Real-Time Analysis, Visualization, and Steering of Microtomography Experiments at Photon Sources

Gregor von Laszewski[*] Joseph A. Insley[*] Ian Foster[*] John Bresnahan[*]
Carl Kesselman[†] Mei Su[†] Marcus Thiebaux[†] Mark L. Rivers[‡]
Steve Wang [§] Brian Tieman[‡] Ian McNulty[‡]

Abstract

A new generation of specialized scientific instruments called synchrotron light sources allow the imaging of materials at very fine scales. However, in contrast to a traditional microscope, interactive use has not previously been possible because of the large amounts of data generated and the considerable computation required translating this data into a useful image. We describe a new software architecture that uses high-speed networks and supercomputers to enable quasi-real-time and hence interactive analysis of synchrotron light source data. This architecture uses technologies provided by the Globus "computational grid" toolkit to allow dynamic creation of a reconstruction pipeline that transfers data from a synchrotron source beamline to a preprocessing station, next to a parallel reconstruction system, and then to multiple visualization stations. Collaborative analysis tools allow multiple users to control data visualization. As a result, local and remote scientists can see and discuss preliminary results just minutes after data collection starts. The implications for more efficient use of this scarce resource and for more effective science appear tremendous.

1 Introduction

The use of x-rays as a nondestructive tool for investigating the internal structure of materials at the micron length scale has grown rapidly over the past decade as a result of the advent of synchrotron radiation sources. In a typical computed microtomography (CMT) experiment, a sample is illuminated by a colliminated beam of x-rays, and data is collected for multiple sample orientations by using a charge-coupled device (CCD).

A time-consuming reconstruction process is then used to obtain a three-dimensional (3-D) raw data with spatial resolution of as little as 1 μm. The three-dimensional image contains quantitative information on the x-ray attenuation coefficient at a particular x-ray energy.

The many orders of magnitude increase in brilliance now available at third-generation sources such as the Advanced Photon Source (APS) allows dramatic improvements in temporal resolution and makes it feasible to record fully 3-D, high-resolution tomographic data on time scales of less than a second per image. This new capability of synchrotrons makes it possible to perform quasi-real-time 3-D imaging.

[*]MCS, Argonne National Laboratory, Argonne, IL, http://www.mcs.anl.gov/xray
[†]USC/Information Sciences Institute, Marina del Rey, CA
[‡]Consortium for Advanced Radiation Sources, The University of Chicago, IL
[§]APS, Argonne National Laboratory , Argonne, IL

However, the effective use of these capabilities requires the ability to archive, analyze, and visualize the collected data at orders of magnitude more than is currently possible.

This paper describes the innovations in methods, algorithms, and software that allow this data to be utilized fully by scientists.

The data rates and compute power required to address this Grand Challenge problem are prodigious, easily reaching one gigabit per second and a teraflop per second. We illustrate this statement with a scenario. A 3-D raw data set generated by a typical detector will comprise 1000 1024×1500 two-byte slices (3 GB); detectors with significantly higher resolutions will soon be available. If we assume current reconstruction techniques and make fairly optimistic scaling assumptions, reconstruction of this dataset requires about 10^{13} floating-point operations (10 Tflops). On a 100 Mflop/sec workstation, this translates to 32 hours; on a 1 Tflop/sec computer, it would take 10 seconds. With current detector technologies, this dataset might take 1500 seconds to acquire; however, new detectors will improve readout times considerably.

We have developed a high-performance parallel implementation of reconstruction algorithms for microtomographic datasets, based on the filtered backprojection techniques currently used for reconstruction. Filtered backprojection techniques parallelize nicely, in that each slice in a dataset can be processed independently. Hence, the principal challenge is to develop efficient techniques for moving data between detector, secondary storage, and computer and for visualizing the results. A general framework has been developed to map the high-performance parallel application in a portable fashion on a grid-enabled environment.

The paper is structured as follows: First, we analyze the specific requirements of x-ray microtomography experiments. This includes an exemplary walk-through of an experiment conducted at a beamline. Then, we describe a general framework for processing data gathered during such experiments. The unique characteristics of the experimentation environment make it necessary to use distributed and remote resources, as available in computational grids. We explain the concept of the computational grid and describe how we implemented our framework with the help of the Globus metacomputing toolkit. The capabilities of the resulting analysis environment enable new acquisition and reconstruction strategies, as introduced next. The details of the collaborative tomography environment are described in the next section. Finally, we conclude the paper by summarizing our contributions to the research community.

2 X-Ray Microtomography Requirements

The design of the computational environment is shaped by an analysis based on the requirements of the x-ray microtomography experiments.

