

Grid Computing Workloads

In the mid 1990s, the grid computing community promised the “compute power grid,” a utility computing infrastructure for scientists and engineers. Since then, a variety of grids have been built worldwide, for academic purposes, specific application domains, and general production work. Understanding grid workloads is important for the design and tuning of future grid resource managers and applications, especially in the recent wake of commercial grids and clouds. This article presents an overview of the most important characteristics of grid workloads in the past seven years (2003-2010). Although grid user populations range from tens to hundreds of individuals, a few users dominate each grid’s workload both in terms of consumed resources and the number of jobs submitted to the system. Real grid workloads include very few parallel jobs but many independent single-machine jobs (tasks) grouped into single “bags of tasks.”

In the mid 1990s, grid computing engineers formed a new vision – one in which the grid operated as a ubiquitous and uninterrupted computing and data platform that offered uniform user access, similar to a power grid.¹ Grids such as the Enabling Grids for E-Science grid (EGEE, which is global although mostly European-based), the Open Science Grid (OSG, which is also global although mostly US-based), Teragrid (US), Naregi (Japan), Grid’5000 (France), and the Distributed ASCI Supercomputer (DAS, the Netherlands), have grown from serving tens of scientists to hundreds. These grids are used for many application areas, including physics, bioinformatics, Earth sciences, life sciences, finance, space engineering,

and so on. They assist the fields of science and engineering and couple theory and experimentation with computation and data-intensive discovery.^{2,3} Grids still pose many research challenges—among them, the high and variable job wait times. To continue evolving and tuning grids for production work, it’s important to understand the characteristics of entire grid workloads.

Grids are collections of resources ranging from clusters to supercomputers. Often, grid resource providers and grid resource consumers (users) are different entities. Providers determine resource management policies and offer only minimal, generic job management services. To simplify management, virtual organizations (VOs) group

**Alexandru Iosup
and Dick Epema**
Delft University of Technology

Table 1. Summary of the properties of the studied traces.

Trace ID (GWA index)	Source system		Number of ... in system	
	Name (country, type)	Period	Sites	CPUs
GWA-T-1	DAS-2 (NL, academic)	Feb 05 – Mar 06	5	400
GWA-T-2	Grid'5000 (FR, academic)	May 04 – Nov 06	15	2,500+
GWA-T-3	NorduGrid (EU, academic/production)	May 04 – Feb 06	75+	2,000+
GWA-T-4	AuverGrid (FR, production)	Jan 06 – Jan 07	5	475
GWA-T-5	NGS (UK, production)	Feb 03 – Feb 07	1	400+
GWA-T-6	LCG, RAL cluster (UK, production)	May 05 – Jan 06	1*	880
GWA-T-7	GLOW (US, production)	Sep 06 – Jan 07	1*	1,400+
GWA-T-8	Grid3 (US, academic/production)	Jun 04 – Jan 06	29	2,200+
GWA-T-9	TeraGrid-I, ANL cluster (US, production)	Aug 05 – Mar 06	1*	96
GWA-T-10	SHARCNET (CA, production)	Dec 05 – Dec 06	10	6,828
GWA-T-11	EGEE/LCG (EU, production)	Nov 05 – Dec 05	220+	24,000+
GWA-T-12	Condor U.Wisc.-Madison (US, production)	Oct 06 – Nov 06	5	2,100+
GWA-T-13	TeraGrid-2, NCSA cluster (US, production)	May 06 – Jan 07	1*	1,000
GWA-T-14	DAS-3 (NL, academic)	July 06 – Oct 08	5	544
GWA-T-15	Austrian Grid (AT, academic/production)	Sep 06 – Oct 07	8	250
GWA-T-16	Austrian Grid 2 (AT, academic/production)	May 07 – Nov 07	8	250
GWA-T-17	TeraGrid-2, NCSA cluster (US, production)	Jan 10 – May 10	1*	930

*The trace only represents a part of the system.

administratively users and/or resource providers. Over time, scientists and engineers have tried many types of jobs on grids – from sequential to parallel, from compute-intensive to data-intensive, and from coordinated applications to bags of independent tasks. A contemporary grid-based experiment may require the repeated execution of a computational task on different sets of input parameters or data.

