The Grid Workloads Archive

Alexandru Iosup [a,*], Hui Li [b], Mathieu Jan [a], Shanny Anoep [a], Catalin Dumitrescu [a], Lex Wolters [b], and Dick H.J. Epema [a]

[a] *Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, The Netherlands*

[b] *LIACS, University of Leiden, The Netherlands*

Abstract

While large grids are currently supporting the work of thousands of scientists, very little is known about their actual use. Because of strict organizational permissions, there are few or no traces of grid workloads available to the grid researcher and practitioner. To address this problem, in this work we present the Grid Workloads Archive (GWA), which is at the same time a workload data exchange and a meeting point for the grid community. We define the requirements for building a workloads archive, and describe the approach taken to meet these requirements with the GWA. We introduce a format for sharing grid workload information, and tools associated with this format. Using these tools, we collect and analyze data from nine well-known grid environments, with a total content of more than 2000 users submitting more than 7 million jobs over a period of over 13 operational years, and with working environments spanning over 130 sites comprising 10000 resources. We show evidence that grid workloads are very different from those encountered in other large-scale environments, and in particular from the workloads of parallel production environments: they comprise almost exclusively single-node jobs, and jobs arrive in "bags-of-tasks". Finally, we present the immediate applications of the GWA and of its content in several critical grid research and practical areas: research in grid resource management, and grid design, operation, and maintenance.

Key words: Grid computing, grid workloads, grid traces

1 Introduction

Current grids bring together (tens of) thousands of resources for the benefit of thousands of scientists, in infrastructures such as CERN's LHC Computing Grid

* Corresponding author.
Contact: A.Iosup@tudelft.nl (Alexandru Iosup), gwa@tudelft.nl (group).

(LCG) [1], NorduGrid [2], TeraGrid [3], and the Open Science Grid [4]. Very little is known about the real grid users' demand, in spite of the tools that monitor and log the state of these systems and traces of their workloads. Indeed, because of access permissions, almost no grid workload traces are available to the community that needs them. The lack of grid workload traces hampers both researchers and practitioners. Most research in grid resource management is based on unrealistic assumptions about the characteristics of the workloads. Thus, performance evaluation studies lack a comparable basis, and researchers often fail to focus on the specifics of grids, e.g., the "bag-of-tasks" job arrival behavior. Most grid testing is in practice performed with unrealistic workloads, and as a result the middleware is optimized for the wrong use case, and often fails to deliver good service in real conditions [5–8]. There is little quantitative data for establishing best practices in grid design, and for grid comparison in the resource procurement process. In this work we present the design and the current status of the Grid Workloads Archive, which is an effort to collect grid workload traces and to make them available to this community.

The goal of the Grid Workloads Archive (GWA) is to provide a virtual meeting place where practitioners and researchers can exchange grid workload traces. Extending established practice [9–12], we define the requirements for an archive of large-scale distributed system workload traces [1] (Section 2). We design the GWA around building a grid workload data repository, and establishing a community center around the archived data (Section 3). We further design a grid workload format for storing job-level information, and which allows extensions for higher-level information, e.g., "bag-of-tasks". We develop a comprehensive set of tools for collecting, processing, and using grid workloads. We give special attention to non-expert users, and devise a mechanism for automated trace ranking and selection. We have collected so far for the GWA traces from nine well-known grid environments, with a total content of more than 2000 users submitting more than 7 million jobs over a period of over 13 operational years, and with working environments spanning over 130 sites comprising 10000 resources. Thus, we believe that the GWA already offers a better basis for performance evaluation studies.

With the GWA we target as a first step people involved in grid computing research, industry, and education. We have already used the contents of the Archive for a variety of applications, from research in grid resource management to grid maintenance and operation (Section 4). However, we believe that the data, tools, and even the approach taken for building the GWA will be of immediate use to the broader community around resource management in large-scale distributed computing systems.

The Grid Workloads Archive effort was initially motivated by the success of the

[1] Throughout this work, we use the terms "grid workload trace", "grid workload", and "grid trace" interchangeably.

Parallel Workloads Archive (PWA [12]), the current de-facto standard source of workload traces from parallel environments. We also draw inspiration from a number of archival approaches from other computer science disciplines, e.g., the Internet [9–11] and clusters-based systems [13]. In comparison with the other efforts, the GWA is the major source of grid-related data, and offers more tools to the community of workload data users (Section 5).

2 Requirements for a Grid Workload Archive

In this section we synthesize the requirements to build a grid workloads archive. Our motivation is twofold. First, grid workloads have specific archival requirements. Second, in spite of last decade's evolution of workload archives for scientific purposes (see Section 5), there is still place for improvement, especially with the recent evolution of collaborative environments such as Wikis.

We structure the requirements in two broad categories: requirements for building a grid workload data repository, and requirements for building a community center for scientists interested in the archived data. **Requirement 1: tools for collecting grid workloads**. In many environments, obtaining workload data requires special acquisition techniques, i.e., reading hardware counters for computer traces, or capturing packets for Internet and other network traces. Obtaining grid workloads data is comparatively easy: most grid middleware log all job-related events. However, it is usually difficult to correlate information from several logs. This problem is starting to be solved by the use of unique job identifiers. Second, to keep the size of the logs small, fixed-size logs are used, and old data are archived or even removed. Third, due to political difficulties, parts of a data set may be obtained from several grid participants. Fourth, to provide uniformity, a workload archive provides a common format for data storage. The format must comprehensively cover current workload features, and also be extensible to accommodate future requirements. To conclude, there is a need for tools that can collect and combine data from multiple sources, and store it in a common grid workload format (*requirement 1*).

Requirement 2: tools for grid workload processing. Following the trend of Internet traces, sensitive information must not be disclosed. For grids, environment restrictions to data access are in place, so it is unlikely that truly sensitive data (e.g., application input) can be obtained or published. However, there still exists the need to anonymize any information that can lead to easily and uniquely identifying a machine, an application, or a user (*requirement 2a*).Time series analysis is the main technique to analyze workload data in computing environments. While many generic data analysis tools exist, they require specific configuration and policy selection, and input data selection and formatting. In addition, the data in the archive is often subjected to the same analysis: marginal distribution estimation and analysis, second and higher order moment analysis, statistical fitting, and time-

based load estimation. In addition, grids exhibit patterns of batch submission, and require that workload analysis is combined with monitoring information analysis. To assist in these operations, there is a need for grid-specific workload analysis tools (*requirement 2b*). The data donors and the non-expert users expect a user-friendly presentation of the workload analysis data. The GWA community needs tools that facilitate the addition of new grid workloads, including a web summary. Thus, there is a need for tools to create workload analysis reports (*requirement 2c*).

Requirement 3: tools for using grid workloads. The results of workload modeling research are often too complex for easy adoption. Even finding the parameter values for another data set may prove too much for the common user. By comparison to previous computing environments (e.g., clusters), grid models need to include additional (i.e., per-cluster, per-group) and more complex (e.g., batching) information. There is a need for tools to extract for a given data set the values of the parameters of common models (*requirement 3a*). The common user may also find difficult to generate traces based on a workload model. There is a need to generate synthetic workloads based on models representative for the data in the archive (*requirement 3b*). Since the grid workload format can become complex, there exists also a need for developer support (i.e., libraries for parsing and loading the data) (*requirement 3c*).

Requirement 4: tools for sharing grid workloads. Over time, the archive may grow to include tens to hundreds of traces. Even when few traces are present, the non-expert user faces the daunting task of trace selection for a specific purpose. There is a need for ranking and searching mechanisms of archived data (*requirement 4a*). There is a need to comment on the structure and contents of the archive, and to discuss on various topics, in short, to create a medium for workload data exchange (*requirement 4b*). One of the main reasons for establishing the grid workloads archive is the lack of data access permission for a large majority of the community members. We set as a requirement the public and free access to data (*requirement 4c*).

Requirement 5: community-building tools. There are several other community-building support requirements. There is a need for creating a bibliography on research on grid (and related) workloads (*requirement 5a*), a bibliography on research and practice using the data in the archive (*requirement 5b*), a list of tools that can use the data stored in the archive (*requirement 5c*), and a list of projects and people that use grid workloads (*requirement 5d*).

3 The Grid Workloads Archive

In this section we present the Grid Workloads Archive. We discuss its design, detail three distinguishing features, and summarize its current contents.

