physics experiments. The infrastructure was composed of more than 30 sites and 4500 CPUs; the participating sites were the main resource providers under various conditions [6]. We have analyzed traces recorded by the Grid-level scheduler corresponding to one of the largest VOs: the Grid3/USATLAS [12]; there are three major VOs in the system, the others being iVDgL and USCMS.

The third testbed is the TeraGrid system, a comprehensive architecture for scientific research, with more than 13.6 TeraFLOPS of computing power, and facilities capable of managing and storing more than 450 TeraBytes of data [7]. We have analyzed traces recorded by the interface between the Grid level scheduler and the local resource manager of one of the TeraGrid sites: the UC/ANL [7].

The fourth Grid testbed considered in our work is the DAS-2 environment [8], a wide-area distributed system consisting of 400 CPUs located at five Dutch Universities. DAS-2 is a research testbed, with the workload composed of a large variety of applications, from simple single CPU jobs to complex co-allocated Grid MPI [13] or IBIS [14] jobs. Jobs can be submitted directly to the local resource managers (i.e., by *system users*), or to Grid interfaces that interface with the local resource managers (i.e., by *Grid users*—the fifth testbed). Note that the DAS-2 and the DAS-2 Grid testbeds are operating in the same environment, and that the DAS-2 testbed includes the DAS-2 Grid testbed. In the DAS there exists an unwritten *reduced use* rule: researchers try to leave the system as free as possible, to allow critical research to be done in time.

III. WORKLOADS ANALYSIS

In this section we present the characteristics of three traces coming from four real Grid environments (see Section II for a description of the traces). We first present system-wide characteristics, then we detail the characteristics per VO/group and per user, and we end with insights into the evolution of the systems under study.

In Table I we summarize the content of the studied traces. Our workload analysis covers more than four million jobs submitted during a period of three years. Note that the traces obtained from the LCG, Grid3, and the TeraGrid testbeds are restricted in content, due to their provenance (see Section II). Specifically, the LCG and the TeraGrid traces comprise data from one site each, and the Grid3 trace reports information about one VO.

In the rest of this work we will use the terms DAS and DAS-2 interchangeably. We will also interchangeably refer to the LCG trace as the *LCG system* or the *LCG environment*; similar formulations will be used from hereon for the Grid3 and for the TeraGrid traces.

A. System-wide characteristics

Figure 1 shows the system utilization over time, for the production environments LCG, Grid3, TeraGrid (upper row), and for the research environments DAS-2, and DAS-2 Grid (lower row). To ease the comparison, the bottom-right-most graph depicts the hourly system utilization during the busiest month, for each environment. The production environments display high (60%) and very high utilizations (+80%), while the research environment has a very low utilization, around 10%. The average LCG system utilization is extremely high, at over 85% on average. This unusually high utilization is caused by the nature of jobs run in LCG: embarrassingly parallel, with long durations relative to the environment overhead. The Grid3 utilization caused by one of the top VOs (see Section II) is also quite high, with an average just above 15%. Speculating on the use incurred by the other major and minor Grid3 VOs, this would lead to a system utilization of roughly 60-70%. The DAS system has a very low average utilization, below 10%. This is a consequence of the unwritten rule of reduced usage (see Section II). The DAS Grid usage is roughly 50% of the overall DAS system utilization, for an average utilization below 5%. We conclude that in Grids there currently is a wide range of expected system utilizations, with expected values above 60% for production systems, and below 15% for research environments.

Figure 2 shows the hourly job arrival rate. Note that the vertical axis of each individual site graph is clipped for better visibility. The average hourly arrival rates for the LCG, Grid3, TeraGrid, DAS-2, and DAS-2 Grid systems are 183, 90, 167, 64, and 32 jobs/hour, respectively. While the traditional yearly, weekly, and daily patterns can be easily observed, it is noteworthy that the number of arrivals in the systems varies greatly, with a peak value of 8322 jobs/hour for LCG, and from 500 to 1000 for the other systems. The number of visible spikes is high, which shows that, with high probability, bags of tasks are arriving in the system (this conclusion is also supported by other findings reported in this section). In LCG, there is no correlation between the number of arrivals and the observed utilization: as the system is over-utilized, new job arrivals do not directly influence the utilization, as they remain in the queues. A similar effect but with different reasons can be observed for the DAS: as jobs have relatively short runtimes (see also Figure 3), their effect on hourly utilization is smoothed-out.

