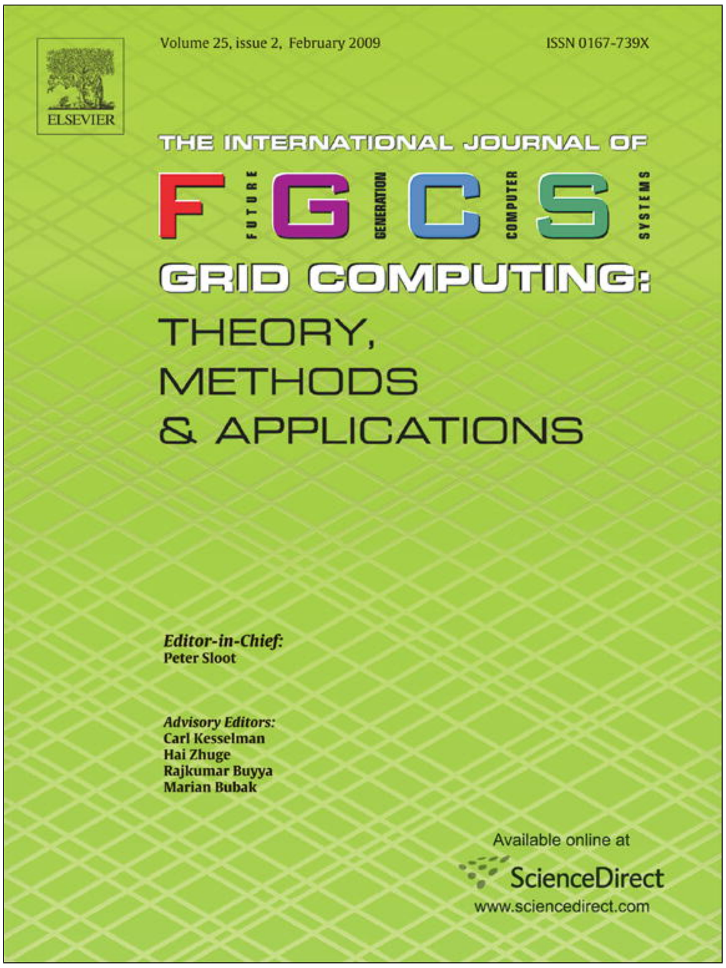

Volume 25, issue 2, February 2009
ISSN 0167-739X
ELSEVIER
THE INTERNATIONAL JOURNAL OF
FGCS
FUTURE GENERATION COMPUTER SYSTEMS
GRID COMPUTING:
THEORY,
METHODS
& APPLICATIONS
Editor-in-Chief:
Peter Sloot
Advisory Editors:
Carl Kesselman
Hai Zhuge
Rajkumar Buyya
Marian Bubak
Available online at
ScienceDirect
www.sciencedirect.com

Future Generation Computer Systems 25 (2009) 161–168

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Enabling high resolution collaborative visualization in display rich virtual organizations

Luc Renambot *, Byungil Jeong, Hyejung Hur, Andrew Johnson, Jason Leigh

Electronic Visualization Laboratory, University Illinois at Chicago, United States

ARTICLE INFO

Article history:
Received 5 February 2008
Received in revised form
19 June 2008
Accepted 9 July 2008
Available online 24 July 2008

Keywords:
Collaborative computing
Visual
Network communication

ABSTRACT

We predict that high-resolution displays will penetrate scientific laboratories first, and will become pervasive at the office, in the cubicle, and meeting room, and ultimately at home. In this paper, we present a novel approach to distribute high-resolution images to multiple tiled displays in real-time using a high-performance PC cluster. This approach allows users to share their visualization applications and videos on multiple tiled displays, repositioning and resizing each application window independently. These operations require high-resolution image multicasting from a rendering cluster to multiple display clusters. A high-performance bridging system called SAGE Bridge performs the image multicasting on a high-performance PC cluster. We show in several experiments that this system enables sharing of high-resolution scientific animations and high-definition videos (with audio) between multiples international sites, hence creating virtual research laboratories (i.e. virtual organizations), as observed in a figure given in this article.

1. Introduction

High-resolution displays that are greater than 4 Megapixels are becoming a standard part of scientific research (See Fig. 2). For scientific disciplines, large displays are the only means by which they can see data from their instruments. With the advent of low-cost LCDs, research labs are now using tiled display walls as "mash up" environments where they can juxtapose a variety of data so that they can look at them as a whole [22]. For large-scale scientific research, there is no other way to look at the data. These projects routinely deal with time-varying data on the order of terabytes to petabytes. It is impossible to manage this information by printing out static diagrams on sheets of paper and pinning them to a wall. Today's microscopes and telescopes are no longer simple optical instruments, but are integrated with complex computing systems that perform noise filtering, mosaicing, and feature detection. High-resolution displays are the windows into those instruments [2].

