Grid Performance Workshop 2004 Meeting Report

http://www.mcs.anl.gov/~jms/GPW2004

Jennifer M. Schopf, on behalf of the Steering group

Sept. 23, 2004, with final edits August 2005

Mathematics and Computer Science Division Technical Memorandum ANL/MCS-TM-285,
Argonne National Laboratory
National eScience Centre Technical Report UKeS-2005-05, available from
http://www.nesc.ac.uk/technical_papers/UKeS-2005-05.pdf

Executive Summary

Grid computing—the use of multiple resources in different administrative domains to solve a single coordinated problem—is becoming more prevalent in the scientific community. Many applications are taking advantage of larger data stores and more computing power to solve large problems or to solve the same problems faster. However, many of the applications being adapted to run on a Grid rarely achieve even a fraction of the possible performance of the underlying systems. To begin to address this need, we held a two-day workshop that focused on the science of performance and the Grid. A number of open problems were identified at the workshop.

Measurements

The first step in understanding any system is measuring it. Without proper measurement, we cannot understand how changes over time affect the system, nor can we make decisions about how changes should be made.

- A set of standard, basic performance measurements is needed for individual components
    - Standard data sets for experimental work
    - Standard ways to compare scalability of tools
- Grid integration tests are also needed for end-to-end performance data
- Scalability must be considered when taking the measurements, when storing them, and when accessing them

Measurement/Monitoring System Deployment

The lack of systematic deployment and integration between tools was an issue of concern to many. Issues include the following:

- Common interfaces to probes and common schemas
- Ability to tune the configuration (probe size, frequency) for different uses over time

Analysis

Given a set of measurements, one must be able to understand them. From an applications point of view, the simple numbers reported by a benchmark often aren't helpful; additional interpretation is needed to understand their meaning. Open issues include the following:

- Correlation of probes (measurements)
- Understanding sensitivity of performance data to changes in parameters, understanding baseline performance, and prediction.

Simulation

One cannot always run experiments on live systems. Often, the experiments are too intrusive, or the level of control needed is not possible in a Grid environment that by nature lacks global control. Hence, simulation continues to play a role in any discussion of performance and performance predictability. This topic was noted as a possible area of discussion for the follow-on workshop. In addition, several speakers commented the need for "classic" data sets, workload traces, and ways to compare results.

1 Introduction

Grid computing—the use of multiple resources in different administrative domains to solve a single coordinated problem—is becoming more prevalent in the scientific community. Many applications are taking advantage of larger data stores and more computing power to solve large problems or to solve the same problems faster.

However, many of the applications being adapted to run on a Grid rarely achieve even a fraction of the possible performance of the underlying systems. In part, this situation arises because users do not know what performance they could achieve. No means exists for estimating baseline performance and distinguishing between how an application is currently performing and what is possible in practice. In addition, Grids are becoming more complex as larger numbers of components in multiple administrative domains are working together; hence, understanding the performance of a single system is no longer enough. Service agreements are being written with no way to verify that the performance goals are being met, and troubleshooting errors between sites has become a major issue to most Grid users.

To address the need for better understanding of Grid performance issues, we held a two-day workshop for 35 invited participants, with twelve talks and four breakout sessions on current approaches and application needs for the science of performance and the Grid. The workshop identified the key areas required for research as well as community leadership and application engagement that will lead to metrics and measures enabling us to track the improvement of Grid infrastructures and to more efficiently focus resources on the most critical issues.

In this report we first enumerate the visions of the workshop participants of how collected performance data could be used, in terms of integration with current tool approaches, error analysis, understanding service level agreements, and basic analysis of the data (prediction, correlation of data, etc.). In Section 3 we discuss what data should be collected (and Appendix A includes a list of the data currently collected by the tool builders at the meeting), and a discussion of the second-order data that is also needed, items such as variance andrefresh rates. Section 4 details issues of deployment and standardization so that current tools can better interact with one another. Section 5 discusses simulation, a side topic of this workshop, but one possible focus for the following year. We conclude in Section 6 with a summary of the state of the field and the open issues, highlighting those for discussion at next year's workshop.