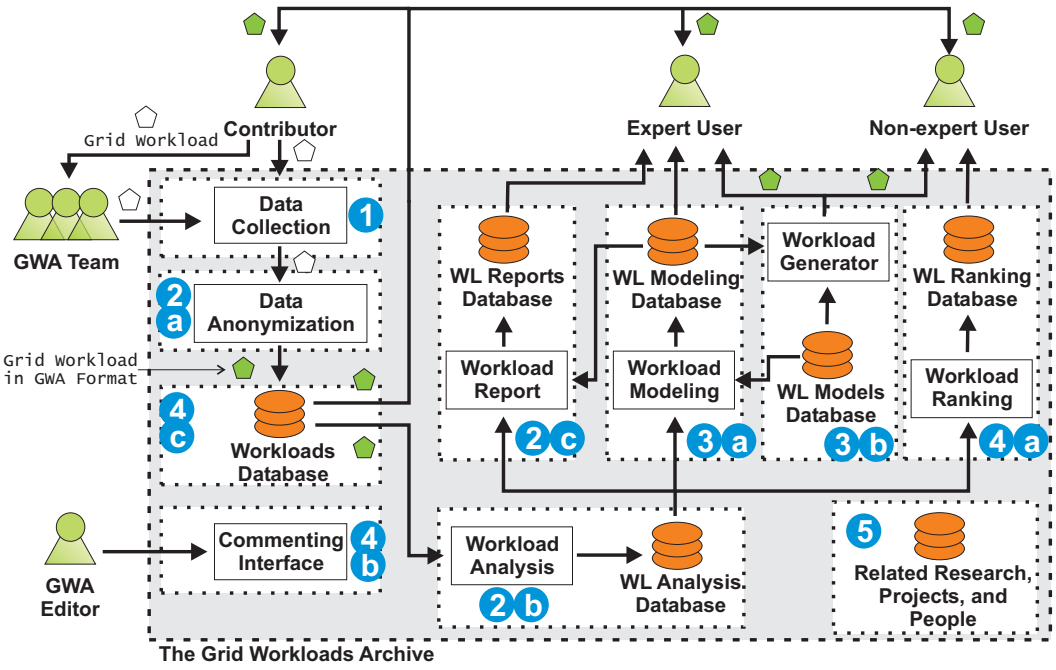

Fig. 1. An overview of the Grid Workloads Archive design. The design requirements (see Section 2) are marked on the figure.

3.1 The Design

We envision five main roles for the GWA community member. The *contributor* is the legal owner of one or more grid workloads, and offers them to the GWA community. The *GWA team* helps the contributors to add data to the GWA archive. The *non-expert user* is the typical user of the archived data. This user type requires as much help as possible in selecting an appropriate trace. The *expert user* uses the archived data in an expert way. Mainly, this user type requires detailed analysis reports and not automatic ranking, consults the related work, and may develop new analysis and modelling tools that extend the GWA data loading and analysis libraries. The *GWA editor* contributes to the community by commenting on the contents of the archive, and by adding related work. One of the major design goals of the GWA is to facilitate the interaction between these five types of members.

Figure 1 shows the design of the Grid Workloads Archive, with the requirements expressed in Section 2 marked on the figure. The arrows represent the direction of data flows.

There is one module for collecting grid workload data. The Data Collection module receives grid workloads from the contributor (or from the GWA Team, if the contributor delegates the task). There are several potential sources of data: grid resource managers (e.g., the logs of the Globus GRAM Gatekeeper), local resource managers (e.g., the job logs of SGE), web repositories (e.g., the GridPP Grid Operations Center), etc. The data are usually not in the GWA format, but in the format

specific to the grid from which they were obtained. The main tasks of the Data Collection module are to ensure that the received data can be parsed, to eliminate wrongly formatted parts of the trace, and to format data provenance information.

There are three modules for processing the acquired data. The Data Anonymization module anonymizes the content received from the Data Collection module, and outputs in the Grid Workloads Archive format (see Section 3.2). If the Contributor allows it, a one-to-one map between the anonymized and the original information is also saved. This will allow future data to be added to the same trace, without losing identity correlations between added parts. The Workload Analysis module takes in the data from the Workloads Database, and outputs analysis data to the Workload Analysis Database. More details about this component are given in Section 3.5. The Workload Report module formats for the expert user the results of the workload analysis and sometimes of workload modeling.

There are three modules supporting the use of the archived data. The Workload Modeling module attempts to model the archived data, and outputs the results (i.e., parameter values) to the Workload Modeling Database. The input for this process is taken from the Workload Analysis Database (input data), and from the Workload Models Database (input models). Several workload models are supported, including the Lublin-Feitelson model [14]. The Workload Generator module generates synthetic grid workloads based on the results of the Workload Analysis and Modeling results, or on direct user input. The third module (not shown in Figure 1) is a library for parsing the stored data.

The GWA contains three modules for data sharing. The Workload Ranking module classifies and ranks the stored traces, for the benefit of the non-expert user. This process is further detailed in Section 3.3. The GWA editor uses the Commenting Interface to comment on various aspects presented in the Grid Workload Archive's web site. The Workloads Database stores data in Grid Workloads Archive format. To enable quick processing, the data is stored as raw text and as a relational database. This module has a web interface to allow the public and free distribution of the data within.

The Grid Workload Archive contains various additional community-building support, e.g., bibliographies on previous and derivative related research, and links to related tools, projects, and people.

3.2 The Format for Sharing Grid Workload Information

One of the main design choices for the Grid Workloads Archive was to establish a common format for storing workload data. There are two design aspects to take into account. First, there are many aspects that may be recorded, e.g., job characteristics, job grouping and inter-job dependencies, co-allocation [15], advance reserva-

Table 1
A model for automated trace ranking. The quality level is given by the number of stars ($\star$ sign).

Category	Sample	$\star$	$\star\star$	$\star\star\star$	$\star\star\star\star$	$\star\star\star\star\star$
System Sites	-	1	2-5	6-10	11-20	>20
System Cores	0-100	101-1k	1k-5k	5k-10k	10k-25k	>25k
No. Users	0-50	51-100	101-200	201-500	0.5k-1k	>1k
No. Jobs	0-15k	15k-100k	100k-200k	200k-500k	500k-1M	>1M
Utilization	0-10%	11-20%	21-40%	41-60%	61-75%	>75%
Reported work	0	1	2-5	6-10	11-20	>20

tions [16], etc. Second, grid workload data owners are reluctant to provide data for a format they have not yet approved. Thus, one must provide the simplest possible format for the common user, while designing the format to be extensible. We have designed a *standard Grid Workload data Format* (GWF) [17], which records detailed information about submitted jobs. To further ease the adoption of our format, and as a step towards compatibility with related archives, we base it on the PWA workload format (SWF, the de-facto standard format for the parallel production environments community) [12]. We add to this format several grid-specific aspects (e.g., job submission site, etc.) and extension capabilities. We specifically design the language for the following extensions, which we have identified in our previous work ([18]) as the most relevant for grid workload modeling: batches and workflows, co-allocation, malleability and flexibility, checkpointing, migration, reservations, failures, and economic aspects (e.g., user-specified utility). From a practical perspective, the format is implemented both as an SQL-compatible database (GWF-SQLite), which is useful for most common tasks, and as a text-based version, easy to parse for custom tasks.

Since grids are dynamic systems, using the workload data in lack of additional information (e.g., resource availability) may lead to results that cannot be explained. To address this issue, we have already designed and used a minimal format for resource availability and state [19].

3.3 The Trace Ranking and Selection Mechanism

The non-expert user faces a big challenge when faced with a large database of traces: *which trace to select?* We design a trace mechanisms that ranks traces and then selects the most suitable of them, based on the requirements of the experimental scenario.

We devise a classifier that evaluates a workload according to six categories: number of sites, number of virtual processors, number of users, number of jobs, average utilization, and number of reported publications using the traces. Note that the categories describe user-specific aspects, system-specific aspects, user-system interac-

tion (utilization), and community relevance (reported work). For a given workload, the classifier assigns a number of stars for each category, from 0 to 5, higher values are better. The classifier is completely described by Table 1, which shows the mapping between value ranges and the number of stars for GWA's six categories. We denote by W_s a hypothetical workload that has sample-like characteristics, i.e., the values for its six categories are nearly 0.

We define the *workload signature* as the set of six values for workload's characteristics as output by the classifier. We denote by C_0 the set of all six categories. Consider an experimental scenario in which only some of the categories from C_0 are relevant for workload selection, e.g., the number of system cores and the average utilization for a resource provisioning scenario. Let C be the subset of C_0 that includes only the categories relevant for the scenario at hand; we call such C a scenario-dependent subset of C_0. We define the *partial workload signature for the characteristics in* C as the workload signature from which the characteristics not in C (with $C \subseteq C_0$) have been eliminated. Then, we define the *distance between two workloads* W_1 *and* W_2 as

$$D(C, W_1, W_2) = \frac{\sum_{k=1}^{|C|} (c_1^k - c_2^k)^2}{25 \times |C|}, \quad (1)$$

where C is the scenario-dependent set of characteristics, and $W_i = (c_i^1, c_i^2, ...)$ is the partial workload signature of workload i for the characteristics in C. The denominator in Equation 1 normalizes the values; the constant included in the divisor, 25, takes into account that the values are between 0 and 5 stars. In particular, we call $D(C_0, \cdot, \cdot)$ the *scenario-independent distance*, and $D(C_0, W_i, W_s)$ the *scenario-independent value* (short, *value*) *of workload* W_i.

The ranking and selection mechanisms use the distance between workloads. We present online the ranking table that uses the scenario-independent value of the traces present in the GWA. The GWA users can select traces using directly this table, or can obtain a different table by specifying a new C.

3.4 The Contents of the Workloads Database

Table 2 shows the nine workload traces currently included in the GWA. Note that several are under processing, or have pending publication rights. The data sources for these traces range from local resource managers (e.g., PBS) to grid resource managers (e.g., Globus GRAM) to user- and VO-level resource managers (e.g., Condor Schedd). In several cases, incomplete data is provided, e.g., for NorduGrid the trace does not include locally submitted (non-grid) jobs. The traces include grid applications from the following areas: physics, robotics, rendering and image pro-

Table 2
The GWA content (status as of August 2007). The ⋆ sign marks restrictions due to data scarcity (see text). The ◇ sign marks traces under processing. The ‡ sign marks traces with pending publication rights.

