ADAPTING GRID SERVICES FOR URGENT COMPUTING ENVIRONMENTS

Jason Cope and Henry M. Tufo
Department of Computer Science, University of Colorado, 430 UCB, Boulder, CO, 80304-0430, U.S.A.

Keywords: Data Grids, Urgent Computing, Grid and Web Services, Service-Oriented Architectures.

Abstract: Emerging urgent computing tools can quickly allocate computational resources for the execution of time critical jobs. Grid applications and workflows often use Grid services and service-oriented architectures. Currently, urgent computing tools cannot allocate or manage Grid services. In this paper, we evaluate a service-oriented approach to Grid service access and provisioning for urgent computing environments. Our approach allows resource providers to define urgent computing resources and Grid services at a much finer granularity than previously possible. It accommodates new urgent computing resource types, requires minimum reconfiguration of existing services, and provides adaptive Grid service management tools. We evaluate our service-oriented, urgent computing approach by applying our tools to Grid services commonly used in urgent computing workflows and evaluate management policies through our urgent service simulator.

1 INTRODUCTION

Recent research into urgent computing systems has improved several dynamic data-driven workflows that perform emergency computations. The goal of urgent computing is to provide a cohesive infrastructure to support time-critical computations. Examples of these applications and workflows include severe weather forecasting workflows such as those provided by Linked Environments for Atmospheric Discovery (LEAD) (Droegemeier et al., 2004), the Southern California Earthquake Center's (SCEC) TeraShake earthquake simulation applications (Cui et al., 2007), the Southeastern Universities Research Association (SURA) Coastal Ocean Observing and Prediction (SCOOP) storm surge modeling applications (Bogden et al., 2007), epidemic transmission simulations using tools like Virginia Tech's EpiSims application (Eubank et al., 2006), and the Data Dynamic Simulation for Disaster Management project's Coupled Atmosphere-Fire (CAF) wildfire forecasting workflow (Mandel et al., 2007).

The LEAD and SCOOP projects use the Special PRiority and Urgent Computing Environment (SPRUCE) to obtain high-priority access to the shared computing resources available on the TeraGrid (Catlett et al., 2007), a distributed supercomputing environment in the US similar to DEISA (Lederer et al., 2007). SPRUCE provides project users with elevated and automated access to TeraGrid computational resources so that high-priority applications run immediately or as soon as possible (Beckman et al., 2007). SPRUCE currently provides capabilities for allocating computational resources but does not yet support urgent storage or data management capabilities. Urgent storage and data management capabilities provide cohesive infrastructure for prioritized usage of storage resources, such as file systems, data streams, and data catalogs. Since several of these applications, such as LEAD and TeraShake, have significant data requirements, providing urgent storage and data management is an essential and currently absent capability for urgent computing workflows. Supporting end-to-end urgent computing workflows requires support for common data capabilities, such as data storage, access, search, and manipulation capabilities, required by these workflows.

We are developing an Urgent Data Management Framework (UDMF) to address the data requirements of urgent computing workflows. This framework will manage data related tasks and provision storage resources for urgent workflow usage. Several urgent computing workflows use service-oriented architectures for resource and application integration. These application deployments typically operate in multiple modes and share services between project deploy-

Cope J. and M. Tufo H. (2008).
ADAPTING GRID SERVICES FOR URGENT COMPUTING ENVIRONMENTS.
In *Proceedings of the Third International Conference on Software and Data Technologies - PL/DPS/KE*, pages 135-142
DOI: 10.5220/0001893301350142

ments. While the current SPRUCE infrastructure can negotiate access to compute resources for these systems, it does not support similar capabilities for the Grid services utilized by these workflows. In order to support end-to-end urgent workflow scheduling and data management, the Grid services that provide data capabilities to these workflows should also support authorization and usage capabilities similar to other urgent computing resources.

