Grids for Experimental Science: The Virtual Control Room

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

[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*

[A]*keahey@mcs.anl.gov*, [B]*papka@mcs.anl.gov*, [C]*peng@fusion.gat.com*,
[D]*schissel@fusion.gat.com*, [E]*abla@fusion.gat.com*, [F]*araki@mcs.anl.gov*,
[G]*burrus@fusion.gat.com*, [H]*sfeibush@pppl.gov*, [I]*lane@mcs.anl.gov*, [J]*sklasky@pppl.gov*,
[K]*leggett@mcs.anl.gov*, [L]*dmccune@pppl.gov*, [M]*lranderson@pppl.gov*
,

Abstract

The National Fusion Collaboratory focuses on enabling fusion scientists to explore Grid capabilities in support of experimental science. Fusion experiments are structured as a series of plasma pulses initiated roughly every 20 minutes. In the between-pulse intervals scientists perform data analysis and discuss results to reach decisions affecting changes to the next plasma pulse. This interaction can be made more efficient by performing more analysis and engaging more expertise from a geographically distributed team of scientists and resources. In this paper, we describe a virtual control room experiment that unites collaborative, visualization, and Grid technologies to provide such environment and shows how their combined effect can advance experimental science. We also report on FusionGrid services whose use during the fusion experimental cycle became possible for the first time thanks to this technology. We also describe the Access Grid, experimental data presentation tools, and agreement-based resource management and workflow systems enabling time-bounded end-to-end application execution. The first virtual control room experiment represented a mock-up of a remote interaction with the DIII-D control room and was presented at SC03 and later reviewed at an international ITER Grid Workshop.

1. Introduction

Computational Grids are of proven value to the scientific community. The National Fusion Collaboratory (NFC) [1, 2] has benefited from Grid capabilities by allowing the adoption of the application service provider (ASP) model so that codes can be shared by members of a virtual organization (VO) [3] as remotely accessible network services. NFC, as well as other communities, is benefiting from Grid tools developed for resource management, data management, and transport, security, and information services. Although the existing infrastructure works well, additional capabilities must be developed to enable Grid computing to take on another challenge: implementing interactions for experimental science.

Leveraging Grid capabilities for experimental sciences poses unique 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 (QoS). This task involves resolving issues of control over resources shared by more or less controlled communities as well as finding ways to deal with uncertainty and dynamic behaviors always present in a distributed environment. No less important is the issue of providing satisfactory communication between 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 poses unique challenges but creates the possibilities of high reward in the form of increased efficiency of experiments.

In this paper, we describe a virtual control room experiment that unites collaborative, visualization, and Grid technologies and shows how their combined effect can advance experimental science. We discuss advances in technology driven by the National Fusion Collaboratory that made it possible. Specifically, we describe the AG, 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 fusion experimental cycle became possible for the first time thanks to this technology and discuss its potential future impact on fusion science.

This paper is organized as follows. Section 2 describes the nature of fusion experiments and motivates the vision of a virtual control room. Section 3 describes our implementation of a virtual control room as an experiment in collaborative science. Section 4 describes the technology that was developed to make this vision possible, discusses its merits and points to areas of future growth. We conclude in Section 5 with a brief discussion of future work.

2. Setting the Stage: Interactions within the Control Room and Their Requirements

Magnetic fusion experiments operate in a pulsed mode. In 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 kHz to MHz, representing about a gigabyte of data. Throughout the experimental session, hardware/software plasma control adjustments are made as required by the experimental science. These adjustments are debated and discussed among the experimental team. Decisions for changes to the next pulse are informed by data analysis 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 to the actual experimental results in real time.

This mode of operation places a large 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. In order to be fully functional, the collaborative control room requires (1) secured computational services that can be scheduled as required, (2) the ability to rapidly compare experimental data with simulation results, (3) a means to easily share individual results with the group 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.

3. Virtual Control Room Experiment

The vision of the virtual control room was developed 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 in the experiment. The experiment was demonstrated at SC03. The interactions are depicted in Figure 1 and described below.

