Grid Support for Collaborative Control Room in Fusion Science

K. Keahey,[1] M. E. Papka,[1] Q. Peng,[2] D. Schissel,[2]
G. Abla,[2] T. Araki,[1,3] J. Burruss,[2] E. Feibush,[4] P. Lane,[1]
S. Klasky, [4] T. Leggett,[1] D. McCune,[4] L. Randerson, [4]

[1]Argonne National Laboratory, Argonne, Illinois 60439-4844

[2]General Atomics, P.O. Box 85608, San Diego, California 92186-5608

[3]NEC Internet Systems Research Laboratories, Kanagawa 216-8555, Japan

[4]Princeton Plasma Physics Laboratory, Princeton, New Jersey 08543

Abstract

The National Fusion Collaboratory project seeks to enable fusion scientists to exploit Grid capabilities in support of experimental science. To this end we are exploring the concept of a collaborative control room that harnesses Grid and collaborative technologies to provide an environment in which remote experimental devices, codes, and expertise can interact in real time during an experiment. This concept has the potential to make fusion experiments more efficient by enabling researchers to perform more analysis and by engaging more expertise from a geographically distributed team of scientists and resources. As the realities of software development, talent distribution, and budgets increasingly encourage pooling resources and specialization, we see such environments as a necessary tool for future science.

In this paper, we describe an experimental mock-up of a remote interaction with the DIII-D control room. The collaborative control room was demonstrated at SC03 and later reviewed at an international ITER Grid Workshop. We describe how the combined effect of various technologies—collaborative, visualization, and Grid—can be used effectively in experimental science. Specifically, we describe the Access Grid, experimental data presentation tools, and agreement-based resource management and workflow systems enabling time-bounded end-to-end application execution. We also report on FusionGrid

services whose use during the fusion experimental cycle became possible for the first time thanks to this technology, and we discuss its potential use in future fusion experiments.

Introduction

Over the past decade computational Grids have become a successful tool for the secure and coordinated execution of distributed scientific applications. The National Fusion Collaboratory [1, 2] has adopted the Grid computing paradigm in order to simplify and reduce the costs of sharing codes that are hard to port and maintain. By means of Grid technology, such codes can be remotely executed by members of a virtual organization (VO) [3], using single sign-on to integrate secure access to fusion data in remote databases and other resources.

Nevertheless, leveraging Grid capabilities for experimental sciences poses several challenges. For example, to assist in an ongoing experiment, we need to find ways of delivering results, such as time-critical execution in the Grids, within promised quality of service. This task involves resolving issues of control over resources shared by controlled communities, as well as finding ways to deal with uncertainty and dynamic behaviors typically present in a distributed environment. No less important is the issue of providing satisfactory communication among distributed participants. In fact, to dramatically improve the efficiency of experimental sciences, we need to combine Grid computing with collaboration technologies such as the Access Grid (AG) and application sharing.

The combination of these technologies into a unified scientific research environment is challenging but creates the possibility of increased efficiency of experiments.

In this paper, we describe a collaborative control room experiment, developed as part of the National Fusion Collaboratory project, that unites collaborative, visualization, and Grid technologies and shows how their combined effect can advance experimental science. Specifically, we describe the Access Grid, experimental data presentation tools, and agreement-based resource management and workflow systems enabling time-bounded end-to-end application execution. In addition, we report on fusion services whose use during the fusion experimental cycle became possible for the first time and discuss its potential future impact on fusion science.

This paper is organized as follows. In Section 2 we describe the nature of fusion experiments and both the motivation and the requirements for a collaborative control room. In Section 3 we present our implementation of such a control room as an experiment in collaborative science. In Section 4 we describe the technology developed for this collaborative environment, evaluate its merits, and point to areas of future growth. In Section 5 we conclude with a brief discussion of future work.

Fusion Experimental Cycle

Magnetic fusion experiments operate in a pulsed mode. On any given day, 25–35 plasma pulses are taken with approximately 10 to 20 minutes between each ~10-second pulse. For every plasma pulse, up to 10,000 separate measurements versus time are acquired at sample rates from kilohertz to megahertz, representing about a gigabyte of data. Throughout each experiment, hardware and software plasma control adjustments are made in order to ensure better convergence of an experiment. These adjustments are typically made right before the start of a plasma pulse and are based on data analysis and discussions conducted within the roughly 20-minute between-pulse interval.

