Enabling petascale science: data management, troubleshooting and scalable science services

A. Baranovski[1], K. Beattie[6], S. Bharathi[2], J. Boverhof[6], J. Bresnahan[3,4], A. Chervenak[2], I. Foster[3,4,5], T. Freeman[3,4], D. Gunter[6], K. Keahey[3,4], C. Kesselman[2], R. Kettimuthu[3,4], N. Leroy[7], M. Link[3,4], M. Livny[7], R. Madduri[3,4], G. Oleynik[1], L. Pearlman[2], R. Schuler[2], and B. Tierney[6]

[1]Fermi National Accelerator Laboratory
[2]Information Sciences Institute, University of Southern California
[3]Computation Institute, University of Chicago and Argonne National Laboratory
[4]Mathematics and Computer Science Division, Argonne National Laboratory
[5]Department of Computer Science, University of Chicago
[6]Lawrence Berkeley National Laboratory
[7]Department of Computer Science, University of Wisconsin, Madison

Contact Author: annc@isi.edu

1. Introduction

DOE science applications in such diverse areas as astrophysics, biology, chemistry, combustion, fusion, high energy physics, nanoscience, and nuclear physics are generating and analyzing up to petabytes of data per year. The Center for Enabling Distributed Petascale Science (CEDPS) project [1, 2] works with DOE application science communities to provide services required to move data sets where they are needed; to enable analysis near the data; and to detect and recover from failures in the distributed environment. Each of these tasks is challenging in a petascale environment, because of the need to coordinate numerous shared resources, including CPUs, storage, and networks.

The three main components of the CEDPS project are:

- **Data Placement:** We are developing tools and techniques for reliable, high-performance, secure, and policy-driven placement of data within a distributed science environment.
- **Troubleshooting:** We are also addressing cross-cutting issues relating, in particular, to *failure detection and diagnosis* in distributed systems.
- **Scalable Services:** A third component of the CEDPS project supports programs that are packaged as scalable science services that process many requests to access and analyze data.

In this paper, we describe the ongoing work of the CEDPS project in each of these areas.

2. Data Management

We are developing tools and techniques for reliable, high-performance, secure, and policy-driven placement of data within a distributed science environment. This work includes end-to-end data distribution services that implement different data placement behaviors. In addition, we are significantly enhancing the existing GridFTP data transfer service to provide management of the space, bandwidth, connections, and other resources needed to transfer data to and from a storage system. We are working with a range of DOE application communities to deploy, apply, and evaluate these new capabilities.

2.1. GridFTP Enhancements

In this section, we discuss new functionality and recent developments that we have incorporated into the Globus GridFTP framework [1]. Specifically, we describe the following four initiatives.

1. GridFTP over SSH, which enables GridFTP client "globus-url-copy" [2] to use SSH security for authentication and for exchanging commands and responses on the GridFTP control channel.
2. GridFTP multicasting, which allows efficient transfer of a single data set to many locations, as well as a GridFTP overlay network with many GridFTP servers act as routers forwarding packets.
3. A plugin for GridFTP that enforces usage policies on storage systems managed by Lotman.
4. Concurrency, a client side optimization to improve the performance of lots of small files transfers

2.1.1. GridFTP over SSH

GridFTP traditionally uses Grid Security Infrastructure (GSI) [4] for establishing secure connections as a strong, robust, and flexible means of securing messages. In some situations, however, it is preferable to use the SSH security mechanism. We enhanced the GridFTP server and the globus-url-copy client to use SSH for authentication and command/response exchange. GridFTP-over-SSH leverages the fact that an SSH client can remotely execute programs by forming a secure connection with SSHD. In this case, all of the standard I/O from the remote program is routed back to the client through the secure SSH channel. The standard I/O is used as the secure channel to exchange control channel messages.

2.1.2. GridFTP multicasting

The multicasting capability allows for efficient transfer of a single data set to many locations. The goal is to use available network cycles effectively to move the total amount of data (destinations * file size) at network speeds. The Globus GridFTP server uses Globus XIO [5] for its I/O. We added a multicasting capability by creating a new XIO drive. Clients may add the new XIO multicast driver to their GridFTP server's I/O stack. Globus XIO provides a modular driver abstraction. As the GridFTP server writes data blocks to its file system, the data blocks first pass through the multicast driver, which passes the data block to other GridFTP servers.

The multicast driver can be configured to forward data to the next server and not to write that data to disk. This configuration allows creation of a network overlay where many GridFTP servers act as routers, forwarding packets until they reach their final destination, where they are written to disk.