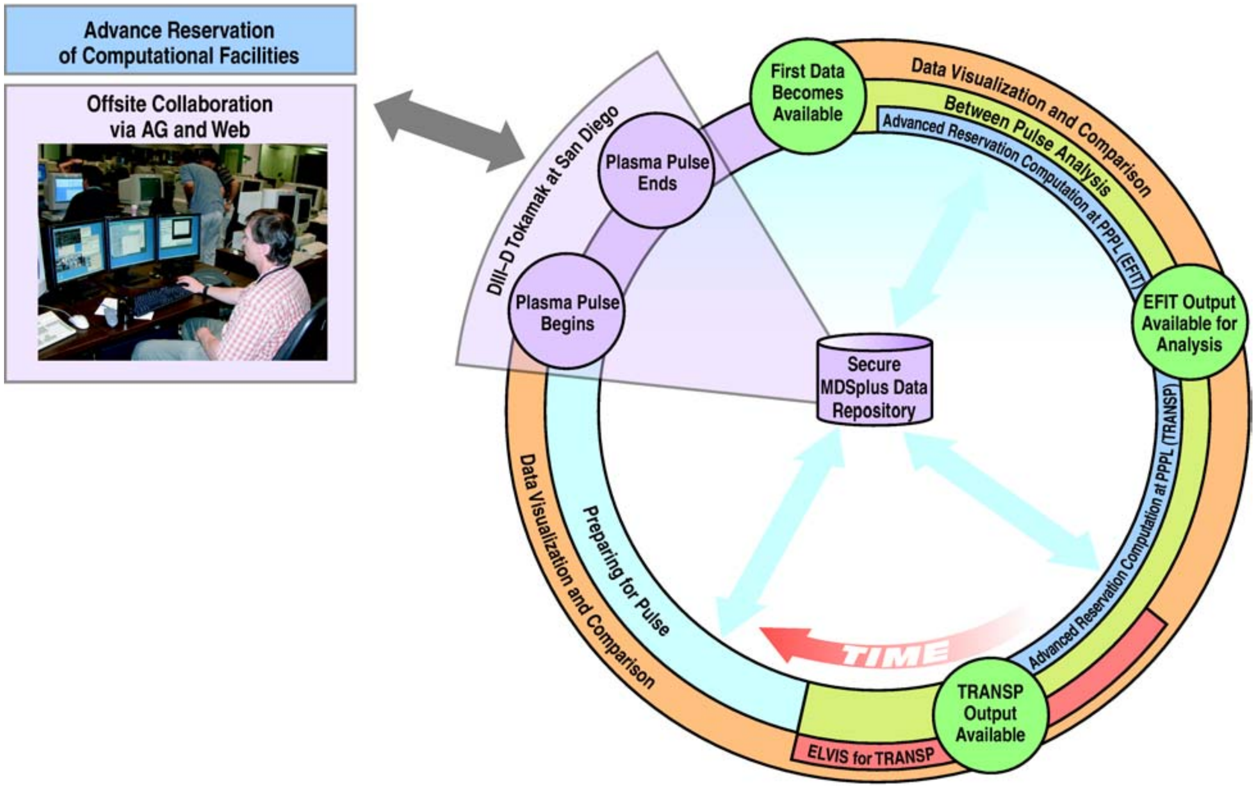

Figure 1: SC03 experiment: the virtual control room interaction cycle

Offsite collaborators (SC floor, Phoenix) joined in a mockup of a DIII–D experiment [4] located in San Diego. 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 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. Between-pulse data analysis of the plasma shape (EFIT [6] running at PPPL) was conducted on FusionGrid through a computational reservation system that guaranteed a specific analysis to be completed within a set time window. Additionally, the TRANSP [7] 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. The offsite team members were able to collaborate more efficiently by being able to share their personal display with the room's shared display. This capability allowed visualizations to be efficiently compared for debate before reporting results back to the DIII-D control room. The results of this demonstration and the feedback from fusion scientists has helped sharpen the requirements for a truly collaborative control room for fusion experiments.

4. Technology behind the Control Room

This section describes the technology used to implement the interactions in the control room.

4.1. Interactions over the Access Grid

With the goal of showing a mockup of tokamak experimental operation, we were trying to illustrate how a remote scientist could participate fully in the experiment without actually being at the experimental facility. The Access Grid was used to give the remote scientist the feeling of being part of the control room at a distance. The Access Grid is an ensemble of network, computing, and interaction resources that support group-to-group human interaction across the Grid [8]. 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 into "designed spaces" that explicitly support the high-end audio and visual technology needed to provide a high-quality compelling and productive user experience [9]. Access Grid nodes are connected via the high-speed Internet, typically using multicast video and audio streams.

First and foremost the Access Grid enabled the remote scientist to talk to and see the control room at GA, enabling the remote scientist to ask questions of the operators there as well as see what was going on in the control room as it was happening. This could never be achieved with just a telephone call. However, there is still some need for more fine-grained interaction. Remote scientists do not always need to communicate with the whole control room. Instead, they might need to coordinate with only one or two scientists who are working in their specialized field. Currently, the control room audio is of the whole room. What that means is that everyone in the control room can be heard by the remote scientist and everyone in the control room hears the remote scientist. As a future effort aimed at the specific needs of the control room scenario, the Access Grid teams 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, remote scientists were able to see the state of the pulse cycle, how much time was left before the next pulse, and what data had been acquired from the current pulse. Because of the short time frame of each pulse, it is crucial for the remote scientists to get this information as soon as it's available so that they can process it and suggest any changes in parameters for the next pulse. On the SC03 show floor within the ANL booth a temporary AG node was built that had three 50" plasma screens as a display surface. This provided us with a good amount of screen real estate. However, many remote scientists will not have the luxury of these and might have only 2 or 3 LCD or CRT monitors. That means that all of this information needs to be presented clearly but also use as little screen real estate as possible so the scientist can still effectively do research without feeling cramped and cluttered. Instead of using multiple Web pages to display all of this, an easily created custom application needs to be created that has more real estate-friendly UI using 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 dumping data collected into the AG venue.

