Available online at www.sciencedirect.com
SCIENCE DIRECT
Future Generation Computer Systems 22 (2006) 1004–1010
FGCS FUTURE GENERATION COMPUTER SYSTEMS
www.elsevier.com/locate/fgcs

Distributed and collaborative visualization of large data sets using high-speed networks

Andrei Hutanu[a,*], Gabrielle Allen[a], Stephen D. Beck[a], Petr Holub[b,c], Hartmut Kaiser[a], Archit Kulshrestha[a], Miloš Liška[b,c], Jon MacLaren[a], Luděk Matyska[b,c], Ravi Paruchuri[a], Steffen Prohaska[d], Ed Seidel[a], Brygg Ullmer[a], Shalini Venkataraman[a]

[a] *Center for Computation and Technology, 302 Johnston Hall, Louisiana State University, Baton Rouge, LA 70803, United States*
[b] *CESNET z.s.p.o., Zikova 4, 16200 Praha, Czech Republic*
[c] *Masaryk University, Botanick 68a, 62100 Brno, Czech Republic*
[d] *Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany*

Available online 22 May 2006

Abstract

We describe an architecture for distributed collaborative visualization that integrates video conferencing, distributed data management and grid technologies as well as tangible interaction devices for visualization. High-speed, low-latency optical networks support high-quality collaborative interaction and remote visualization of large data.

Keywords: Collaborative visualization; Distributed applications; Co-scheduling; Interaction devices; Video conferencing

1. Introduction

The study of complex problems in science and engineering today typically involves large scale data, huge computer simulations, and diverse distributed collaborations of experts from different academic fields. The scientists and researchers involved in these endeavors need appropriate tools to collaboratively visualize, analyze and discuss the large amounts of data their simulations create. The advent of optical networks, such as the 40 Gbps research network currently being deployed across Louisiana,[1] opens doors to lower latency, higher bandwidth approaches that allow these problems to be addressed, particularly in highly interactive environments, as never before.

This article describes recent work to develop generic, novel techniques that exploit high speed networks to provide collaborative visualization infrastructure for such problems. The particular driving problem is provided by a numerical relativity collaboration between the Center for Computation & Technology (CCT) at LSU, and colleagues in Europe, including the Albert Einstein Institute (AEI) in Germany. The task at hand is to collaboratively visualize the gravitational wave output from simulations of orbiting and colliding binary black holes. The collaborators already hold joint meetings each week using AccessGrid technologies. However, much higher image quality, resolution, and interactivity are needed to support collaborative visualization and deep investigations of data.

Some of the most important issues a collaborative visualization environment needs to tackle are: minimizing latency in the interaction loop, maximizing the performance and quality of the visualization and effectively coordinating the use of various resources. These can be conflicting goals leading to different design decisions. A possible strategy to minimize interaction latency is to replicate sections of the visualization pipeline (data access $\Rightarrow$ filtering $\Rightarrow$ rendering $\Rightarrow$ display) to each of the participating users. However, having more visualization processing taking place on local machines leads to increasingly difficult synchronization between the distributed users. Moreover, the performance of the visualization is

* Corresponding author. Tel.: +1 2255780811.
E-mail address: ahutanu@cct.lsu.edu (A. Hutanu).
[1] Louisiana Optical Network Initiative. http://www.loni.org.