ID	System	Period	Number of observed Sites	CPUs	Jobs	Groups	Users
GWA-T-1	DAS-2	02/05-03/06	5	400	602K	12	332
GWA-T-2	Grid'5000	05/04-11/06	15	∼2500	951K	10	473
GWA-T-3◇	NorduGrid	05/04-02/06	∼75	∼2000	781K	106	387
GWA-T-4◇	AuverGrid	01/06-01/07	5	475	404K	9	405
GWA-T-5◇	NGS	02/03-02/07	4	∼400	632K	1	379
GWA-T-6◇	LCG	05/05-01/06	1⋆	880	1.1M	25	206
GWA-T-7‡	GLOW	09/06-01/07	1⋆	∼1400	216K	1⋆	18
GWA-T-8‡	Grid3	06/04-01/06	29	2208	1.3M	1⋆	19
GWA-T-9‡	TeraGrid	08/05-03/06	1⋆	96	1.1M	26	121
	Total	13.51 yrs	136	>10000	>7M	191	2340
	Average	1.35 yrs	15	1151	787K	21	260

cessing (graphics), collaborative and virtual environments (v-environments), computer architectures simulations (CAS), artificial intelligence (AI), applied mathematics (math), chemistry, climate and weather forecasting (climate), medical and bioinformatics (biomed), astronomy, language, life sciences (life), financial instruments (finance), high-energy physics (HEP), aerospace design (aero), etc.; three of the nine GWA traces include only HEP applications. Note that due to trace anonymization it is not possible to map the jobs included in the GWA traces to specific applications or application areas.

The GWA-T-1 trace is extracted from DAS-2 [20], 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 [21] or IBIS [22] jobs. Jobs can be submitted directly to the local resource managers (i.e., by *system users*), or to Grid gateways that interface with the local resource managers. To achieve low wait time for interactive jobs, the DAS system is intentionally left as free as possible by its users. The traces collected from the DAS include applications from the areas of physics, robotics, graphics, v-environments, CAS, AI, math, chemistry, climate, etc. In addition, the DAS traces include experimental applications for parallel and distributed systems research.

The GWA-T-2 trace is extracted from Grid'5000 [23], an experimental grid platform consisting of 9 sites geographically distributed in France. Each site comprises one or several clusters, for a total of 15 clusters inside Grid'5000. The main objective of this reconfigurable, controlable, and monitorable experimental platform is to allow experiments in all the software layers between the network protocols up to the applications. We have obtained traces recorded by all batch schedulers handling Grid'5000 clusters (OAR [24]), from the beginning of the Grid'5000 project up to

November 2006. Note that most clusters of Grid'5000 were made available during the first half of 2005. The traces collected from Grid5000 include applications from the areas of physics, biomed, math, chemistry, climate, astronomy, language, life, finance, etc. In addition, the Grid5000 traces include experimental applications for parallel and distributed systems research.

The GWA-T-3 trace is extracted from NorduGrid [2], a large scale production grid. In NorduGrid, non-dedicated resources are connected using the Advanced Resource Connector (ARC) as Grid middleware [25]. Over 75 different clusters have been added over time to the infrastructure. We have obtained the ARC logs of NorduGrid for a period spanning from 2003 to 2006. In these logs, the information concerning the grid jobs is logged locally, then transferred to a central database voluntarily. The logging service can be considered fully operational only since mid-2004. The traces collected from NorduGrid include applications from the areas of CAS, chemistry, graphics, biomed, and HEP.

The GWA-T-4 trace is extracted from AuverGrid, a multi-site grid that is part of the EGEE project. This grid employs the LCG middleware as the grid's infrastructure (same as the GWA-T-6 trace). We have obtained traces recorded by the resource managers of the five clusters present in AuverGrid. The traces collected from AuverGrid include applications from biomed and HEP.

The GWA-T-5 trace is extracted from the UK's National Grid Service (NGS), a grid infrastructure for UK e-Science. We have obtained traces from the four dedicated computing clusters present in NGS. The traces collected from the NGS include applications from biomed, physics, astronomy, aero, and HEP.

The GWA-T-6 trace is extracted from the LHC Computing Grid (LCG) [1]. LCG is a data storage and computing infrastructure for the high energy physics community that will use the Large Hadron Collider (LHC) at CERN. The LCG production Grid currently has approximately 180 active sites with around 30,000 CPUs and 3 petabytes storage, which is primarily used for high energy physics (HEP) data processing. There are also jobs from biomedical sciences running on this Grid. Almost all the jobs are independent computationally-intensive tasks, requiring one CPU to process a certain amount of data. The workloads are obtained via the LCG Real Time Monitor [2] (RTM). The RTM monitors jobs from all major Resource Brokers on the LCG Grid therefore the data it collects are representative at the Grid level. In particular, the GWA-T-6 is a long-term trace coming from one of the largest LCG sites, comprising 880 CPUs. The traces collected from the LCG include only HEP applications.

The GWA-T-7 trace is extracted from the Grid Laboratory of Wisconsin (GLOW), a campus-wide distributed computing environment that serves the computing needs

[2] The Real Time Monitor is developed by Imperial College London http://gridportal.hep.ph.ic.ac.uk/rtm/.

of the University of Wisconsin-Madison's scientists. This Condor-based pool consists of over 1400 machines shared temporarily by their rightful owners [26]. We have obtained a trace comprising all the jobs submitted by one Virtual Organization (VO) in the Condor-based GLOW pool, in Madison, Wisconsin. The trace spans four months, from September 2006 to January 2007. The traces collected from GLOW include only HEP applications.

The GWA-T-8 trace is extracted from the 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 [27]. We have obtained traces recorded by the Grid-level scheduler corresponding to one of the largest VOs: the Grid3/USATLAS; there are three major VOs in the system, the others being iVDgL and USCMS. These traces capture the execution of workloads of physics working groups: a single job can run for up to a few days, and the workloads can be characterized as directed acyclic graphs (DAGs) [28]. The traces collected from Grid3 include only HEP applications.

The GWA-T-9 trace is extracted from the TeraGrid system, a system for e-Science, with more than 13.6 TeraFLOPS of computing power, and facilities capable of managing and storing more than 450 TeraBytes of data [3]. We have obtained traces recorded by the interface between the Grid level scheduler and the local resource manager of one of the TeraGrid sites: the UC/ANL. In the analyzed traces, workloads are composed of applications targeting high-resolution rendering and remote visualization; ParaView, a multi-platform application for visualizing large data sets [3], is the commonly used application.

3.5 The Toolbox for Workload Analysis, Reporting, and Modeling

The GWA provides a comprehensive toolbox for automatic trace analysis, reporting, and modeling. The toolbox provides the contributors and the expert users with information about the stored workloads, and can be used as a source for building additional workload-related tools.

The workload analysis focuses on three aspects: system-wide characteristics (e.g., system utilization, job arrival rate, job characteristics; comparison of sequential and parallel jobs' characteristics), user and group characteristics (i.e., similar to system-wide characteristics, but for all and top users), and performance analysis (e.g., resource consumption, waiting and running jobs, and throughput). The analysis enables a quick comparison of traces for the expert user, or a detailed view of one grid, for the contributor (that is, the grid administrator); we detail in Section 4.1 several such uses. Figures 2, 3, and 4 show a sample of the workload analysis results.