In this work, we discuss the characteristics of grid workloads, with a focus on the past seven years. We look at four main grid workload axes: system usage (utilization and task arrivals), user population (number of users and VOs), general application characteristics (CPU, memory, disk, and network), and characteristics of grid-specific application types (presence, structure, and so on). We show that grid workloads are very different from the workloads of other environments used by scientists and engineers, and emphasize the emergence of workflows and bags of many tasks as important application types.

General Workload Characteristics

Our analysis is based on grid workload traces collected from more than 15 real grids. Grid owners and users kindly provided these traces; some

of the traces are now publicly available via the Grid Workloads Archive (GWA; <http://gwa.ewi.tudelft.nl/>).⁴ Table 1 summarizes two properties of the studied traces: duration and system size. The values illustrate our study’s breadth. Nine of the traces are long term (one year of operation or more) and 13 are medium term (six months or more); we collected the traces from several large grids (2,000 CPUs or more), including EGEE, Grid’5000, Grid3 (the precursor of OSG), and NorduGrid. The traces also include examples of system replacement (DAS-2 was phased out and replaced with DAS-3, and traces GWA-T-15 and GWA-T-16 represent the replacement of the job manager), system evolution (traces GWA-T-13 and GWA-T-17 come from the same system but at a 3.5-year interval), and detailed/coarse views of the same system (for example, the traces GWA-T-6/GWA-T-11 for EGEE).

Grid workloads exhibit several features that we examine in this article; more information appears in our previous studies.⁴

System Utilization

The long-term average grid utilization ranges from very low (10 to 15 percent in research grids DAS and Grid’5000) to very high (more

Table 2. Summary of the content of the studied traces.

Trace ID (GWA index)	Source system Name (country/region)	Number of observed					
		Jobs	GRP	USR	CPU time	Arrivals	Spike
GWA-T-1	DAS-2 (NL)	602K	12	332	69y	71	19,550
GWA-T-2	Grid'5000 (FR)	951K	10	473	128y	—	—
GWA-T-3	NorduGrid (EU)	781K	106	387	2,444y	28	7,953
GWA-T-4	AuverGrid (FR)	404K	9	405	278y	46	823/h
GWA-T-5	NGS (UK)	632K	1	379	270y	23	4,994/h
GWA-T-6	LCG, RAL cl. (UK)	1.1M	25	206	751y	—	22,550
GWA-T-7	GLOW (US)	216K	1*	18	120y	—	6,590
GWA-T-8	Grid3 (US)	1.3M	1*	19	240y	—	15,853
GWA-T-9	TeraGrid-1, ANL cl. (US)	1.1M	26	121	—	—	7,561
GWA-T-10	SHARCNET (CA)	1.2M	—	412	3,782y	127	22,334/h
GWA-T-11	EGEE/LCG (EU)	188K	28	216	54y	504	1,638/h
GWA-T-12	Condor U.Wisc.-Madison (US)	765K	—	—	22,370y	—	—
GWA-T-13	TeraGrid-2, NCSA cl. (US)	140K	—	—	222y	—	—
GWA-T-14	DAS-3 (NL)	2.0M	12	333	—	—	—
GWA-T-15	Austrian Grid (AT)	141k	—	—	152d	—	—
GWA-T-16	Austrian Grid 2 (AT)	46k	—	—	41d	—	—
GWA-T-17	TeraGrid-2, NCSA cl. (US)	28K	—	83	—	—	—

*The trace only represents a part of the system. GRP and USR are acronyms for number of groups (VOs) and of users, respectively. The “Arrivals” column lists the average number of arrivals per hour. The “Spike” column lists the maximum number of jobs running per day.

than 85 percent in production grids such as the Large Hadron Collider (LHC) Computing Grid (LCG), Condor University Wisconsin-Madison, and AuverGrid). Short-term utilization can be very high, and every grid we investigated in this work experienced week-long overloads (full-capacity utilization and excess demand) during their existence. Load imbalance between grid sites and submission spikes happen often.

Workload Size

Table 2 summarizes the size characteristics of grid workloads. A single grid cluster can provide more than 750 CPU years per year (for example, the RAL cluster in LCG), whereas a single user VO can consume more than 350 CPU years per year in combined use (for example, the Atlas VO in Grid3). On average, grid systems complete more than 4,000 jobs per day in LCG’s RAL cluster, and between 500 and 1,000 in Grid3 and DAS-2. Although the number of hourly job arrivals is generally small, the number of jobs running in a grid can spike to more than 20,000 per day for a single cluster (for example, in DAS-2 and the LCG RAL cluster traces), and to more than 20,000 per hour for a whole grid (Sharcnet).