2.1.3. Lotman plug-in for GridFTP

Lotman is a storage management tool from University of Wisconsin. We created a plug-in for GridFTP to communicate with Lotman to enforce space usage for data transferred to storage systems that are managed by Lotman. Lotman supports disk quotas for users on a storage system in the form of lots. Before writing data on to the disk, GridFTP server communicates with the Lotman and ensures that the user quota limit is not violated. One advantage of this plug-in is that transfer failures due to a destination storage system running out of space during a transfer can be avoided if the file size is known in advance; the GridFTP server can check whether there is enough space at the destination before starting the transfer.

2.1.4. Concurrency

In [1, 2] , we discussed the pipelining optimization in GridFTP to improve the performance of many small files transfers. In modern high-speed networks, data sets less than 100 MB are too small for the underlying protocols like TCP to utilize the maximum capacity of the network. To further enhance the performance of small files transfers, we added a feature called "concurrency" to the GridFTP client. This feature helped DOE's Advanced Photon Source user facility at Argonne to transfer more than a terabyte of data to Australia at a rate 30 times faster than traditional data transfer mechanisms such as SCP.

2.2. Data Placement Services

The CEDPS team's staged development of data placement service functionality began with the implementation of a priority-based bulk transfer service for managing many concurrent data transfer

operations. This functionality is needed to provide reliable distribution of data according to the plan decided by the higher-level placement service. In addition to the bulk transfer service, we have implemented a web service interface for submitting data placement requests. A paper describing the initial data placement service implementation was published at the Petascale Data Storage Workshop in November 2007 [3].

2.2.1. Integration of Data Placement and Workflow Management

In addition to development of the functionality of the data placement services, a significant focus of our work has been to understand the interaction between data placement services and workflow management systems, which are increasingly being used to manage sophisticated scientific analyses consisting of thousands of interdependent tasks. Through measurements and simulations, we have done substantial work in understanding how data placement can improve the performance of data-intensive computational workflows running on distributed computing resources, clusters and supercomputers.

The CEDPS team did an initial study of placement service options that evaluated the performance of a data placement service in combination with the Pegasus workflow management system. We performed initial measurements that demonstrated the performance advantages of prestaging data sets onto computational resources before execution of workflows; this work was presented at the Grid2007 Conference [4]. In addition, we modified an existing Grid simulator to experiment with different data placement algorithms operating in conjunction with a workflow manager. A paper describing these simulation results for a variety of data placement algorithms was submitted to the SC2008 conference [5]. Finally, in a paper presented at the 3rd International Workshop on Workflow Systems in e-Science (WSES 08), we take a broad view of the challenges of data placement and data management at all stages of the workflow lifecycle [6].

2.2.2. Simplified Data Replication Tools

To make data replication services more accessible to CEDPS users, we have designed and developed a set of tools to simplify key data replication usage scenarios. Existing tools for data replication include the Globus RLS replica catalog and the Globus GridFTP Server. Our new tools integrate the capabilities in the replica catalogs and GridFTP server for common data operations such as getting, putting and copying replicas.

3. CEDPS Troubleshooting

3.1. Introduction

Many of today's Grids have ongoing performance and reliability problems that have yet to be addressed. As far back as 2003, the grid Grid2003 project, now Open Science Grid (OSG) [7], saw a 30% job submission failure rate [8], with 90% of the failures caused by problems such as disk filling errors, gatekeeper overloading, and network interruptions. Although many improvements have been made, the Grid continues to grow in size and complexity. The net improvement is often small: for example, the LCG Grid [9] is still reporting about a 25% error rate [10],.

Troubleshooting Grid middleware can be difficult because of the large number of interconnected components. For example, a single action, such as reliably transferring a directory of files, can result in the coordination of a wide suite of loosely coupled software tools. Each of these systems typically creates its own log files using their own log format, semantics, and identifiers. To troubleshoot a problem as it cascades from one component into the next, this information must be combined to form a logically consistent trail of activity. The immediate goal of the CEDPS troubleshooting area is to replace the current project-specific ad-hoc techniques used in distributed computing with a more powerful unified toolkit. The long-term goal is that any software that navigates multiple layers of interfaces can leverage the CEDPS troubleshooting tools to more simply and definitively analyze failure and performance issues.

3.2. Logs

We have written a recommendations guide for Logging Best Practices [11] that combines good instrumentation practices with log format guidelines. The practices and log format go hand-in-hand so

that there are standard representations for all the essential elements: timestamps, *start* and *end* events to wrap operations, unique identifiers, and status codes. Arbitrary additional named attributes can be added by the application. The log event format is simply a line of ASCII "name=value" pairs: this is highly portable, human-readable, and works well with line-oriented tools and syslog-ng.