There is a long history of imaging with x-rays. Three-dimensional x-ray imaging (CAT scan) was developed in radiology in the 1960s and 1970s. With the introduction of synchrotron x-ray sources, microtomography is now routinely performed by using hard x-rays at a few μm resolution, while nanotomography has been developed with soft x-rays for imaging biological[12, 15] and microfabricated objects[10] at 10-100 nm resolution.

A typical computed microtomography (CMT) experiment is structured as follows:

1. Conduct the experiment planning.

2. Prepare the samples and the tomography hardware at the beamline.

3. Perform the data acquisition.

4. Choose parameters for the reconstruction.

5. Perform a reconstruction.

   (a) Select a reconstruction algorithm.

   (b) Perform a preprocessing step, which does normalization and prepares so-called sinograms[12]

   (c) Perform the actual reconstruction.

6. Visualize the reconstructed data.

7. Go to step 4, or, if no parameters can be found, repeat the experiment and go to step 3.

The traditional usage model to perform these steps features hands-on use and batch processing. A major difficulty with current practice is the turnaround time between the data acquisition and the reconstruction, often due to lack of available computing power. This is especially problematic for synchrotron-based experiments because only a limited amount of beam time is available for a user. The use of supercomputing power can reduce this turnaround time to a few hours or minutes, as demonstrated at the SC'98 conference in Florida, allowing the users to view the results in quasi-real time gives the opportunities to alter experiment conditions on the fly. This capability greatly improves the capabilities of a synchrotron radiation facility.

To characterize the experiment requirements more precisely, we identified two operation modes: (1) the use of an online operation mode as described above and (2) the use of a post-processing mode, which reconstructs the dataset with a varying parameter set. If enough computational power is available, recalculation can also be performed during an experiment. To enable fast processing, the data must be shipped from the acquisition hardware to the computer doing the reconstruction, preferably on storage media with fast access. At the same time it is important to enable an archival service for the experimentalist in order to allow for data recovery in the case of data loss. As pointed out previously, beam time is limited, and the goal for an experimentalist is to achieve the most in this limited time. A fast reconstruction algorithm can be used to help decide whether the current experiment has to be interrupted prematurely because of an error in the setup. This will allow for an increase in the number of experiments to be conducted per hour. In order to handle the complicated and diverse supercomputing environments, it is essential to provide a simple interface giving the beamline experimentalist control over the parameter set, as well as the possibility to terminate the current calculation at any time.

Besides the requirements driven by the computational aspect of the application, organizational aspects benefit from the framework introduced in this paper. Because of the hazardous and often unpleasant environment, remote operation is desirable. With remote operation, the facility can maintain a small but well-trained team of beamline staff experimentalists. This approach offers several benefits. It reduces the operational and user-specific cost and minimizes travel cost to the unique facility. It allows new user groups to gain access to a unique facility such as the APS. Furthermore, it increases the access time to the beamline while minimizing the effort required by trained experts to set up experiments.

With the availability of a collaborative and remote steering environment, new user communities in commercial and educational facilities are likely to utilize the supercomputing-enhanced light sources in remote fashion. Multiple users at geographically disperse locations should be able to collaborate easily with each other. A "plug-and-play" framework as shown in Figure 1, makes it possible for authorized participants in different geographical locations with different visualization engines to attend a collaborative computed microtomography experiment session.

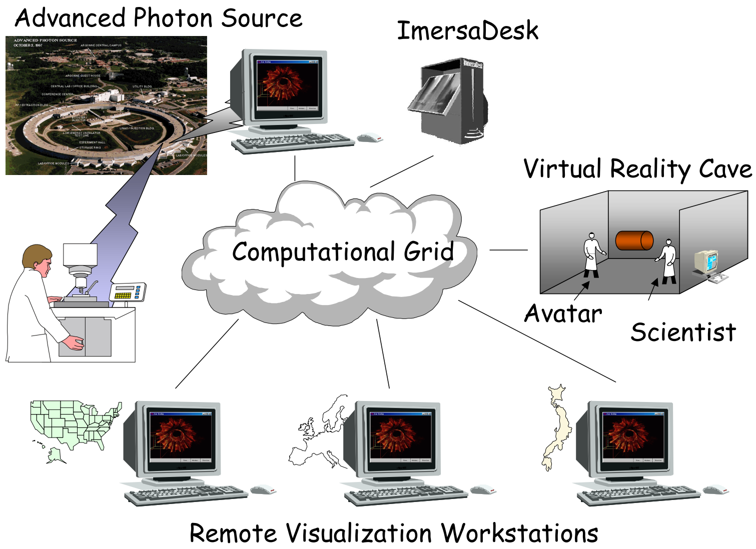

FIG. 1. The "grid-enabled" CMT application allows researchers to display the same state of the visualized object on all display stations participating in a collaborative session[14]. Remote computation and steering become possible.