Data analysis to support experimental operations includes between pulse analysis of raw acquired data as well as the merging of numerous data sources for whole-device simulation of the experimental plasma. Results of more detailed, computationally demanding predictive simulations, carried out during the planning phase prior to the experiment, are made available for comparison with the actual experimental results in real time.

This mode of operation places a high premium on rapid data analysis that can be assimilated in near-real time. The experimental science can be made more efficient by pushing the boundaries in two directions. First, by running codes on geographically dispersed resources, we can increase the amount and detail of both analysis and simulation results. Second, by bringing in expertise from geographically remote teams of experts, we can increase the depth of interpretation and improve the assimilation of those results.

Computational Grids offer the opportunity to do both; however, new capabilities need to be developed to ensure the completion of time-critical execution within the allotted time frame of the experimental cycle and to deepen the sense of presence shared with remote experts. Thus, in order to be fully functional, the collaborative control room requires (1) secured computational services that can be scheduled to deliver results within the critical time window, (2) the ability to rapidly compare experimental data with simulation results, (3) a means to easily share individual results with a distributed group of experts by moving application windows to a shared display, and (4) the ability for remote scientists to be fully engaged in experimental operations through shared audio, video, and applications.

Moving toward the Collaborative Control Room

The concept of a collaborative control room was formulated in answer to the requirements discussed above. We developed a prototype implementation of the required functionality and conducted a mock-up simulation of the control room interactions as an experiment in collaborative science. The interactions involved remote codes, resources, and scientific teams (see Fig. 1). The experiment was demonstrated at SC03 with collaborators on the SC floor in Phoenix interacting with researchers conducting a mock-up of a DIII-D experiment [4] located in San Diego.