3.3. Logging Framework

A logging framework is required to bring all the various types of log files together and make sense of them. There are four main components to our logging framework: log collector, log parser, database loader, and database. Loading a subset of logs into a relational database enables sophisticated data mining. Historical queries can provide baseline performance information, and this database can be used for post-mortem anomaly detection.

For the log collector we use *syslog-ng* [12]. *syslog-ng* allows us to aggregate the data and filter logs based on program name, log level, and even a regular expression on message contents. Syslog-ng is very scalable: if a particular collector gets over-loaded, one can just bring up another collector on another machine, and send half the logs to each collector.

To convert logs into a common format we have written a log parser component. This log parser is implemented with a plug-in architecture to make it easy to add new parsers. Parsed logs are then fed to our database loader, which loads the records into a MySQL database.

3.4. OSG Deployment

The Open Science Grid is a distributed computing infrastructure for large-scale scientific research. It is built and operated by a consortium of universities, national laboratories, scientific collaborations and software developers. The CEDPS project has been working closely with OSG to design and deploy a centralized log collection framework based on syslog-ng. We have written a pacman [13] package for the Virtual Data Toolkit (VDT) [14], which is deployed on OSG.

After much discussion with OSG site administrators, it was determined that sending a large of amount of logs to the Grid Operations Centers (or GOC) would not work due to security and privacy concerns at several sites. If sites choose, they can forward their logs to the GOC directly to aid in troubleshooting. The OSG logging architecture is shown in Figure 1.

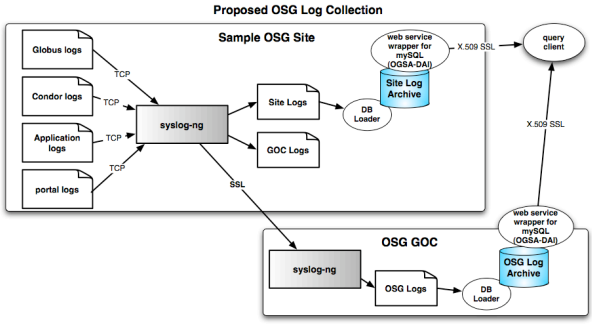

Figure 1: Sample OSG syslog-ng deployment

ET MSD 7/1/08 2:22 PM
Deleted: Figure 1

3.5. Sample Results

We have reworked the logging for Globus v4.2 to use the new "best practices" format for most Globus Services. We have found a number of examples where this has made troubleshooting much easier. For example, the GridFTP server [15] now wraps authentication, authorization, and transfers with *start* and *end* events. If we see the following logs events: authn.start, auth.end, authz.start, authz.end, transfer,start; but are missing the transfer.end log and there are no other error messages, then we now can tell this is likely a firewall issue on the data port. Before this additional logging, it was very difficult to determine why the GridFTP transfer was hung.

While our initial motivation was to build this framework for troubleshooting issues, we have found it to be very useful for more detailed accounting as well. For example, the Southern California Earthquake Center (http://www.scec.org/) uses the Grid to run several seismological modeling codes that simulate earthquake processes. They need to run many thousands of small independent jobs (800K in a typical run) over a period of one to two days. They wanted to better understand their load balancing across the

available resources, and used our logging framework to load their 'kickstart' logs [16] into our database. We were then able to easily extract information such as the number of jobs and job run-time broken down by hour for a typical day long run.

3.6. Availability and Next Steps

All the components of our logging framework are available as part of our NetLogger Toolkit, available at: http://acs.lbl.gov/NetLogger/. Our next focus will be on the integration of OSGA-DAI [17]. We hope that OSGA-DAI will solve the authorization problem of allowing users to see their own logs, but not the logs of others. Performance issues with OSGA-DAI will also be explored. We also plan to focus on further improvements to our Logging Best Practices recommendations and implementation. Currently, BP-formatted logs simplify the process of diagnosing problems that occur within a single service, or among multiple services in a single hosting environment (such as a Globus container). We plan to expand our logging efforts to include information exposing the relationships between subtasks of workflows that span multiple sites.

4. CEDPS Scalable Services

We are developing tools and techniques for the construction, operation, and provisioning of scalable science services. We are creating service construction tools that take application code (simulation or data analysis, program or library) and wrap it as a remotely accessible service, with appropriate interfaces, authorization, persistence, provisioning, and other capabilities. We are constructing provisioning tools for dynamic management of the computing, storage, and networking resources required to execute a service, and the configuration of those resources to meet application requirements.

