1 Auction-Based Scheduling for the TeraGrid

The TeraGrid dream Supercomputing centers provide an invaluable national service: They enable large-scale data-intensive and compute-intensive research which could not take place if researchers had to rely on their individual lab resources alone. Examples of such research include the Cosmic Evolution Simulation Project at the Pittsburgh Supercomputing Center (PSC) which uses 1024 processors to simulate billions of simulation years [2]; the Gordon Bell Prize-winning (Earth)Quake project at the PSC, requiring 3000 simultaneous processors [3, 1]; the Protein DataBank at the SDSC [5], which leverages that center's superlative data warehousing infrastructure; and the Telerobotics/Telepresence at the Argonne National Lab (ANL), which would not be possible without the extensive animation/visualization capabilities in place at ANL. The TeraGrid is the NSF's flagship project aimed at integrating the nation's supercomputing resources [4, 10] via a 40 Gbit/sec Extensible Backplane network.

The scheduling nightmare While the NSF has invested hundreds of millions in computing hardware, they have invested very little in understanding how to schedule these powerful resources. Currently, scheduling of jobs at supercomputing centers is largely done in an *ad hoc* fashion, with surprisingly *little automation* and a surprisingly high number of *phone calls* and *human intervention*. Users are not given any guarantee of *queueing delay* at a supercomputing site, which can range from a few hours to two weeks.

There are many difficulties inherent to scheduling in a grid environment which make this scheduling problem extremely challenging. First, supercomputing centers are highly encouraged to follow various NSF mandates arising from different solicitations, such as *giving priority to large jobs* (jobs requiring a large number of processors), or maximizing throughput. To fit in these large jobs requires regular "*drains*" to be scheduled to clear small jobs from the system, so that there is room for the large jobs, which can result in *underutilization* of resources. A second problem is that it is very hard to predict *when* a job will get to run; even under prioritized placement of jobs in a grid, there is a lot of *reordering* since small jobs can be used to fill "holes." Predictability of wait times is made far worse however by the fact that frustrated users can call the center to get their job "bumped up" in the queue, leading to a lot of *randomness* in the ordering of jobs. There is also *no automated reservation mechanism* in place, and *co-scheduling* of jobs across sites is currently done by hand.

The unpredictability described above and lack of a clearly-stated uniform scheduling policy leads to a perception of unfairness, resulting in a loss of potential supercomputing users. As the TeraGrid grows to include new member sites, the inefficiencies present at the existing sites will only be compounded. These issues are stated clearly in the Grid Interest Group's 2005 RAT report [19], as well as the charter for the 2006 TeraGrid Metascheduling RAT [17].

Our proposed 4-year solution The key idea behind our new scheduling concept for the TeraGrid is the use of a *market-based auction mechanism*, creating a *uniformly fair* environment, whereby all users have the same opportunity to *reveal* which jobs are actually time-critical, thus eliminating "squeaky wheel" effects. We propose to run an auction every 48 hours, where users bid using virtual tokens for a scheduling slot. In contrast to standard auctions, our procedure will allow the prioritization of users not only by the number of tokens bid, but also, for example, by the size (# processors) of the job, or any other NSF mandated criteria. Within a priority level (size grouping),