As the data is gathered into the MDSPlus [10] system, the remote scientist was able to open standard data processing and viewing applications such as ReviewPlus or EFIT viewer to start the analyzing process. Once remote scientist identified data points of interest, they were able to "warp" the application to a region that was shared between Access Grid node and the control room. This area can be seen and interacted with by both parties. This kind of interaction is a huge leap forward from the present situation where the scientist calls the control room on the telephone and describes what he or she is interested in to the operator. VNC is 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 and advance VNC for better compression to enable even the slowest participants to feel like they're local to the control room.

4.2. Experimental Data Presentation

Prior to defining the collaborative control room and presenting the SC03 demonstration, a collaborative experiment was conducted between a remote scientist at MIT and the DIII-D control room. Using an AG node and VNC for sharing applications, the remote MIT scientist tried to effectively participate with a live DIII-D experiment. What was 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 that is displayed there needs to be made available in real time to off-site participants. As a result, this information has now been made available through a Web interface to remote participants. The tokamak control computer has the pulse cycle information and plasma control parameters. As it sends the information to the large LED display in the control room, it also writes the same data directly to a Web server that parses them 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 indicating where in a pulse cycle, requested magnetic field, requested plasma current, and countdown if applicable to the state. Integrated in the same display is a quick view of the data acquisition and analysis status. Whenever a group of data becomes available, the corresponding indicator changes color. The statuses are made available by the MDSplus event system that drives the analysis cycle.

The detailed data analysis status including every stage of particular analyses and fault detections can be tracked in real time using the Data Analysis Monitor (DAM) [11]. The monitoring system uses Java Servlet technology to accept information from an HTTP post request, which allows the user interface to be provided as an easy-to-use Web page. When the monitoring system receives a new post, it dynamically creates the HTML and automatically updates the user clients via server push. The monitoring system is also built with the Java Expert System Shell, Jess [12]. Jess is a rules-based expert system shell that utilizes the C Language Integrated Production System, CLIPS [13], in order 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 provides the monitoring system with reasoning capabilities. The functionality enables a wide range of custom design to be done with the monitor, such as customized 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, thereby giving the administrator the ability to recover or update the monitoring system without losing information.

At present 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 exact signals, generates an image and makes it available on the Web server. The implementation makes the signals available on the Web in a quasi-real-time fashion. Alternatively, the same plasma control signals can be made available to remote participants in real time by "reflecting" what is displayed in the control room. The "reflecting" can be accomplished with VNC. This requires separate hardware lest the VNC server may 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. It has links to the proposals of the experiments. It displays the summary 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.

4.3. Agreement-based Between-Pulse Execution System

Agreements-based interactions enable negotiation enable negotiation approach to resource and service management [14-16]. 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. A committed agreement can be viewed as a concretization of use policy representing a relationship between a client and a provider. From the providers perspective, an agreement represents an adaptation and optimization target; from the client's it represents a form of guarantee that future services will happen as required and when required. This mode of interaction has high potential for resolving problems of provisioning in Grid computing and has received much interest lately; a WS-Agreement draft specification [9] is currently under discussion at the GRAAP working group of the GGF.

According to the specification, agreements are represented as Grid services [17]: they are created by factories, 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. Those terms are assumed to be domain-specific as needed for resource brokering, data transfer or application-specific constructs. Once created based on a set of initial terms, an agreement is subject to negotiation between client and provider till both parties commit. Claiming an agreement involves performing the promised actions either by creating an application service or by influencing events already in progress.

We implemented agreement-based interactions in Globus Toolkit 3 (GT3) to enable fusion scientists to negotiate end-to-end guarantees on execution of remote codes between the experimental pulses. While our implementation influenced, and was influenced by, WS-Agreement [16], our use case did not require a full implementation of it. We adopted a simplified negotiation and commitment model and focused on defining terms and functionality required by our application and practical experiences with the system.

In our system, a client can make agreements for four kinds of services: 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 approach similar to GARA [18] using DSRT [6] to reserve and later claim a CPU slice. The job execution service depends on the CPU reservation and makes agreements of 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 using GT3's GRAM job execution service. The data transfer service is implemented using GT3's reliable file transfer service (RFT) [19], and its agreements for data transfer times likewise depend on prediction. The workflow service combines the projected execution and data transfer times to provide an end-to-end execution time.

