

Research Statement

Daniel Nurmi

Large scale distributed computational architectures are more ubiquitous today than ever before, and are rapidly becoming a key element of next-generation computer systems. In the mainstream, novel Internet services are being developed on data warehouse architectures (Google) and new business models are being supported by so-called cloud computing systems (Amazon EC2). In the scientific computing arena, projects such as the TeraGrid represent a continuing interest in wide area distributed scientific computing at a national level. These architectures are highly heterogeneous, composed of hundreds of thousands of parallel, desktop, network and storage components. Due to their rapidly increasing complexity, brought upon by an ever growing number of individual, coordinated components, we have found that these architectures are exhibiting unprecedented levels of dynamism in their network, computational and storage systems, often leading to poor overall performance, availability and usability. In order to mitigate the negative effects of this dynamism, we propose that system software services that can be incorporated into large scale applications, middleware and operating systems is needed. We have found that the application of traditional, parametric modeling and prediction techniques is becoming increasingly difficult due to the sheer number of individual components that must be parameterized and modeled, particularly when the goal is to provide system software that must be both on-line and highly responsive.

Thus, I believe that a new approach to the observation, analysis and mitigation of large scale system complexity is needed. My research takes the perspective of treating modern large scale computational systems as entities that can be studied and treated phenomenologically. That is, instead of attempting to model every component of systems that can be simulated in a laboratory environment, my approach has been to measure and observe existing computer systems, analyze their response statistically, and propose solutions to the problems of resource behavior characterization and prediction. My research efforts have thus focused on the observation of high performance distributed system behaviors, identification of existing problems with performance, reliability, and usability of these systems, performing statistical analysis of resource response. The goal of my research efforts has been to provide software services that next generation operating systems can incorporate to help them manage the problems of scale. In this light, I have explored high performance system design and implementation, resource performance forecasting and monitoring systems, wide area application workflow scheduling, application load balancing and resource failure analysis and prediction.

1 Batch Queue Delay Prediction

In order to manage demanding user workload, most HPC centers today employ some form of batch queuing software to which users submit jobs that await execution until resources become available. While users typically have an accurate notion of how long their jobs will execute once they start, there was no satisfactory methodology for predicting how long their jobs would wait in queue before they began execution. This delay is both significant (hours or days) and highly variable (six to seven orders of magnitude when measured in seconds). The inability to predict when, from the time a job is submitted to a batch queue until it finishes execution, has been a significant problem for over a decade. While a large amount of work had been done in this area, most attempts to provide an accurate model of batch queuing systems, schedulers, and/or user behavior. We have found that it is difficult if not impossible to employ these typically parametric models to build system software that can be used, in real time, by HPC middle-ware and operating systems.

We have taken a different approach to the problem, with great success. First, we observe that for most of the problems queuing delay imposes on users and automated scheduler systems, a *precise* prediction of queue delay would be ideal, but is often more information than is required. For example, a hurricane forecasting application does not need to know exact time in the future when it will complete, but rather the *latest* time that it will finish in order for the results to be meaningful. If the job finishes early, there is no detriment; finishing late, however, is unacceptable. Thus, we have created a methodology to predict statistical upper bounds on future batch queue wait time that is composed of four interacting fundamental components; a batch queue job data measurement system, a non-parametric quantile predictor, an empirical on-line time series change-point detector, and a model-based job clustering algorithm. The methodology fundamentally relies on historical traces of user job data on real production HPC systems, which we gather in real time from over 20 super-computers around the world. With this data, we apply our novel non-parametric quantile predictor to the traces, giving us predictions of future population quantiles using a small number of empirical observations. Further, we note that batch queue delay prediction traces not only exhibit highly variable behavior within a narrow time band, but also shows drastic changes in behavior at certain points in time. Based on the level of autocorrelation found in the recent history, we determine when an improbable number of consecutive events have occurred, at which point we throw away any data in the time series observed before the flagged *change-point*. Finally, we observe that we have extra information about jobs that we believe influences their wait times; the number of processors and the amount of time the job requests. Intuitively, a job requesting one processor for ten seconds will most likely wait in queue for less time than a job requesting two thousand processors for a day. To take advantage of this extra information, we employ an automatic model-based clustering algorithm that determines an optimal number of clusters (using the Bayesian information criteria) as well as those clusters' ranges (using parametric statistical modeling). When used in concert, these four

components form a powerful methodology that we have shown to accurately and correctly predict statistical queue delay bounds for individual jobs [1, 2, 4], and can improve scientific application performance by a factor of three or more [5] when integrated into a wide area distributed computing workflow scheduler.

Having invented, extensively tested and verified our novel methodology with great success, we felt compelled to go one step further and provide our solution to the user community at large. The batch queue wait time prediction system [7] has been extremely well received in the scientific computing community (over 2000 predictions made per day from 50 to 100 unique sites), illustrated by the fact that I am now responsible for monitoring the batch queues of over twenty supercomputers around the world. In addition, it has been integrated into several production and research software efforts, and is an influential component of a number of graduate student projects and publications.