2 Why Implies What (and How)

Performance and benchmark data on a distributed system is needed for several reasons. The expected use of the data will greatly influence what data should be collected and how data should be acquired. The twelve invited talks and various breakout discussions often shifted focus as different uses of performance data were identified. In this section we group the different uses of performance data, albeit somewhat arbitrarily, into integration with current tools, error analysis, understanding service agreements, and basic analysis of the data (prediction, correlation of data, etc.). For each group we identify the data needed and the tools in common use today, as well as open areas of study.

2.1 Integration with Current Tools

By far the most common use envisioned for performance data was to make "smarter" tools available to users. The two largest classes discussed included smarter schedulers and smarter applications, although smarter monitoring systems were also mentioned.

Smarter schedulers use performance data to better match available resources to the needs of a job, whether that job is the computing part of an application, transferring a file, or some other task. Given additional details about trends of current use, one can make more informed choices with respect to where to start a job or file transfer and when to migrate that job or transfer to another resource due to poorer performance than expected. Some current scheduling approaches use dynamic monitoring data [DBG+03, EDG04, RSL04], but most of today's production quality schedulers use only the most basic of information to make scheduling decisions. Current research shows that additional data will strongly improve scheduling decisions [BD97, Mitz98, RF03], but details on exactly what data is needed and how accurate it needs to be is an open question.

Self-healing and adaptable applications are currently being investigated by several groups [SHA02]. For example, Beckman detailed the TeraGyroid project, a Lattice-Boltzmann simulation of defect dynamics in amphiphilic liquid crystals [TGP], eing studied at several institutions in the UK and running on the TeraGrid resources. This application uses computational steering to execute within a parameter space based on previous results; but with additional data, smarter resource allocation decisions could be made, and additional tradeoffs in terms of accuracy versus compute time could be offered to the user. Several applications are also being developed to be more "network-aware" [Ste99]. These applications are dynamically adjusting their TCP buffer sizes and the number of parallel streams in use based on performance feedback data.

In addition, several attendees envisioned their own monitoring systems being made smarter in terms of the data being collected and how often. For example, the Inca system [SOE+04] is being extended to allow dependencies in terms of the probes that are run. If one probe fails— for example, a basic file transfer—then more complicated file transfer probes should not also run; instead more low-level tests to understand basic network functionality should begin. NetLogger [GTK+02] also allows different users to subscribe to different levels of monitoring data on an individual data-stream basis.

2.2 Error Analysis

Trouble shooting and fault detection are two areas receiving a fair amount of attention now. These are the silver bullets that some say are just out of reach [TSWS02]; however, the general feeling at our workshop was that both of these areas required significantly more work than other general researchers believed.

Troubleshooting can be broadly defined as detecting errors in application runs, preferably before they happen, and resolving them, preferably without the user being aware that they occurred in the first place. Related to this is the concept of anomaly detection—both the detection and the prediction of anomalies in the system. The amount of information needed for these use cases is not at all well understood. End-to-end monitoring of all the components would be required: application, middleware, operating system, hardware, and network. Some research has focused on understanding whether subsets of data can be used to predict overall performance [KSD02], but this work is still very preliminary.

Related and yet distinct from troubleshooting during an application run is the need for data to aid in debugging applications as they are being deployed on Grid systems. Even simple applications are difficult to debug over multiple platforms, let alone those that are complex or multithreaded. To our knowledge, no tool is successfully addressing this problem yet, nor is the data needed to achieve it well understood.

2.3 Understanding Service Agreements: Is the Grid Up?

As Grids are growing in size and complexity, organizations are defining more extensive service agreements for the partnerships involved. These can vary from agreements on software stack to detailed contracts of uptime and service provisions. The primary goal of these agreements is to understand what it means for the Grid to be up, and provide guarantees to partners that this state will occur with a predefined regularity.