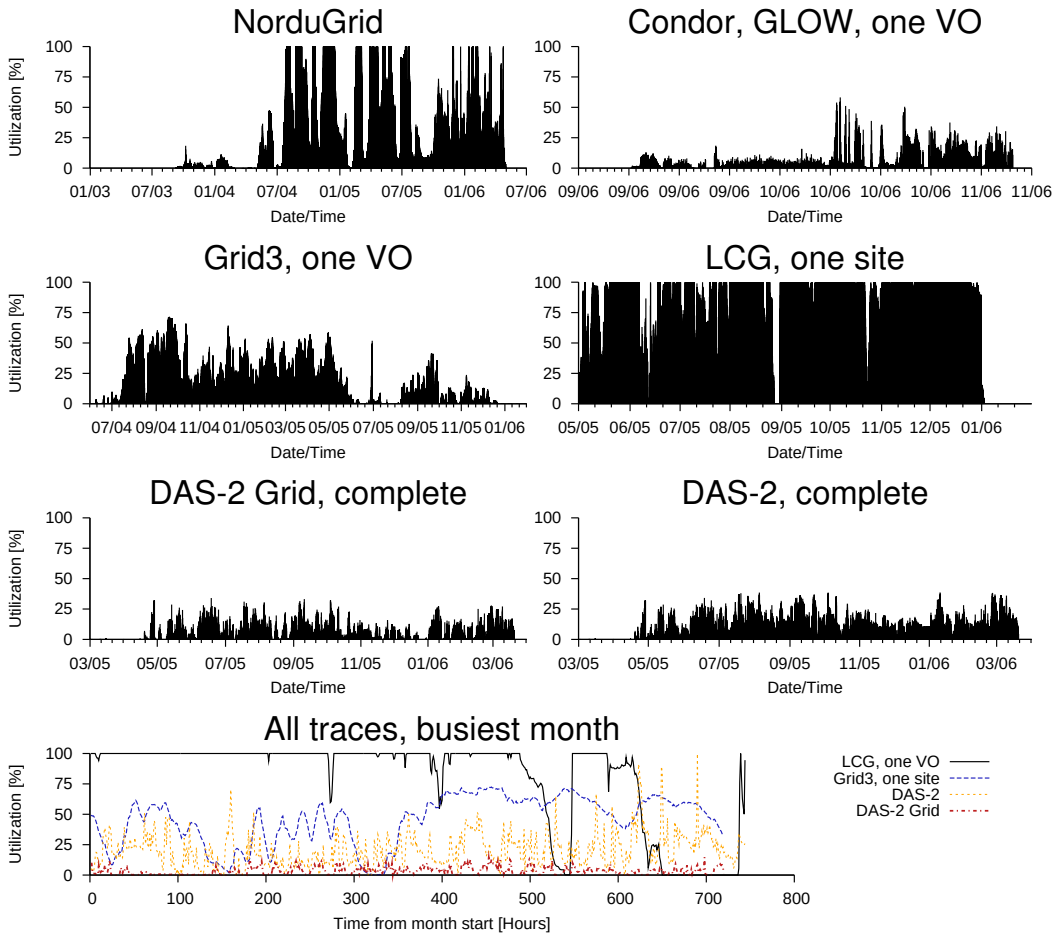

Fig. 2. System utilization over time for NorduGrid, Condor GLOW, Grid3, LCG, DAS-2, and DAS-2 Grid. The busiest month may be different for each system.

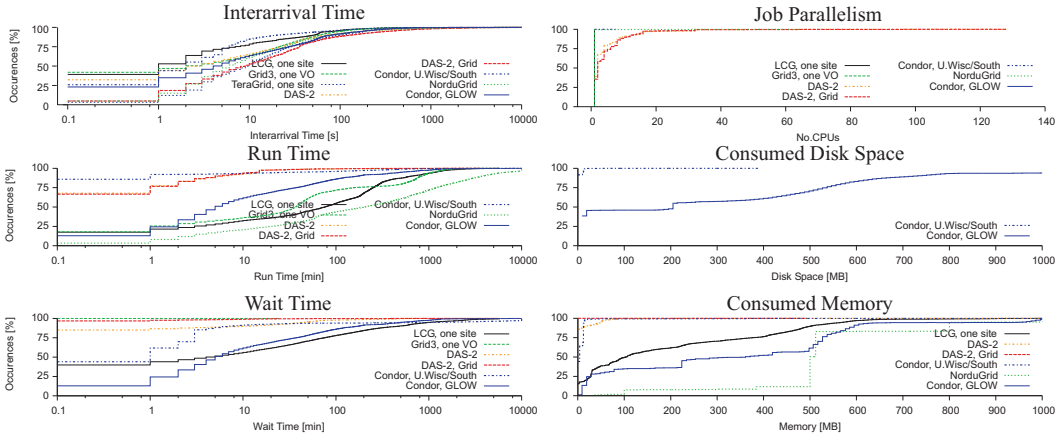

Fig. 3. CDFs of the most important job characteristics for NorduGrid, Condor GLOW, Condor UWisc-South, TeraGrid, Grid3, LCG, DAS-2, and DAS-2 Grid. Note the log scale for time-related characteristics.

Figure 2 shows that grid utilization ranges from very low (below 20% for DAS) to very high (above 85% for one cluster in LCG (trace `GWA-T-6`)).

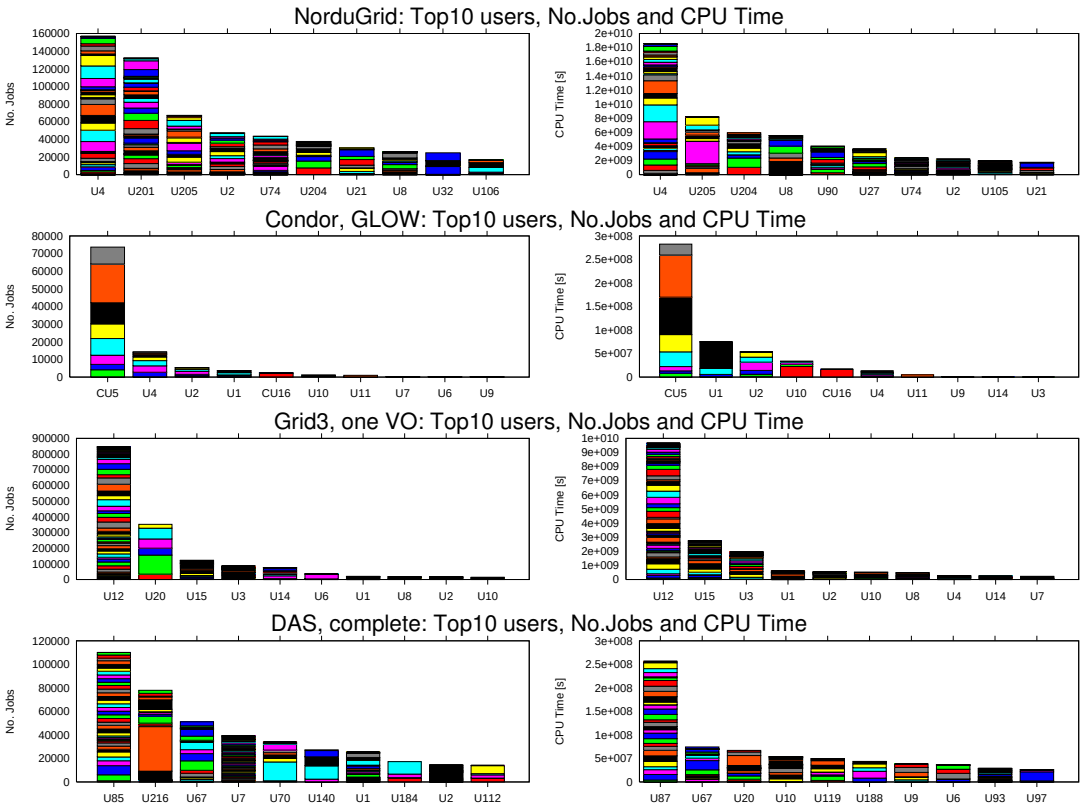

Fig. 4. The number of submitted jobs (left) and the consumed CPU time (right) by user per system: NorduGrid (top row), Condor GLOW (second row), Grid3 (third row), and DAS-2 (bottom). Only the top 10 users are displayed. The horizontal axis depicts the user's rank. The vertical axis shows the cumulated values, and the breakdown per week. For each system, users have the same identifiers (labels) in the left and right sub-graphs.

Figure 3 depicts the cumulative distribution function (CDF) for the most important job characteristics for the GWA workloads: the inter-arrival time between consecutive jobs, the wait time, the runtime, the memory consumption, the consumed CPU time, and the job parallelism (number of CPUs per job). For all traces, in 90% of the cases a new job arrives at most 1 minute after the previous job. The runtime and wait time distributions confirm that the grids where the GWA traces were collected serve widely different categories of users, with application use from interactive (DAS) to computing-intensive/batch (NorduGrid). Similarly, the resource consumption characteristics are very different across traces. Finally, for six of the GWA traces, over 90% of the jobs are single-processor; for four of them the percentage is 100%. We believe that this corresponds to the real use of many grids for the following reasons. First, many grids offer facilities for sending bags of similar tasks (e.g., parameter sweeps) with a single command; the user's task of running large numbers of jobs is thus greatly simplified. Second, there are few parallel applications in the GWA traces relative to single-processor jobs. Even though three GWA traces include more than 10% parallel jobs (`GWA-T-1`, `GWA-T-2`, and `GWA-T-9`), two of them (`GWA-T-1` and `GWA-T-2`) correspond to grids that run experimental applications for parallel and distributed systems research. For example, in DAS, three of the top five and six of the top ten users ranked by the number of submitted jobs are parallel and distributed systems researchers, which explains the high percent-

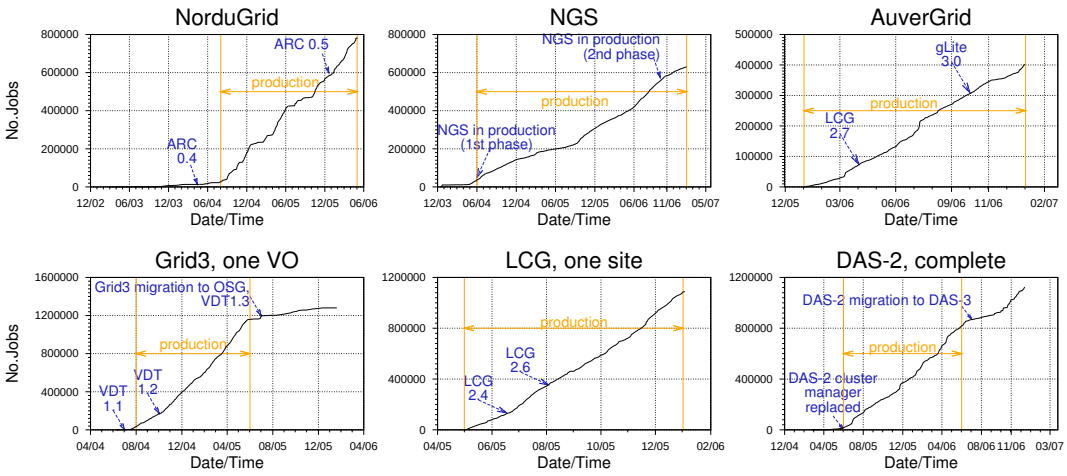

Fig. 5. The evolution of the cumulative number of jobs submitted over time for six grids. Special events such as middleware change are marked with dotted lines. The production period is also emphasized. ARC, LCG, gLite, and VDT are the key grid middleware packages used by the depicted grids.

age of parallel jobs in the DAS traces. Third, for the periods covered by the traces, there is a lack of deployed mechanisms for parallel jobs, e.g., co-allocation and advance reservation. Co-allocation mechanisms were available only in the DAS, and, later, in Grid5000. The first co-allocation mechanisms that do not require advance reservation have been implemented in the DAS [21], which explains why 10% of the jobs present in the DAS traces are co-allocated jobs. Even with the introduction of co-allocation based on advance reservation in several of the other grids (i.e., Grid500), there is no evidence showing that co-allocation has become mainstream (the percentage of co-allocated jobs in the Grid5000 traces is below 1%). However, parallel jobs are still important in grids, e.g., for `GWA-T-5`, the parallel jobs account for 5% of the number of jobs, but 85% of the consumed CPU time.