0167-739X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2006.03.026

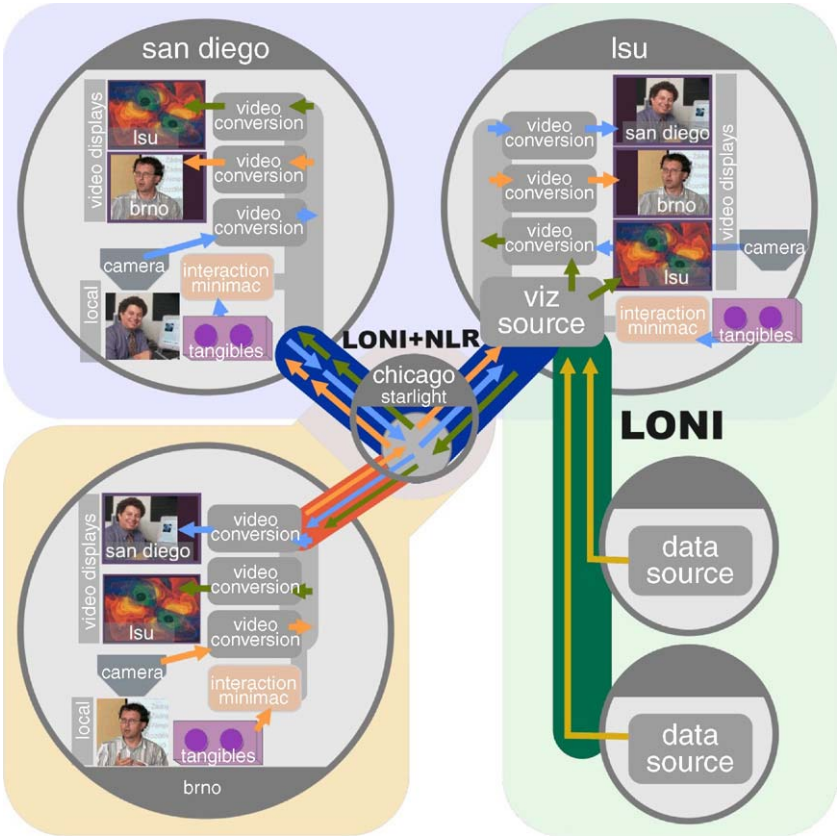

Fig. 1. Illustration of the visualization server-based collaborative environment which combined Brno, Baton Rouge, and San Diego at iGrid 2005.

constrained by the capacity of the hardware available at each of the participating sites. In this article we show how the latest technologies in video conferencing, interaction, distributed visualization, grid computing and networking are utilized to create a powerful, high-quality visualization server-based collaborative environment. We also present details of a practical experiment performed during iGrid 2005.

2. Collaborative environment

In our system high-definition video and audio transport software and hardware is used to connect sites in a video conference session. One site serves as the host for the visualization application, whose output (rendered images) is connected to the video conference. Remote collaborators follow, interact with and steer the visualization using custom-made interaction devices deployed at all the sites (see Fig. 1).

2.1. Video transport system

Compression of video streams is useful for reducing the data rate [1] but it comes at the cost of inducing additional latency and having to deal with the issue of quality degradation due to data loss in the network. Using standard videoconferencing technologies may require separation of the interactive and collaborative part from the high-resolution visualization [2].

For video transport we are using our custom solution based on uncompressed high-definition video described in more detail in an article titled "High-Definition Multimedia for Multiparty Low-Latency Interactive Communication" in this issue. This system captures HD-SDI video with full 1080i resolution (1920×1080 image, 60 fps interlaced) and sends the data over the network resulting in 1.5 Gbps per each video stream. For three video streams (one visualization, two video conference) this totals 4.5 Gbps required at each of the participating sites. In addition to the network bandwidth requirements, it is advantageous to use dedicated network circuits with very low jitter and small packet reordering that eliminate the need for substantial video buffering. Optical paths or "lambdas" meet all of these requirements, and their usage reduces the latency originating in the network to the minimum.

In this setup the end-to-end latency from camera capture to display without taking the network latency into account is approximately 175 ms. In order to integrate visualization into the system we use a Doremi XDVI 20s box that converts the output of the visualization server (DVI format) to HD-SDI. According to the product specifications this converter induces a latency of 30 ms.[2]

An alternative to hardware video capture is to use pixel readback on the rendering machine(s) as in Griz [3]. Video transmission performance is in this case negatively influenced by the rendering overhead and modifications of the visualization application are required.

[2] At 1920×1080 @ 65 Hz DVI input and 1080i HD-SDI output.