Performance data is playing a key role in understanding what a service agreement is and whether it is being met. One example discussed at the workshop was the use of the Inca test harness for the TeraGrid project [SOE04]. This software checks basic software stack data and some service data, and researchers are incorporating additional performance data for each of the TeraGrid sites. Similar monitors are being used by the UK eScience Project [GITS], Grid3 [Grid3], and GrADS [CDC+04]. The measurements used in these systems vary widely but focus on data about common services; those systems that collect more data do so because they are more developed, not because additional data isn't needed in the other systems. How much data is enough is an open question, as is a way to integrate these tools to allow shared information sources.

2.4 Basic Analysis—Prediction, Evaluation, Correlation

Given a set of measurements, one must be able to understand them. From an applications point of view, the simple numbers reported by a benchmark often aren't helpful; additional interpretation is needed to understand their meaning.

In our original proposal for the workshop we believed that a main discussion point would be analysis—how collected data was being used in predictions and for higher-level services. While these issues were discussed, the workshop participants generally took a more theoretical focus, as opposed to the pragmatic uses discussed in the previous subsections. However, many important open questions were expressed, and solutions will be needed in order to have operational higher-level services currently being envisioned by many application scientists.

One analysis of measurement data is the evaluation of the gap seen between what an application achieves in terms of performance, what the application scientist expects to achieve, and the level of performance that is possible to achieve with tuning. In today's systems, applications often have no information with which to base any prediction of future performance or possible achievable performance. Hence, resources go underutilized, and applications remain far short of their achievable promise. With better performance data, application scientists could look at performance tuning for their applications. The overall effectiveness of Grid systems could be significantly improved by examining better program design for the systems they will run on, for example, understanding whether better pipelining of I/O and computation can increase performance.

Another secondary analysis of data is understanding baseline performance. Given a set of standard measurements deployed on a number of systems, researchers could make basic performance comparisons and thus have a starting point for understanding what kind of performance was possible from a system and how performance on the system changes over time. In addition, decisions on which platform is best suited to a specific application could be more easily made if there were standard ways to compare performance capabilities between Grids.

One of the most prevalent forms of analysis envisioned by the workshop participants was prediction of future behavior. As stated in Section 2.1, prediction engines can work with existing tools to have smarter schedulers and better overall resource management of the system. Another vision for predictions is to help predict when faults will occur—given ongoing monitoring of the systems, one may be able to predict upcoming faults based on current condition data.

Another form of analysis discussed was the need for correlation of probes in order to limit the impact on the monitored system. Cotrell hypothesized being able to extrapolate network data for unmonitored sites from that available at hubs; Wolski discussed the overhead of current probes in detail; and Beckman suggested that if probe data could be correlated, higher-level probes could be used more often, with lower-level probes collecting data only in special cases. More research is needed to understand these issues in detail.

In general, it is not well understood how low-level probe data relates to higher-level services. For example, Schopf and Vazhkudai found that Network Weather Service 64K probes are not correlated to the end-to-end throughput of GridFTP transfers for large files [VS03], and users need a way to interpret when a job will start running when the only information source is queue length. These are only two open questions of many. The general field of mapping low-level data to user-level information is an open research question.

3 Measurements

The discussion of how measurement data could be used lead to a better understanding of why current performance benchmarks are not sufficient, what data needed to be collected, and how it might be managed. Appendix A lists the data currently collected by the systems represented at the workshop, and it was felt that this list was at least a good starting point for what data should be gathered and corresponds closely to common practices [Schopf03]. What was less well understood were the secondary measurement problems: how often the data should be collected, how it should be archived, what metadata was needed, and so forth.

3.1 Why What We Have Is Not Enough

The HPC community has explored performance techniques for decades, from sequential to vector, massively parallel architectures. Participants in the workshop explored which benchmarks and methodologies learned from examining such architectures could apply to Grid environments. Several talks described how Grid computing was different from standard cluster or distributed computing. There is a physical separation of the resources, and not just compute resources but data resources, instruments, visualization, and so on. There is a lack of control over the entire set of resources, unlike previous computing approaches, including client-server. There are, in general, many more components in play to address each applications needs. And the gap between the application (and software) and the hardware can be much greater because of the many layers of middleware often involved to make the disparate components appear as a whole to the problem being solved. These factors—physical separation, lack of control, large number of components, and gap between application and hardware—significantly complicate understanding the performance of the resulting system and are unlike any system studied in the past.