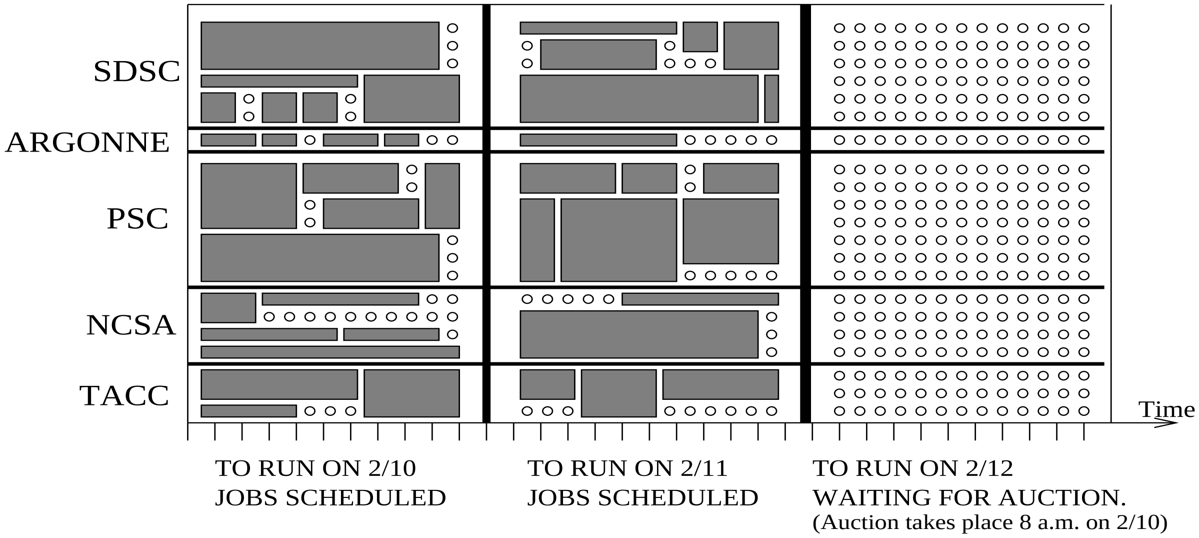

Figure 1: *Proposed interface seen by users when scheduling their jobs on the TeraGrid.*

we prioritize jobs by “best” bid first.

There are many benefits to using an auction. First, the bidding process gives users a mechanism for revealing the true importance of their job. Users can *prioritize* their jobs by time-criticality by bidding more tokens for these jobs. Second, our auction procedure will greatly improve *efficiency/utilization* of resources by dictating that a *drain* is run every 48 hours, in a planned way, allowing the use of mathematically efficient bin-packing algorithms to maximize utilization of processors in the hours just before the drain. Furthermore, our auction mechanism allows for the most efficient re-packing of jobs when jobs end early. Our auction procedure will also greatly increase user satisfaction. Because our system will generate schedules every 48 hours, the start times of the winning jobs will be entirely *predictable*. Our scheme will also enable *reservations* for future slots by establishing a limited market for future processing capacity. All users will know the auction procedure and rules, so our system will provide users with *transparency*. Both users and the supercomputing centers will also greatly benefit from a system that is *automated*. Finally, our auction scheme will also be beneficial to the TeraGrid as a whole. Use of a single, automated, resource allocation algorithm will achieve *standardization* across TeraGrid member sites. Similarly, users will be able to bid for multiple bundled resources at the same time, which will elegantly enable *co-scheduling* of their jobs across sites, without human intervention. Lastly, our framework allows us to extend the use of supercomputing facilities beyond batch jobs to allow for more interactive, experimental users.

We have a very extensive plan for a 4-year implementation, involving live test users on a subset of the PSC processors. Our project is novel in its incorporation of auction mechanism design, user behavior studies, workload characterization, bin-packing/scheduling algorithms, and queueing theory applied to a supercomputing grid environment. Our team includes: *David O’Hallaron* (CMU, CS), a daily user of the PSC for his Gordon Bell Prize-winning earthquake experiments; *Vincent Conitzer* (Duke, CS and ECON), a specialist in auction scheduling; *Sergiu Sanielevici* (PSC), head of the TeraGrid Scheduling RAT 2005 report and member of the 2006 TeraGrid Metascheduling group; *Alan Scheller-Wolf* (CMU, Tepper School), an expert in queueing theory and performance modeling; and *Avrim Blum* (CMU, CS), an expert on machine learning and bin-packing.

How we differ from prior work While previous research has proposed a number of market-based approaches to grid scheduling[6, 7, 15, 24, 25, 20, 8, 13, 18, 11, 14, 27, 23, 22, 21, 9, 16, 12, 26], these are not appropriate for scheduling *supercomputing* resources. First, in grid computing, the supply of computing resources is generally not fixed, and sellers of resources must be compensated for making them available. Thus, the appropriate model is that of an *exchange* in which both buyers and sellers participate. By contrast, in the supercomputing setting, resources are fixed, and their owners do not need to be compensated by users. Here, the appropriate model is that of a (potentially combinatorial) auction. The supercomputing setting is also unique because of NSF-mandated prioritizations, for example prioritization in favor of large jobs. Finally, prior work does not incorporate supercomputing job characteristics and workload distributions, does not address the need for drains, and does not allow for advanced reservations and for co-scheduling across sites.