Figure 3 depicts the *cumulative distribution function* (CDF) for the most important job characteristics for the LCG, the Grid3, the DAS-2, and the DAS-2 Grid environments: the inter-arrival time, the wait time, the runtime, the memory consumption, the consumed CPU time, and the job parallelism (number of CPUs per job). In more than 50% of the cases there
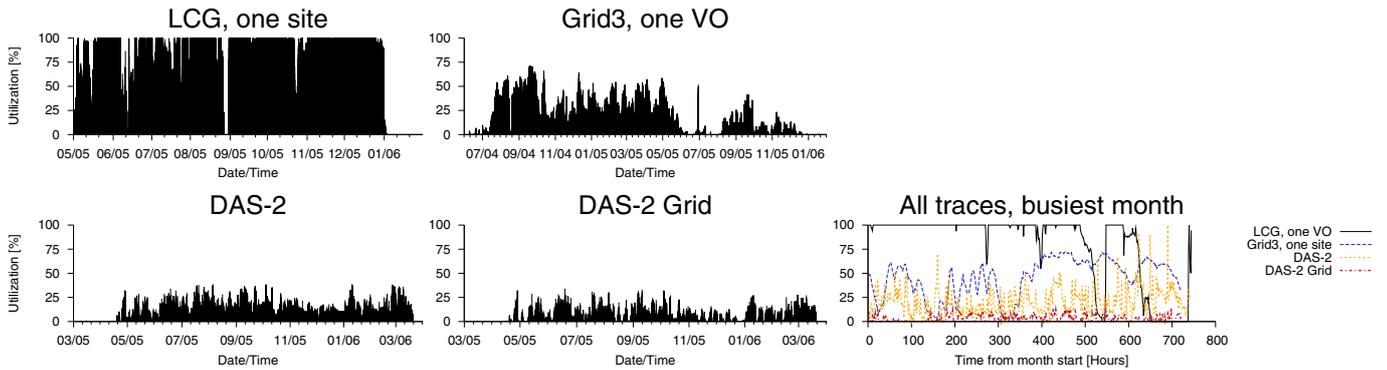
Fig. 1. System utilization over time for LCG, Grid3, DAS-2, and DAS-2 Grid. The busiest month may be different for each system.
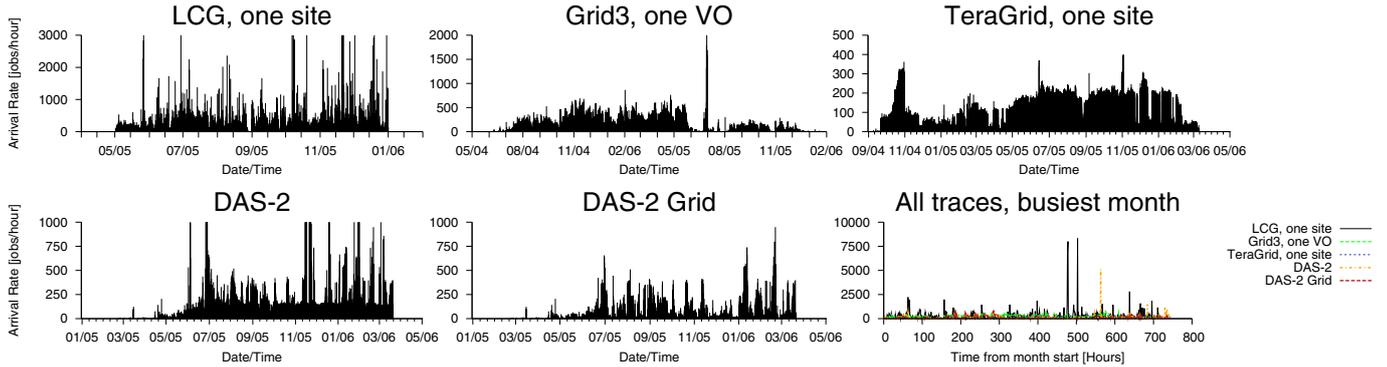

Fig. 2. Job arrival rate during hourly intervals for LCG, Grid3, TeraGrid, DAS-2, and DAS-2 Grid. The vertical axis of each individual site graph is clipped for better visibility. The busiest month may be different for each system.
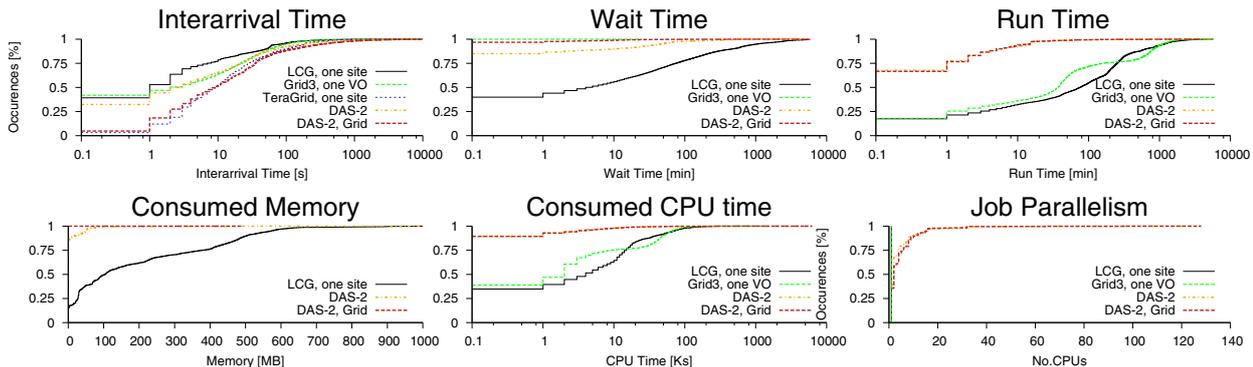

Fig. 3. CDFs of the most important job characteristics for LCG, Grid3, TeraGrid, DAS-2, and DAS-2 Grid. Note the log-like scale for time-related characteristics.