Figure 4 shows that a small number of users (below 10) dominate the workloads in both number of submitted jobs and consumed CPU time, for all the analyzed traces. In DAS, the top user by the number of submitted jobs is an automated verification tool: jobs are constantly generated every two hours; Figure 4 depicts this situation with equally sized stripes.

The results depicted in Figures 2, 3, and 4 represent the outcome of an analysis on the complete traces. However, some of the systems were not in production from the beginning to the end of the period for which the traces were collected. Moreover, the middleware used by a grid may have been changed (e.g., upgraded or replaced) during the production period. To support the validity of our analysis results, we show below that the non-production periods captured in our traces include few jobs, and that the middleware changes do not significantly affect the properties of the submitted jobs. Figure 5 shows the evolution of the cumulative number of submitted jobs over time. The period during which the grids are in production and

the main events affecting them (e.g., middleware change, system evolution, system closure) are specially marked. We observe four main trends related to the rate of growth for the cumulative number of submitted jobs (the *input*). First, the production period (marked on the image with ”production” for each depicted trace) has a homogeneous aspect, with the input much higher than for the non-production periods. Second, for the periods covered by our traces, the change of the middleware version does not have a significant impact on the input. Third, the period before entering production exhibits a low input (i.e., few job submissions) relative to the production period. Fourth, a system at the end of its production cycle loses its users in favor of the system that replaces it (a ”migration” event occurs), starting about a month before the migration event; this situation is captured in the `GWA-T-1` (DAS-2) and the `GWA-T-8` (Grid3) traces. The first two trends, and the observation that most of the middleware changes are minor version increments (with the notable exception of AuverGrid, which switched from 2.7 to 3.0–gLite is the successor of LCG), indicate that there is no significant change in the characteristics of the jobs that is due to the system change. The last two trends show that the characteristics of the jobs present in our traces are mostly influenced by the jobs submitted during the production period.

4 Using the Grid Workloads Archive

The GWA can also be beneficial in many theoretical and practical endeavors. In this section, we discuss the use of the GWA in three broad scenarios: research in grid resource management, for grid maintenance and operation, and for grid design, procurement, and performance evaluation.

4.1 Research in grid resource management

There are many ways in which the GWA can be used for research in grid resource management. We have already used the archived content to understand how real grids operate today, to build realistic grid workload models, and as real input for a variety of resource management theory (e.g., queueing theory).

The study in [29] shows how several real grids operate today. The authors analyze four grid traces from the GWA, with a focus on virtual organizations, on users, and on individual jobs characteristics. They further quantify the evolution and the performance of the Grid systems from which the traces originate. Their main finding is that the four real grid workloads differ significantly from those used in grid simulation research, and in particular that they comprise mostly single processor jobs. In another work, we have investigated the existence of batches of jobs in grids, and found that in several real grids batches are responsible for 85%–95% of the

jobs, and for 30%–96% of the total consumed CPU [30]. The imbalance of job arrivals in multi-cluster grids has been assessed using traces from the GWA in another study [31].

Hui Li et al. conduct statistical analysis of cluster and grid level workload data from LCG, with emphasis on the correlation structures and the scaling behavior [32,33,37]. This leads to the identification and modeling of several important workload patterns, including pseudo-periodicity, long range dependence, and "bag-of-tasks" behavior with strong temporal locality. The performance impact of the correlations between the workload characteristics is shown in simulations to influence significantly the system performance, both at the local and at the grid level [34]. This gives evidence that realistic workload modeling is necessary to enable dependable grid scheduling studies.

The contents of the GWA has been used to characterize grids as queues, by assessing the jobs' wait and run time marginal distributions, by estimating the number of jobs arriving and exiting the system over time, and by computing the resource utilization rate [29]. Similarly, the traces have been used to characterize grids as service-oriented architectures, by assessing the jobs' goodput and throughput [29]. Finally, the traces have been used to show that grids can be treated as dynamic systems with quantifyable [35] or predictable behavior [36,37]. These studies show evidence that grids are capable to become a predictable, high-throughput computation utility.

The contents of the GWA has also been used to evaluate the performance of various scheduling policies, both in real [5] and simulated [35,31,**?**] environments. Finally, the tools in the GWA have been used to provide an analysis back-end to a grid simulation environment [31].

4.2 Grid maintenance and operation

The content of GWA can be used for grid maintenance and operation in many ways, from comparing real systems with established practice (represented in the archive), to testing real systems with realistic workloads. We detail below two such cases.

A system administrator can compare the performance of a working grid system with that of similar systems by comparing performance data extracted from their traces. Additionally, the performance comparison over time (e.g., for each week represented in the trace) may help understanding when the operated system has started to behave outside the target performance level. Since grids represent new technology for most of their users, a lower performance in the beginning may represent just a learning period; to distinguish between this situation and system misconfiguration, the beginning of the traces of other starting systems (e.g., `GWA-T-1`) can be compared with the system under inquiry.

In large grids, realistic functionality checks must occur daily or even hourly, to prevent that jobs are assigned to failing resources. Our results using data from the GWA show that the performance of a grid system can rise when availability is taken into consideration, and that human administration of availability change information may result in 10-15 times more job failures than for an automated solution, even for a lowly utilized system [19]. To perform realistic functionality testing, tools like the GRENCHMARK can ”replay” selected parts of the traces from the the GWA in the real environments [5]. Similarly, functionality and stress testing are required for long-term maintenance. Again, tools like GRENCHMARK make use of the data stored in the GWA to run realistic tests.

Grid monitoring systems like Caltech/CERN’s MonALISA [39] and GridLab’s Mercury [40] are now common tools that assist in the grid maintenance and operation. Based on a trace and tools from the GWA, we have assessed the trade-off between the quality of information and the monitoring overhead; our simulation results show that a reduction of 90% in monitoring overhead can be achieved with a loss in accuracy of at most 10% [41]. Similarly, a system manager may choose to setup the monitoring system based on a similar analysis, using the same tools and his system’s data.

4.3 Grid design, procurement, and performance evaluation

The GWA has already been used in many grid design, procurement, and performance evaluation scenarios.

The grid designer needs to select from a multitude of middleware packages, e.g., resource managers. Oftentimes, the designer uses ”what-if” scenarios to answer questions such as *What if the current users would submit 10 times more jobs in the same amount of time? Or 50 times, or 100 times...*, or *If the users of another environment could submit their workload to our environment, what would be the success rate of the jobs submitted by these combined communities?* Using workloads from the GWA, and a workload submission tool such as GRENCHMARK, the designer can answer these questions for a variety of potential user workloads. We have shown a similar use of the GWA’s content for the DAS environment in our previous work [5].

During the procurement phase, a prospective grid user may select between several infrastructure alternatives: to rent compute time on an on-demand platform, to rent or to build a parallel production environment (e.g., a large cluster), or to join a grid as a resource user. The reports published by the GWA show evidence that grids already offer similar or better throughputs and can handle much higher surges in the job arrival rate, when compared with large-scale parallel production environments (see Table 3 and Figure 6 for a summary of these reports).

Table 3
Grid vs. Parallel Production Environments: processing time consumed by users, and highest number of jobs running in the system during a day. The ”Type” column shows the environment type: PProd for parallel production, or Grid for grid computing.

Environment		Data Source	System	Goodput	Spike
Name	Type	/Analysis	Processors	[CPUYr/Yr]	[Jobs/Day]
NASA iPSC	PProd	[12,42]	128	92.03	876
LANL CM5	PProd	[12,43]	1,024	808.40	5358
SDSC Par95	PProd	[12,44]	400	292.06	3407
SDSC Par96	PProd	[12,44]	400	208.96	826
CTC SP2	PProd	[12,45]	430	294.98	1648
LLNL T3D	PProd	[12]	256	202.95	445
KTH SP2	PProd	[12,46]	100	71.68	302
SDSC SP2	PProd	[12,47]	128	109.15	2181
LANL O2K	PProd	[12]	2,048	1,212.33	2458
OSC Cluster	PProd	[12]	57	93.53	2554
SDSC BLUE	PProd	[12]	1,152	876.77	1310
LCG, 1 Cluster	Grid	[29]	880	750.50	22550
Grid3, 1 VO	Grid	[29]	2208	360.75	15853
DAS-2	Grid	[29]	400	30.34	19550
NorduGrid	Grid	this work	∼3,100	770.20	7953
TeraGrid, 1 Site	Grid	[29]	∼200	n/a	7561
Condor, GLOW, 1 VO	Grid	this work	∼1,400	104.73	6590

Similarly to system design and procurement, performance evaluation can use content from the GWA in a variety of scenarios, e.g., to assess the ability of a system to execute a particular type of workload [5,8], to find the throughput of the system for the common usage patterns, or to measure the power consumption and failure rate under different workload patterns. Note that the same approach may be used during procurement to compare systems using trace-based grid benchmarking.