The video data is distributed to the other participating sites using UDP packet reflector technology [4].[3] The switching capacity required by this setup is equal to the number of participants × number of video streams × 1.5 Gbps. For three sites and three video streams this adds up to 13.5 Gbps required in switching capacity.

2.2. Interaction

In the initial stages of the iGrid experiment, we saw how remote mouse control (e.g., via the Synergy program) can grow practically unusable over high-latency (>1 s) image-streaming pipes. Even with lower latency, there are major practical challenges in allowing numerous users to collaboratively manipulate a shared visualization via mouse-based interaction (whether with one or many cursors).

In response, we made experimental use of physical interaction devices called "viz tangibles". We planned to deploy both a "parameter pad", with two embedded knobs; and an "access pad", allowing the parameter knobs to be rebound to different parameters using RFID-tagged cards. In practice, we deployed only the parameter pads, each statically bound to two parameters: object rotation and global timestep.

For iGrid, four sets of parameter pads were deployed: two in San Diego, and one apiece in Baton Rouge and Brno. Each pad contained two PowerMate USB knobs. These were connected to Mac Mini machines running Linux Fedora Core 4, using routed TCP sockets via 100 MB Ethernet. The synchronization between users is reduced to a simple aggregator software that uses clutching to transform the four sets of incoming wheel updates to one set of visualization parameters.

3. Distributed visualization

Responsive interaction is critically important for collaborative visualization systems, and one approach suitable for high speed optical networks is to move appropriate parts of the system to remote computers.

In our case we separate the visualization front-end from a distributed data access server with a 10 Gb network connecting the machines running these components. This improves performance in two ways. First, by distributing the data server we can access multiple storage resources and parallelize the data access operations. Second, by using the remote machines to pre-cache the large data set to be visualized improves responsiveness since main memory access is faster than disk access. High-speed networks that provide bandwidths larger than disk transfer rates make transferring data from remote memory faster than reading data from the local disk. In effect, we are using a large pool of memory distributed over multiple remote computers, similar to LambdaRAM/Optiputer [5].

With distributed resources network latency can become an issue for the application. In order to limit the effect of latency on the visualization system, we build upon a remote data access system that separates data selection from data transport [6,7]. This allows pipelining and asynchronous execution and reduces the overhead of a remote data access operation to a maximum of one network Round-Trip Time (or RTT). For our experiment, network transport performance was more important as the data transfer time is about one order of magnitude larger than the network RTT.

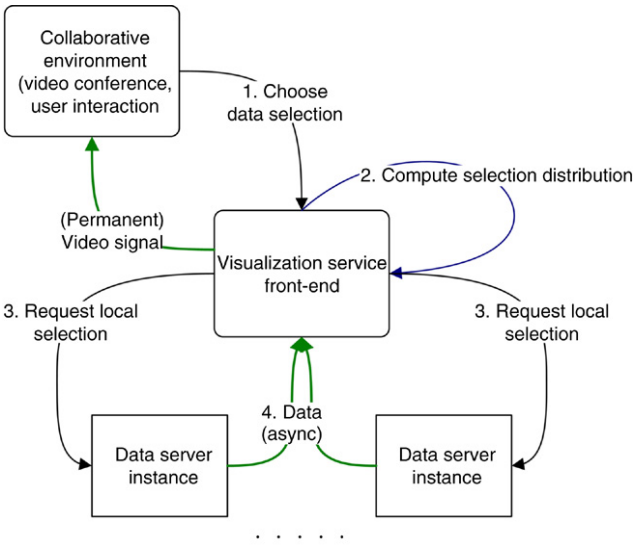

Fig. 2. Requesting visualization of new data.

Fig. 2 illustrates the interactions between the components of the visualization system as triggered by a user request for new data to be visualized. When one of the users participating to the collaborative session requests that a new portion of the dataset should be visualized, the visualization application determines which section of the data needs to be supplied by each data server and communicates the individual selection to the servers. Upon receiving the requests the servers will start delivering their data in parallel to the visualization application. The data servers are also capable of performing subsampling operations, a feature that allows multi-resolution rendering approaches.