is at least one job arriving 10 seconds after the previous one, and with very high probability a new job will come at most 20 minutes after the previous one, for all systems. The LCG site behaves as a batch processing environment: over 50% of the jobs must wait 10 minutes and more before starting. As expected, the waiting time for the DAS-2 and DAS-2 Grid environments is very small: the systems can be considered as interactive. Surprisingly, we observe this feature for the Grid3 environment as well. We explain this by the fact that the value was reported by the last middleware tool, which is in this case an on-node job wrapper instead of the off-node global or even local scheduler (the *last middleware report* effect). As expected, the jobs run in production environments are much longer (factor of 100) than their research environment's counterparts. Evidence shows that only sequential jobs are

run in the LCG and Grid3 production environments. This means that a high percent of applications still employ an embarrassingly parallel model; arguably, this situation will change with the penetration of new programming models (e.g., Ibis [14]), and with the maturity of the systems (see the *learning curve* effect detailed in Section III-C). The DAS-2 and DAS-2 Grid environments host mostly parallel applications, with almost 50% of the job requests being for power-of-two sizes. But the tendency of preponderantly serial jobs is starting to show even here: the number of serial jobs in the DAS has gone from 2.5% [15] in 2003, to about 30% in our trace. The average size of jobs in DAS-2 and DAS-2 Grid is 4.1 and 4.6, respectively. The CPU time consumption also shows a discrepancy between the production and the research environments: production jobs consume much bigger

amounts than research jobs, with a factor of about 20. The memory consumption is much higher (factor 30-1000) for production jobs, when compared to research jobs. The average job memory consumption for LCG, DAS-2, and DAS-2 Grid is 195.4 MB, 6.5 MB, and 0.1 MB, respectively. The CDF of the memory consumption shows the existence of specific values around which most applications are clustered.

B. VO, group, and user characteristics

Figure 4 depicts the number of submitted jobs and the consumed CPU time by group and by user. To catch the submission patterns, stripes of different colors (shades of gray) group statistics for different weeks. A small number of groups (usually below 5, always below 10) dominate the workloads in both number of submitted jobs and consumed CPU time, for all the analyzed traces. There is usually one dominant group in terms of number of submitted jobs, and two dominant groups in terms of consumed CPU time. In the DAS, the dominant three groups seem to have a rather continuous activity, with few and small demand spikes. In LCG (not shown in the figure), the weekly slices of consumed CPU time vary greatly: there are group spikes in the resource demand. The situation is similar in the case of users: a small number of users dominate the workload; in most cases, the top user is by far dominating the others. In DAS, the top user by the number of submitted jobs is an automated verification tool. Besides the original setup and verification (the two bottom-most stripes in column $U85$), jobs are constantly generated every two hours; the graph depicts this situation with equally-sized stripes. The second most important user in terms of number of submitted jobs has had a bursty activity: during one week he launched more than 30,000 jobs, and for the rest of the time he submitted relatively few jobs.

We make the distinction between jobs submitted by a user to its local site (*local jobs*) and jobs submitted by a user to a site outside the local reach (*non-local jobs*). We argue that for the Grid to become universally adopted, the satisfaction obtained by users when submitting jobs locally should be similar to or better than when submitting jobs remotely. We quantify the satisfaction as a tuple of three metrics: the waiting time, the throughput, and the total number of jobs done metrics. Figure 5 shows the satisfaction obtained for local and non-local job submissions. We observe that in the DAS the number of completed local and non-local jobs are roughly the same, while the throughput is similar for the two classes of jobs. This means that, while the users have a higher probability to submit to their own site than to any other in the system, when they do submit to a non-local cluster they obtain the same satisfaction as when submitting to their local site. We do not have enough information to quantify the user satisfaction given locality for the other traces. Also, note that satisfaction metrics cannot be generalized for jobs with very specific requirements, i.e., jobs that can only run at a specific site, due to data dependencies or licence issues.

C. System evolution

As is the case with any system, Grids evolve with time. We identify three types of evolution: infrastructure evolution,

project (defined as a common goal sought after by a group of users, who submit a large number of jobs to achieve it) evolution, and user evolution. The evolution may be reflected in any of the metrics presented in this analysis; in Figure 6 we present evidence for the evolution of number of completed jobs for all three types of evolution. Our results confirm and complement the findings of Hotovy [16], who shows that traces from parallel environments change as the system matures.

For the infrastructure evolution, we use old traces from the DAS [15], and we observe the total number of completed jobs. The old DAS system trace suffers from the *learning curve* phenomenon: the majority of users do not really know the infrastructure, which leads to a small number of jobs being submitted in the first month (until around day 300, for the DAS). After that period, the users submit at the same rate as the one observed in the DAS trace analyzed in this paper.

For the project evolution, the graph displaying the Grid3 project (*Grid3-PRJ-1*) has a knee around day 350 (date: 06/2005): first there is a steep ascend in the number of completed jobs, then, after 06/2005, there is a milder one. We attribute this difference to a project change of course: as the Grid3 evolved to Open Science Grid (OSG), fewer jobs were submitted to the former. We encounter the same situation, on a larger scale, in the second most important project present in the LCG trace (*LCG-PRJ-1*).