In this paper, we describe our use of monitoring, authorization, and provisioning infrastructure to provide urgent computing capabilities to Grid services. We provide these capabilities by leveraging past research on Grid authorization and monitoring infrastructure and using Service Level Agreements (SLA) for establishing and sustaining Quality of Service (QoS). Providing these urgent Grid service capabilities allows resource providers to directly apply urgent computing policies to Grid services. These policies can indirectly provision resources used by a Grid service. This feature is especially useful for providing urgent computing capabilities to software or resources that can not directly support urgent computing capabilities. Our approach is reusable by other Grid services and requires minimal reconfiguration of existing services.

The remainder of this paper describes our current work on urgent Grid services. Section 2 describes the urgent computing paradigm and the current urgent computing infrastructure. Section 3 describes related areas of research. Section 4 describes our urgent service provisioning and policy management framework. Section 5 details our initial evaluation of this framework through its use by Grid resource management and data services. In the final sections of this paper, we present future work and conclusions.

2 BACKGROUND

2.1 The State of Urgent Computing

The Special PRioirty and Urgent Computing Environment (SPRUCE) (Beckman et al., 2007) enables on-demand resource allocation, authorization, and selection for urgent computing applications and workflows. This environment provides on-demand access to shared Grid computing resources with a token-based authorization framework. By utilizing shared resources, SPRUCE allows data centers and virtual organizations to utilize existing computing infrastructure for emergency computations instead of allocating dedicated resources for these tasks. Users submitting SPRUCE jobs specify a color-coded urgency parameter with their job description. SPRUCE authorizes the urgent job execution request by verifying that a user is permitted to execute jobs with the specified urgency on the target resource. Each resource provider defines policies for how the urgent tasks are handled on a per-resource basis. For example, a resource provider may choose to preempt non-urgent jobs for high-priority tasks or to give the urgent tasks next-to-run privileges. The infrastructure is currently deployed on several TeraGrid resources, including resources at NCAR, UC/ANL, SDSC, and TACC, and is used by several Grid computing workflows, such as LEAD and SCOOP.

While SPRUCE provides access to computational resources, it does not provision or negotiate access to other common Grid services and capabilities, such as data management services, resource management services, and Grid credential management services. To completely support urgent computing in Grids, these other services must be supported so that urgent workflows do not bottleneck against these unmanaged services.

2.2 Service Requirements of Urgent Computing Workflows

Recent service-oriented systems have provided many capabilities for users to integrate into their workflows. For example, LEAD provides a large set of services hosted across the project partner sites. The available services include data management services that can assimilate data, reference metadata catalogs, stream data from remote sensing equipment, and execute data mining operations. LEAD leverages the core Globus Grid services (Foster, 2005), such as the Grid Resource Allocation and Management (GRAM) service, to interact with computational and storage resources. Similar to LEAD, SCOOP provides a series of data services exposed as Grid and Web Services. These services provide capabilities to access catalog data, transform data into compliant formats, and visualize data sets. Like LEAD, SCOOP implements these services as Web or Grid services.

Since one of the benefits of Web and Grid services is software reuse, these services are used by projects and users other than SCOOP or LEAD. SCOOP has stated that their tools are available for community use. Furthermore, these systems also provide multiple operational modes and use cases. Not only is LEAD capable of forecasting severe weather events such as tornadoes, it is also used as an educational and training tool. The combination of multimodal use and the lack of urgent computing infrastructure to provision these services necessitates infrastructure to manage

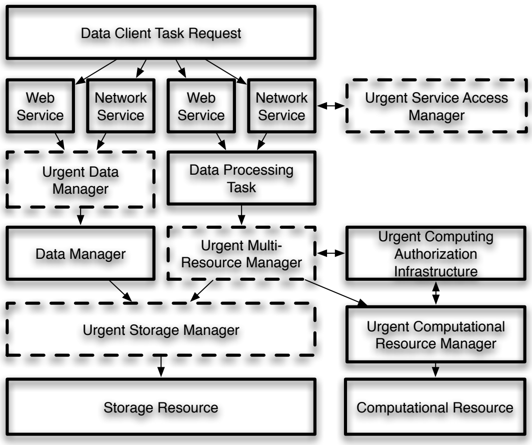

Figure 1: UDMF architecture.

and provision access to these services. In the event of an urgent workflow, it is not acceptable for the workflows performance to be degraded because of congestion at these services and the resources they manage.