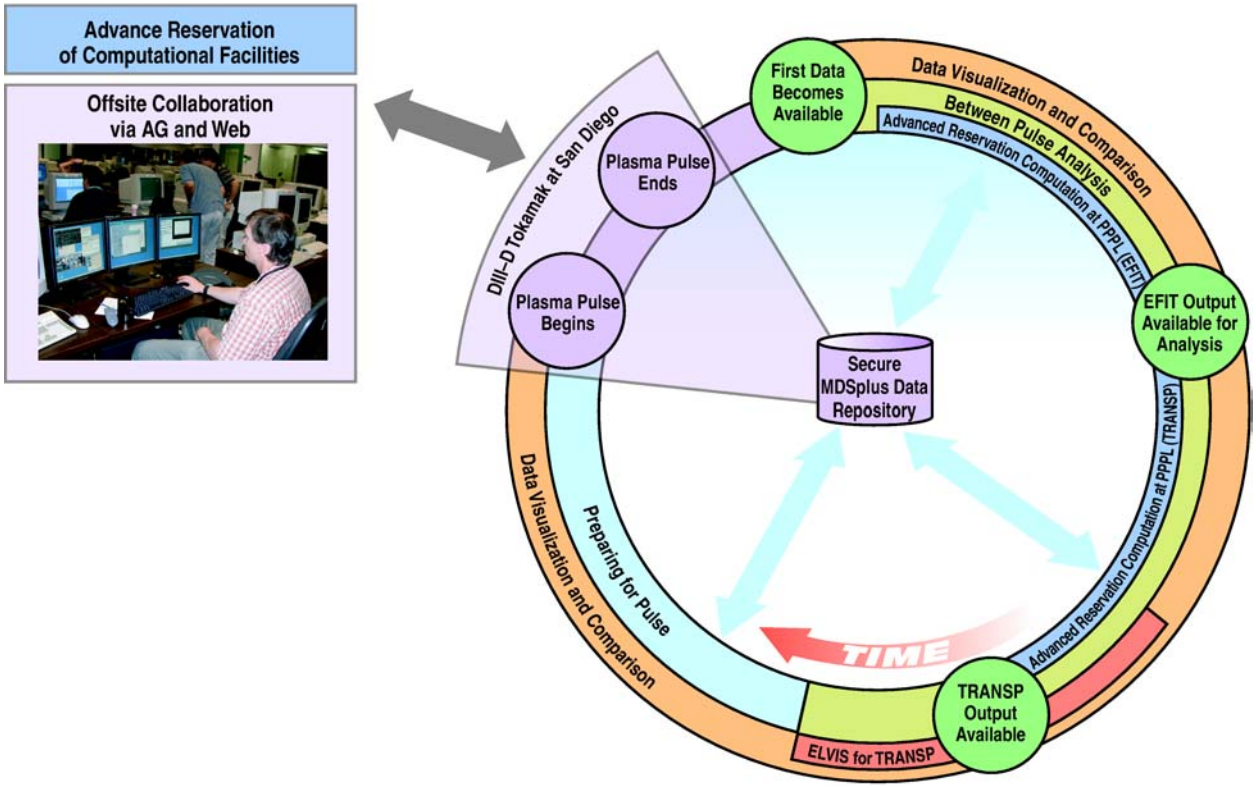

Figure 1: Fusion experimental cycle in the collaborative control room demonstration

AG technology allowed for shared audio and video as well as shared applications. The offsite collaborators could hear DIII-D announcements from both the scientist and the engineer in charge, as well as see via a Web interface the state of the pulse cycle and the status of data acquisition and between-pulse data analysis. The fusion visualization application ReviewPlus [5] was shared between the two sites, allowing for joint scientific exploration of experimental data. Further, the team members were able to share their personal display with the room's shared display, which allowed visualizations to be efficiently compared for discussion.

After each plasma pulse ended, the data was deposited in MDSplus [6], a fusion data repository cataloguing data based on provenance in the fusion experimental cycle. Between-pulse data analysis of the plasma shape (using EFIT [7] running at the Princeton Plasma Physics Laboratory, PPPL) was then conducted on the FusionGrid through an agreement-based system that guaranteed a specific analysis to be completed within a set time window based on a reservation block made before the experiment. After EFIT completed, the TRANSP [8] service was run at PPPL for the first time between pulses, giving the scientists data that was previously available only after the experimental day had ended. Throughout the experiment, relevant raw and processed data was available through MDSplus and used as input to codes, visualization tools, and discussions and comparisons conducted by experts trying to decide on adjustments for the next pulse.

As a result of applying Grid technologies, the fusion scientists were able to run more codes and interact more effectively on reaching decisions affecting steering of the experiment before reporting results back to the DIII-D control room. Although these capabilities were not yet able to influence science, they constituted an important proof of concept and an indication of how future experiments can be transformed. They also enabled fusion scientists to provide valuable feedback on how the technology needs to develop. In the following sections we discuss the collaborative technology and future developments.

Support for the Collaborative Control Room

The collaborative control room was made possible by aggregating existing Grid technologies such as Access Grid and the Globus Toolkit, customizing these technologies for use in the fusion experimental cycle, developing domain-specific tools, and adapting fusion codes not previously used in an experimental cycle to run between pulses. In this section we discuss these technologies and their use.

1.1. Interactions over the Access Grid

The mock-up of the tokamak experimental operation was intended to illustrate how remote scientists can participate fully in the experiment without being at the experimental facility. The Access Grid was used to give the remote scientists the feeling of being part of the control room. The Access Grid is an ensemble of network, computing, and interaction resources that support group-to-group human interaction across the Grid [9]. It consists of large-format multimedia displays, presentation and interactive software environments, interfaces to Grid middleware, and interfaces to remote visualization environments. Access Grid nodes are deployed as “designed spaces” that explicitly support the high-end audio and visual technology needed to provide a high-quality compelling and productive user experience [10]. Access Grid nodes are connected via the Internet, typically using multicast video and audio streams.

The Access Grid enabled the remote scientists interact with the control room at General Atomics (GA) in a natural manner, asking questions of the operators at GA as well as seeing what was going on in the control room as it was happening. This interaction could not have been achieved with a telephone call. However, finer-grained interaction is still needed. Remote scientists do not always want to communicate with the whole control room; instead, they might want to coordinate with only one or two scientists in their specialized field. Currently, the control room audio involves the whole room: the remote scientist can hear everyone in the control room, and everyone in the control room hears the remote scientist. The Access Grid team now is working to enable multiple audio streams within an Access Grid session, allowing one-to-one communication between a remote participant and a control room operator or operators.