For the user evolution, we look to the DAS users rankings by submitted jobs, and select the highest ranked user (*U85*), and two other users from top10 (*U67* and *U140*). User *U85* has an initial period of roughly 100 days, during which its behavior is unstable, then evolves towards an *automated tool* situation (starts submitting jobs at a constant pace). Not surprisingly, this corresponds to a two-stage user project, in which the development phase is followed by a production phase. User *U67* has a *learning curve*: the first 60-70 days since his first job submission are characterized by few job submissions, then the user starts submitting at full pace. Both users *U67* and *U140* display several *spike periods* (small number of days in which a large number of jobs are submitted); the rest of the time they submit very few or no jobs.

IV. GRID PERFORMANCE ANALYSIS

Figure 7 depicts the daily number of waiting and running jobs over time. The average values of the number of running jobs are between 776 (DAS, grid) and 5144 (LCG) jobs/day. The average values of the number of waiting jobs are between 6.5 (DAS, grid) and 2345 (LCG) jobs/day.

Figure 8 depicts the total number of completed jobs over time. In a typical production system, this graph should show a linear dependency between the number of completed jobs and the time. This is indeed the case for the LCG, the TeraGrid, and the DAS, but not for the Grid3 (in Section III-C we attribute the unusual Grid3 graph shape to the system being discontinued).

We further evaluate the overall performance of each system. From the multitude of potential metrics, we select metrics
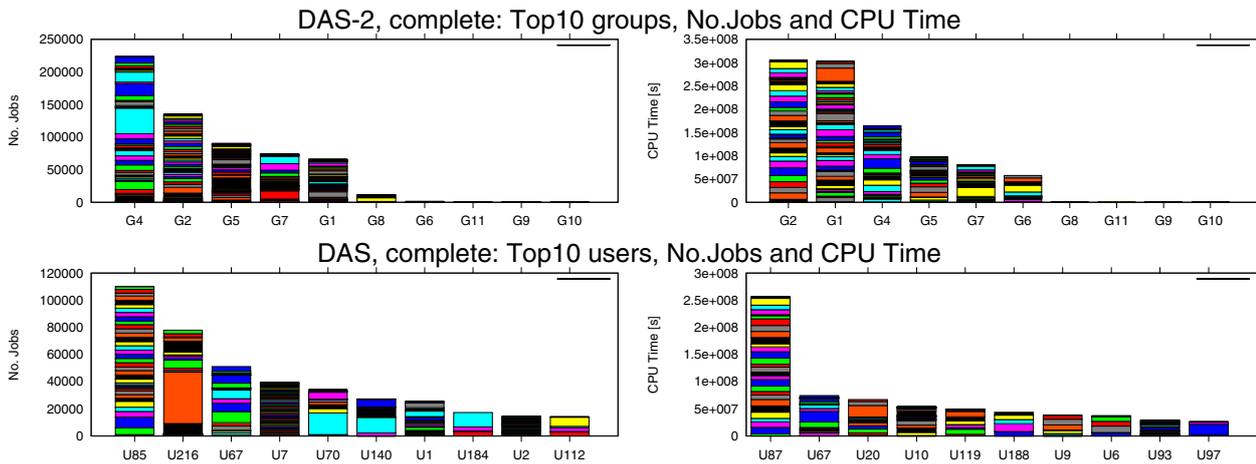
Fig. 4. The number of submitted jobs (left) and the consumed CPU time (right) by group (top) and user(bottom) for DAS-2. Only the top 10 groups / users are displayed. The horizontal axis depicts the group's or user's rank. The vertical axis shows the cumulated values, and the breakdown per week. For each system, the same groups and users have the same labels in the left and right sub-graphs.
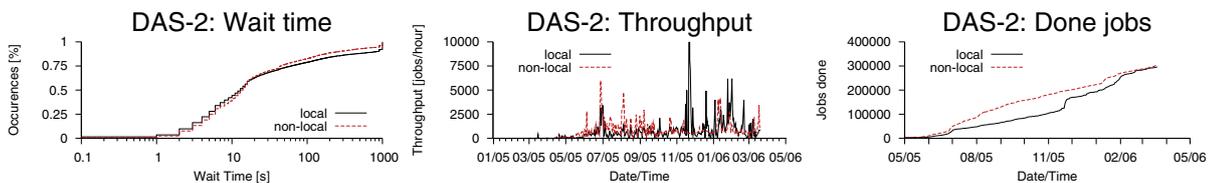

Fig. 5. Satisfaction obtained by users when submitting to local and non-local environments: the waiting time CDF, the throughput over time, and the total number of jobs done over time.
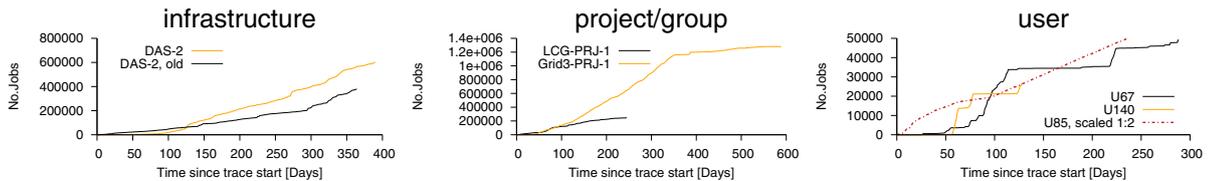