2.3 Urgent Data Management

We are developing an Urgent Data Management Framework (UDMF) that provides the necessary data management capabilities for urgent computing workflows (Cope and Tufo, 2008). This framework provides users with priority based access to data capabilities used in urgent computing workflows. These tools provide storage and network resource provisioning, autonomic resource management, and resource management policies that address urgent computing data requirements. Figure 1 illustrates the components and interactions of UDMF. UDMF leverages existing infrastructure (illustrated as the solid components in Figure 1) when possible. UDMF contains several software managers that can execute the resource management policies and negotiate urgent user access to the data resources. The UDMF components are illustrated as the dashed components in Figure 1. Many of the existing software components are extensible, such as the Data Storage Interface to GridFTP (Allcock et al., 2002) or the Authorization and Authentication Framework in the Globus Web service container.

The focus of this paper is on the Grid service components of UDMF. These Grid service specific UDMF components provide monitoring and instrumentation for Grid service invocations, user authorization consistent with other urgent computing tools, and adaptive resource provisioning tools and polices to manage user access and interaction with Grid services.

3 RELATED WORK

The research we present in this paper draws upon the experiences from several related areas of work, including Grid security, Service Level Agreements (SLA) for Grid services, and Grid Quality of Service (QoS). Several authorization and authentication tools exist for use in Grids. The Globus Toolkit provides the bulk of the authorization capabilities required (Foster, 2005). We use the Globus Policy Decision Point (PDP) and Policy Information Point (PIP) authorization chains to negotiate access to Grid services during the invocation of a service operation. Several other projects use the same authorization infrastructure for integration of security policies and authorization tools. GridShib integrates Shibboleth within the Globus authorization frameworks using custom PDP and PIP infrastructure (Chadwick et al., 2006; Lang et al., 2006). The Virtual Organization Management System (VOMS) also provides similar capabilities to GridShib using the Globus authorization and authentication framework (Alfieri et al., 2003).

In this paper we leverage past work on Web service QoS and Service Level Agreements (SLA) techniques for supporting urgent computing environments. In the context of service-oriented architectures, our work on Grid service provisioning is most similar to past research on SLAs. The use of SLAs is still an active area of research, but they have not yet been used to support urgent computing applications and resources. The infrastructure to support SLAs in UDMF is similar to other SLA management tools that use brokers to advertise, negotiate, and establish SLAs between clients and services (Dan et al., 2004).

Grid QoS research is often associated with SLA research. QoS monitors typically observe the properties of systems and SLA brokers use these observations to determine attainable and sustainable QoS for a specific SLA. Several projects have used this model for Grid systems (Liu et al., 2004; Wang et al., 2005; Truong et al., 2006; Al-Ali et al., 2004). Our work is similar to request prioritization and QoS differentiation techniques that adapt service executions to priority-based policies (Sharma et al., 2003; Zhou et al., 2007; Benkner et al., 2007; Erradi and Maheshwari, 2007). Other tools, such as the Network Weather Service (NWS), use collected observations to forecast expected QoS for a system (Wolski et al., 2005; Nurmi et al., 2007).