4.1. Service Construction Tools

The CEDPS team has been working on developing tools to easily wrap applications. This year marked some changes in the technologies to be used for wrapping legacy applications into Grid Services. We started working on a Java based tool called Grid Remote Application Virtualization Interfaces (gRAVI). This tool is built on top of a Grid Service building tool called Introduce. The following are the current capabilities of the gRAVI:

- Auto generation of services using standard Grid Security and flexible authorization models
- Integration of gRAVI with centralized registries to enable automatic publishing of services enabling discovery
- Development of strongly typed services

4.2. Application outreach

- **APS:** The Advanced Photon Source scientists used gRAVI to wrap applications and created a workflow that orchestrates two gRAVI-created application services that run analytics.
- **caBIG:** Using gRAVI, the cancer Biomedical Informatics Grid project was able to quickly wrap an Application called Hierarchical Clustering into a Grid service and successfully demonstrate this at the caBIG conference. The tool will let the users create a strongly typed grid service and would enable the execution of the application as a Grid job on a cluster.

4.3. Future Work:

We are working on generating code automatically that would enable the transfer of input files needed for the execution of the application from the user's machine and also transfer the generated output files back to the user's machine. We are also developing provisioning services using Virtual workspaces and WS-GRAM services.

Acknowledgments

This work was supported in part by the Department of Energy's Scientific Discovery through Advanced Computing II program under grant DE-FC02-06ER25757 and by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific

Computing Research, Office of Science, U.S. Department of Energy, under contracts DE-AC03-76SF00098 and DE-AC02-06CH11357.

References

[1] A. Baranovski, S. Bharathi, J. Bresnahan, A. Chervenak, I. Foster, D. Fraser, T. Freeman, D. Gunter, K. Jackson, K. Keahey, C. Kesselman, D. E. Konerding, N. Leroy, M. Link, M. Livny, N. Miller, R. Miller, G.Oleynik, L. Pearlman, J. M. Schopf, R. Schuler, and B. Tierney, "Enabling distributed petascale science " *Proceedings of SciDAC 2007 Conference, Boston, MA, June 2007. Also appeared in Journal of Physics: Conference Series* vol. 78, 2007.

[2] J. M. Schopf, A. Chervenak, I. Foster, D. Fraser, D. Gunter, and B. Tierney, "End-to-End Data Solutions for Distributed Petascale Science," *CTWatch Quarterly*, vol. 3(4), 2007.

[3] A. L. Chervenak and R. Schuler, "A Data Placement Service for Petascale Applications," presented at Petascale Data Storage Workshop, Supercomputing '07, Reno, Nevada, 2007.

[4] A. Chervenak, E. Deelman, M. Livny, M. Su, R. Schuler, S. Bharathi, G. Mehta, K. Vahi, "Data Placement for Scientific Applications in Distributed Environments," presented at 8th IEEE/ACM Int'l Conference on Grid Computing (Grid 2007), Austin, Texas, 2007.

[5] S. Bharathi and A. Chervenak, "Data Staging Strategies and Their Impact on Workflow Performance," submitted to SC2008 Conference, 2008.

[6] E. Deelman and A. Chervenak, "Data Management Challenges of Data-Intensive Scientific Workflows " presented at 3rd International Workshop on Workflow Systems in e-Science (WSES 08), in conjunction with CCGrid 2008 Conference, Lyon, France, 2008.

[7] "Open Science Grid Consortium, http://www.opensciencegrid.org/," 2006.

[8] I. Foster, J. Gieraltowski, S. Gose, N. Maltsev, E. May, A. Rodriguez, D. Sulakhe, A. Vaniachine, J. Shank, and S. Youssef, "The Grid 2003 Production Grid: Principles and Practice," presented at 13 th IEEE International Symposium on High Performance Distributed Computing (HPDC), 2004.

[9] "LCG Grid: http://www.gridpp.ac.uk."

[10] M. Aggarwal, D. Colling, B. McEvoy, G. Moont, and O. v. d. Aa, "A Statistical Analysis of Job Performance within LCG Grid," presented at CHEP06, Mumbai, India, 2006.

[11] CEDPS Project, "Logging best practices guide, http://www.cedps.net/wiki/index.php/LoggingBestPractices," 2007.

[12] "syslog-ng: http://www.balabit.com/products/syslog-ng/."

[13] "Pacman: http://physics.bu.edu/~youssef/pacman/."

[14] "VDT: http://vdt.cs.wisc.edu/."

[15] W. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu, I. Foster, "The Globus Striped GridFTP Framework and Server," presented at IEEE Supercomputing (SC05) Conference, Seattle, WA, 2005.

[16] J. S. Vöckler, G. Mehta, Y. Zhao, E. Deelman, and M. Wilde, "Kickstarting Remote Applications," presented at 2nd International Workshop on Grid Computing Environments, Tampa, FL, 2006.

[17] University of Edinburgh, "Open Grid Services Architecture Data Access and Integration (OGSA-DAI), http://www.ogsadai.org.uk/," 2008.