4.4 Education

The reports, the tools, and the data included in the GWA can greatly help the educators. We target courses that teach the use of grids, large-scale distributed computer systems simulation, and computer data analysis. The reports included in the GWA may be used to better illustrate concepts related to grid resource management, such as resource utilization, job wait time and slowdown, etc. The tools may be used to build new analysis and simulation tools. The data included in the archive may be used as input for demonstrative tools, or as material for student assignments.

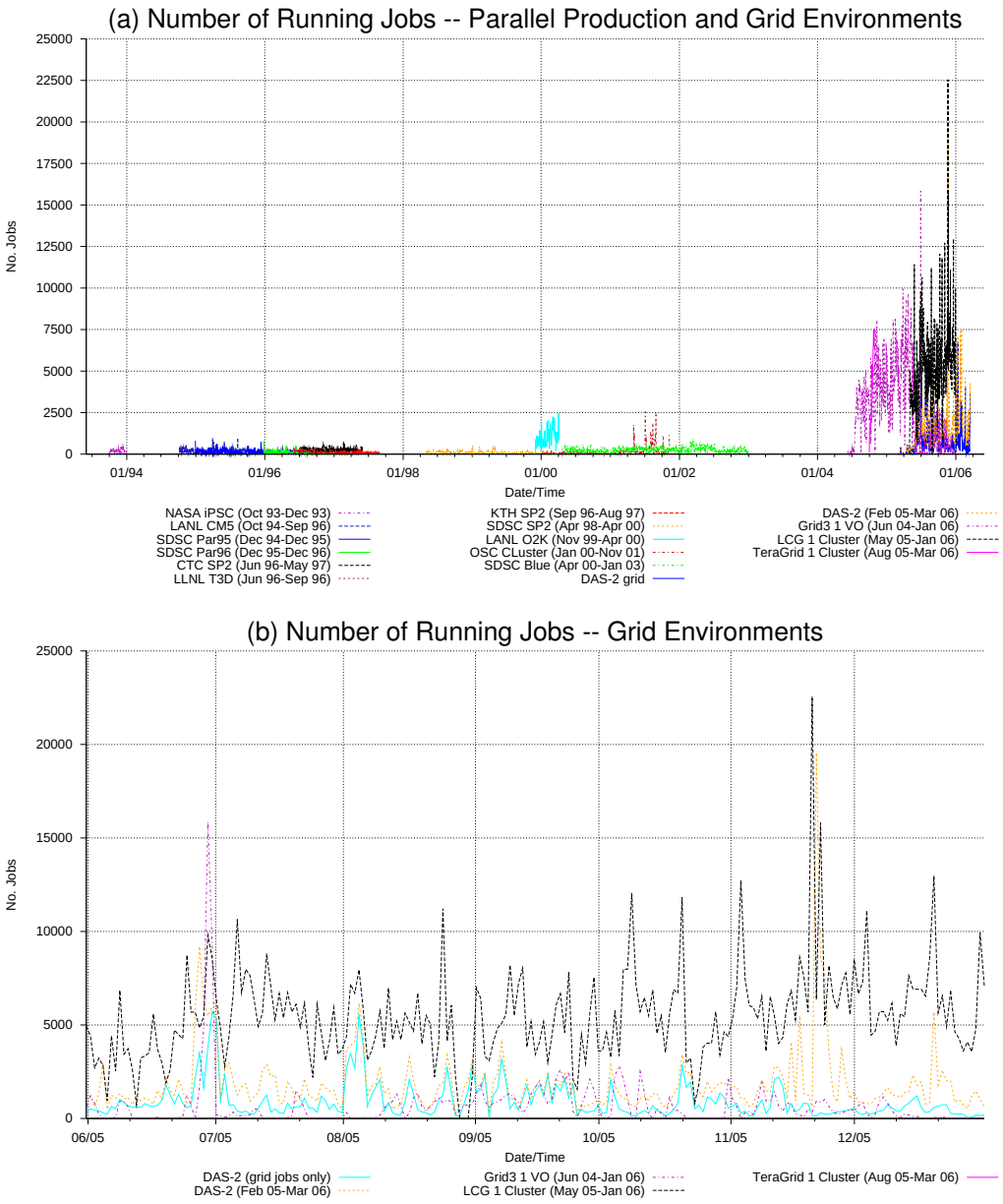

Fig. 6. Running jobs during daily intervals for grid and parallel environments: (a) comparative display over all data for grid and parallel environments; (b) comparative display for data between June 2005 and January 2006, for grid environments only.

5 A Survey of Workload Archives in Computer Science

In this section we survey several archival approaches in computer science areas, e.g., Internet, clusters, grids. We assess the relative merits of the surveyed approaches according to the requirements described in Section 2; Table 4 summarizes our survey. In comparison with the Internet community efforts, the GWA contains tools to generate and use synthetic grid workloads, besides the raw grid workload data. In comparison with the other efforts, the GWA offers more tools for processing, using, and sharing the stored data.

Table 4
A summary of workload archives in Computer Science. The +, ∼, and - signs denote a feature that is present, for which an insufficient approach has been taken, or which lacks, respectively.

Workload Archive (Since)	1 collect	2 process			3 use			4 share			5 build community			
		a	b	c	a	b	c	a	b	c	a	b	c	d
The Internet														
ITA [9] (1995)	-	+	+	+	-	-	-	-	+	+	-	+	+	-
WIDE [48] (1999)	-	+	+	+	-	-	-	-	+	+	-	+	+	-
CAIDA (2002)	+	+	+	+	-	-	-	+	+	+	+	+	+	+
CRAWDAD [11] (2005)	-	+	+	+	-	-	-	+	+	∼	+	+	+	+
NTTT [49] (2006)	-	-	+	+	-	+	-	-	-	+	∼	∼	-	-
Single-computer Systems														
BYU [50] (1999)	+	-	+	-	-	-	+	-	+	+	-	+	+	-
NMSU [51] (2002)	-	-	-	-	-	-	-	-	-	∼	+	-	+	-
CADRE [52] (2003)	+	-	+	+	-	-	-	+	-	+	∼	∼	+	-
Cluster-based Systems														
PWA [12] (1999)	+	+	-	-	-	+	+	∼	-	+	+	+	-	+
MAUI HPC [53] (2001)	-	-	-	-	-	-	-	-	-	+	-	∼	-	-
CFDR [13] (2007)	-	-	-	-	-	-	-	-	-	∼	-	-	-	-
Grids														
DGT [54,55] (2005)	-	-	-	-	-	-	-	-	-	+	-	∼	+	-
Other Archives of Interest														
RAT [56] (2007)	-	-	-	-	-	-	-	-	-	+	-	+	-	+
GWA (2006)	+	+	+	+	+	+	+	+	+	+	+	+	+	+

The research community has started to understand the importance of computer systems' performance evaluation based on real(istic) traces at the beginning of the '70s [57]. By the beginning of 1990s, this shift in practice had become commonplace [58,59]. In beginning of the 1990s the invention of the world-wide web [60], and the gradual lowering of the bandwidth and disk storage costs, paved the way for the first workload archives.

In 1995, the Internet community assembled the first publicly and freely available workload archive: the Internet Traffic Archive (ITA). ITA has undergone several updates over the years, which show why it still is surprisingly modern: it has tools for collecting and processing data, mailing lists for commenting, its data is publicly and freely available, etc. The Internet community has since created several other archives, i.e., WIDE, CAIDA's archives, CRAWDAD, and NTTT. The CAIDA archives [10,61,62] are combined the largest source of Internet traces. CRAWDAD is the first archive dedicated to the wireless networks community. These archives have gradually evolved towards covering most of the requirements expressed in Section 2 for the Internet community. Notably, with the exception of NTTT, they do not offer tools for using the offered data; NTTT offers tools for generating syn-

thetic workloads.

Contrary to the Internet community, the computer systems communities are still far from addressing Section 2's requirements. The computer architectures community started its first and most successful database (BYU) at the end of the 1990s. Since then, several other archives have started, e.g., NMSU and CADRE, but have yet to improve on the results of the BYU archive. For the cluster-based communities, the Parallel Workloads Archive (PWA) covers many of the requirements, and has become the de-facto standard for the parallel production environments community. The MAUI HPC archive started separately, but has since been included in the PWA. The DGT, CFDR, and RAT archives only resource availability and failure traces.