User interactions that do not require any modifications of the visualization data, such as viewpoint changes (rotations) do not trigger any communication between the visualization front-end and the data servers.

4. Grid technologies

We consider that scheduling of resources (as opposed to having the servers run permanently) for running the data servers is required if the data selection/filtering operations are non-trivial (i.e. they are CPU intensive) or if any type of caching is used on the server side (as is the case in this experiment). To execute the distributed visualization application, a component was needed that could co-schedule the required compute and network resources. To this end, the HARC (Highly Available Robust Co-scheduler) framework was developed and deployed.[4]

[3] During iGrid, packet reflectors were running on machines located at StarLight, Chicago, where all the network links met.

[4] The HARC/1 implementation, which was used here, is available for download at http://www.cct.lsu.edu/personal/maclaren/CoSched/.

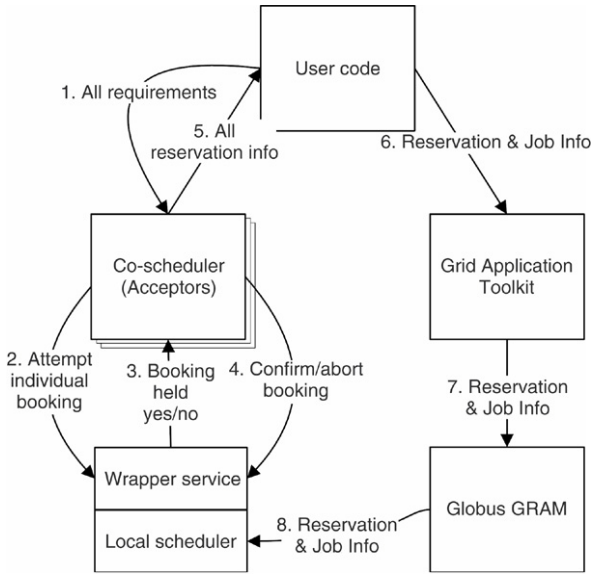

Fig. 3. Job scheduling and submission.

To ensure that all the resources are made available for the same time period, HARC uses a phased commit protocol; we assume that each resource has a scheduler capable of making and honoring advance reservations.[5] The co-scheduler asks each resource to make a tentative reservation for the required time (*prepare*); if, and only if, all resources respond in the positive (*prepared*), are the reservations confirmed (*commit*). In all other situations, the tentative reservations are removed (*abort*).

To avoid the blocking problems of the classic two-phase commit protocol, where the *Transaction Manager* is a single point of failure, HARC is based on applying Lamport's Paxos Consensus Algorithm to the transaction commit problem [8]. Multiple *Acceptors* co-operatively play the role of the Transaction Manager; a deployment with $2n + 1$ Acceptors can tolerate failure of n Acceptors. Even with conservative estimates of Mean-Time to Failure and Mean-Time to Repair, it is possible to deploy a set of seven Acceptors with a Mean-Time to Failure measured in years.

After successful co-scheduling, the Grid Application Toolkit (GAT) [9], which provides a simple generic job-submission interface, is used to submit the jobs to the compute resource reservations, through the chosen Grid resource management system. For the iGrid demonstration, Globus GRAM was used to access PBSPro schedulers and the Globus middleware (GRAM client) was used to build the underlying adaptor that implemented the job submission functionality of the GAT API.

Fig. 3 shows the interactions between the various Grid components for data job scheduling and submission. Only one set of local scheduler/wrapper service/Globus GRAM is shown but multiple instances of these services are involved in these interactions (one for each compute resource).

5. Results and conclusions

5.1. iGrid scenario

For the demonstration in San Diego, CCT/LSU (Baton Rouge, Louisiana), CESNET/MU (Brno, Czech Republic) and iGrid/Calit2 (San Diego, California) participated in a distributed collaborative session (see Fig. 4). For the visualization front-end we used a dual Opteron 252, 2.6 GHz, 8 Gbyte RAM, NVidia Quadro FX 4400 graphic card (512 Mbyte video memory) at LSU running a modified version of Amira [10] for the 3D texture-based volume rendering.