Using a set of Web-based scripts, the remote scientists were able to see the state of the pulse cycle, the time left before the next pulse, and the data acquired from the current pulse. Because of the short time frame of each pulse, remote scientists need to get this information as soon as it is available so that they can process it and suggest any changes in parameters for the next pulse. On the SC03 show floor within the Argonne booth, the temporary AG node used three 50” plasma screens as the display surface. However, many remote scientists will not have the luxury of these larger screens and might have only two or three LCD or CRT monitors. Thus, all of the information needs to be presented clearly but also use as little screen real estate as possible so the scientist can still do research effectively without feeling cramped orcluttered. Instead of using multiple Web pages to display the information, a custom application needs to be created that uses tabs and nested windows. This application should also use the Access Grid Toolkit to provide a secure information channel as well as a way of easily putting data collected into the AG venue.

As the data was gathered into the MDSPlus [6] system, the remote scientists were able to open standard data processing and viewing applications, such as ReviewPlus or EFIT viewer, to start the analysis. Once the data points of interest had been identified, the scientists were able to “warp” the application to a region that was shared between Access Grid node and the control room. This area could be seen and interacted with by both groups. This kind of interaction is a huge leap forward from the typical situation in which the scientist calls the control room on the telephone and describes to the operator what is of interest.

VNC was used to handle the remote desktop sharing. An active area of work for the Fusion Collaboratory Project is better integration of the shared desktop into the Access Grid architecture. The objective is twofold: to increase the security associated with the shared session and to improve the response of the shared session, so that the participants feel like they are local to the control room.

1.2. Experimental Data Presentation

The development of a system for experimental data presentation was motivated by an experiment conducted between a scientist at MIT and the DIII-D control room using an AG node and VNC for sharing applications. The lesson learned from this experiment was that connecting only via the AG node is not sufficient. In addition to video and audio from the actual control room, all real-time data displayed there needs to be made available in real time to off-site participants.

To remedy this situation, we developed a Web interface displaying this information to remote participants. The pulse cycle information displayed on the large LED display in the control room by the tokamak control computer is now also written to a Web server that transforms it into a format suitable for display on the Web page. The Web client checks with the Web server periodically and updates the status accordingly. Initial parameters include pulse number, pulse type, state of a pulse cycle, requested magnetic field, requested plasma current, and countdown to the state (if applicable). Integrated in the same display is a quick view of the status of the data acquisition and analysis. Whenever a group of data becomes available, the corresponding indicator changes color.

The status fields are made available by the MDSplus event system that drives the analysis cycle.

Every stage of particular analyses and fault detections can be tracked in real time by using the Data Analysis Monitor [11]. The monitoring system uses Java Servlet technology to accept information from an HTTP posted request; the user interface is provided as an easy-to-use Web page. When the monitoring system receives a new posting, it dynamically creates the HTML and automatically updates the user clients via a server push. The monitoring system is built with the Java Expert System Shell, JESS [12], an expert system shell that utilizes the C Language Integrated Production System, CLIPS [13], to define a set of rules. Each fact that is posted to the monitoring system can then be evaluated by the rules defined in CLIPS. This strategy provides the monitoring system with reasoning capabilities enabling a wide range of customization (for example, for error detection). In addition, the facts being declared are logged to a relational database using Java's JDBC and Sybase's dblib client. The information not only allows for overview evaluation of monitored resources but also enables the monitor to recover information whenever the servlet is reinitialized. Thus, the administrator can recover or update the monitoring system without losing information.

Currently at DIII-D, a few simple plasma waveforms from the plasma control system are displayed in real time in the control room. Upon completion of the pulse, these signals are immediately available in MDSplus. A visualization tool retrieves the signals, generates an image, and makes it available on the Web serve in quasi-real time. Alternatively, the same plasma control signals can be made available to remote participants in real time by “reflecting,” with VNC, what is displayed in the control room. This approach requires separate hardware to ensure that the VNC server not interfere with the performance of the plasma control system.

Furthermore, users have access to the overview and summary information of the experiments of the day from a Web interface. This interface displays the parameters, the contents of the electronic log book that records the status comments made by chief operator, the comments from the scientists who lead the experiments, and so on.