Table 3. Summary of the job characteristics for the studied traces.*

System	Overall job characteristics		
	Size (CPUs)	Runtime (s)	Memory (Mbytes)
GWA-T-1	4.3 (6.3)	370 (3,938)	46 (346)
GWA-T-3	1.1 (1.0)	89,274 (284,300)	200 (307)
GWA-T-4	1.0 (0.0)	25,186 (40,780)	296 (343)
GWA-T-5	1.4 (2.8)	2,925 (17,908)	39 (226)
GWA-T-6	1.0 (0.0)	14,599 (28,641)	195 (206)
GWA-T-7	1.0 (0.0)	4,705 (14,488)	332 (276)
GWA-T-8	1.0 (0.0)	13,797 (25,201)	n/a
GWA-T-10	1.5 (6.2)	31,964 (117,088)	81 (466)
GWA-T-11	1.0 (0.0)	8,971 (32,833)	n/a

*Standard deviations are in parentheses.

Often, fewer than five users dominate the workload, both in terms of number of jobs submitted to the grid and resources consumed.⁷

Submission Patterns

Grid workloads exhibit strong time patterns, including seasonal, workday, and hourly. Most grids are used less during holidays, weekends, and midday. Many academic grids are overloaded during periods preceding major conferences.

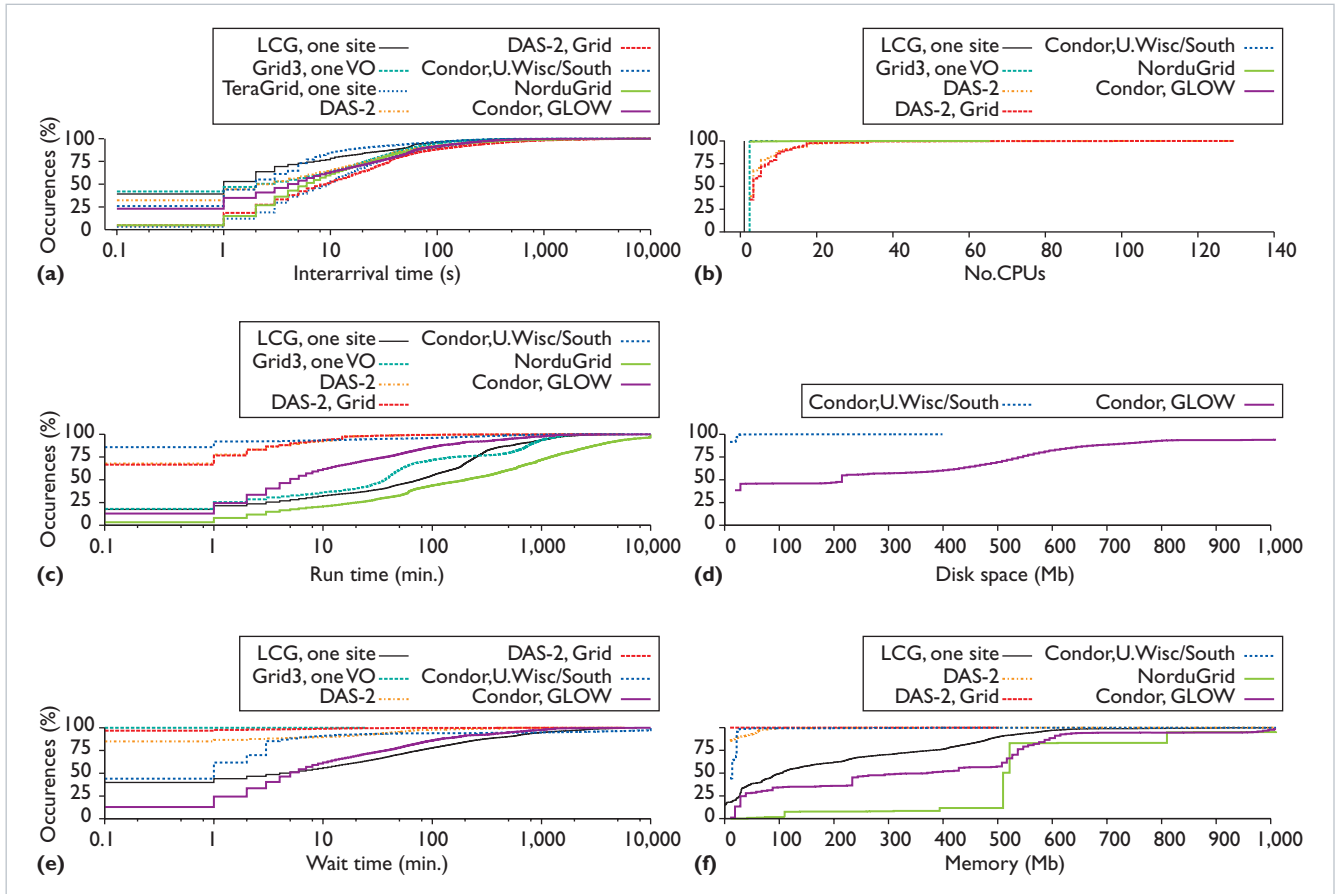


Figure 1. CDFs of the most important job characteristics for NorduGrid, Condor GLOW, Condor UWisc-South, TeraGrid, Grid3, LCG, DAS-2, and DAS-2 Grid. Time-related characteristics are in logscale. (a) Interarrival time. (b) Job parallelism. (c) Run time. (d) Consumed disk space. (e) Wait time. (f) Consumed memory.

The submission behavior of individual users varies greatly, but top users often replace irregular (manual) submission with tools that submit jobs at regular intervals.

Grids vs. Parallel Production

Compared to the clusters and low-end supercomputers of the late 1990s and early 2000s, grids over the past seven years exhibit similar resource consumption, complete more jobs per day, demonstrate higher spikes in the number of concurrently running jobs, and can reach much higher utilizations (see Table 2). Specifically, parallel production environments (PPEs) offer 50 to 1,300 CPU years per year and have (on average) fewer than 500 jobs completed per day, spikes of 300 to 5,400 jobs, and utilization often in the mid-60th percentile. (These results hold for each individual PPE trace in the Parallel Workloads Archive at <http://cs.huji.ac.il/labs/parallel/workload/>.)

General Job Characteristics