The PWA is the closest project to our GWA both in target (parallel computing environments) and realization. Recently, the PWA has added several grid traces to its content. However, the PWA workload format [63] was not designed for grid workloads, and loses information on many grid-specific aspects, including the number of used nodes (which may be different from the number of used processors, for multiprocessor nodes), co-allocation, submission site (which may be different from the execution site), and job exit status. The Grid Workloads Archive is the first archive to accommodate the requirements of grid workloads, and thus complements the PWA and other approaches. To the best of our knowledge, the GWA is also the first to satisfy all the requirements of a workloads archive.

6 Conclusion and Future Work

While many grids are currently serving as e-Science infrastructure, very little is known about the real users' demand. The lack of grid workloads hampers the research on grid resource management, and the practice in grids design, management, and operation. To collect grid workloads and to make them available to this diverse community, we have designed and developed the Grid Workloads Archive. The design focuses on two broad requirements: building a grid workload data repository, and building a community center around the archived data. For the former, we provide tools for collecting, processing, and using the data. For the latter, we provide mechanisms for sharing the data and other community-building support. We have collected so far traces from nine well-known grid environments.

Figure 7 shows the timeline of the GWA project. We are currently extending this work in two directions: extending the content and inter-connecting the GWA with other grid tools. For extending the content, our current focus is on supporting online addition of traces with minimal support from the GWA team. We are also working towards a monthly contribution mechanism, which includes anonymizing information at the contributor's site. We are currently inter-connecting the GWA with ServMark, a grid testing tool that extends GrenchMark [64]. ServMark will

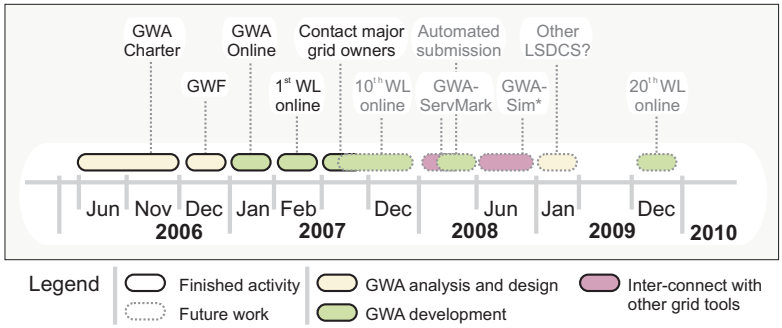

Fig. 7. The GWA project timeline. LSDCS is an acronym for large-scale distributed computing systems.

use tools and the contents of the GWA to generate new workloads. By replaying the existing or the new grid traces on several grids, we hope to prove that the GWA enables comparable grid performance testing. We are also continuing our effort to provide libraries for using GWA contents in several grid simulators, e.g., GridSim [65], SimGrid [66], GangSim [67], and DGSim [31].

For the future, we plan to bring the community of resource management in large-scale distributed computing systems closer to the Grid Workloads Archive. We believe that our archive will be useful for many scientific directions including, but not limited to, scheduling in and performance evaluation of such systems. We also plan to develop a grid workload model based on the data collected in the archive, and to include it in the workload models database.

Data Availability and Contributions

The Grid Workloads Archive can be reached online at

`http://gwa.st.ewi.tudelft.nl`

We continue to look for contributors who would donate grid workloads for the benefit of the grid community. On the one hand, many organizations view this data as revenue-generating (in the industry), or critical for obtaining grants (the academia), and are reluctant to make the data public. On the other hand, collecting unrepresentative traces, either because of reduced size or because of the source (e.g., jobs specific to only one user), is not a goal for the Grid Workloads Archive.

We are looking for two types of contributions: one-time and monthly. We look for one-time contributions with traces that have an average value of $\star\star\star$ or higher (see Section 3.3). For monthly contributions, we invite contributors which collect month-worth traces with a value of at least $\star$ on average.

Acknowledgements

This research 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). Part of this work was also carried out under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265). We are grateful to the GridPP project team at Rutherford Appleton Lab (RAL) who graciously provided us with the LCG data. We are equally thankful to the USATLAS/Grid3 experiment, to the UC/ANL TeraGrid site, to the Grid'5000 team (in particular to Dr. F. Cappello), to the DAS-2 team (in particular to K. Verstoep and P. Anita), to the NorduGrid team (in particular to Dr. B. Kónya), to the NGS team (in particular to Dr. R. Sakellariou), and to the Condor UWisc and GLOW team (in particular to D. Bradley, to G. Thain, and to Dr. M. Livny) who provided their respective Grid traces for our work. Finally, we would like to thank our anonymous reviewers for their comments and suggestions.

References

[1] EGEE Team, LCG (2004).
URL lcg.web.cern.ch/LCG

[2] P. Eerola, B. Kónya, O. Smirnova, T. Ekelöf, M. Ellert, J. R. Hansen, J. L. Nielsen, A. Wäänänen, A. Konstantinov, J. Herrala, M. Tuisku, T. Myklebust, F. Ould-Saada, B. Vinter, The NorduGrid production grid infrastructure, status and plans., in: Int'l. Conf. on Grid Computing (GRID), IEEE Computer Society, 2003, pp. 158–165.

[3] The TeraGrid Project , Npaci (March 2006).
URL www.teragrid.org

[4] The Open Science Grid Project , OSG (Jul 2007).
URL www.opensciencegrid.org/

[5] A. Iosup, D. H. J. Epema, GRENCHMARK: A framework for analyzing, testing, and comparing grids., in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2006, pp. 313–320.

[6] O. Khalili, J. He, C. Olschanowsky, A. Snavely, H. Casanova, Measuring the performance and reliability of production computational grids, in: Int'l. Conf. on Grid Computing (GRID), IEEE Computer Society, 2006.

[7] H. Li, D. Groep, L. Wolters, J. Templon, Job failure analysis and its implications in a large-scale production grid., in: IEEE Int'l. Conf. on e-Science and Grid Computing (e-Science), IEEE Computer Society, 2006, pp. 27–27.

[8] A. Iosup, D. Epema, P. Couvares, A. Karp, M. Livny, Build-and-test workloads for grid middleware: Problem, analysis, and applications, in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2007, pp. 205–213.

[9] P. Danzig, J. Mogul, V. Paxson, M. Schwartz, The Internet Traffic Archive (Aug 2007).
URL ita.ee.lbl.gov/

[10] CAIDA Team, The Cooperative Association for Internet Data Analysis (Aug 2007).
URL www.caida.org/data/

[11] J. Yeo, D. Kotz, T. Henderson, CRAWDAD: a community resource for archiving wireless data at dartmouth, SIGCOMM Comput. Commun. Rev. 36 (2) (2006) 21–22.

[12] The Parallel Workloads Archive Team , The parallel workloads archive logs (June 2006).
URL www.cs.huji.ac.il/labs/parallel/workload/logs.html

[13] B. Schroeder, G. Gibson, The computer failure data repository (CFDR) (Aug 2007).
URL cfdr.usenix.org/

[14] U. Lublin, D. G. Feitelson, The workload on parallel supercomputers: modeling the characteristics of rigid jobs., J. Parallel Distrib. Comput. 63 (11) (2003) 1105–1122.

[15] K. Czajkowski, I. T. Foster, C. Kesselman, Resource co-allocation in computational grids., in: IEEE Int'l. Symp. on High Performance Distributed Computing (HPDC), IEEE Computer Society, 1999.

[16] W. Smith, I. T. Foster, V. E. Taylor, Scheduling with advanced reservations., in: Int'l. Parallel & Distributed Processing Symposium (IPDPS), IEEE Computer Society, 2000, pp. 127–132.

[17] A.Iosup, H. Li, C. Dumitrescu, L. Wolters, D.H.J.Epema, The Grid Workload Format (Nov 2006).
URL gwa.ewi.tudelft.nl/TheGridWorkloadFormat_v001.pdf

[18] A. Iosup, D. H. J. Epema, C. Franke, A. Papaspyrou, L. Schley, B. Song, R. Yahyapour, On grid performance evaluation using synthetic workloads., in: E. Frachtenberg, U. Schwiegelshohn (Eds.), Int'l. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), Revised Selected Papers, Vol. 4376 of Lecture Notes in Computer Science, Springer, 2007, pp. 232–255.

[19] A. Iosup, M. Jan, O. Sonmez, D. H. Epema, On the dynamic resource availability in grids, in: Int'l. Conf. on Grid Computing (GRID), IEEE Computer Society, 2007.

[20] H. Bal et al., The Distributed ASCI Supercomputer project, Operating Systems Review 34 (4) (2000) 76–96.

[21] H. Mohamed, D. Epema, The design and implementation of the KOALA co-allocating grid scheduler, in: P. M. A. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, M. Bubak (Eds.), European Grid Conference (EGC), Vol. 3470 of Lecture Notes in Computer Science, Springer, 2005, pp. 640–650.

[22] G. Wrzesinska, R. van Nieuwpoort, J. Maassen, H. E. Bal, Fault-tolerance, malleability and migration for divide-and-conquer applications on the grid., in: Int'l. Parallel & Distributed Processing Symposium (IPDPS), IEEE Computer Society, 2005.

[23] R. Bolze, F. Cappello, E. Caron, M. Dayd , F. Desprez, E. Jeannot, Y. Jgou, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.-G. Talbi, T. Irena, Grid'5000: a large scale and highly reconfigurable experimental grid testbed., International Journal of High Performance Computing Applications 20 (4) (2006) 481–494.