1.3. Agreement-Based Execution

Agreement-based interactions enable a negotiation approach to resource and service management [14-17]. The negotiation process can be viewed as a discovery phase in which clients and providers represent their needs and capabilities to each other. This phase ends when both sides commit. For example, an advance reservation allows a user who has the right to execute on a CPU to claim the execution rights for a specific period of time. From the provider’s perspective, an agreement represents an adaptation and optimization target; from the client’s perspective, it represents a guarantee that future services will be available as required and when required. This mode of resource management has high potential for resolving problems of provisioning in Grid computing and has received much interest lately; a WS-Agreement draft specification [17] is under discussion at the GRAAP working group of the GGF.

To enable fusion scientists to negotiate end-to-end guarantees for the execution of remote codes between experimental pulses, we implemented an agreement-based system loosely based on WS-Agreement. Specifically, our implementation is based on Globus Toolkit 3 (GT3). Agreements are represented as Grid services [18], they are created by factories, are subject to soft-state lifetime management, and enable access to state exposed as service data elements (SDEs). In particular, one of the SDEs exposes the agreement terms describing qualities related to resource brokering, data transfer, or application-specific constructs needed for services. A client can negotiate these terms by trading off, for example, execution time for the number of timesteps an application will run. We adopted a simplified model that negotiates on the level of the whole agreement rather than on specific terms. The negotiation process ends in commitment and the creation of an agreement.

In our system, a client can make agreements for four kinds of service: CPU reservation, job execution, data transfer, and a workflow service that coordinates these services to provide end-to-end execution. The *CPU reservation service* uses an approach similar to GARA [19], using DSRT [20] to reserve and later claim a CPU slice. The *job execution service* depends on the CPU reservation and makes agreements for job execution time based on prediction relying on history of previous runs and resources available as per the

CPU reservation. Agreements for job execution are claimed by using GT3's GRAM job execution service. The *data transfer service* is implemented by using GT3's reliable file transfer service (RFT) [21]. Although in this experiment we integrated data transfer agreements depending on simple prediction, we have also successfully explored a more sophisticated approach combining prediction, rate limiting, and adaptation [22], which would provide more control in future implementations of this system. The *workflow service* is a custom implementation that combines the projected execution and data transfer times to provide an end-to-end execution time.

Agreement-based negotiations are used in the collaborative control room as follows. Before the experiment starts, a scientist negotiates the end-to-end time for a remote execution of a fusion service, such as EFIT. The end-user negotiation is conducted with the workflow service, which in turn negotiates execution times with subsidiary services such as data transfer and job execution. The CPU reservations are made as needed by the job execution service. By tuning the arguments in the end-to-end service description of EFIT, such as the number of timesteps for which the program will execute, the client can influence transfer and execution times; hence, these subsidiary agreements may also have to be renegotiated. This complex renegotiation with multiple services is handled automatically by the workflow service and may, but need not, be exposed to the user. When an acceptable execution time is reached, the end-user commits and obtains the agreement handle, which is then integrated with scripts triggering automatic agreement claiming and execution of the requisite services during the between-pulse interactions.

Our simplified negotiation model worked well in the context of this application. Although most of our agreements are advisory (that is, the provider does not actually commit to resource management or adaptation actions), they still benefit the scientist, who does not have to manually experiment with quantities for remote execution in an environment made more complex by the use of Grids.

We concluded from this experiment that any agreements for interaction in the Grid will require a well-defined set of guarantees. This requirement is universal: even the guarantee of an advance reservation on a specific resource will ultimately depend on the reliability of that particular resource (e.g., its uptime). Especially in a situation where we cannot rely on prior reservation actions, such as is the case with actions strongly or even exclusively relying on prediction, those guarantees have to be quantified. For this reason, we introduced "levels of confidence," in this case modeled as prediction errors or a weighted combination of errors in the workflow case.

We note that some agreements can be made only with a very low level of guarantee (for example, execution times of many applications are hard to predict). In the case of such "underperforming" services, a scientist may want to take charge and, for example, disassociate a resource reservation agreement from an unpromising execution in order to apply it elsewhere. We are now integrating this additional functionality with our system.