In this article, we characterize the jobs present in grid workloads, regardless of their application domain or structure. Table 3 summarizes the averages and standard deviations of the number of processors allocated to jobs, job runtimes, and memory consumption of jobs. Figure 1 depicts the cumulative distribution functions (CDFs) associated with various job characteristics. Both indicate the high variability of job characteristics in the grid.

Not every grid workload trace we use in this study contains information about all characteristics. In particular, only a few contain memory-, I/O-, and network-related information. For I/O and network information, we use the Condor-based system traced in GWA-T-12 and independently analyze five data subsets, each coming from a traced resource pool. Subsets t1 and t2 comprise mostly engineering and computer science jobs, respectively; subsets t3, t4,

Table 4a. Average data per job in Condor-based grids (I/O in operations and volume).

T-12 part	I/O (KOps)				I/O Traffic (Mbytes)		
	Total	Rd	Wr	Wr %	Total	Rd	Wr %
t1	28	18	6	20	469	174	63
t2	957	770	187	20	144	114	21
t3	904	881	23	3	161	130	19
t4	13,058	9	13,049	100	389	33	92
t5	11,128	8	11,121	100	330	31	91

Table 4b. Average data per job in Condor-based grids (network traffic).

T-12 part	File Transfer (Mbytes)				Remote Sys. Calls (Mbytes)		
	Total	In	In %	Out %	Total	In	In %
t1	10,865	8,259	76	24	28	71	59
t2	1,736	1,542	89	11	77	28	40
t3	1,938	1,738	90	10	6	32	42
t4	1,043	653	63	37	44	6	100
t5	671	432	64	36		40	91

and t5 comprise exclusively high-energy physics (HEP) jobs of different characteristics. Other studies⁴⁻⁶ offer a more detailed examination of typical jobs present in grid workloads.

Parallel Jobs

Grid workloads exhibit little intra-job parallelism (see Figure 1b); in contrast to PPE workloads, which include mostly parallel jobs – that is, jobs that require more than a single node to operate – grid workloads are dominated by loosely coupled jobs. Moreover, in many grid workload traces, no parallel jobs exist. However, we did find two exceptions, Sharcnet and TeraGrid, both of which run scientific applications as parallel jobs. Even for the few grids that do run parallel jobs, job parallelism is low: mostly fewer than 32 processors per job (the most is 800 [Sharcnet] and 128 [others]). These small parallel job sizes match the parallel workloads of early 2000s PPEs.