[24] N. Capit, G. D. Costa, Y. Georgiou, G. Huard, C. Martin, G. Mouni, P. Neyron, O. Richard, A batch scheduler with high level components, in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2005, pp. 776–783.

[25] M. Ellert et al., Advanced Resource Connector middleware for lightweight computational grids, Future Generation Computer Systems 23 (2) (2007) 219–240.

[26] D. Thain, T. Tannenbaum, M. Livny, Distributed computing in practice: the Condor experience., Concurrency - Practice and Experience 17 (2-4) (2005) 323–356.

[27] I. Foster et al., The Grid2003 production grid: Principles and practice, in: IEEE Int'l. Symp. on High Performance Distributed Computing (HPDC), IEEE Computer Society, 2004, pp. 236–245.

[28] M. Mambelli, al., Atlas data challenge production on Grid3 (2005).
URL `griddev.uchicago.edu/download/atgce/doc.pkg/presentations/chep04-503-usatlas-dc2.pdf`

[29] A. Iosup, C. Dumitrescu, D. H. Epema, H. Li, L. Wolters, How are real grids used? The analysis of four grid traces and its implications., in: Int'l. Conf. on Grid Computing (GRID), IEEE Computer Society, 2006, pp. 262–269.

[30] A. Iosup, M. Jan, O. Sonmez, D. Epema, The characteristics and performance of groups of jobs in grids, in: Int'l. European Conference on Parallel and Distributed Computing (Euro-Par), Lecture Notes in Computer Science, Springer, 2007, pp. 382–393.

[31] A. Iosup, D. Epema, T. Tannenbaum, M. Farrellee, M. Livny, Inter-operating grids through delegated matchmaking, in: Proc. of the ACM/IEEE Conference on High Performance Networking and Computing (SC), ACM Press, 2007.

[32] H. Li, R. Heusdens, M. Muskulus, L. Wolters, Analysis and synthesis of pseudo-periodic job arrivals in grids: A matching pursuit approach., in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2007, pp. 183–196.

[33] H. Li, M. Muskulus, L. Wolters, Modeling job arrivals in a data-intensive grid., in: E. Frachtenberg, U. Schwiegelshohn (Eds.), Int'l. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), Revised Selected Papers, Vol. 4376 of Lecture Notes in Computer Science, Springer, 2007, pp. 210–231.

[34] H. Li, Long range dependent job arrival process and its implications in grid environments., in: Proc. of MetroGrid Workshop, Int'l. Conference on Networks for Grid Applications (GridNets07), ACM Press, 2007, (in press).

[35] A. Iosup, P. Garbacki, D. H. Epema, Provisioning and scheduling resources for world-wide data-sharing services., in: IEEE Int'l. Conf. on e-Science and Grid Computing (e-Science), IEEE Computer Society, 2006, pp. 84–84.

[36] H. Li, D. L. Groep, J. Templon, L. Wolters, Predicting job start times on clusters., in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2004, pp. 301–308.

[37] H. Li, L. Wolters, Towards a better understanding of workload dynamics on data-intensive clusters and grids., in: Int'l. Parallel & Distributed Processing Symposium (IPDPS), IEEE Computer Society, 2007, pp. 1–10.

[38] H. Li, D. L. Groep, L. Wolters, Mining performance data for metascheduling decision support in the Grid, Future Generation Computer Systems 23 (1) (2007), pp. 92–99.

[39] H. B. Newman, I. C. Legrand, P. Galvez, R. Voicu, C. Cirstoiu, Monalisa : A distributed monitoring service architecture, in: Computing in High Energy and Nuclear Physics (CHEP03), 2003.

[40] Z. Balaton, G. Gombas, GridLab Monitoring: Detailed Architecture Specification, EU Information Society Technologies Programme (IST) TR GridLab-11-D11.2-01-v1.2, http://www.gridlab.org/Resources/Deliverables/D11.2.pdf (2001).

[41] C. Stratan, C. Cirstoiu, A. Iosup, On the accuracy of off-line monitoring information in grids, in: Proc. of the 17th Intl. Conference on Control Systems and Computer Science (CSCS-17), 2007, may, Bucharest, Romania.

[42] D. G. Feitelson, B. Nitzberg, Job characteristics of a production parallel scientific workload on the NASA Ames iPSC/860, in: D. G. Feitelson, L. Rudolph (Eds.), Int'l. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), Revised Selected Papers, Vol. 949 of Lecture Notes in Computer Science, Springer, 2007, pp. 337–360.

[43] D. G. Feitelson, Memory usage in the LANL CM-5 workload, in: D. G. Feitelson, L. Rudolph (Eds.), Int'l. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), Revised Selected Papers, Vol. 1291 of Lecture Notes in Computer Science, Springer, 1997, pp. 78–94.

[44] K. Windisch, V. Lo, R. Moore, D. Feitelson, B. Nitzberg, A comparison of workload traces from two production parallel machines, in: 6th Symp. Frontiers Massively Parallel Comput., 1996, pp. 319–326.

[45] S. Hotovy, Workload evolution on the Cornell Theory Center IBM SP2, in: D. G. Feitelson, L. Rudolph (Eds.), Int'l. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), Revised Selected Papers, Vol. 1162 of Lecture Notes in Computer Science, Springer, 1996, pp. 27–40.

[46] D. G. Feitelson, A. Mu'alem Weil, Utilization and predictability in scheduling the IBM SP2 with backfilling, in: 12th Intl. Parallel Processing Symp. (IPPS), 1998, pp. 542–546.

[47] A. W. Mu'alem, D. G. Feitelson, Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling, IEEE Trans. Parallel & Distributed Syst. 12 (6) (2001) 529–543.

[48] K. Cho, K. Mitsuya, A. Kato, Traffic data repository at the wide project, in: ATEC'00: Proceedings of the Annual Technical Conference on 2000 USENIX Annual Technical Conference, USENIX Association, Berkeley, CA, USA, 2000, pp. 51–51.

[49] NTTT Team, Network Tools and Traffic Traces (Aug 2007).
URL `www.grid.unina.it/Traffic/`

[50] Performance Evaluation Laboratory, BYU trace distribution center (Aug 2007).
URL `tds.cs.byu.edu/tds/`

[51] Performance and Architecture Research Lab, NMSU trace database (Aug 2007).
URL `tracebase.nmsu.edu/`

[52] Pablo Research Group, CADRE: A national facility for I/O characterization and optimization (Aug 2007).
URL `www.renci.org/pablo/Project/CADRE/`

[53] Center for HPC Cluster Resource Management and Scheduling, Maui HPC workload/resource trace respository (Aug 2007).
URL `www.supercluster.org/research/traces/`

[54] D. Kondo, G. Fedak, F. Cappello, A. A. Chien, H. Casanova, The desktop grid availability traces archive (Aug 2007).
URL `vs25.lri.fr:4320/dg/`

[55] D. Kondo, G. Fedak, F. Cappello, A. A. Chien, H. Casanova. Characterizing resource availability in enterprise desktop grids, Future Generation Computer Systems 23 (7) (2007) pp. 888–903.

[56] B. Godfrey, I. Stoica, Repository of availability traces (RAT) (Aug 2007).
URL `www.cs.berkeley.edu/~pbg/availability/`

[57] D. Ferrari, Workload characterization and selection in computer performance measurement, IEEE Computer 5 (4) (1972) 18–24.

[58] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling,, 1991, winner of "1991 Best Advanced How-To Book, Systems" award from the Computer Press Association.

[59] M. Calzarossa, G. Serazzi, Workload characterization: a survey, Proc. of the IEEE 81 (8) (1993) 1136–50.

[60] T. Berners-Lee, R. Cailliau, J.-F. Groff, B. Pollermann, World-wide web: The information universe., Electronic Networking: Research, Applications and Policy 1 (2) (1992) 74–82.

[61] DatCat Team, The DatCat Internet Measurement Data Catalog (Aug 2007).
URL imdc.datcat.org/

[62] MOAT Team, The NLANR Measurement and Network Analysis Group (Aug 2007).
URL mna.nlanr.net/

[63] S. J. Chapin, W. Cirne, D. G. Feitelson, J. P. Jones, S. T. Leutenegger, U. Schwiegelshohn, W. Smith, D. Talby, Benchmarks and standards for the evaluation of parallel job schedulers., in: D. G. Feitelson, L. Rudolph (Eds.), Int'l. Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP), Revised Selected Papers, Vol. 1659 of Lecture Notes in Computer Science, Springer, 1999, pp. 67–90.

[64] M. Andreica, N. Tapus, A. Iosup, D. Epema, C. Dumitrescu, I. Raicu, I. Foster, M. Ripeanu, Towards servmark, an architecture for testing grids, Tech. Rep. TR-0062, Institute on Resource Management and Scheduling, CoreGRID - Network of Excellence (November 2006).

[65] R. Buyya, M. M. Murshed, GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing., Concurrency and Computation: Practice and Experience 14 (13-15) (2002) 1175–1220.

[66] A. Legrand, L. Marchal, H. Casanova, Scheduling distributed applications: the simgrid simulation framework., in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2003, pp. 138–145.

[67] C. Dumitrescu, I. T. Foster, GangSim: a simulator for grid scheduling studies., in: IEEE/ACM Intl. Symp. on Cluster Computing and the Grid (CCGrid), IEEE Computer Society, 2005, pp. 1151–1158.