The visualization backend (data server) ran on an IBM Power5 cluster (14 Nodes, 112 1.9 GHz Power5 processors, 256 Gbyte overall main memory) and a SGI Prism Extreme (32 Itanium processors, 128 Gbyte shared main memory, 10 Gb network interface), at LSU. We ran nine data processes each on one node of the P5 cluster, each process configured to cache approximately 12 Gbytes of data and one process on the Prism configured to cache approx. 15 Gbytes of data. The data set used, a scalar field from a binary black hole simulation, had a size of 120 Gbytes with 400^3 data points at each timestep (4 bytes data/point for a 256 Mbyte/timestep).

In our demonstration, three HARC Acceptors were deployed. In addition to scheduling ten compute jobs, two Calient DiamondWave switches (one at LSU, the other at MCNC) were also scheduled.[6]

5.2. Results and discussion

The latency induced by the video system is approximately 200 ms.[7] Even with network round-trip times of up to 150 ms for the transatlantic connection to Brno the distributed collaborative environment remains interactive.

The distributed visualization system shows that using a pool of networked memory can improve the responsiveness of the visualization application. Our initial measurements showed a reduction in load time from 5 s and more when using a single locally mounted filesystem to 1.4–1.5 s per timestep when using the distributed cache. This is currently limited by the network transport performance and by the fact that we used only one processor on the visualization machine for data transfer while keeping the other one dedicated for visualization.

For data transport to the visualization, our original plan was to use the GAT streaming API as well as the RBUDP protocol [11] interfaced by the GAT. This would have enabled us to hot swap the network protocol during the application runtime. Unfortunately we encountered a few issues. As described in [12], RBUDP is not suitable for many-to-one communications, and as we found out, it is practically unusable for many-to-one communications when using a single processor for receiving the data. This is possibly caused by the fact

[5] In the case of the Calient DiamondWave switches, this had to be constructed. The scheduler consists of a timetable for each port in the switch; a reservation requests a number of connections which should be active during the reservation.

[6] At the time of the demo, these were not connected to any relevant resources.

[7] At iGrid we had to use an alternative setup using compressed video resulting in approximately 2 s latency in the video system. The issue was solved in time for another demonstration in Seattle during Supercomputing 2005.