Job Runtimes, Arrival Times, and Wait Times

In general, grid jobs require several hours to complete, with per-grid averages ranging from about one hour to one day. HEP jobs were designed for approximately 12 hours of processing on low-end machines; thus, many run for between six and seven hours on regular grid nodes.⁶ The DAS-2 and DAS-3 grids were designed to promote the use of small, interactive

jobs, which explains the DAS-2 outlier average-job runtime of 370 seconds. Although the averages are relatively high, most grid workloads contain large numbers of much shorter or much longer jobs. Notably, Figure 1c shows a runtime of 2 minutes or less for a quarter of the jobs in many grids. The high variability of grid job runtimes and arrival times (see also Figures 1a and 1c) is an important factor in the high and variable wait times of grid jobs (see Figure 1e).

Memory Requirements

On average, grid jobs require tens to hundreds of Mbytes of memory. Most HEP jobs require machines with at least 2 Gbytes of memory per processor, although they might use less in practice. On average, production grid jobs require more memory than academic grid jobs. The CDF of memory consumption in Figure 1f shows the existence of preferred memory consumption sizes; the NorduGrid trace has a distribution mode of approximately 500 Mbytes.

I/O Requirements

Many grid jobs are compute-intensive and have modest I/O requirements. Table 4a summarizes the I/O consumption for five subsets of the GWA-T-12 trace – one for each resource pool in the system. The total number of operations and the total I/O traffic averaged by grid jobs are both higher than for typical scientific applications.⁵ The variability of observed values remains high. The sizes

Table 5. Summary of BoT presence in grid traces.

Trace ID	Observed BoTs	Percentage from total	
		Jobs (%)	CPUTime (%)
GWA-T-1	57k	92	78
GWA-T-2	26k	85	30
GWA-T-3	50k	94	90
GWA-T-6	43k	95	95
GWA-T-7	13k	95	96
GWA-T-8	302k	94	98
GWA-T-10	16k	93	92
GWA-T-11	5k	96	97
GWA-T-12	135K	94	96
GWA-T-13	68K	96	86

and rates of various I/O operations exhibit pronounced distributional modes, so system designers can optimize for common cases. However, the high fraction of `writes`, from all I/O operations, might make caching difficult. HEP jobs have much larger I/O requirements⁶ (see Figure 1d). At roughly 65 MBps for single experiments, these jobs process about 2.2 Pbytes of data per year; their mean file size is 300 Mbytes, almost 5 percent of their files are larger than 1 Gbyte, and each job accesses more than 100 files on average.

Network Requirements

In grids, network traffic can be generated by I/O file transfers – to and from the processing nodes – and by remote system calls. Table 4b summarizes the network consumption per job for the same five subsets of the GWA-T-12 trace we used for the I/O analysis (see Table 4a). The input varies widely among these subsets and, in all traces, represents more than 60 percent of file traffic. The traffic used for remote system calls is much lower than for files; here, the fraction of output traffic ranges from 0 to 60 percent of the total traffic.

Bags of Tasks

Bags of tasks (BoTs) are loosely coupled parallel jobs in which a set of tasks are executed to produce a meaningful, combined result. Many grid workload traces are missing information about the job-to-BoT mapping, and the use of BoT managers can make the automatic identification of BoTs in such traces very difficult. For example, many BoT managers delay task submissions to ensure that a limited number of tasks run concurrently in the grid; thus, tasks belonging

to the same BoT become grid jobs with different submission times. When job-to-BoT mapping information is missing from the trace, we identify BoTs using a method^{7,8} in which jobs submitted by the same user are grouped according to their relative arrival time. Table 5 summarizes the presence of BoTs in several selected grids. In most grid traces, BoT submissions account for more than 75 percent of the tasks and consumed CPU time (see Table 5). For some grids, BoTs are even responsible for over 90 percent of the total consumed CPU time. The average number of tasks per BoT in the different grid traces we investigated ranges between 2 and 70; most averages for these traces fall between 5 and 20.

A model accounting for the highly variable data we observed for grid BoTs⁸ could focus on four aspects: the submitting user, BoT arrival patterns, BoT size, and intra-BoT (individual task) characteristics. The probability of a specific user submitting a grid job is well modeled by a Zipf distribution. For most systems, BoT inter-arrival time, BoT size, and variability of BoT task runtimes are best modeled by a Weibull distribution. For most systems, the normal distribution models the average BoT task runtime well.