The agreement-based negotiations are used in the virtual 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 service description of EFIT, such as the number of timesteps for which the program will execute, the user can effect transfer and execution times so that these agreements may 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 specific adaptation and other resource management 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. Moreover, the use of agreements presents a framework for building up the resource management capabilities required to provide stronger guarantees for between-pulse execution. At the same time, one of the conclusions prompted by this experiment is that any agreements for interaction in the Grid will require a well-defined set of guarantees. Especially in a situation where we cannot rely on prior reservation actions, such as is the case with data transfer over the Internet, those guarantees have to be strongly quantified. For this reason, we introduced levels of confidence, modeled as prediction errors, or a weighted combination of errors in the workflow case and we are currently trying to generalize this concept in discussions on WS-Agreement.

4.4. 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 (MHD) equilibrium fitting code EFIT [20], 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. It is written in FORTRAN and 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 virtual 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 makes direct use of 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 super-thermal 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 super-thermal 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, which if undetected can cause the day's experimental results to be rendered 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 places further limitation on the amount of time that could 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 6 minutes, which was found to be acceptable for an experimental run. The actual TRANSP run execution time was slightly over 3 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 [21], 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; 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 an analysis of 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, and, 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 can to make a better use of the access to computational resources to run numerous (say, 10–20) timeslice simulations, all of which are independent and could be carried out in parallel.

5. Conclusions and Future Work

This prototype of the virtual control room was demonstrated at the SC03 conference in November 2003 and later reviewed at an international ITER Grid Workshop. The experiment was well received and we are planning to make adjustments and improvements moving toward the future deployment of the described infrastructure in the control room. At the same time, we are aware that much research still needs to be done to generalize the developed infrastructure and make it applicable to more than a fixed number of cases.

The U.S. participation in ITER opened another challenge and at the same time an opportunity to use systems such as described in this paper. With an international effort to build an experimental facility, collaborative use of this facility will have to follow. We are hoping that our work will contribute to the development of an internationally viable model for widely distributed collaborations.

Acknowledgments

This work was supported by the DOE SciDAC project, 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 (to appear), 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. J.L. Luxon, N.H. Brooks, L.G. Davis, R. K. Fisher, F.J. Helton, M.A. Mahdavi, J. Rawls, R.D. Stambaugh, T.S. Taylor, J.C. Wesley, 11th European Conference on Controlled Fusion and Plasma Physics, 5–9 September 1983, Aachen, Federal Republic of Germany (European Physical Society, Aachen, 1983).
5. J. Schachter, Q. Peng, D.P. Schissel, "Data Analysis Software Tools for Enhanced Collaboration at the DIII–D National Fusion Facility," published in the proceedings of the 2nd IAEA Technical Committee Meeting on Control, Data Acquisition, and Remote Participation on Fusion Research (1999).
6. 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.
7. R.J. Hawryluk, "An Empirical Approach to Tokamak Transport," Phys. Plasmas Close to Thermonuclear Conditions, ed. by B. Coppi, et al., (CEC, Brussels, 1980), Vol. 1, p. 19.
8. R. Stevens and the Futures Lab, "Access Grid: Enabling Group Oriented Collaboration on the Grid," in *The Grid: Blueprint for a New Computing Infrastructure*, *Grid Computing*, I. Foster and C. Kesselman, Eds., Second ed: Morgan Kaufman, 2003, pp. 191-199.
9. L. Childers, T. L. 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 *Parallelldatorcentrum Kungl Tekniska Högskolan Seventh Annual Conference (Simulation and Visualization on the Grid)*, vol. 13, *Lecture Notes in Computational*

Science and Engineering, B. Engquist, L. Johnsson, M. Hammill, and F. Short, Eds. Stockholm, Sweden: Springer-Verlag, 1999, pp. 64-80.

10. T. Fredian, J. Stillerman, "MDSplus Remote Collaboration Support–Internet and World Wide Web," Fusion Engineering and Design **43** (1999) 327.
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.* IAEA.
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. Czajkowski, K., A. Dan, J. Rofrano, S. Tuecke, and M. Xu, *Agreement-based Grid Service Management (OGSI-Agreement) Version 0.* https://forge.gridforum.org/projects/graap-wg/document/Draft_OGSI-Agreement_Specification/en/1/Draft_OGSI-Agreement_Specification.doc, 2003.
17. 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,.
18. 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.
19. Ravi K. Madduri, Cynthia S. Hood, William E. Allcock: *Reliable File Transfer in Grid Environments.* LCN 2002: 737-738
20. 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.
21. http://w3.pppl.gov/~efeibush/elvis