These environments will also extend to the trade-show conference floor where the scientist presents his/her results to peers, the office and the meeting room. It is now quite common to see large panel displays at trade-shows and technology-related venues. Panel displays can be found in both commercial booths as well as university research booths. In the future, one can imagine that conferences will replace traditional pin-up boards in poster sessions with digital poster boards. A presenter could simply walk up to a wall, enter a login ID or plug in a thumb drive, and immediately give a multimedia presentation. The content of the presentation could be retrieved from his/her office over high-speed networks.

Similarly, one would expect that, cubicle walls will have light-weight, low-power, low-heat, displays imbedded in them that have both near-print-quality resolution and can be controlled with touch. Higher resolution is necessary in a cubicle because of the close proximity a person is to the walls/screens. One can imagine that the display-enabled cubicle walls are modular, lightweight, and can snap together to provide daisy-chained power and networking as well as physical configuration information so that an intelligent software controller can manage them as one continuous surface. The computing to support the intelligence could be built directly into the walls. Cameras would also be embedded in the walls, to be used for seamless video conferencing. Many knowledge workers, such as analysts, covers almost all the wall surfaces of their offices with documents (printed and hand-written); it is essentially a project room within one's office where one can externalize all one's thoughts on the walls.

In the workplace, displays will pervade meeting rooms where ultra-large flat-panel displays will replace projectors. This makes practical sense when the cost of maintaining a projection-based system exceeds that of maintaining a flat-panel display system. Flat panels also have the added benefits that the content is clearly visible without dimming the room lights, and they consume less power than projectors. As more flat panel displays are deployed in meeting rooms, the desire to use them for more than just

* Corresponding author. Tel.: +1 312 996 3002; fax: +1 312 413 7585.
E-mail address: renambot@uic.edu (L. Renambot).

0167-739X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2008.07.004

Fig. 1. High-resolution scientific visualization being shared by various groups around the world (US, Czech Republic, the Netherlands) at the same instant. Each group maintains a different display configuration and local control of the content.

PowerPoint presentations will grow. They will be used for multi-site video conferencing (such as in Cisco, Polycom, Sony and HP's recent telepresence products) as well as for poster boards on which digital information can be posted and shared (locally and remotely). The traditional paper-based War Room/Project Room will be forever transformed.

In traditional Project Rooms, the walls are usually covered with notes and drawings from intense brainstorming sessions. Teasley et al., at the University of Michigan's School of Information, found that engineering teams who worked in these environments enjoyed considerable performance improvements over teams that worked in more traditional work environments [20,21]. A person's working memory can hold approximately 6–7 pieces of information at a time. Externalizing one's thoughts on a wall expands one's working memory and enables a team to collectively organize hundreds of thoughts at a time.

In this paper, we show how this vision is reality today for a group of scientists connected by international research and education networks. One of this group refers to themselves as the Global Lambda Visualization Facility (GLVF) where scientists live the dream of persistent connectivity among their sites supported by high-speed networks (such as NLR and GLIF). The goals of GLVF are:

- To create integrated tools for real-time, interactive visualization and distance collaboration,
- To work with global domain science teams on the social science of collaboration, to both learn from and educate them on how to use these new technologies to transform how they do science,
- To train students and junior faculty, the next-generation workforce.

GLVG participants are among the following countries: Canada, Japan, Netherlands, South Korea, USA and new partners from Brazil, China, Czech Republic, Russia, India, Mexico, Taiwan.

While such environments raise numerous issues and research topics about user interaction and collaboration, we will focus in this paper on the media delivery problems at high-resolution between globally distributed high-resolution tiled displays and rendering resources interconnected by a global LambdaGrid (a grid of deterministic high-speed networks) which is intensively researched by the international partners of the OptIPuter project [3,4,17].

Fig. 1 shows the high-resolution tiled displays of the national and international GLVF collaborators. They are primarily enabled by the Scalable Adaptive Graphics Environment (SAGE) [5,6,18]. SAGE is a specialized middleware for managing real-time streams of high-definition video and extremely high-resolution graphics from remotely distributed rendering clusters to scalable display walls over ultra-high-speed networks. SAGE supports multiple visualization applications at the same time allowing free moving, resizing and overlapping of application windows. This provides users with a full multitasking environment on a tiled display.