In his talk, Gropp gave three cautionary tales from previous benchmarking techniques to show why previous approaches would not meet the needs of the Grid community:

1) LINPACK overemphasized the raw flop rate on algorithms by having N^3 work on N^2 data
2) SPEC (and other vendor chosen tests) are being used to design tomorrow's hardware for yesterday's algorithms
3) Latency and bandwidth for message passing is often misrepresented in current performance benchmark approaches

One point of agreement among the participants was that application requirements should guide today's benchmarks, unlike some initial approaches [NGB04]. However, the lack of canonical Grid applications, or even classes of applications, makes the definition of benchmarks that much more complicated. Unless performance data can be mapped to application-level behavior, it will not be useful to most application scientists. It is quite likely that different application types will require very different sources of performance data, as well.

Current benchmarking approaches were developed for other kinds of systems and application, so they do not capture the needed level of detail. We are not able to collect the level of data to ensure that a change in the system is reflected in the data collected—a fundamental flaw. Another problem with current approaches is that data is collected at one level, when for many Grid applications different levels of data are needed depending on the application type. For example, many Grid applications have complex paths for messaging, multiple transport types, and high latency; hence, simplified message-passing performance data does not adequately reflect this richer environment. Work is needed to better understand how to cover the basic performance aspects of Grid systems.

3.2 Measurements Needed

The participants agreed that the measurements needed in a Grid environment had to reflect the end-to-end path of the system: software, middleware services, hardware, network, and the full I/O path needed to be instrumented to gather the data assumed to be available by many of the analysis techniques discussed in Section 2. It is especially important to capture the "last meter" effect seen by poor I/O to disk, saturated memory systems, and other close-to-the-application problems not captured by more high-level performance data. The list of basic data that can be collected was fairly well understood. Appendix A lists the data currently collected by a broad set of tools in common use today. This list may not meet all the envisioned needs of the performance community, but it a first pass at the general needs.

While there was general agreement about the lower-level atomic data to be collected, there were many open questions about what kind of higher-level information was needed. For example, while the researchers agreed that bandwidth and latency data should be collected, it was recognized that what was actually needed was an understanding of large file transfer performance, which includes many other factors and for which there is no agreed upon measurement technique. The researchers also agreed that critical application characteristics should be identified and data collected to correspond to each so that the performance impacts could be better understood; however, no one was sure of the best way to more forward with this approach.

Similarly, logged data—such as the error logs currently seen in many systems with a job failure—was discussed and identified as an important category of higher-level data but not sufficient by itself for understanding application behavior or failure. Tierney reiterated the need for propagating unique id's across all components of a Grid application to aid in correlating log data to performance affects.

3.3 Metadata Needed

The larger open question was what secondary statistics should also be gathered when measurement data was taken. Properties considered included uncertainty, variance, refresh rate, sensitivity and security, data summaries, user access to data, and archiving of information. There is a well-known tradeoff between the accuracy of a piece of data, the lifetime of that data, and the overhead of collecting and storing the data. The sweet spot for this tradeoff is not well understood, however, especially since it will vary widely with the type of data being collected.

The most frequently mentioned type of metadata needed was a measurement of usability. Usability greatly effects the performance of the system, but very little has been done to measure it in any repeatable way [Wilson94, WB96].

These topics need further discussion at the next workshop.

3.4 Accessing Measurement Data

As more and more data is being collected, concern was expressed about the right way to manage the sets of measurements. While several of the tools represented at the workshop were currently using relational database approaches, these solutions were too heavy weight for many of the others.

There was some discussion about the use of global unique id's that can be associated with all the components of an application or related set of tasks. This approach is especially useful for netlogger and similar systems that show timelines of associated actions. Without some kind of identifier that ties together the separate pieces, however, events that are related to one another can only be found using statistical techniques, and never with certainty.

The lack of "classic" data sets in the Grid environment was also commented on. In most fields there are canonical data sets that are used to compare different approaches and to evaluate new work in a known setting. Without this standard of comparison, new results cannot be evaluated against other approaches.