3 The Computational Processing Pipeline Framework

The general framework of the CMT application is based on the concept of a pipeline. The pipeline consists of a series of components that can communicate with each other via input and output channels. The different components of the system are introduced by following through the steps of a typical experiment as depicted in Figure 2. Each of the components can be mapped in principle on different computers. Thus the framework supports a distributed, networked compute environment.

After the experiment is started on a beamline control PC, the data is transferred to a supercomputer for further processing. In addition, the data is stored locally on an external storage tape in HDF format. After the image is constructed with a parallel algorithm based on the filtered backprojection algorithm, the resulting images are shipped in real time to a visualization unit to gradually update a 3-D rendered image during the experiment. This gradual update is important to allow decisions as to whether the experiment should continue. If the experiment does not perform as expected, it is terminated. Once a better experiment setup is found, the experiment is run to completion. Often the data must be postprocessed and the results discussed with other scientists. One goal of our project is to enable researchers to participate in an experiment at their home institution rather than to travel to the APS. A beamline scientist will be able to handle the experiment locally while communicating with the scientist designing the experiment. A remote video conferencing tool allows such collaboration. In addition, the 3-D image analysis tool contains a control component that enables collaborative visualization on a variety of output devices including graphics workstations, ImmersaDesks, and CAVEs, which are described in a later section. The same capability can be used to achieve real-time computational steering of the experiment.

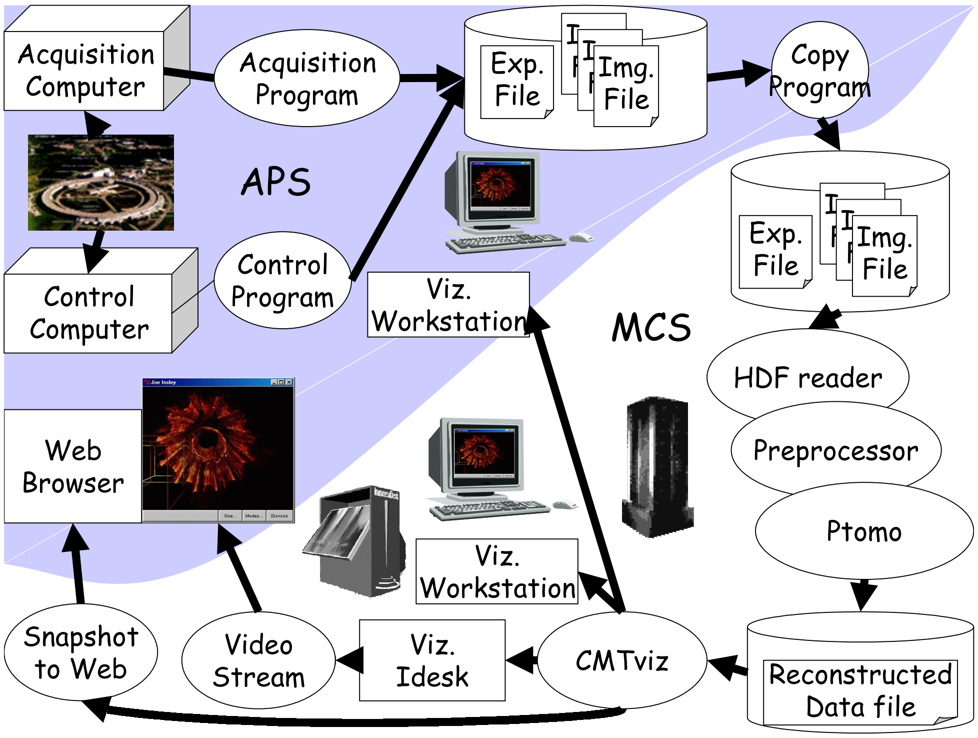

FIG. 2. The details of the processing pipline as used in the SC'98 experiment setup.

The details of this infrastructure will be hidden from the end user, the microtomography scientist, who will be aware only of the environment as depicted in Figure 1. For this user it is irrelevant where and how the result is achieved, as long as time and accuracy requirements are met.

In summary, the current selected environment contains the acquisition hardware, a large compute server to perform the reconstruction in real time, a storage robot for storing the data for postprocessing, graphic workstations, and an ImmersaDesk (Idesk) or CAVE for data visualization. The data flow between these compute resources is coordinated via the Globus metacomputing toolkit. In initial work, the primary compute server for obtaining the reconstructed images is an Origin2000 with 128 nodes. Nevertheless, we plan to use other compute resources, since the code is portable over a variety of resources. We plan to utilize a computing resource broker to decide which compute resource should be used for executing the reconstruction.