2 Resource Failure Mitigation

Another research area that I have explored is the development of methods to manage the problem of resource and software failures that are becoming increasingly significant as system sizes grow and failure rates skyrocket, with the goal again being to eventually provide the ability to use any successful resource availability prediction methodologies as the basis for new system tools.

The preponderance of resource failures in large scale distributed systems is a major impediment to users wishing to execute highly distributed applications. Augmenting the Network Weather Service (NWS) measurement infrastructure, I have developed a methodology for gathering data, based on the idea of combing through existing data repositories such as local logs and accounting databases and automatically extracting and storing the data of interest. In [3] we develop techniques to automatically fit and test statistical models to a subset of the resource availability data and show that many previously assumed models (exponential and Pareto distributions) of failure behavior are inferior to slightly more complex but still very usable alternatives (Weibull distribution). Based on this finding, we show in [6] that the use of our automatic model fitting system provides a better solution to the problem of finding optimal checkpoint intervals for distributed applications than when we rely on simpler model assumptions.

This work has had a significant impact, prompting a number of colleagues in the high performance computing field to inquire as to whether the same data gathering infrastructure and modeling techniques could be applied to high performance and super-computing resources. Toward this end, I have been working on the deployment of resource availability sensors on super-computers distributed throughout the world, and have made available a web service interface to real time failure model information; a service that was recently utilized in emerging technology demonstrations at SuperComputing 2007, using several of the NSF TeraGrid core super-computers to execute a fault tolerant version of the Linked Environments for Atmospheric Discovery (LEAD) weather forecasting application.

3 Future Research Directions

The research that I've discussed thus far are representative of how I plan to conduct future research projects. They begin with an observation that there is an obvious need, we carefully observe and study the relevant entities, invent novel characterization and forecasting methodologies, and finally test, verify and implement our solutions in the form of usable system software. Along the way, we reveal exciting possibilities and questions, publish our findings in top conferences, and create an environment where real users interact with our research solutions, providing us a valuable source of data to be used in future research.

In particular, I'm interested in starting new research within the context of *cloud computing*, since it is an exciting new technology with great potential, and that it provides a rich environment to within which many aspects of computer science systems research can be explored. National labs, universities and companies are installing "clouds" of resource pools that are similar to standard compute clusters, but provide the extra feature of OS level virtualization that can be used to, among other things, dynamically grow and shrink the resource pools available to applications at runtime. While management and control technology is being developed for this new style of architecture, there has been very little work done regarding the programming models, software APIs, and runtime methodologies that will be needed once these systems are in place. I believe my work and research perspective is uniquely suited to this new environment. Specifically, I believe that exploring predictive methodologies for triggering resource pool expansion/contraction at run-time is interesting and would draw heavily from the network modeling and performance evaluation fields. I'm also interested in new parallel/distributed programming models that this architecture will encourage, especially given that many of the novel features of cloud computing architectures provide malleable levels of parallelism at runtime. Cloud computing will be relying on OS virtualization, and so I am interested studying predictive methodologies for determining when and how best to distribute virtual OS images within a cluster, an idea that has direct operating systems implications. I believe that the future of computational systems relies on ever increasing numbers of individual components, increased parallelism from single chips to Internet scale, faster networks, and more dynamic software. My primary research agenda is to illuminate, address and solve the problems and scientific questions that this oncoming trend imposes.

References

- [1] J. Brevik, D. Nurmi, and R. Wolski. Quantifying machine availability in networked and desktop grid systems. In *Proceedings of CCGrid04*, April 2004.
- [2] J. Brevik, D. Nurmi, and R. Wolski. Predicting bounds on queuing delay for batch-scheduled parallel machines. In *Proceedings of PPOPP 2006*, March 2006.

- [3] D. Nurmi, J. Brevik, and R. Wolski. Modeling machine availability in enterprise and wide-area distributed computing environments. In *Proceedings of Europar 2005*, August 2005.
- [4] D. Nurmi, J. Brevik, and R. Wolski. QBETS: Queue bounds estimation from time series. In *Proceedings of 13th Workshop on Job Scheduling Strategies for Parallel Processing (with ICS07)*, June 2007.
- [5] D. Nurmi, A. Mandal, J. Brevik, C. Koelbel, R. Wolski, and K. Kennedy. Evaluation of a workflow scheduler using integrated performance modelling and batch queue wait time prediction. In *Proceedings of SC06*, November 2006.
- [6] D. Nurmi, R. Wolski, and J. Brevik. Model-based checkpoint scheduling for volatile resource environments. In *Proceedings of Cluster 2005*, September 2004.
- [7] The nws batch queue prediction home page – <http://nws.cs.ucsb.edu/batchq>.