Workflows, Pilots, and Others

Although grids already support (small) bags of tasks, the performance of their generic job and resource management services can be improved via user- and application-specific tools and policies. Motivated by high rates of system and middleware failure,⁹ high job management overhead, and slow job failure detection, the grid community has built tools and mechanisms for improved execution and coordination of specific types of jobs in grids.

Grid Workflows

Grid workflows are jobs with a graph structure, in which nodes are computing or data-transfer tasks, and edges are dependencies between the tasks. A common workflow would consist of preprocessing, simulation, and postprocessing steps, each consisting of several tasks (more details appear elsewhere²).

It's difficult to identify the presence of workflows in most of the grid workload traces used in this study – at this level of tracing, we found little data concerning workflows. In a recent study,¹⁰ five scientific workflows covering

astronomy, Earth sciences, and bioinformatics had sizes of tens to tens of thousands of tasks; the same authors reported cases of even larger instances. The sums of task runtimes in these workflows range from hours to weeks, which makes workflows equivalent to long-running grid jobs. As we've recently shown using the GWA-T-15 and GWA-T-16 traces,¹¹ engineering and scientific workflows can have very different characteristics. For engineering workflows, the average number of tasks is in the low tens, with 75 percent of the workflows having at most 40 tasks and 95 percent having at most 200. Tasks in these engineering workflows can be very short, with more than 75 percent of them taking less than 2 minutes to complete. A possible explanation for the small size of engineering workflows is that common grid workflow schedulers incur execution overheads that increase quickly with the size and complexity of the workflow.¹²

Pilot Jobs (BoTs with Many Tasks)

For performance and reliability reasons, pilot jobs install the user's own job management system on the resources provisioned from the grid and then execute (pilot) through this new system a stream or a bag of tasks. Common pilot job tools are Condor (through its glide-in features), Falkon,¹³ and GridBot.¹⁴ For pilot jobs, a common performance metric is throughput (defined as the number of tasks completed per second [tps]); the Falkon system has achieved a throughput of approximately 500 tps, two orders of magnitude better than regular – that is, non-pilot-job-enabled – grid job-management systems.¹³

Currently, no study of a pilot job workload exists. With pilot jobs, grid systems can record jobs that run for days or even weeks; in reality, such jobs run streams of short tasks that can each take anywhere from a few minutes to an hour. Researchers have used GridBot¹³ to execute the workload of a real bioinformatics community through pilot jobs – hundreds to millions of tasks were able to go through per pilot job (stream), and each pilot job averaged approximately 4,000 tasks. These pilot jobs took an average of 0.5 CPU years, with the average task runtime being 15 minutes for medium-sized jobs and between 0.5 and 5 minutes for small-sized jobs; the largest pilot jobs consumed each more than 100 CPU years.

Others

Coallocation – the simultaneous allocation of resources from different grid clusters or even sites for a single grid job – was one of the first grid mechanisms designed for user-specific resource management.¹⁵ Another was malleable allocation – the dynamic allocation and deallocation of resources for single grid jobs. No study of the actual use of either mechanism in real grid workloads exists; however, only about 6,000 coallocated (parallel) jobs exist in Grid'5000, fewer than 2 percent of the jobs recorded in that trace.

Understanding grid workloads is important for tuning existing grids and for designing future grids. With the possibility of grid workloads moving to clouds (several high-performance computing centers are currently installing private clouds for their user communities), this understanding might drive the design and tuning of clouds as well. Ultimately, it may affect the way scientists work and even think about their work.

Our overview of seven years of grid workloads reveals several emerging trends (such as the prevalence of pilot jobs) that require further investigation. Will interdependent, many-task jobs become daily scientific tools? Will parallel jobs see an increase in the number of multicore grid nodes? Will scientists rely on increasingly interactive jobs? New data and new studies of the characteristics and evolution of grid and cloud workloads are required to answer these questions. With the clear trend of increasing the number of resources and application types, new challenges arise in collecting and mining workload data – will new methods and technologies be able to respond? □

Acknowledgments

We thank all the contributors to the Grid Workloads Archive for making their data available.

References

1. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J High Performance Computing Applications*, vol. 15, no. 3, 2001, pp. 200–222.
2. E. Deelman et al., "Workflows and E-Science: An Overview of Workflow System Features and Capabilities," *Future Generation Computer Systems*, vol. 25, no. 5, 2009, pp. 528–540.