4 Grid-Enabled Computing

In the previous section we described a typical experiment needing a lot of resources to enable real-time analysis. Unfortunately, resources, such as

- supercomputers, to fulfill the need for fast compute power,
- storage robots, to fulfil the need for large and persistent storage, and
- high speed networks, to connect the distributed resources

motivate the development of a grid enabled application as described next.

The requirements of the CMT applications can be fulfilled with the infrastructure provided by a computational grid[9]. The term "grid" is chosen in analogy with the electric power grid, which provides pervasive access to power and, like the computer and a small number of other advances, had a dramatic impact on human capabilities. By providing pervasive, dependable, and consistent access to advanced computational capabilities, the application of supercomputing-enhanced CMT algorithms allows the real-time operation mode during a CMT experiment while using nonlocal computational resources to allow for a certain level of quality of service.

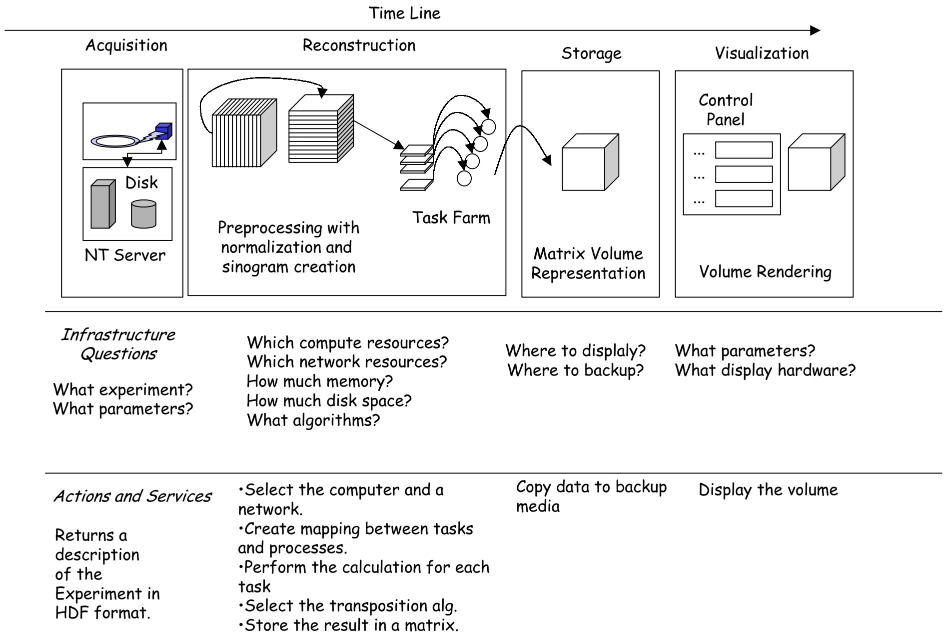

FIG. 3. The tasks that must be performed in order to enable a reconstruction in a grid-enabled environment.

Since the pipeline framework is designed to meet the unique requirements of the dynamic behavior of the "grid," a few questions as outlined in Figure 3 have to be answered to properly execute the program in a grid-enabled environment. These questions concern which hardware should be used for the calculations, which algorithms should be chosen, and which parameters should be used to instantiate the algorithms. To be more precise, we illustrate in Figure 4 the typical demand on a compute infrastructure.

A typical request issued to fulfill the requirements of the parallel CMT algorithm is formulated as follows:

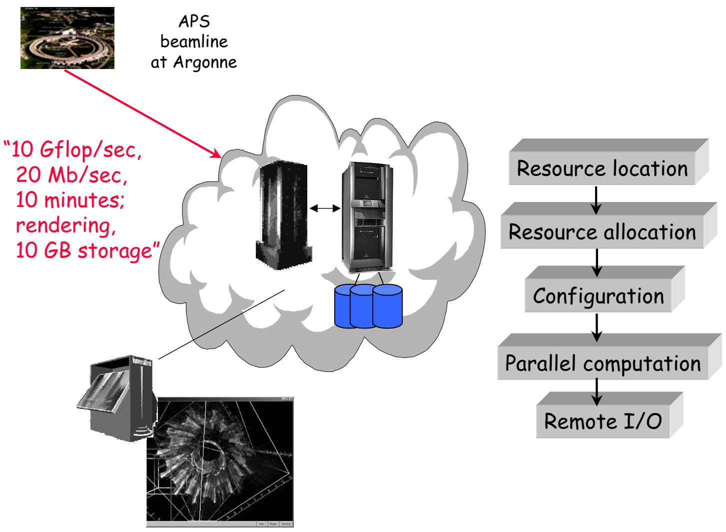

FIG. 4. A selection process of resources to fulfill the real-time requirements of the parallel x-ray microtomography program.