System support of distance collaboration is a unifying and fundamental requirement of the GLVF community. The high-resolution visualization system of each member must be capable of multicasting visualizations to all collaborating end-points, so that all participants can simultaneously see and interact with the data. Multicasting of high-definition video and audio is also required for effective communication among collaborators. We propose a novel visualization multicasting scheme, called Visualcasting, in order to extend SAGE to support distance collaboration with multiple end-points. Visualcasting is a scalable real-time image distribution service for multiple high-resolution tiled displays. A visualization application streamed to multiple Visualcasting end-points (tiled displays) can have different window layouts on each tiled display. The window operations (moving or resizing) on each tiled display are independent.

As windows on the tiled display are resized or repositioned, SAGE performs non-trivial reconfigurations of the multiple streams from the rendering source to the PC nodes that drive the tiled displays. Visualcasting makes this problem even more complex because SAGE must consider multiple window layouts per application on heterogeneous tiled displays when it performs stream reconfigurations.

To solve this problem we introduce SAGE Bridge into the SAGE architecture. SAGE Bridge is a high-speed bridging system that duplicates and splits pixel streams received from rendering clusters for each end-point. This allows each rendering node to stream whole images without considering the window layouts and tiled display configurations of multiple end-points. SAGE Bridge is deployed on high-performance PCs equipped with 10-gigabit network interfaces.

There are two different approaches in duplicating and partitioning images on SAGE Bridge. One is to copy images onto separate

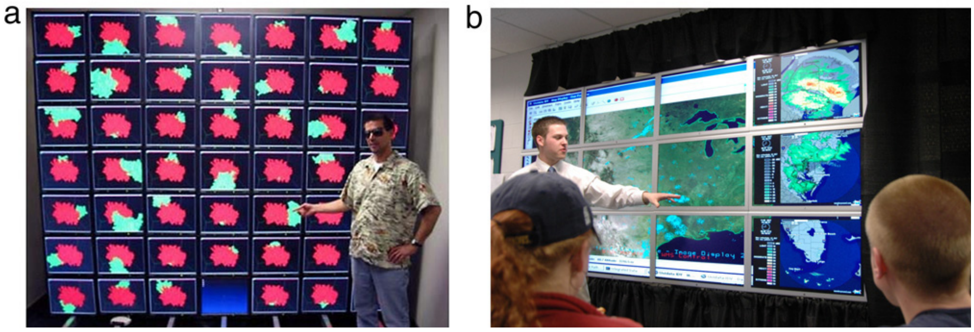


Fig. 2. The LCD HyperWall at NASA Goddard is on the left. NASA uses the individual tiles to show simulation models run with different initial conditions so researchers can quickly compare the results of multiple simulation conditions. A tiled display at the University of Michigan's Atmospheric, Oceanic and Space Sciences department is on the right. Undergraduate students use it to give presentations on class projects. The wall is used as a storytelling environment, much like a traditional poster board.

Table 1
Comparison of system features

	SAGE	SGE	XDMX	Chromium
Multitasking	Y	Y	Y	–
Window move/resize	Y	Y	Y	–
High-speed WAN support	Y	–	–	–
Scalable parallel application	Y	Y	–	Y
Scalable image multicasting	Y	–	–	–

network buffers for each end-point. This approach is associated with image frame based streaming used in the previous version of SAGE [6]. The other is pixel block selection, which is associated with a new pixel block based streaming method. This approach selects and streams different groups of pixel blocks for each end-point. The two approaches are compared and discussed in detail in Section 4.

The main contributions of this paper are:

- Proposing the Visualcasting approach to support distance collaboration in the SAGE framework,
- Enhancing Visualcasting performance by pixel block based streaming, and
- Showing by experiments that SAGE Bridge can support high-performance Visualcasting.