Fig. 6. System evolution: infrastructure evolution (left), project/group evolution (middle), and user evolution (right).
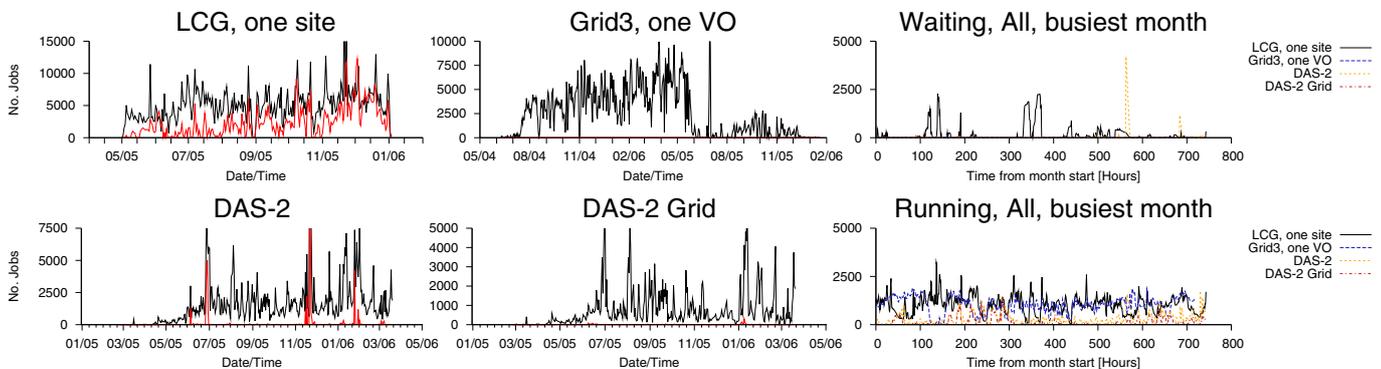

Fig. 7. The number of running and of waiting jobs during hourly intervals for LCG, Grid3, DAS-2, and DAS-2 Grid. The vertical axis of each individual site graph is clipped for better visibility. The busiest month may be different for each system.

for overall system utilization, user satisfaction, and system fairness. For the user satisfaction component, we select the throughput, the *average weighted run time* (AWRT), the *average weighted wait time* (AWWT), and the *average weighted slowdown* AWSD, the latter three defined as in Yahyapour *et al.* [17]. For system fairness, we use the *average wait time deviation* (AWTD), defined in Shan and Oliker [10]. The main advantage of this selection is that it is composed only from well-known metrics and, besides the useful insight into the

performance of the analyzed systems, it allows us an indirect comparison with results obtained for parallel production environments in the past 15 years, and with simulation results for grid environments [10], [17].

Table II presents a summary of the performance characteristics of all the studied environments. We have discussed in previous sections the specific and average values of the utilization and throughput metrics (see Section III-A). It can also be seen again that the top VO/group and user contribute decisively
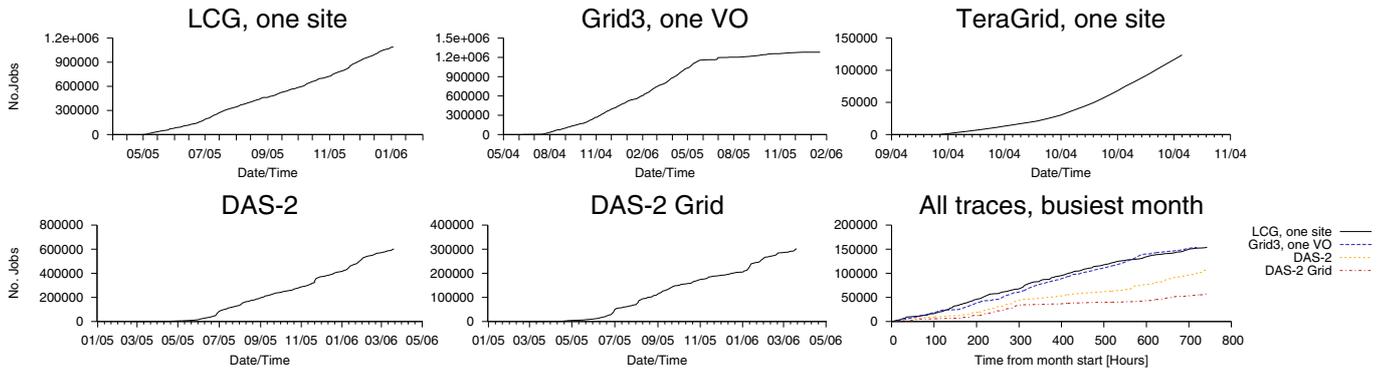
Fig. 8. The number of completed jobs during hourly intervals for LCG, Grid3, DAS-2, and DAS-2 Grid. The busiest month may be different for each system.

TABLE II

SUMMARY OF PERFORMANCE CHARACTERISTICS OF THE LCG, GRID3, DSA-2 AND DAS-2 GRID SYSTEMS, WHEN RUNNING THE ANALYZED TRACES.