1.4. Computational Services

To assess the progress of the experiment, fusion scientists run analysis and simulation codes during the between-pulse period. The core analysis code is the magnetohydrodynamics equilibrium fitting code EFIT [7], first developed in 1985 to perform magnetic and optionally kinetic-magnetic analyses for Doublet III, the predecessor to DIII-D. It was later adapted for the DIII-D National Fusion Facility and many other tokamaks around the world. Written in FORTRAN, it translates measurements from plasma diagnostics, such as external magnetic probes, external poloidal flux loops, and the motional stark effect, into useful information such as plasma geometry, stored energy, and plasma current profiles.

The collaborative control room experiment leveraged access to remote codes and resources enabled by the use of Grids to include runs of the TRANSP code for the first time in a fusion experiment. The between-pulse TRANSP analysis has two main benefits for the experimental physicist: (1) validation of plasma diagnostic measurements and (2) quick assessment of plasma performance. TRANSP directly uses plasma measurements wherever possible; it then simulates expected signals for plasma diagnostics that cannot be used directly. For example, typically, profiles of temperatures and densities of the main thermal plasma species are available, but details of the velocity distribution of superthermal species are not directly measured. The total plasma neutron production, an indicator of the total fusion reaction rate, is measured and depends on the superthermal distribution. By using the measured temperatures and densities, TRANSP can often simulate the superthermal distribution with accuracy sufficient to match the observed neutron rate. However, the match will work only if all the input data are correct. Thus, failure to match can be an early indicator of diagnostic problems that, if undetected, can render the day's experimental results unusable. If the match succeeds, then TRANSP's assessment of plasma performance can be used with confidence.

TRANSP relies on the mapping from "real space" coordinates to "magnetic flux space" coordinates performed by EFIT and therefore has to follow EFIT execution in the cycle of codes run between pulses. This situation further limits the amount of time that can be budgeted for those codes. Thus, in preparation for the experiment, significant work was done to reduce TRANSP run production time, through both software and hardware changes, to about six minutes, which was found to be acceptable for an experimental run. The actual TRANSP run time was slightly over three minutes; the balance of the time was due to network data transfers. These data transfer delays will be reduced through further optimization of the software.

As was demonstrated at SC03, an Internet-accessible Java-based graphical monitoring tool, ElVis [23], is available to display results from remote simulations as they are computed. The ElVis monitoring not only shows that the remote computational service is operating but it also allows select results to be made available in the control room or at collaborator sites even before the run is completed.

In this first attempt, only one timeslice of the experimental data was run. In principle it would be best to run a fully time-dependent TRANSP simulation. However, such calculations cannot be parallelized over time, moreover, they require fully prepared time-dependent input datasets and produce fully time-dependent output datasets—fairly complicated objects that would be a challenge to digest between pulses even if all technical barriers were overcome. However, we could make a better use of the access to computational resources by running numerous (say, 10–20) timeslice simulations, all of which are independent and could be carried out in parallel.

Conclusions and Future Work

The collaborative control room demonstration described here represents a step forward in a series of experiments striving to harness the computational power of Grids for experimental science. While the demonstration was well received by the fusion community, much research still needs to be done to generalize the infrastructure and make it applicable to real-life experiments. The lessons learned and requirements described here define directions for future research.

As we move toward a scientific future that requires pooling of resources on international scale, collaborative environments such as the National Fusion Collaboratory described here will become a necessity for many communities. Thus, the exploration and scaling of such environments acquire a new urgency and significance.

Acknowledgement

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, SciDAC Program, U.S. Department of Energy, under Contract W-31-109-ENG-38.

References

1. Keahey, K., T. Fredian, Q. Peng, D.P. Schissel, M. Thompson, I. Foster, M. Greenwald, and D. McCune, *Computational Grids in Action: the National Fusion Collaboratory.* Future Generation Computing Systems, October 2002. **18**(8): p. 1005-1015.
2. Schissel, D.P., *Building the U.S. National Fusion Grid: Results from the National Fusion Collaboratory Project.* 4th IAEA Tech. Mtg. on Control, Data Acquisition, and Remote Participation for Fusion Research, July 21-23, 2003, San Diego, California, to be published in Fusion Engineering and Design.
3. Foster, I., C. Kesselman, and S. Tuecke, *The Anatomy of the Grid: Enabling Scalable Virtual Organizations.* International Journal of High Performance Computing Applications, 2001. **15**(3): p. 200-222.
4. Luxon, J.L., N.H. Brooks, L.G. Davis, R.K. Fisher, F.J. Helton, M.A. Mahdavi, J. Rawls, R.D. Stambaugh, T.S. Taylor, and J.C. Wesley. in *11th European Conference on Controlled Fusion and Plasma Physics*. 1983. Aachen, Federal Republic of Germany: European Physical Society.