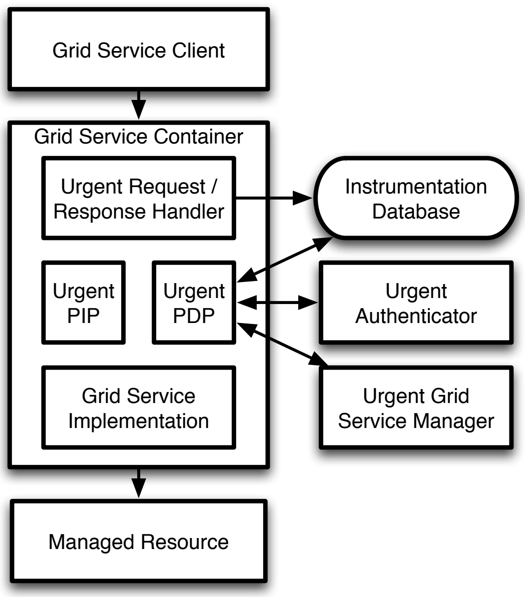

Figure 2: Implemented urgent service authorization framework.

4 IMPLEMENTATION OF THE URGENT SERVICES FRAMEWORK

Our urgent Grid service framework consists of information services to communicate the state, capabilities, and policies associated with available urgent services. It also monitors the usage of urgent Grid services and other services collocated with an urgent Grid service. We provide authorization infrastructure to determine the level access to a service a user is permitted. Figure 2 illustrates these components and their interactions.

4.1 Advertising and Monitoring Urgent Grid Services

Information services are a critical component of service-oriented architectures so that end-users or other software components can recognize the properties and capabilities of deployed services. To help identify available urgent computing services hosted on a resource, we developed a mechanism to register and describe urgent computing services with a local Globus Monitoring and Discovery System v4 (MDS4) (Schopf et al., 2006). A Perl script produces XML descriptions of supported urgent services that describe service usage policies. This information can be used in conjunction with QoS information to model the behavior of the service under current or expected conditions. These tools provide valuable information for time critical applications. It is essential for end users to determine their authorized urgency level so they can select the service that best accommodates their needs. The registration of a Grid service's urgency parameters with information services allows end users to verify service properties before a service is used.

Grid service invocations are monitored through our tools. All Grid service interactions, regardless of their urgency, are monitored. We extended a Grid service message handler distributed with Globus WS CORE to log Web service communication into our instrumentation database. We capture several fields form each SOAP header sent or received by the service container, including the messages UUID, the target service address, the service operation, the caller's Distinguished Name (DN), the UUID of a related message, and the time the message was delivered to our message handler. This handler is installed on the service containers input and output message handler chain so that every SOAP message is logged. This information is used by our policy and service management tools to adapt policies based on current service usage characteristics.

4.2 Urgent Grid Service Authorization

The framework that authorizes urgent access to Grid services consists of custom Globus Grid service authorization framework that integrates with urgent computing authorization tools, such as SPRUCE. Our authorization infrastructure consists of a custom Policy Decision Point (PDP) and a custom Policy Information Point (PIP) that supports urgent access to these services. The PDP is responsible for authorizing client access to Grid services and communicates with our service management tools. When the PDP evaluates the incoming service request, the DN for the user invoking the service is verified and the appropriate urgency is applied to their invocation of the service. This verification occurs through execution of the Urgent Authenticator, which is similar to the SPRUCE Token Authentication script. Once the urgency level has been verified, the PDP contacts the Urgent Service Manager. The Urgent Service Manager determines which policy to execute for the service and returns this information to the PDP. The PDP then executes the appropriate policy and permits the service to execute.

4.3 Policy Enforcement and Adaptation

The Urgent Service Manager is responsible for determining which service management policy associates with a particular user and urgency level. Once the appropriate policy is identified and adapted to the current operating environment, the Urgent Service Manager communicates the policy to the Urgent PDP. The Urgent Service Manager can simulate the execution of Web services. We integrated the SimPy discrete event simulation package into the Urgent Service Manager client so that we can quickly simulate and experiment with anticipated Web service requests flows. We envision that the service management and simulation components can integrate with other autonomic computing tools to foster an automated, self-managing, urgent computing environment.

We have currently defined and implemented three service management policies:

- Unmanaged. Requests are permitted as they are received regardless of the urgency of the request.
- Exclusive. Only the most urgent requests are allowed. This policy is similar to strict priority scheduling.
- Shared. A mix of requests are permitted at variable rates per urgency level.