4 Deployment Issues

A topic that came up unexpectedly at the workshop was the deployment and integration of monitoring and performance tools.

A complete monitoring system includes many pieces: probes or sensors to publish data, schemas and access policies for that data, management of the probes, data collection points, archiving, and analysis tools. It is unrealistic to expect multiple administrative domains to deploy the same infrastructure, so tool integration becomes a factor. Adapting configurations and managing infrastructure also need to be considered.

One step toward the interoperability of monitoring frameworks would be to define a set of common interfaces and schemas. With a well-defined request and response protocol/schema, probes from one system could be used with another without changes. Some work has been down toward this, including the GLUE schema [GS03] for basic cluster data, and the schemas being defined by the GGF Network Measurement Working Group [NM], but these are only small initial steps. In part we also need to be cautious about standardizing too soon. Standards without enough practical experience will cause more problems than they prevent.

For the tools that are commonly deployed, the three biggest problems that workshop attendees identified were lack of a way to tune configurations or customize clients, poor usability, and lack of reliability or scalability. In general, configuring current systems was an involved process that was not easily changed after the initial deployment. Management of deployed sensors was seen as extremely difficult, and client tools did not often allow the customization that was needed to make the data more useful. Both of these factors made the systems significantly harder to use than many people wanted. Most systems also had reliability issues—although without common test data and environments, it was extremely difficult to compare systems. Some work has begun to examine scalability of monitoring systems [ZFS03, JLK+03], but this is limited in scope.

5 Simulation

Simulation was a topic identified for the following workshop. Simulation is an invaluable tool to counter the lack of reproducibility that we currently see in most Grid environments. With valid statistical approaches to simulations we can better understand the effects of performance changes, provide a way to evaluate new methods, and allow much easier ways to test the effects of adding or removing resources and services to an environment.

Several approaches to Grid simulation are being studied [LD03, LMC03, RF03, XDC+04], but this is still an open topic of research. At next year's workshop we plan to have a session to discuss the needs of the performance community for Grid simulations.

6 Summary and Open Issues

Many facets of performance in a Grid environment are not yet well understood. While many uses of the data can be envisioned, only preliminary work has been done to collect basic data. Metadata is needed, as is a better understanding of how to access large data sets of this nature. Tools need to be easier to use and to work together more easily.

Several topics were identified as especially important for the next workshop. These include the following:

- How Grid operators make use of performance data
- Performance and infrastructure—what are better ways to build performance hooks into middleware
- Performance engineering on the Grid—which parameters have been used
- What are the right ways to compare monitoring systems
- Where do simulations fit into performance studies

- How should "classic data sets" be defined

This workshop provided a venue where performance tool builders, researchers, and application scientists could come together to discuss performance issues in Grid environments. Many open problems were addressed, and the participants were able to clearly identify major areas where technologies are needed to realize the full potential offered by Grid computing. The area of Grid performance is in its infancy, and follow-on is critical if Grids are to deliver reliable, application-level performance.

References

[BD97] M. Harchol-Balter and Allen Downey, "Exploiting Process Lifetime Distributions for Dynamic Load Balancing", IEEE Transactions on Computer Systems, 15(3), pages 253–285, August 1997.

[CDC+04] Greg Chun, Holly Dail, Henri Casanova, and Allan Snavely, "Benchmark Probes for Grid Assessment", Proceedings of the High-Performance Grid Computing Workshop, April 2004.

[DBG+03] Ewa Deelman, James Blythe, Yolanda Gil, and Carl Kesselman, "Workflow Management in GriPhyN", in *Grid Resource Management*, J. Nabrzyski, J. Schopf, and J. Weglarz, editors, Kluwer, 2003.