> To conduct the calculation, I need a compute environment that can perform 10 Gflops/sec; my data can be sent over a network link that utilizes 20 Mb/sec. The total time to reserve a specialized rendering pipe is 10 minutes for the rendering process. I also need 10 GB of storage to store the final result.

The Globus metacompiting toolkit provides components or allows for the construction of components that address the issues typically found in a grid-enabled application, such as the CMT Grand Challenge: resource location, allocation, resource configuration, and the parallel computation and use of remote I/O operations (see left side of Figure 4). In particular, the Globus toolkit provides a uniform communication layer, an information infrastructure that describes the state of the computational grid environment to support resource reservation and scheduling, as well as the program submission, file staging, and uniform program execution. For the CMT application we used the Globus components listed in Table 1 and describe their role in the pipeline framework.

TABLE 1

Globus components used in the CMT application

Component	Usage
GRAM	Component startup
GSI	Uniform access to instrument and processing machines
Nexus	Communication between pipeline components
GASS	Access to remote archived data, storage of processed data
Nexus	Pipeline control
GASS	Executable staging
HBM	Monitoring of pipeline status

At present, we have not exploited all the features that the processing pipeline allows. These include issues related to the dynamic behavior of the environment, fault tolerance, and process migration.

5 New Acquisition and Reconstruction Strategies

The immediate access to a large amount of compute power allows the use of new acquisition and reconstruction strategies. Traditionally, data is collected at microtomographic beamlines at constant angle offsets: for example, 0, 1, 2, ... degrees if 360 samples are to be taken. In an interactive environment such as we describe here, it becomes attractive instead to use an interleaved angle list. For example, we may first gather images at 60-degree offsets (0, 60, 120, 180, 240, 300), then collect additional images to provide a 30-degree sampling, and so on until a complete 1- degree dataset is obtained. The advantage of this strategy is that the reconstruction algorithm can be run repeatedly, once for each more detailed set of data; hence, the scientist obtains a series of more refined images and may be able to detect a flawed experimental setup early in the data collection process.

Another interesting direction that is enabled by the availability of supercomputer resources is the following. In principle, reconstruction quality can be improved by performing multiple reconstructions with different algorithms and parameter settings. We are hopeful that the enhanced compute power made accessible by grid environments will initiate a new area in the development of reconstruction algorithms for computed microtomography and other disciplines.

6 Visualization and Collaborative Tomography Environment

To support the requirements driven by the CMT community, we decided to integrate a three-dimensional visualization tool into the collaboration environment for the structures reconstructed by the parallel microtomography algorithm. The tool provides fast visualization and high resolution, thereby enabling researchers to interact with large datasets in real time. Moreover, the tool is portable on a variety of display stations and operating systems, and its exposed interface is easily used by nonexpert programmers. The resultant general visualization framework can be reused by scientists from different disciplines with similar visualization requirements (e.g., computed microtomography, electron microscopy, and astrophysics), as demonstrated at the SC98 Conference in Florida[4].

The tool uses a hardware-optimized volume-rendering [8] library from Silicon Graphics called *Volumizer*. Volumizer takes advantage of hardware-accelerated texture mapping [5], which enables the visualization to be fast enough to allow for real-time interaction. This reliance on SGI-specific hardware for the rendering does limit the application's portability somewhat, but as we explain later, its usability can be extended beyond the SGI platform.

One of the ways that this tool gives valuable qualitative information about the datasets that it renders is by providing insight about the inner structure of the volume. While two-dimensional tools also allow the user to see the inner structure of the data, they can do so only one slice at a time. Our 3-D tool, on the other hand, allows the user to take a cross section of the volume and see the inner makeup as it relates to the rest of the overall structure. Parts of the volume can also be stripped away by isolating only those values that fall within a particular range, making the rest of the volume transparent. In addition, one can also zoom in on a selected range of values to get finer detail.

As previously mentioned, the size of the datasets generated in these experiments can be quite large. A typical dataset can be on the order of 512^3 floating- point values. Currently, even the accelerated graphics hardware used by our application has trouble keeping up with volumes of this size, and the dataset needs to be subsampled down to 256^3 floating-point values in order to maintain its frame rates allowing interactive usage.

The tool can be used in a variety of environments. The one that provides the most powerful viewing capability is the immersive, three-dimensional virtual environment known as the CAVE[6, 7]. The CAVE is a $10 \times 10 \times 10$ foot room with computer-generated images projected onto three of the walls and the floor. It uses a stereoscopic, user-centered perspective to enable the user to see the images in three dimensions. The software can also be displayed on a smaller, single-screen display device known as ImmersaDesk. A motion-sensible three-dimensional user interface device, called a wand, allows the user to interact with and manipulate the data in the CAVE or Idesk. The CAVE and Idesk both rely an SGI workstation to perform their rendering. Nevertheless, a CAVE simulator, which can be run directly on any SGI workstation, eliminates the need for these high-end display devices. In the CAVE a control panel equipped with buttons and sliders and other high-level widgets is used to manipulate the parameters for controlling the volume rendering.