The unmanaged policy allows all requests to complete. The exclusive access policy only permits the most urgent requests and stalls all other requests until the urgent requests complete. The stalled requests are not denied so that concurrent, less urgent workflows do not fail. The shared policy allows multiple levels of urgent requests to execute and arbitrates the frequency that the requests are permitted to execute. Like the exclusive access policy, requests are not outright denied but are stalled to achieve allowable service execution frequencies. Our PDP throttles service requests at the authorization point and shapes the Grid service request traffic.

Each of the defined policies has several configuration parameters. For example, all policies have an execution window parameter to define how long the policy should remain active since the last request of that policy occurred. This can block non-urgent requests if more urgent requests are expected. The shared policy defines traffic percentages and rates for each urgency level. This parameter prevents non-urgent traffic from starving urgent traffic. Multiple policies can be mixed per service. For example, the most urgent policy for a service can implement an exclusive policy for a specific user while the other policies can be applied to other users.

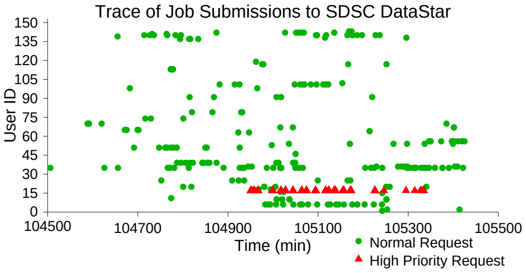

Figure 3: SDSC DataStar trace illustrating jobs submitted per user over a 16 hour interval.

5 EVALUATION OF UDMF GRID SERVICES

Our evaluation of this service provisioning infrastructure occurred in two phases. In the first phase, we evaluated service provisioning and request prioritization in our service management simulator. In the second phase, we deployed our service provisioning and management tools into the Globus Web service container and evaluated the capabilities of the service management tools with several services.

5.1 Policy Simulation

We performed more detailed analyses of our provisioning software using our simulator and data from the Parallel Workloads Archive (Feitelson, 2008a). We used the trace of jobs submitted to SDSC's DataStar TeraGrid computational resource from March 2004 to April 2005. For our evaluation, we assume that all jobs submitted to DataStar were using the Globus WS GRAM Grid service so that we could approximate a workload trace for a Grid resource using Grid services. WS GRAM manages Grid applications running on Grid computational resources and invokes other Grid services to transfer data between Grid resources. Figure 3 illustrates job submission times for a segment of the trace. Over the 16 hour period of the trace, 1010 jobs were submitted to DataStar from 40 different users. From this trace, we studied the performance and impact of our software by varying the urgency of users within the trace. As an example, when user 17 has high priority and exclusive access to the resource while all other users have low priority access, our software limits the low priority requests as illustrated in Figure 4. In this example, the window parameter of the high priority policy was set to one hour and prevented low priority tasks from executing a service within one hour of the last high priority request. Low priority tasks were also limited to four

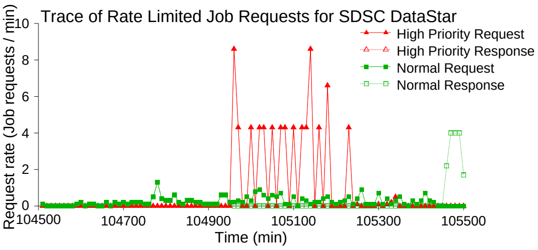

Figure 4: SDSC DataStar trace illustrating the effects of urgent request throttling on job throughput over a 16 hour interval.

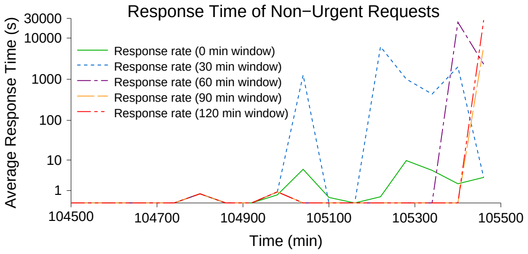

Figure 6: Response time of normal priority SDSC DataStar traffic delayed by higher priority traffic.

service accesses per hour for all users.