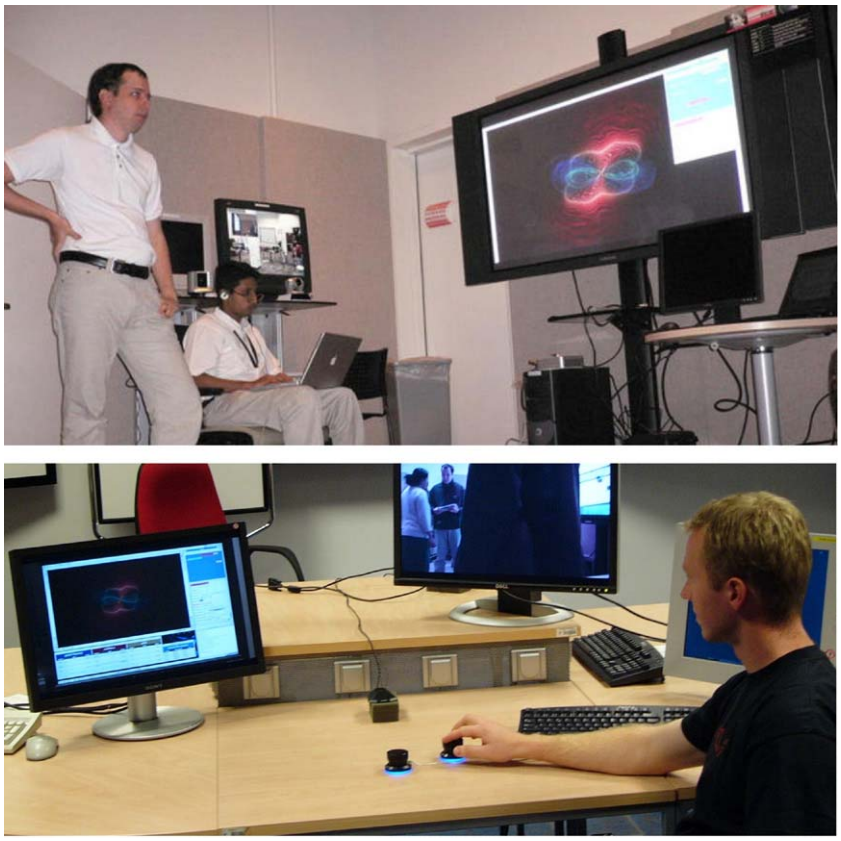

Fig. 4. Remote visualization at iGrid (top). Remote interaction from Brno (bottom).

that the current implementation creates a separate socket for each incoming connection requiring a process or thread to be active on each connection at any given time. We will further investigate this issue to exactly determine the cause of the problems we encountered. The end-to-end bandwidth observed by the application (including network transfer using TCP, data request, endian conversions) was approximately 1.2 Gbps.

5.3. Conclusions and future work

We have developed a collaborative application that exploits high speed optical networks for interactive, responsive visualization of huge data sets, over thousands of kilometers, with high image quality. Co-scheduling of network and computing resources has been used to guarantee resource availability.

While currently the data transfer does take most of the update time when changing a timestep (1.4 s compared to 0.35 s for transfer to video memory), further optimizations in the networking implementation might reverse this situation. Also, as the data size increases beyond the rendering capacity of a single video card, investigating distributed rendering front-ends for the visualization becomes a necessity.

One of the lessons learned while using the GAT as well as the BSD socket API for TCP was that a byte-level streaming API is not optimal for the block-wise type of data transfer we are doing. Future efforts will lean towards defining and incorporating message-based APIs as well as related network protocols.

Acknowledgments

We thank many people who helped make this possible: Boyd Bourque, Fuad Cokic, Jiří Denemark, Peter Diener, Lukáš Hejtmánek, Ralf Kaehler, Gigi Karmous-Edwards, Olivier Jerphagnon, Michael Lambert, Lonnie Leger, Honggao Liu, Charles McMahon, Sasanka Madiraju, Andre Merzky, Yaaser Mohammed, Seung Jong Park, Jan Radil, Tomáš Rebok, Sean Robbins, Brian Ropers-Huilman, Rajesh Sankaran, William Scullin, John Shalf, Jeremy Songne, Steve Thorpe, Cornelius Toole, Isaac Traxler, Alan Verlo and Sam White. This work was supported by the Center for Computation and Technology at LSU, the Enlightened project (NSF grant 0509465); the NSF MRI (grant 0521559); and the Louisiana Board of Regents. The Czech authors were supported by the CESNET research intent (MŠM 6383917201). The loan of two 10GE T210 network cards from Chelsio is highly appreciated.

References

[1] A. Breckenridge, L. Pierson, S. Sanielevici, J. Welling, R. Keller, U. Woessner, J. Schulze, Distributed, on-demand, data-intensive and collaborative simulation analysis, Future Gener. Comput. Syst. 19 (6) (2003) 849–859.

[2] N.T. Karonis, M.E. Papka, J. Binns, J. Bresnahan, J.A. Insley, D. Jones, J.M. Link, High-resolution remote rendering of large datasets in a

collaborative environment, Future Gener. Comput. Syst. 19 (6) (2003) 909–917.

[3] L. Renambot, T. van der Schaaf, H.E. Bal, D. Germans, H.J.W. Spoelder, Griz: experience with remote visualization over an optical grid, Future Gener. Comput. Syst. 19 (6) (2003) 871–881.