The graphics can also be displayed remotely on even a low-end desktop computer. This remote machine relies on the graphics hardware of the CAVE/Idesk application. A virtual camera placed in the scene generates a nonstereo compressed video stream that is broadcast over the Internet. A number of remote clients can view this video using readily available video conferencing software (see Figure 6). In addition to multiple researchers at remotely distributed sites being able to view the same visualization of the data, collaborative control of the rendering is also possible (see Figure 5 and 6). Using shared controls, multiple CAVE/Idesks attached to a visualization session can manipulate what the others in that session are viewing.

Figure 5 shows such a control panel (CP) at the bottom of the image. On the right of the CP there are a number of sliders. These sliders are used to manipulate the curves drawn on the left-hand side of the CP. The four curves (red, green, blue and alpha) determine how colors are assigned to the values in the volume. The buttons near the center of the CP (Cent[er], Width, Amp[litude]) allow the user to change which component of the color curves that the sliders affect. All of these sliders and buttons are shared controls; as a user in one CAVE adjusts one of the sliders, the rest of users attached to that session see the slider on their CP move, along with the curves and colors in the volume changing accordingly.

On the left side of Figure 5 is a list of the users currently attached to the session. On the right of the image is another shared control. The two concentric boxes allow the user to select a subregion of the volume to visualize. The user can also use buttons on another panel of the CP to manipulate the position and orientation of the volume in the scene.

The remote clients viewing the video on their desktops can also control the visualization by using a 2-D Java version of the ControlPanel. Figure 6 shows the view of the desktop client attached to the same visualization session as the CAVE user in Figure 5. The desktop client has all of the same controls available to the CAVE user, although in some cases the interface for changing those controls is modified to be more appropriate for the 2-D environment.

These shared controls are all implemented by using the CIF Shared State library for data synchronization.

The Collaborative Infrastructure Framework (CIF) is a set of communication libraries written on top of Nexus, the communication component of Globus. These libraries provide the functionality for facilitating the creation of collaborative tools. CIF Shared State is a

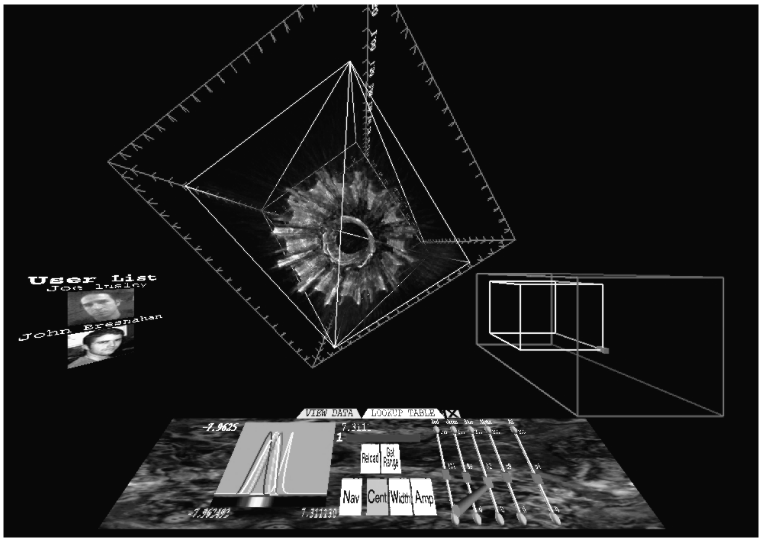

FIG. 5. A screen shot of the ImersaDesk taken during a collaborative session with two users.