The parameters that define higher priority policies impact the traffic throughput of the lower priority policies. Figure 5 illustrates how the window parameter of the high priority policy affects all lower priority Grid service traffic. Expanding or contracting the high priority window affects what traffic is permitted to execute. Smaller windows allow lower priority requests to execute closer to the original service request. For example, a non-urgent traffic stream executed concurrently with a high priority request configured with a 30 minute window exclusive access policy. Lower priority stream interleaved the higher priority request stream. Larger windows delay non-urgent requests longer and can prevent the interleaving behavior of the shorter windows.

While the window parameter can reserve access to the service if more requests are expected, it degrades the response time of the lower priority requests. Figure 6 illustrates the response time of low priority request over the hourly intervals in the simulation. Without a window, small delays of less than one second still occur since the high-priority service requests have exclusive access to the service until the request completes. As the window values increase in size for this simulation, the response time for the lower priority requests increases as the requests execute later in the simulation. Adapting the request scheduling parameters, such as the access window, to urgent workflows is necessary so that the services are still highly utilized by all priorities.

Configuring the policies is left to the service hosting site. Since workloads and service usage patterns can change as the workflows change, we are developing tools to automatically configure these policy parameters. We have begun to implement several service request rate, load, and usage pattern forecasters within the Urgent Service Manager. These forecasters use time-series analysis techniques (Feitelson, 2008b) and more recent forecasting techniques (Nurmi et al., 2007) to predict the expected usage or response of a Grid service for urgent computing workflows. These prediction tools are still under development, but the initial results from these tools for tuning policy parameters are promising.

5.2 Deployment Experiences

To evaluate our urgent service provisioning framework, we integrated our tools into several services, evaluated our service models and provisioning policies with our simulator, and performed end-to-end tests using our authorization and provisioning tools on several deployed services. We first integrated our authorization framework into several services, including the Globus DefaultIndexService, a data catalog service that utilizes the Globus Replica Location Service (RLS), and a database query Grid service. With all of these services, we were able to control access to the Grid service with only minimal configuration. The reconfiguration of the services included adding the PDP and PIP to the service security configuration so that the priority of the request could be validated. The service manager was also configured so that the priority validation requests from services running in the container could check and schedule access to the service.

6 FUTURE WORK

We are currently developing forecasting tools to automatically tune the service policies. This automatic tuning and configuration work leverages techniques that mine for patterns in Grid service usage and forecast expected Grid service behavior or load. The work we present in this paper does not address supporting non-service oriented access to urgent data resources and we leave this topic for future work. Additionally, we are developing an autonomic data management system to adapt and reconfigure storage managers for urgent data requirements.

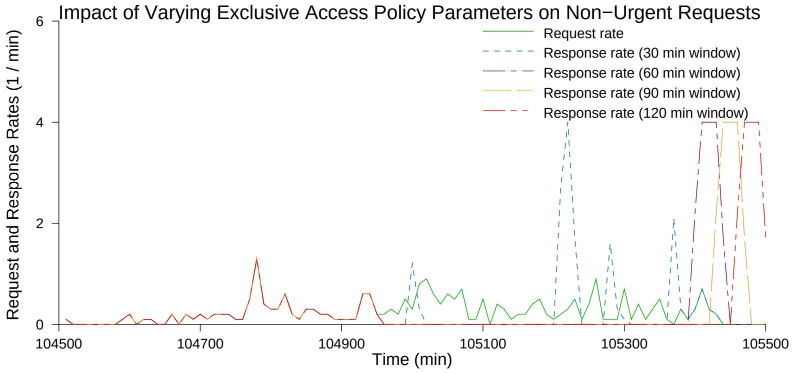

Figure 5: SDSC DataStar trace illustrating the effects of urgent policy parameters on non-urgent request traffic.