[4] E. Hladká, P. Holub, J. Denemark, An active network architecture: Distributed computer or transport medium, in: 3rd International Conference on Networking, ICN'04, Gosier, Guadeloupe, March 2004, pp. 338–343.

[5] C. Zhang, J. Leigh, T.A. DeFanti, M. Mazzucco, R. Grossman, Terascope: distributed visual data mining of terascale data sets over photonic networks, Future Gener. Comput. Syst. 19 (6) (2003) 935–943.

[6] S. Prohaska, A. Hutanu, Remote data access for interactive visualization, in: 13th Annual Mardi Gras Conference: Frontiers of Grid Applications and Technologies, 2005.

[7] R. Kähler, S. Prohaska, A. Hutanu, H.-C. Hege, Visualization of time-dependent remote adaptive mesh refinement data, in: Proc. IEEE Visualization'05, 2005, pp. 175–182.

[8] L. Lamport, J. Gray, Consensus on transaction commit, Technical Report MSR-TR-2003-96, Microsoft Research, January 2004. http://research.microsoft.com/research/pubs/view.aspx?tr_id=701.

[9] G. Allen, K. Davis, T. Goodale, A. Hutanu, H. Kaiser, T. Kielmann, A. Merzky, R. van Nieuwpoort, A. Reinefeld, F. Schintke, T. Schütt, E. Seidel, B. Ullmer, The grid application toolkit: Towards generic and easy application programming interfaces for the grid, in: Grid Computing, Proceedings of the IEEE 93 (3) (2005) (special issue).

[10] D. Stalling, M. Westerhoff, H.-C. Hege, Amira: A highly interactive system for visual data analysis, in: C.D. Hansen, C.R. Johnson (Eds.), The Visualization Handbook, Elsevier, 2005, pp. 749–767.

[11] E. He, J. Alimohideen, J. Eliason, N.K. Krishnaprasad, J. Leigh, O. Yu, T.A. DeFanti, Quanta: a toolkit for high performance data delivery over photonic networks, Future Gener. Comput. Syst. 19 (6) (2003) 919–933.

[12] X.R. Wu, A.A. Chien, Evaluation of rate-based transport protocols for lambda-grids, in: Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, HPDC'04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 87–96.

Andrei Hutanu graduated in 2002 and received his engineering diploma in Computer Science from Politehnica University of Bucharest. Currently he is a researcher in the Grid and Visualization departments of the Center for Computation and Technology at Louisiana State University. Prior to this he was employed for two years as a researcher in the Visualization department of the Zuse Institute Berlin. His research interests are distributed interactive visualization, data management and optical network applications.

Gabrielle Allen is an associate professor in computer science and focus area head of the core computational science focus area at CCT. She received her Ph.D. in computational astrophysics from Cardiff University and completed a postdoctoral fellowship at the Max Planck Institute for Gravitational Physics and led its efforts in computational science for a number of years before joining CCT.

Dr. Stephen D. Beck is the director of the Laboratory for Creative Arts and Technologies (LCAT), and also Professor of Composition and Digital Music in LSU's School of Music. He received his Ph.D. in music composition and theory from UCLA, and held a Fulbright Fellowship (1985) at the Institut du Recherche et Coordination Acoustique/Musique (IRCAM) in Paris. His research interests include: immersive audio, data sonification, auditory display, sound diffusion, and interactive computer music.

Petr Holub graduated at Faculty of Sciences, Masaryk University in Brno and received his Ph.D. from Faculty of Informatics MU in informatics, focusing at high-speed networks, multimedia, and parallel and distributed systems. Currently he works at Institute of Computer Science MU in the Laboratory of Advanced Networking Technologies and participates on its scientific leadership. He is also a researcher with CESNET. His professional interests include high-speed networks and suitable protocols, active networks, user-empowered overlay networks, advanced collaborative environments, grid environments, and computational quantum chemistry and its implementation on distributed systems.