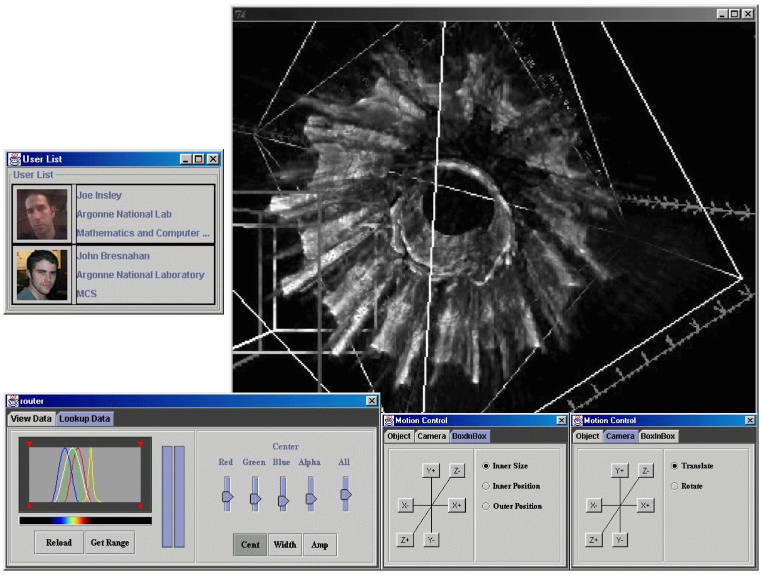

FIG. 6. A screen shot of a low-resolution graphics workstation taken during a collaborative session with two users.

component of CIF that allows for shared control of abstract states in collaborative space across multiple platforms. The library was used in this experiment for the creation of the shared control set that allowed multiple users to collaboratively control sliders, buttons, toggles, and other widgets across multiple C++ and Java platforms as seen in Figure 6.

Each client in the collaborative space has a set of abstract shared states that it is interested in using. Each CIF_Shared_State is an object created in the client's local memory space. When one client changes the value of a CIF_Shared_State, the CIF_Shared_State_Set associated with it handles all the messaging necessary to ensure state synchronization in total order. Therefore, when one state changes its value in the client's local memory, the Shared State library guarantees that every Set with that CIF_Shared_State in it updates its value.

The CIF_Shared_State object is an abstract class. Implementations of this class must provide a method to pack its value into an array of bytes (the library provides functionality to easily accomplish this), as well as a method to unpack its value. Each set identifies individual CIF_Shared_States by a character string unique to that state. So when a state of a given name is changed, its value is packed, via the pack method, and sent to all CIF_Shared_State_Sets in the message space. If a CIF_Shared_State_Set has a state identified by the same character string, that state's unpack method is called.

The library provides a set of CIF_Shared_State implementations. These implementations are in the form of arrays of the following data primitives: floating point, double-precision number, Boolean, byte, integer, short integer, and long integer. When any element of these shared arrays is changed, the state is packed up and propagated though the message space. The architecture of the CIF_Shared_State for multiple clients is shown in Figure 7.

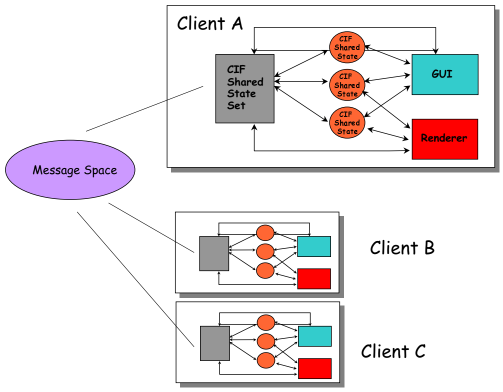

FIG. 7. The architecture of the CIF_Shared_State.

In an environment where many collaborators are working with and sharing a large amount of data it is important to agree upon a common data format. We have decided to store and distribute the data using the Hierarchical Data Format (HDF)[2]. HDF is a multiobject file format for sharing scientific data in a distributed environment. It was created at the National Center for Supercomputing Applications (NCSA) and designed to address many requirements for storing scientific data, including: support for the types of data and metadata commonly used by scientists, efficient storage of and access to large data sets, as well as, platform independence. Based on the HDF format a data format for the exchange of neutron and synchrotron scattering data between facilities and user institutions has been defined[11, 3] and is in general use. While HDF provides a data format, which is portable, binary, extensible and self-describing, NeXus defines the structure and contents of these HDF files in order to facilitate the visualization and analysis of neutron and X-ray data. We developed an extended version of the NeXus data format, referred to as CMT-NeXus, to allow portable and easy data exchange between CMT scientists. Currently, we are actively pursuing the improvement of the CMT-NeXus format and provide a C++ API that enables access to data stored in this format conveniently. In addition, we included in this API function calls that allow sending and receiving of the objects represented by the data format via the Globus communications module to and from remote computers. This library is already a defacto standard between the groups participating in the Grand Challenge and their collaborators. This standard will be implemented on at least three beamlines at the APS during the next year. The visualization environment as described in this section is able to utilize this data format.

7 Conclusion

In this paper we have demonstrated a grid-enabled real-time analysis, visualization, and steering environment for microtomography experiments. Specifically, we have provided a portable parallel framework that allows different reconstruction algorithms to be ported on a distributed networked supercomputing environment. A standard format based on HDF is defined to distribute the data among scientists in a meaningful way. This format will be implemented at several beamlines at the Advanced Photon Source during the next year.