TWO ADDITIONAL DATA SUMMARIES ARE SHOWN FOR LCG, GRID3, AND DAS-2: FOR THE TOP GROUP'S JOBS, AND FOR THE TOP USER'S JOBS.

| | LCG, one site | | | Grid3, one VO | | | DAS-2 | | | DAS-2 Grid |
	all	top GRP	top USR	all	top GRP	top USR	all	top GRP	top USR	all
Utilization [%]	84.3	26.2	12.6	15.7	15.7	9.8	7.5	2.8	2.4	3.5
Throughput [jobs/hour]	183.0	74.7	8.0	90.4	90.4	68.0	64.2	15.5	0.6	32.3
AWWT [s]	24692.3	7058.1	121018.3	0.1	0.1	0.1	2457.0	657.2	774.0	3416.6
AWRT [s]	95479.2	44263.9	219962.2	59825.5	59825.5	55757.2	36724.6	25038.6	21665.0	41374.2
AWSD [s]	1.7	1.2	3.6	1.0	1.0	1.0	1.6	1.1	1.0	1.4
AWTD [s]	49.7	18.3	1295.2	0.0	0.0	0.0	4.9	5.8	62.3	5.0

to the overall throughput and utilization (see Section III-B).

The AWWT metric shows the ability of the systems to handle interactive requests, in particular for multiprocessor jobs; the lower the AWWT value, the better. The AWWT values obtained for LCG match the middle segment of the range presented by Yahyapour *et al.* [17] (141-75,940) and fall within the lower end of the heavy load range described as by Shan and Oliker [10] (2,779-254,797). The LCG's AWWT shows that the system cannot handle interactive jobs. We observe that the most active VO is favored with regard to the waiting time (below 25% of the system's AWWT), while the most active user is unfairly treated (over 400% of the system's AWWT). Just like the case for job wait time characteristics (Section III-A), The AWWT values obtained for Grid3 are flawed by the *last middleware report* effect and should be disregarded. The AWWT of the DAS shows that the system could be capable of interactivity even for large jobs. The most active group and user seem to be favored, as a result of running mostly small and/or single-CPU jobs in a system in which even medium-sized parallel jobs lead to fragmentation, as the DAS is not a very large system. Slightly higher values are observed for the DAS-2 Grid, but the observations for the DAS-2 system hold.

The AWRT metric (weighted turnaround) shows the ability of the Grid to finish (large) jobs in time. All AWRT results are higher or match the higher half of the the range predicted by Yahyapour *et al.* [17] (8,644-75,940), but fall within the lower half of the range for heavy load reported by Shan and Oliker [10] (7,756-260,010). This is a result of both job lengths and system middleware; for the latter, we conclude that more work remains to be done in Grid scheduling, with the goal of reducing the AWRT.

The AWSD metric shows the overhead of the Grid; for an AWSD of 1.7, it takes 70% more time to run jobs in the Grid rather than in a dedicated environment. Our results show that for complete systems, the Grid overhead ranges from below 1% to below 70%. This shows that Grid systems are viable alternatives to traditional production environments, which display overheads of 70-548%, cf. Yahyapour *et al.* [17]). The observed Grid AWSD values are in concordance with the expected margins of 1-548% predicted by Yahyapour *et al.* [17] for coupling up to six clusters, and those of 0-66% obtained by Shan and Oliker [10]. Despite the reduced system overhead, some users are disadvantaged: the top user of LCG is experiencing an unusually high overhead of 260%.

Finally, the AWTD metric quantifies the overall fairness, and the performance variability [10]. The LCG's AWTD values give more insight into the fairness of waiting. While the most active VO in LCG will have a low waiting time variability relative to the rest of the system, the most active user can expect to see a high variability in waiting time. This hints to the inability in the latter case to predict the waiting time, which could lead in turn to a reduced scheduling performance. The values for Grid3 are unreliable, due to the *last middleware report* effect. We observe a similar behavior in the DAS-2 as in the LCG, though at a different scale: it is much harder to predict accurately the waiting time for the most active user, than for the most active group or for the system. Note that the predictability of the DAS-2 environment is much higher than that of the LCG. The AWTD values obtained for LCG are consistent with the values for heavy load obtained by Shan and Oliker [10], while the AWTD values obtained for the DAS-2 and DAS-2 Grid systems are consistent with the values for light load obtained by Shan and Oliker [10].

V. Discussion

In the previous sections we have presented an analysis of four relevant grid traces. In this section, we complement our results with a discussion targeting important Grid research topics: workload modeling, benchmarking, and scheduling.

While our analysis gives a first-order estimate on the current use of Grids, more information is needed for full-scale modeling of Grids workloads. We identify two main problems: a *scale* problem, concerning the origin and size of the data that must be collected, and a *methodological* problem, concerning the missing components of the information, if collected with currently available means.

To address adequately the grid's scale problem, the information needs to come from at least three sources at the same time: from the local and grid schedulers, from the grid AAA modules (e.g., authentication), and, sometimes, from the monitoring systems. Additional information from all the used middleware would sometimes be helpful, depending on the goal of the analysis/modeling. Without local scheduling logs, there cannot exist an analysis of site-related performance metrics. Without the grid scheduling logs, job arrival and coupling/dependency information is lost, and we lose the multi-site side from the analysis, that is, the essence of the grid. Without access to AAA logs, the link between jobs and their owners are lost, e.g., when a pool of certificates serves all the users within an organization. Finally, without access to monitoring information it is impossible to understand the true nature of applications running in Grids, in particular the I/O and network consumption patterns, and to quantify the true system utilization. This huge flow of information needs to be collected from all the sites present in the grid, and analyzed as a single, large-scale, workload information database.