[EDG04] EDG Broker, 2004, http://egee.in2p3.fr/html/Public/Documentation/Organigram-0.1/node3.html

[GITS] Grid Integration and Testing Script, 2004, http://www.grid-support.ac.uk/downloads/ppt/GT2L2G_Course_Part_3_GITS_tests_01.ppt

[Grid3] Grid3 Monitoring, http://www.ivdgl.org/grid2003/catalog/

[GS03] GLUE Schema v 1.0, 2003, http://www.cnaf.infn.it/~sergio/datatag/glue/CE/glueCE.htm

[GTK+02] D. Gunter, B. Tierney, K. Jackson, J. Lee, and M. Stoufer, "Dynamic Monitoring of High-Performance Distributed Applications", in Proceedings of the 11th IEEE Symposium on High Performance Distributed Computing, HPDC-11, July 2002, LBNL-49698.

[JLK+03] H. N. Lim Choi Keung, J. R. D Dyson, Stephen Jarvis, and G. R. Nudd, "Performance Evaluation of a Grid Resource Monitoring and Discovery Service", IEEE Proc.-Software, 150(4), pp. 243–251, August 2003.

[KSD02] Michael W. Knop, Jennifer M. Schopf, and Peter Dinda, "Windows Performance Monitoring and Data Reduction Using WatchTower", in Proceedings of Workshop on Self-Healing, Adaptive and self-MANaged Systems (SHAMAN), June 2002.

[LD03] D. Lu and P. Dinda, "GridG: Generating Realistic Computational Grids", Performance Evaluation Review, Vol. 30, no. 4, March 2003.

[LMC03] A. Legrand, L. Marchal, and H. Casanova, "Scheduling Distributed Applications: The SimGrid Simulation Framework", Proceedings of the 3rd IEEE Symposium on Cluster Computing and the Grid (CCGrid'03), May 2003.

[Mitz98] Michael Mitzenmacher, "How Useful Is Old Information", SRC Technical Note 1998-002, Feb. 1998.

[NGB04] NAS Grid Benchmarks Version 1.0, 2004, http://www.nas.nasa.gov/Research/Reports/Techreports/2002/nas-02-005-abstract.html

[NM] Network Measurements Working Group, Global Grid Forum, http://www-didc.lbl.gov/NMWG/

[RF03] Kavitha Ranganathan and Ian Foster, "Simulation Studies of Computation and Data Scheduling Algorithms for Data Grids", Journal of Grid Computing, 1(1), pages 53–62, 2003.

[RSL+04] Rajesh Raman, Marvin Soloman, Miron, Livny, and Alain Roy, "The Classads Language", in *Grid Resource Management*, J. Nabrzyski, J. Schopf, and J. Weglarz, editors, Kluwer, 2003.

[Schopf03] Jennifer M. Schopf , "MDS2 Data for Project Deployment Guide", Globus Project Tech Report, Brief, June 2003.

[SHA02] Workshop on Self-Healing, Adaptive, and self-MANaged systems (SHAMAN), 2002. http://www.cse.psu.edu/~yyzhang/shaman/proc.html

[SOE+04] Shava Smallen, Catherine Olschanowsky, Kate Ericson, Pete Beckman, and Jennifer Schopf, "The Inca Test Harness and Reporting Framework", to appear in SuperComputing '04, April 2004. Also available as SDSC Technical Report #SDSC-TR-2004-3, http://www.sdsc.edu/TR/SDSC-TR-2004-3-IncaTest.pdf.

[Ste99] Steenkiste, P., "Adaptation Models for Network-Aware Distributed Computations," in Proceedings of the 3rd Workshop on Communication, Architecture, and Applications for Network-based Parallel Computing, Orlando, January 1999.

[TGP] TeraGyroid: Grid-based Lattice-Boltzmann simulations of Defect Dynamics in Amphiphilic Liquid Crystals, http://www.realitygrid.org/TeraGyroid.html

[TSWS02] Troubleshooting and Fault Tolerance in Grid Environments Workshop, December 11th 2002, http://www.ppdg.net/mtgs/Troubleshooting/agenda.htm

[WB96] Gregory V. Wilson and Henri E. Bal, "An Empirical Assessment of the Usability of Orca Using the Cowichan Problems", IEEE Parallel and Distributed Technology, 4(3), pages 36–44, Fall 1996.

[Wilson94] Gregory V. Wilson: "Assessing the Usability of Parallel Programming Systems: The Cowichan Problems", in Proceedings of the IFIP Working Conference on Programming Environments for Massively Parallel Distributed Systems, April 1994.