The real-time visualization environment developed fulfills the basic needs of the microtomography scientists as well as scientists from other fields. Moreover, with the availability of this environment, we anticipate that scientists will make algorithm improvements, for example, including a priori knowledge of a previous reconstruction in order to increase the quality of the image.

The next phase of the project will focus on the use of new modalities in real-time reconstruction for interactive use and will explore the collaborative analysis of results. In addition, we will emphasize improvements to the usage of dynamic scheduling of computers, high-speed networking, and collaboration technologies.

8 Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. Globus research and development is supported by DARPA, DOE, and NSF.

We based the grid enabled version of the code from the code provided by Mark Ellisman and Steve Young from the National Center for Microscopy and Imaging Research at San Diego Supercomputing Center[13, 16, 1]. We are especially grateful for their continuos help throughout the project. We like to thank Derrick C. Mancini, and Francesco DeCarlo for conducting the experiments at the beamline. This truly interdisciplinary project would not have been possible without the countless hours of work performed by the Globus team: Joe Bester, Steve Fitzgeral, Brian Toonen, Steve Tuecke, Karl Czajkowski, Jean Tedesco, and all the others who have helped us throughout the years. We thank Mary Westbrook, the MCS and APS staff for helping us in issues related to setting up the network connection between MCS and the beamline floor.

We also thank Gail Pieper for her valuable comments and editorial input.

References

[1] *National center for microscopy and imaging research.* http://www-ncmir.ucsd.edu/CMDA/.

[2] *The ncsa hdf home page, information, support, and software from the hierarchical data format (hdf) group of ncsa.* http://hdf.ncsa.uiuc.edu/, 1998.

[3] *Neutron and x-ray data format www site.* http://www.neutron.anl.gov/NeXus/, 1998.

[4] Werner Benger, Ian Foster, Edward Seidel, John Shalf, Warren Smith, and Paul Walker, *Numerical relativity in a distributed environment*, in 9th SIAM Conference on PArallel Processing for Scientific Computing, April 1999.

[5] B. Cabral, N. Cam, and J. Foran, *Accelerated volume rendering and tomographic reconstruction using texture mapping hardware*, in Proceedings of the IEEE 1994 Symposium on Volume Visualization, October 1994, pp. 91–94.

[6] C. Cruz-Neira, D.J. Sandin, T.A. DeFanti, R.V. Kenyon, and J.C. Hart, *The cave: Audio visual experience automatic virtual environment*, Communications of the ACM, 35 (1992), pp. 65–72.

[7] C. Cruz-Neira, D. J. Sandin, and T.A. DeFanti, *Surround-screen projection-based virtual reality: The design and implementation of the cave*, in Proceedings of SIGGRAPH '93, August 1993, pp. 135–142.

[8] R. Drebin, L. Carpenter, and P. Hanrahan, *Volume rendering*, in Proceedings of SIGGRAPH '88, August 1988, pp. 65–74.

[9] Ian Foster and Carl Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufman, July 1998.

[10] W. S. Haddad, I. McNulty, J. E. Trebes, E. H. Anderson, R. A. Levesque, and L. Yang, *Ultra high resolution x-ray tomography*, Science, 266 (1994), pp. 1213–1215.

[11] P. Klosowski, M. Koennecke, J. Z. Tischler, and R. Osborn, *Nexus: A common format for the exchange of neutron and synchrotron data*, in Physica B 241-243, 1998, pp. 151–153.

[12] J. Lehr, *3D x-ray microscopy: tomographic imaging of mineral sheaths of bacteria Leptothrix ochracea with the Göttingen x-ray microscope at BESSY*, Optik, 104 (1997), pp. 166–170.

[13] P. J. Mercurio, T. T. Elvins, S.J. Young, P. S. Cohen, K.R. Fall, and M. H. Ellisman, *The distributed laboratory: An interactive visualization environment for electron microscopy and three-dimensional imaging*, Comm. Assoc. Comp. Mach., 35 (1992), pp. 54–63.

[14] Gregor von Laszewski, Mary Westbrook, Ian Foster, Edwin Westbrook, and Craig Barnes, *Using computational grid capabilities to enhance the cability of an x-ray source for structural biology*, to be published, (1999).

[15] Y. Wang, *Three-dimensional imaging in soft x-ray microscopy*, PhD thesis, Department of Physics & Astronomy, State University of New York at Stony Brook, 1998.

[16] S.J. Young, G.Y. Fan, D. Hessler, S. Lamont, T.T. Elvins, M. Hadida, G. Hanyzewski, J. W. Durkin, P. Hubbard, G. Kindalman, E. Wong, D. Greenberg, S. Karin, and M. H. Ellisman, *Implementing a collaboratory for digital microscopy*, 2/3 (1996), pp. 170–181.