Hartmut Kaiser received his diploma in Computer Science at the Leningrad Electrotechnical University, Petersburg, Russia, in 1985, a Ph.D. and the habilitation in Computer Engineering, in 1988, both from the Technical University of Chemnitz, Germany. Currently he is a researcher at the Center for Computation and Technology. His research interests include application programming interfaces to the grid, geoinformation systems and the C++ programming language in general.

Archit Kulshrestha is a Grid Administrator at the the Center for Computation and Technology at LSU. He obtained a Masters degree in System Science from the Department of Computer Science at Louisiana State University and a Bachelors degree in Computer Science and Engineering from JNT University, India. His research interest include Grid Computing, Grid Resource Management and Job scheduling. He is currently pursuing a Ph.D. in Computer Science at Louisiana State University.

Miloš Liška is a Ph.D. student at Faculty of Informatics at Masaryk University. He is also a CESNET researcher working on projects concerning multimedia, tools for collaborative environments and collaborative workflows. He is interested especially in tools for multimedia processing and video transmission.

Jon MacLaren is a researcher in Distributed Computing in the Center for Computation and Technology at LSU. His chief research interest is the co-scheduling of distributed resources across multiple administrative domains; he is particularly interested in the co-scheduling of different classes of resources, e.g. compute nodes and optical network connections. Before coming to CCT in 2005, Jon was a researcher in the University of Manchester (UK)'s e-Science Center for four years. Jon MacLaren received his Ph.D. in Computer Science in 2001 from Manchester University, U.K. where he previously attained an MPhil. He also has a BSc from the University of York, U.K.

Luděk Matyska is an Associate Professor in Informatics at Faculty of Informatics, and he also serves as a vice-director of Institute of Computer Science, both at Masaryk University in Brno, Czech Republic. He got a Ph.D. in Chemical Physics from Technical University Bratislava, Slovakia. His research interests lie in the area of large distributed computing and storage systems, with a specific emphasis on their management and monitoring. He also works in high speed network applications, with a specific emphasis on collaborative work support and use of all these technologies in various e-learning activities. He lead national Grid infrastructure projects and participates in several EU funded international projects including the CoreGRID Network of Excellence.

Ravi Paruchuri received his B.Tech. in computer science and engineering from University of Madras and his M.Sc. in systems science from Louisiana State University. Currently he is the manager of IT Operations within the Center for Computation and Technology at Louisiana State University.

Steffen Prohaska received a diploma in theoretical solid state physics from the Technical University Darmstadt. He is now affiliated with Zuse Institute Berlin, where he is working on visualization and data analysis. Currently, he is finishing his Ph.D. on skeletonization of bio-medical image data with applications to analysis of trabecular bone and micro vascular networks.

Ed Seidel is the director of the Center for Computation and Technology at Louisiana State University and the Floating Point Systems Professor in LSU's Departments of Physics and Astronomy, and Computer Science. Seidel is well known for his work on numerical relativity and black holes, as well as in high-performance and grid computing. He earned his Ph.D. from Yale University in relativistic astrophysics. He headed the numerical relativity group as a professor at the Max-Planck-Institut fuer Gravitationsphysik (Albert-Einstein-Institute) in Germany from 1996–2003, where he maintains an affiliation. He was previously a senior research scientist at the National Center for Supercomputing Applications and associate professor in the Physics Department at the University of Illinois.

Brygg Ullmer is an assistant professor at LSU, jointly at CCT and in computer science. He received his Ph.D. from the MIT Media Laboratory. He leads visualization and human-computer interaction efforts at CCT, including the Tangible Visualization group (jointly in CCT and CS). His research interests include tangible user interfaces, visualization, rapid physical prototyping, embedded systems, and programming languages.

Shalini Venkataraman is a research programmer in scientific visualization with the Visualization, Interaction and Digital Arts (VIDA) group at CCT. She graduated with a Master's degree from the Electronic Visualization Lab at the University of Illinois-Chicago in 2004. Prior to that, she was a software engineer at the Institute of High-Performance Computing in Singapore. Her research interests include scalable, distributed graphics and volume visualization using programmable graphics hardware.