5. Schachter, J., Q. Peng, and D.P. Schissel, *Data Analysis Software Tools for Enhanced Collaboration at the DIII-D National Fusion Facility.* proceedings of the 2nd IAEA Technical Committee Meeting on Control, Data Acquisition, and Remote Participation on Fusion Research, 1999.
6. Fredian, T. and J. Stillerman, *MDSplus Remote Collaboration Support --- Internet and the World Wide Web.* Fusion Engineering and Design, 1999. **43**: p. 327-334.
7. Lao, L.L., H. St. John, R.D. Stambaugh, A.G. Kellman, and W. Pfeiffer, *Reconstruction of Current Profile Parameters and Plasma Shapes in Tokamaks.* Nucl. Fusion, 1985. **25**: p. 1611.
8. Hawryluk, R.J., *An Empirical Approach to Tokamak Transport.* Phys. Plasmas Close to Thermonuclear Conditions, ed. by B. Coppi, et al., (CEC, Brussels, 1980), 1980. **1**: p. 19.
9. Stevens, R., *Access Grid: Enabling Group Oriented Collaboration on the Grid*, in *The Grid: Blueprint for a New Computing Infrastructure*. 2003, Morgan Kaufman. p. 191-199.
10. Childers, L., T. Disz, M. Hereld, R. Hudson, I. Judson, R. Olson, M.E. Papka, J. Paris, and R. Stevens, *ActiveSpaces on the Grid: The Construction of Advanced Visualization and Interaction Environments*, in *Simulation and Visualization on the Grid: Lecture Notes in Computational Science and Engineering 13*, M. Griebel, et al., Editors. 2000: Berlin. p. 64-80.
11. Flanagan, S.M., J.R. Burruss, C. Ludescher, D. McCune, Q. Peng, L. Randerson, and D.P. Schissel, *A General Purpose Data Analysis Monitoring System With Case Studies from the National Fusion Grid and the DIII-D MDSplus Between Pulse Analysis System.* Fusion Engineering and Design, 2004. **71**(1-4): p. 263-267.
12. Friedman-Hill, E.J., *Users Guide for Jess, The Java Expert System Shell.* http://herzberg.ca.sandia.gov/jess/, 2001.
13. Giarrantano, J.S., *CLIPS User's Guide.* http://www.ghg.net/clips/CLIPS.html, 1998.
14. K. Czajkowski, I. Foster, C. Kesselman, V. Sander, and S. Tuecke, *SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems.* 8th Workshop on Job Scheduling Strategies for Parallel Processing, July 2002.
15. Ludwig, H., A. Keller, A. Dan, and R.P. King, *A Service Level Agreement Language for Dynamic Electronic Services.* IBM Research Report RC22316 (W0201-112), January 24, 2002.
16. Keahey, K., T. Araki, and P. Lane. *Agreement-Based Interactions for Experimental Science*. in *Europar*. 2004.

17. Andrieux, A., K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, *Web Services Agreement Specification (WS-Agreement) Draft 20*. 2004: https://forge.gridforum.org/projects/graap-wg/.
18. Foster, I., C. Kesselman, J. Nick, and S. Tuecke, *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*. 2002: Open Grid Service Infrastructure WG, Global Grid Forum,.
19. Foster, I., C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy. *A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation*. in *Proc. International Workshop on Quality of Service*. 1999.
20. Nahrstedt, K., H. Chu, and S. Narayan. *QoS-aware Resource Management for Distributed Multimedia Applications*. in *Journal on High-Speed Networking, IOS Press*. December 1998.
21. Madduri, R., C. Hood, and W. Allcock, *Reliable File Transfer in Grid Environments.* LCN, 2002: p. 737-738.
22. Zhang, H., K. Keahey, and B. Allcock, *Providing Data Transfer with QoS as Agreement-Based Service.* submitted to IEEE International Conference on Services Computing (SCC 2004), 2004.
23. Feibush, E., *ElVis*. 2003: http://www.pppl.gov/~efeibush/elvis.