7 CONCLUSIONS

We presented our approach to provisioning Grid services for urgent computing applications and workflows. Our approach requires minimal service reconfiguration for urgent use cases and can support a variety of resources types managed by Grid services. We evaluated our service management tools using the core Globus data and resource management Grid services. We demonstrated that these services can adapt and respond to dynamic urgent computing requirements using our service and policy simulator. The current scheduling parameters for urgent services can provide near immediate access to these services at the cost of increased response times for lower priority service requests.

ACKNOWLEDGEMENTS

Support for this research and University of Colorado computer time was provided by National Science Foundation (NSF) MRI Grant #CNS-0421498, NSF MRI Grant #CNS-0420873, NSF MRI Grant #CNS-0420985, and a grant from the IBM Shared University Research (SUR) program. This research was supported in part by the NSF through support for the National Center for Atmospheric Research and TeraGrid resources provided by UC/ANL and NCAR. We would like to thank the members of the SPRUCE project, including Pete Beckman and Suman Nadella, for their guidance and support of this research.

REFERENCES

Al-Ali, R., Hafid, A., Rana, O., and Walker, D. (2004). An approach for quality of service adaptation in service-oriented Grids. *Concurrency and Computation: Practice and Experience*, 16(5):401–412.

Alfieri, R., Cecchini, R., Ciaschini, V., Dell'Agnello, L., Frohner, A., A. Gianoli, K. L., and Spataro, F. (2003). VOMS, an Authorization System for Virtual Organizations.

Allcock, B., Bester, J., Bresnahan, J., Chervenak, A., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnal, D., and Tuecke, S. (2002). Data Management and Transfer in High Performance Computational Grid Environments. *Parallel Computing Journal*, 28(5):749 – 771.

Beckman, P., Beschatnikh, I., Nadella, S., and Trebon, N. (2007). Building an Infrastructure for Urgent Computing. *High Performance Computing and Grids in Action*.

Benkner, S., Engelbrecht, G., Middleton, S., Brandic, I., and Schmidt, R. (2007). End-to-end qos support for a medical grid service infrastucture. *New Generation Computing, Computing Paradigms and Computational Intelligence, Special Issue on Life Science Grid Computing*, 25(4):355–372.

Bogden, P., Gale, T., Allen, G., MacLaren, J., Almes, G., Creager, G., Bintz, J., Wright, L., Graber, H., Williams, N., Graves, S., Conover, H., Galluppi, K., Luettich, R., Perrie, W., Toulany, B., Sheng, Y., Davis, J., Wang, H., and Forrest, D. (2007). Architecture of a Community Infrastructure for Predicting and Analyzing Coastal Inundation. *Marine Technology Society Journal*, 41(1):53–71.

Catlett, C., Andrews, P., Bair, R., and et al. (2007). TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applications. *High Performance Computing and Grids in Action*.

Chadwick, D., Novikov, A., and Otenko, A. (2006). GridShib and PERMIS Integration. *Campus-Wide Information Systems*, 23(4):297–308.

Cope, J. and Tufo, H. (2008). A Data Management Framework for Urgent Geoscience Workflows. In *Proceedings of the International Conference on Computational Science (ICCS 2008)*.

Cui, Y., Moore, R., Olsen, K., Chourasia, A., Maechling, P., Minster, B., Day, S., Hu, Y., Zhu, J., Majumdar, A., and Jordan, T. (2007). Enabling Very–Large Scale Earthquake Simulations on Parallel Machines. In *Proceedings of the International Conference on Computational Science (ICCS) 2007*, Beijing, China. Springer.

Dan, A., Davis, D., Kearney, R., Keller, A., King, R., Kuebler, D., Ludwig, H., Polan, M., Spreitzer, M., and Youssef, A. (2004). Web services on demand: WSLA-driven automated management. *IBM Systems Journal*, 43(1):136–158.