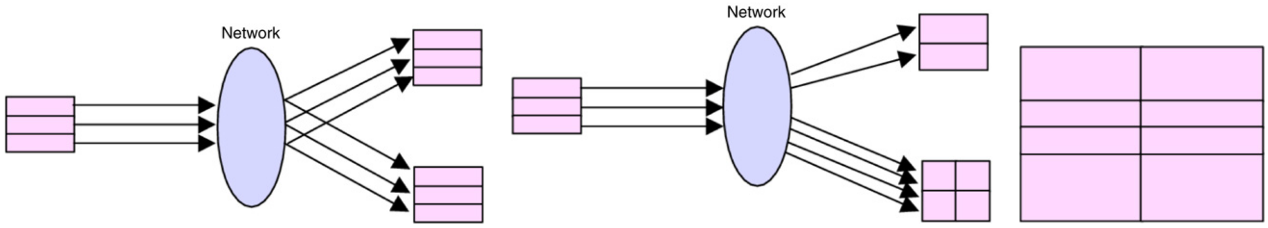
2. Related work

There are several existing systems supporting scalable high-resolution displays with parallel rendering schemes related to SAGE [1]. Perrine et al. [7] and Klosowski et al. [8] presented the merits of high-resolution display for various visualization applications using Scalable Graphics Engine (SGE) developed by IBM. SGE is a hardware frame buffer for parallel computers. Disjointed pixel fragments are joined within the SGE frame buffer and displayed as a contiguous image [7]. SAGE and SGE are similar in receiving graphics data from multiple rendering nodes and routing to high-resolution displays. Flexible scalable graphics systems such as Chromium [9] and Aura [10] are designed for distributing visualization to and from cluster driven tiled-displays. XDMX (Distributed Multi-head X11) is another system that can drive a tiled display. It is a front-end proxy X server that controls multiple back-end X servers to make up a unified large display [11]. Table 1 compares SAGE with the systems discussed so far. This table clearly shows that scalable image multicasting (Visualcasting) and high-speed wide-area network streaming support [6] is the most unique feature of SAGE. No other systems support those features. Our previous work, TeraVision [12], is a scalable platform-independent solution that is capable of transmitting multiple synchronized high-resolution video streams between single workstations and/or clusters. TeraVision also can stream graphics data over wide area networks. However, it has a static application layout on a tiled display. It is suitable for streaming a single desktop to a high-resolution tiled display, but unsuitable for supporting parallel applications or multiple instances of applications. To overcome these drawbacks, we developed SAGE. The Access Grid [13] is a system that supports distributed collaborative interactions over Grids. Although it enables remote visualization sharing, the major focus of the Access Grid lies in distributed meetings, conferences and collaborative work-sessions. Furthermore, the display resolution of remote desktop and Access Grid is limited to a single desktop resolution (at most 1600 × 1200 usually). On the other hand, SAGE can support 100 megapixel display walls and include these systems in the SAGE framework by adding a simple SAGE API to them.

Visualcasting may be implemented by using IP multicasting [14]. However, our problem has significant differences from the traditional multicasting problem as illustrated in Fig. 3. In the case of traditional multicasting, each end-point typically has the same number of end-hosts as the number of data sources as shown in Fig. 3(a). On the other hand, Visualcasting end-points have a different number of end-hosts from the number of data sources as shown in Fig. 3(b). Furthermore, the number of end-hosts can be dynamically changed because of display window operations.

To solve this problem through a multicasting approach, we need more multicast groups than the number of data sources. All data sources are required to partition data considering the window layout on the tiled displays of each end-point and to send each part of the data to the associated multicast groups. Fig. 3(c) shows how the image data has to be partitioned in order to solve the problem in Fig. 3(b) using the multicasting approach. Each image portion in Fig. 3(c) is associated with a multicast group. However, the number of multicast groups may explosively increase with the joining of new end-points or dynamic changes of display window layouts, so this multicasting approach will not be scalable. Moreover, this approach will have long delays in display window operations and joining of a new end-point, because dynamic changes of multicast group membership cause significant latency. Because of these reasons, the multicasting approach is not appropriate to solve our problem.

Our Visualcasting approach using SAGE Bridge is similar to application-layer multicast [15,16] in its basic idea and advantages over IP multicast. Since both Visualcasting and application-layer multicast duplicate data on computing node instead of routers, they do not require router support, so they can be easily deployed on existing networks. On the contrary, large parts of the Internet are still incapable of IP multicast. Application-layer multicast approaches, however, are typically designed for low-bandwidth data streaming applications with large receiver sets [16]. On the other hand, Visualcasting is designed for high-bandwidth large-scale data streaming for multiple tiled display clusters. Moreover,



(a) Multicasting. (b) Visualcasting. (c) Multicasting groups.

Fig. 3. VisualCasting problem.