[XDC+04] Huaxia Xia, Holly Dail, Henri Casanova, and Andrew Chien, "The MicroGrid: Using Emulation to Predict Application Performance in Diverse Grid Network Environments", submitted to the CLADE'04 workshop. Also available from http://www-csag.ucsd.edu/papers/clade04xia.pdf

[ZFS03] [ZF+03] Xuehai Zhang, Jeffrey Freschl, and Jennifer M. Schopf ,"A Performance Study of Monitoring and Information Services for Distributed Systems", in Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing (HPDC-12), July 2003.

Appendix A – Currently Collected Measurements

The tools represented at the workshop compiled a list of currently collected data. This was meant to represent a "state of the field," not a list of all data needed from Grid resources.

BASIC RESOURCE SITE

CPU capability
- Flops count/simple BM (linpack)
- Kernel version
- Architecture type

CPU load
- System load
- User load
- I/O load

Queue info
- Policy (how many jobs at a time, max time)
- Length(residency times?)
- Number running jobs

Inter-processor connections
- BW Between processors in a cluster
- Latency between processors in a cluster
- MPI messaging

Memory/Cache
- Capacity

Disk
- Disk bandwidth (bonnie)

Installation
- Software and versions
- Environment, variables

CROSS SITE DATA – generally related to services
- Job submission service/Gatekeeper
- File transfers
- SSH
- Low-level network data (ping)

Appendix B: Agenda

May 12, 2004

9:00-10:00
- Registration and breakfast

10:00-10:45 Introduction and Overview
- Jennifer Schopf, Welcome
- Tony Hey and Stephen Jarvis, A Roadmap for Grid Performance: Reporting From the Frontline

10:45-12:15 Cautionary Tales
- Bill Gropp, How Not to Measure Performance: Lessons from Parallel Computing
- Pete Beckman, Grid Performance, From LINPACK to Uptime of a National Grid Infrastructure, a Look at Performance and Metrics for Production Grid

Lunch

1:30-3:00 Application Classes and the Measurements Needed for Them
- Ron Perrot, GT2 and GT3 Performance Timings for Projects at the Belfast eScience Centre
- Ani Thakur, Tracking Database Usage and Measuring Data Mining Performance in the SDSS
- Joel Saltz, Cooperative Biomedical Research, Data Virtualization and Grid Computing

3:30-5:00 Breakout - Measurements
- What do we have? What do we need?

5:00-5:30
- Summary of breakouts

6:30
- Banquet at Navarro's, 67 Charlotte Street, London W1T 4PH

May 13, 2004

8:00-9:00
- Breakfast

9:00-10:30 Performance Evaluation and Benchmarking Today
- Brian Tierney, Grid Troubleshooting Using the Netlogger
- Jack Dongarra, A Look at Some Ideas and Experiments
- Marc Snir, Benchmarks and Canonical Data

11:00-12:30 Science of field measurements applied to the Grid
- Les Cotrell, Network Monitoring Today: Why, How, Challenges, Infrastructures, Federations and the Grid
- Graziano Obertelli, Observing Resources in the Wild
- followed by a 30 min discussion of the topic

Lunch

1:30-3:30 - Breakout
- What is the specific data we need to collect?

3:30-4:00
- Workshop Summary (with snacks)

Thanks

This work was supported in part 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 Contract W-31-109-ENG-38, and by NSFAward #0432288, as well as support from the Joint Information Systems Committee (JISC) and Microsoft.

We are also tremendously grateful to Pete Clarke and the UCL crew for their assistance in local arrangements, networking, and planning.

Workshop Steering Committee:

- Jennifer M. Schopf, Argonne National Laboratory
- Mark Baker, University of Portsmouth
- Fran Berman, UCSD
- Jack Dongarra, University of Tennessee
- Ian Foster, Argonne National Laboratory and the University of Chicago
- Bill Gropp, Argonne National Laboratory
- Tony Hey, UK e-Science Centre

Contact for details: Jennifer M. Schopf jms@mcs.anl.gov