Droegemeier, K., Chandrasekar, V., Clark, R., Gannon, D., Graves, S., Joesph, E., Ramamurthy, M., Wilhelmson, R., Brewster, K., Domenico, B., Leyton, T., Morris, V., Murray, D., Pale, B., Ramachandran, R., Reed, D., Rushing, J., Weber, D., Wilson, A., Xue, M., and Yalda, S. (2004). Linked Environments for atmospheric discovery (LEAD): A Cyberinfrastructure for Mesoscale Meteorology Research and Education. In *Proceedings of the 20th Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Seattle, WA. American Meteorological Society.

Erradi, A. and Maheshwari, P. (2007). Enhancing web services performance using adaptive quality of service management. In *Proceedings of the 8th International Conference on Web Information Systems Engineering (WISE 2007)*.

Eubank, S., Kumar, V. A., Marathe, M., Srinivasan, A., and Wang, N. (2006). Structure of Social Contact Networks and Their Impact on Epidemics. *AMS-DIMACS Special Volume on Epidemiology*, 70:181–213.

Feitelson, D. (2008a). Parallel workloads archive, http://www.cs.huji.ac.il/labs/parallel/workload/.

Feitelson, D. (2008b). Workload modeling for computer systems performance evaluations.

Foster, I. (2005). Globus Toolkit Version 4: Software for Service-Oriented systems. In *IFIP International Conference on Network and Parallel Computing*. Springer-Verlag.

Lang, B., Foster, I., Siebenlist, F., Ananthakrishnan, R., and Freeman, T. (2006). A Multipolicy Authorization Framework for Grid Security. In *Proceedings of the Fifth IEEE Symposium on Network Computing and Application*, Cambridge, MA, USA.

Lederer, H., Pringle, G. J., Girou, D., Hermanns, M. A., and Erbacci, G. (2007). Deisa: Extreme computing in an advanced supercomputing environment. *Parallel Computing: Architectures, Algorithms and Applications*, 38:687–688.

Liu, Y., Ngu, A., and Zeng, L. (2004). QoS Computation and Policing in Dynamic Web Service Selection. In *Proceedings of the 13th International Conference on World Wide Web 2004 (WWW2004)*.

Mandel, J., Beezley, J., Bennethum, L., Chakraborty, S., Coen, J., Douglas, C., Hatcher, J., Kim, M., and Vodacek, A. (2007). A Dynamic Data Driven Wildland Fire Model. In *Proceedings of the International Conference on Computational Science (ICCS) 2007*, pages 1024–1049, Beijing, China.

Nurmi, D., Brevik, J., and Wolski, R. (2007). QBETS: Queue Bounds Estimation from Time Series. In *Proceedings of the 13th Workshop on Job Scheduling Strategies for Parallel Processing*.

Schopf, J., Pearlman, L., Miller, N., Kesselman, C., Foster, I., D'Arcy, M., and Chervenak, A. (2006). Monitoring the Grid with the Globus Toolkit MDS4. In *Proceedings of SciDAC 2006*.

Sharma, A., Adarkar, H., and Sengupta, S. (2003). Managing qos through prioritization in web services.

Truong, H., Samborski, R., and Fahringer, T. (2006). Towards a Framework for Monitoring and Analyzing QoS Metrics of Grid Services. In *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing*.

Wang, G., Wang, C., Chen, A., Wang, H., Fung, C., Uczekaj, S., Chen, Y., Guthmiller, W., and Lee, J. (2005). Service level managment using QoS monitoring, diagnostics, and adaptation for networked enterprise systems. In *Proceedings of the Ninth IEEE International EDOC Enterprise Computing Conference*.

Wolski, R., Obertelli, G., Allen, M., Numri, D., and Brevik, J. (2005). Predicting grid resource performance on–line. *Handbook of Innovative Computing: Models, Enabling Technologies, and Applications*.

Zhou, X., Wei, J., and Xu, C. (2007). Quality-of-service differentiation on the internet: a taxonomy. *Journal of Network and Computer Applications*, 30(1):354–383.