The methodological problem is still a subject of active research. While there is usually much work concerning the establishment of an agreed format for storing workload information, such work usually needs to receive a concrete form only once, and can afterwards be used by the whole community for a long period of time. Such an approach is for example that of Feitelson *et al.* [2], which proposes a workload log format for parallel production environments. This work is the basis for the establishment of the Parallel Workloads Archive[1]. For grid environments, initial steps have been taken by Mach *et al.* [18] who propose a Grid Usage Record format. We argue that a number of components need to be added to this approach, of a kind that distinguishes the grid from other large-scale environments. Specifically, *grid logs should additionally contain: detailed grid middleware processing times, the job composition type, error types, and the locality information* (for the latter, see Section III-B).

Just like modeling grid traces, developing grid benchmarks (rules, tools, and data sets) is a difficult task. This and the fact that history shows that the community will select only one benchmark makes investing time in this endeavor an unlikely

[1]The parallel workloads archive, Feb 2005. [Online] Available: http://www.cs.huji.ac.il/labs/parallel/workload/

choice for most academics. With trace-based benchmarking, a large part of the problem is already addressed [2], [5]. Besides the large amount of results existing in the various communities dedicated to computer architecture evaluation (rules already exist), new tools have appeared (e.g., GRENCHMARK, which can accurately generate synthetic traces or replay real ones [19]), and data sets can be obtained in a similar way in which the Parallel Workloads Archive was built (voluntary submission from academic and production environments, and open access to data sets related to published material). More research is needed to assess the capabilities of the tools, specifically the evaluation accuracy, and to address issues like grid-specific evaluation methods, and metrics. However, we argue that a Grid Workloads Archive could become the crystallizing factor of a community oriented not only towards performance evaluation studies, but also to functionality testing, to system tuning, and to middleware design (in particular, scheduling).

We turn our attention to the area of grid scheduling, and answer the question *How can state-of-the-art scheduling systems benefit from this work?* A common problem when developing a new system is the inability to estimate the size of its predicted workload. Considering that the monitoring systems will incorporate our proposed enhancements, the number of events that should be processed by a scheduler could be roughly 10 events per job (e.g., submission, authentication, resubmission, transfer, *etc.*), one monitoring information poll event per minute, and one summary per hour for predictions and advanced features, for a total of 8-10 million events per year per thousand of processors. From the methodological point of view, the *testing procedure* is in many cases flawed; Feitelson *et al.* [5] draw attention on no less than 32 pitfalls concerning the testing procedure for grid schedulers. Most of the pitfalls are related to inadequate workload modeling. To offer guarantees on the grid workloads properties suggested in this work, thus to help overcome these known pitfalls, more data of the type we are presenting in this work are needed.

VI. Related work

The problems of system performance analysis and workload modeling have received a great deal of attention in the passed decade [9], [20], [21], [22], [23], [24], [15], [25]. Our work has been inspired by these approaches, and complements them with results concerning large-scale academic and production Grid environments.

A model for the jobs' inter-arrival time is given by Jann *et al.* [21]. The authors divide the jobs in categories based on size, and fit their characteristics with a hyper-Erlang distribution. Feitelson proposes instead a rigid job model that includes the distribution of job sizes in terms of number of processors, the correlation of runtime with parallelism, and repeated runs of the same job [20]. The model is refined by Lublin and Feitelson to include an arrival pattern with a daily cycle, runtimes that are correlated with the number of nodes, and a distinction between interactive and batch jobs [24]. Downey [22] and Cirne and Berman [23] model the characteristics of malleable jobs submitted to parallel

production environments.

Li *et al.* present a model for a multi-cluster workload, which details jobs characteristics (e.g., runtime, required memory) per cluster and per groups of users, and correlate many of these characteristics [15]. They show time patterns in jobs arrivals, and fit the jobs inter-arrival time to a hyper-exponential of order 2. They also model the jobs' cancelation rate. Similar findings are presented in Medernach [25], for a small cluster working in a Grid computing environment. The latter proposes a Markov model for the user submitting behavior (based on the concept of sessions), and use the model to predict the number of jobs submitted during each session, and their characteristics.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we analyze and compare the resource utilizations of four Grids, one academic and three production, from both a user and a system perspective. We focus our analysis on virtual organizations, users, and individual jobs, and on the evolution of Grids. Our main findings are: (a) a high percent of grid applications still employ an embarrassingly parallel model; (b) a small number of VOs and users dominate each workload both in terms of number of submitted jobs and of consumed resources; (c) users show a small set of behaviors, similar to the one observed on a single cluster; (d) initial results for the locality of jobs and its effects in an unrestricted multi-cluster Grid show that users have a preference for one site of the many available, but this preference is not pronounced if the obtained satisfaction is equal for the local and remote sites; (e) system evolution can manifest itself at the system, VO, and user level, and should therefore be taken into account when provisioning resources; (f) most of the performance metrics confirm the evaluations of simulation research, with the notable exception of the expected wait time.