3. T. Hey, S. Tansley, and K. Tolle, eds., *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, 2009.
4. A. Iosup et al., "The Grid Workloads Archive," *Future Generation Computer Systems*, vol. 24, no. 7, 2008, pp. 672–686.
5. D. Thain et al., "Pipeline and Batch Sharing in Grid Workloads," *Proc. 12th IEEE Int'l Symp. High Performance Distributed Computing*, IEEE CS Press, 2003, pp. 152–161.
6. A. Iamnitchi, S. Doraimani, and G. Garzoglio, "Filocules in High-Energy Physics: Characteristics and Impact on Resource Management," *Proc. 15th IEEE Int'l Symp. High Performance Distributed Computing*, IEEE CS Press, 2006, pp. 69–80.
7. A. Iosup et al., "The Characteristics and Performance of Groups of Jobs in Grids," *International Euro-Par Conference on Parallel Processing*, LNCS, vol. 4641, Springer Verlag, 2007, pp. 382–393.
8. A. Iosup et al., "The Performance of Bags-of-Tasks in Large-Scale Distributed Systems," *Proc. 17th IEEE Int'l Symp. High Performance Distributed Computing*, IEEE CS Press, 2008, pp. 97–108.
9. D. Kondo et al., "The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems," *Proc. 10th IEEE/ACM Int'l Conf. Cluster, Cloud, and Grid Computing (CCGrid)*, IEEE Press, 2010, pp. 398–407.
10. S. Bharathi et al., "Characterization of Scientific Workflows," *Workflows in Support of Large-Scale Science (WORKS)*, ACM Press, 2008, pp. 1–11.
11. S. Ostermann et al., "On the Characteristics of Grid Workflows," *Integrated Research in Grid Computing*, Springer Verlag, 2008, pp. 431–442.
12. O. Sonmez, H. Mohamed, and D. Epema, "On the Benefit of Processor Coallocation in Multicenter Grid Systems," *IEEE Trans. Parallel Distributed Systems*, vol. 21, no. 6, 2010, pp. 778–789.
13. I. Raicu et al., "Falcon: a Fast and Light-Weight Task Execution Framework," *Proc. 2007 ACM/IEEE Conf. Supercomputing*, 2007, p. 43.
14. M. Silberstein et al., "Gridbot: Execution of Bags of Tasks in Multiple Grids," *Proc. 2009 ACM/IEEE Conf. Supercomputing*, ACM Press, 2009, p. 11.
15. C. Stratan, A. Iosup, and D. Epema, "A Performance Study of Grid Workflow Engines," *Proc. 9th IEEE/ACM Int'l Conf. Grid Computing*, IEEE Press, 2008, pp. 25–32.

Alexandru Iosup is an assistant professor with the Parallel and Distributed Systems Group at the Delft University of Technology, the Netherlands. His research interests include large-scale distributed systems, performance analysis and modeling, and scheduling. He is the co-founder of the Grid Workloads, the Peer-to-Peer Trace, and the Failure Trace Archives, which provide open access to workload and resource operation traces from large-scale distributed computing environments. He is a member of IEEE, the ACM, and IGDA. Iosup has a PhD in computer science from the Delft University of Technology, the Netherlands. Contact him at A.Iosup@tudelft.nl.

Dick Epema is an associate professor with the Parallel and Distributed Systems Group at the Delft University of Technology. His research interests include performance analysis and distributed systems, in general, and grids and peer-to-peer systems, in particular. Epema has a PhD in mathematics from Leiden University, the Netherlands. Contact him at D.H.J.Epema@tudelft.nl.



The image shows the cover of the IEEE journal "Design & Test of Computers" and two special issue covers. The main cover features the title in large red and white text on a blue background. Below it are two smaller covers: one for a special issue on "SoC Design and Test" and another for a special issue on "Verifying Physical Testability of ICs and Systems". To the right of the covers is a text box describing the journal's focus and upcoming topics.

IEEE
Design & Test
of Computers

IEEE Design & Test of Computers covers the tools, techniques, and concepts used to design and test electronic product hardware and supportive software. D&T is a leader in analysis of current and near-future practice.

Upcoming: Emerging Interconnect Technologies, New Directions in DFT, Common Language Framework, and Post-Silicon Calibration and Repair

www.computer.org/design