Funding Our work was awarded seed money via a *2007 Microsoft Breakthrough Research* grant. We are seeking larger grants from NSF and DARPA.

References

[1] Carnegie Mellon Researchers and Pittsburgh Supercomputing Center win Prestigious Gordon Bell Prize for High Performance Computing. http://www.psc.edu/publicinfo/news/2003/2003-11-26_bell.html.

[2] Cosmic Simulation Article. http://www.hpcwire.com.breaking/1593.html.

[3] Inside Lemieux, the 1994 Northridge Earthquake Shakes Almost Like the Real Thing. http://www.psc.edu/science/2003/earthquake/big_city_shakedown.html.

[4] The TeraGrid Website. http://www.teragrid.org.

[5] Protein Data Bank, http://www.sdsc.edu/Press/03/01.25.04_pdb.html.

[6] R. Bapna, S. Das, R. Garfinkel, and J. Stallaert. A market design for grid computing. *INFORMS Journal of Computing*, Forthcoming.

[7] F. Berman and R. Wolski. The apples project: A status report. In *8th NEC Research Symposium, Berlin, Germany*, 1997.

[8] F. Berman, R. Wolski, S. Figueira, J. Schopf, and G. Shao. Application-Level Scheduling on Distributed Heterogeneous Networks. In *Proceedings of Supercomputing '96*, Pittsburgh, PA, 1996.

[9] R. Buyya. Economic-based distributed resource management and scheduling f or grid computing. PhD Dissertation, Monash University, 2002.

[10] C. Catlett. The TeraGrid: A Primer. www.teragrid.org/about/TeraGrid-Primer-Sept-02.pdf, September 2002.

[11] W. Cirne and F. Berman. Application scheduling over supercomputers: A proposal. Technical Report CS1999-0631, July 1999.

[12] A. Dogan and F. Ozguner. Scheduling Independent Tasks with QoS Requirements in Grid Computing with Time-Varying Resource Prices. In *GRID 2002 - Third IEEE/ACM International Workshop*, pages 58 – 69, 2002.

[13] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.

[14] A. Geweke. A system for batch-mode economic scheduling of a cluster of workstations. Master's Thesis, UC-Berkeley, 2001.

[15] K. Krauter, R. Buyya, and M. Maheswaran. A taxonomy and survey of grid resource management systems for distr ibuted computing. *Journal of Software Practice and Experience*, 32(2):135–164, 2002.

[16] S. Lalis and A. Karipidis. JaWS: An Open Market-Based Framework for Distributed Computing over the Internet. In *GRID 2000 - First IEEE/ACM International Workshop*, pages 36 – 46, 2000.

[17] Metascheduling Requirements Analysis Team. User Requirements, RP Policies and Metascheduling Technologies – Charter. Available at: www.teragridforum.org/mediawiki/images/4/43/MetaSchedRatCharter-Final.doc, August 2006.

[18] R. Raman, M. Livny, and M. H. Solomon. Matchmaking: Distributed resource management for high throughput co mputing. In *HPDC*, pages 140–, 1998.

[19] Scheduling Requirements Analysis Team. NSF Extensible TeraGrid Facility – Final Report. Available at: www.teragridforum.org/mediawiki/images//005/Sched-rat.pdf, April 2005.

[20] I. Stoica, H. Abdel-Wahab, and A. Pothen. A Microeconomic Scheduler for Parallel Computers. In *Proceedings of the IPPS '95 Workshop on Job Scheduling Strate gies for Parallel Processing*, pages 122–135, April 1995.

[21] P. Tucker. Market mechanisms in a programmed system. Working Paper, UC-San Diego.

[22] P. Tucker and F. Berman. On market mechanisms as a software technique. Technical Report CS96-513, UC-San Diego, 1996.

[23] S. VADHIYAR and J. DONGARRA. A metascheduler for the grid. In *Proceedings of the 11th IEEE Symposium on High-Performance Distributed Computing*, 2002.

[24] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. rey O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18(2):103–117, 1992.

[25] M. P. Wellman. A market-oriented programming environment and its application to di stributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.

[26] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan. Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications*, 15(3):258–281, 2001.

[27] D. Wright. Cheap cycles from the desktop to the dedicated cluster: Combining opportunistic and dedicated scheduling with Condor. In *Proceedings of Linux Clusters: The HPC Revolution*, Champaign-Urbana, IL, 2001.