We believe that the trend of offering high-level programming models and pre-packaged pieces of software for Grids will change the current way of using Grids. Also, we envision that a dynamic, self-organizing, and self-managing Grid system, coupled with the development of intelligent computing services, will represent another major factor in the evolution of Grids. Thus, we believe that as soon as any of these directions will become more mature, major changes will also occur in the results presented in this paper. We firmly conclude that grids are not yet utilized at their full capacity.

AVAILABILITY AND CALL FOR CONTRIBUTIONS

As a first step towards the Grid Workloads Archive, and subject to owner's consent, we provide the traces used in this work at: `http://www.grid-workloads-archive.org/`

We invite other institutions to follow this example, and to submit grid traces for the use of the entire community.

ACKNOWLEDGMENTS

REFERENCES

[1] C. Dumitrescu and I. Foster, "GRUBER: A Grid Resource Usage SLA BrokER," in *Proc. of 11th Intl. Euro-Par Conference, Portugal*, 2005.
[2] S. J. Chapin, D. Feitelson, U. Schwiegelshohn et al., "Benchmarks and standards for the evaluation of parallel job schedulers." in *JSSPP*, ser. LNCS, vol. 1659, 1999, pp. 67–90.
[3] C. Ernemann, B. Song, and R. Yahyapour, "Scaling of workload traces." in *JSSPP*, ser. LNCS, vol. 2862, 2003, pp. 166–182.
[4] D. G. Feitelson, "Metric and workload effects on computer systems evaluation." *IEEE Computer*, vol. 36, no. 9, pp. 18–25, 2003.
[5] E. Frachtenberg and D. G. Feitelson, "Pitfalls in parallel job scheduling evaluation." in *JSSPP*, ser. LNCS, vol. 3834, 2005, pp. 257–282.
[6] I. Foster and al., "The Grid2003 Production Grid: Principles and practice," in *IEEE HPDC*, 2004.
[7] The TeraGrid Project , "Npaci," March 2006. [Online]. Available: www.teragrid.org
[8] Dutch University Backbone, "The distributed ASCI supercomputer 2 (DAS-2)," 2006. [Online]. Available: http://www.cs.vu.nl/das2/
[9] C. Dumitrescu, I. Raicu, and I. Foster, "Experiences in running workloads over Grid3," in *Grid and Cooperative Computing (GCC)*, 2005.
[10] H. Shan, L. Oliker, and R. Biswas, "Job supersched. arch. and performance in computational Grid environments." in *SC*. ACM, 2003, p. 44.
[11] EGEE Team, "LCG," 2004. [Online]. Available: lcg.web.cern.ch/LCG
[12] M. Mambelli and al., "Atlas data challenge production on Grid3," 2005. [Online]. Available: http://griddev.uchicago.edu/download/atgce/doc.pkg/presentations/chep04-503-usatlas-dc2.pdf
[13] H. Mohamed and D. Epema, "The design and implementation of the KOALA co-allocating grid scheduler," in *Proceedings of the European Grid Conference, Amsterdam*, .-. LNCS 3470, Ed., 2005.
[14] G. Wrzesinska, R. van Nieuwpoort, J. Maassen, and H. E. Bal, "Fault-tolerance, malleability and migration for divide-and-conquer applications on the grid." in *Proc. of 19th IPDPS*, April 2005.
[15] H. Li, D. L. Groep, and L. Wolters, "Workload characteristics of a multi-cluster supercomputer." in *JSSPP*, ser. LNCS, vol. 3277, 2004, pp. 176–193.
[16] S. Hotovy, "Workload evolution on the cornell theory center ibm sp2." in *JSSPP*, ser. LNCS, vol. 1162, 1996, pp. 27–40.
[17] C. Ernemann, V. Hamscher, and R. Yahyapour, "Benefits of global grid computing for job scheduling." in *GRID*, 2004, pp. 374–379.
[18] R. Mach, R. Lepro-Metz, B. A. Hamilton, S. Jackson, and L. McGinnis, "Usage Record Format Recommendation," Global Grid Forum, Usage Record WG, Draft Rec-UR-Usage, March 2005. [Online]. Available: https://forge.gridforum.org/projects/ur-wg/
[19] A. Iosup and D. Epema, "GrenchMark: A framework for analyzing, testing, and comparing grids," in *Proc. of the 6th IEEE/ACM Int'l. Symp. on Cluster Computing and the GRID (CCGrid)*, May 2006, (accepted).
[20] D. G. Feitelson, "Packing schemes for gang scheduling." in *JSSPP*, ser. LNCS, vol. 1162, 1996, pp. 89–110.
[21] J. Jann, P. Pattnaik et al., "Modeling of workload in mpps." in *JSSPP*, ser. LNCS, vol. 1291, 1997, pp. 95–116.
[22] A. B. Downey, "A parallel workload model and its implications for processor allocation." in *HPDC*, 1997, pp. 112–126.
[23] W. Cirne and F. Berman, "A model for moldable supercomputer jobs." in *IPDPS*. IEEE Computer Society, 2001, pp. 59–69.
[24] U. Lublin and D. G. Feitelson, "The workload on parallel supercomputers: modeling the characteristics of rigid jobs." *J. Parallel Distrib. Comput.*, vol. 63, no. 11, pp. 1105–1122, 2003.
[25] E. Medernach, "Workload analysis of a cluster in a grid environment." in *JSSPP*, ser. LNCS, vol. 3834, 2005, pp. 36–61.