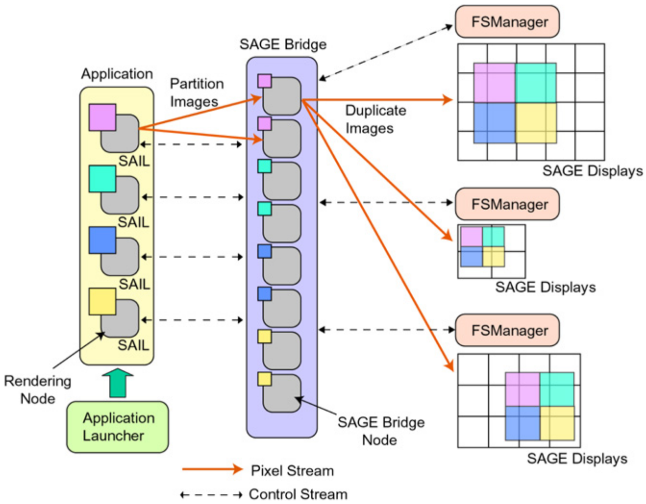


Fig. 4. New SAGE architecture.

existing application-layer multicast approaches have scalability and latency problems as well as IP multicast when they are applied to our problem.

3. Visualcasting architecture

Visualcasting scalably multicasts visualization from multiple rendering clusters to display clusters as shown in Fig. 3. Each rendering cluster can multicast multiple visualization applications, and display clusters can dynamically subscribe or unsubscribe a Visualcasting session. To enable Visualcasting in the SAGE framework, we devised and implemented SAGE Bridge in the SAGE architecture.

3.1. Enhanced SAGE architecture

Visualcasting requires significant changes in the previous SAGE architecture that was described in [6]. Fig. 4 shows the new SAGE architecture. A major difference is that the previous architecture has only one Free Space Manager (FSManager) but the new one has multiple FSManagers, in other words multiple SAGE sessions exist in this architecture. The SAGE Bridge is introduced between the SAGE Application Interface Library (SAIL) and SAGE Displays to intercept pixel streams from SAIL and to duplicate them for each SAGE session.

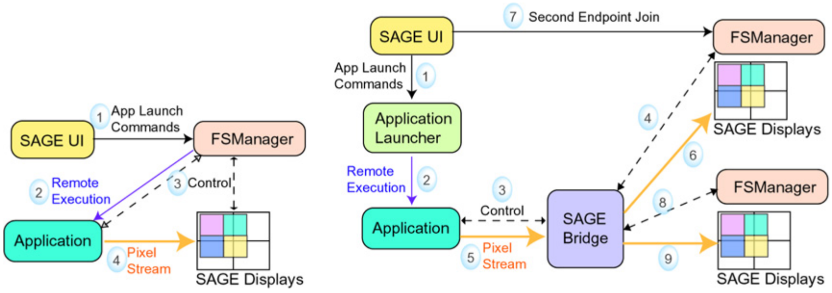
In the example of Fig. 4, a parallel application renders high-resolution images on a four-node rendering cluster. SAIL on each rendering node partitions its rendered images for two SAGE Bridge nodes and streams the sub-images. Each FSManager sends SAGE Bridge the application window layout on the tiled display that it controls. Each SAGE Bridge node receives three window layouts and generates three different image partitions associated with three window layouts. Then, it streams the partitioned images to the appropriate SAGE Display nodes.

For example, if we assume that the SAGE Bridge node on the top receives the top half of the image from SAIL, it partitions the image into four sub-images for the SAGE sessions on the top and the bottom but streams the whole received image without partitioning to the SAGE session in the middle. Whenever users move or resize one of the application windows, the FSManager sends the new window layout to the SAGE Bridge, which dynamically changes the image partition associated with the window and reconfigures the affected network streams. This dynamic pixel stream reconfiguration procedure [6] is performed independently for each SAGE session.

To support visualcasting, the SAGE architecture requires a new procedure to execute applications. Fig. 5 compares the previous and new application launch procedures. The so-called Application Launcher component is introduced to launch applications. The new procedure consists of nine steps:

1. A SAGE User Interface (UI) sends commands with application parameters and information about the SAGE Bridge and the first FSManager to the Application Launcher.
2. The Application Launcher executes an application on appropriate rendering nodes using information from the SAGE UI.
3. SAIL creates a control channel with the SAGE Bridge when the application is launched. The SAGE Bridge allocates SAGE Bridge nodes for the application and configures streams between SAIL and the SAGE Bridge.
4. SAGE Bridge connects to the first FSManager in order to configure the streams between the SAGE Bridge and the SAGE Displays.
5. SAIL starts streaming pixels once all configurations are completed.
6. Application images are displayed in the first SAGE session.
7. In order to allow the second SAGE session join the Visualcasting session, a SAGE UI sends a message that includes information about the second FSManager to the first FSManager.
8. The first FSManager directs the SAGE Bridge to connect to the second FSManager.
9. The pixel streams between the SAGE Bridge and the second SAGE session are configured and started.

As the number of Visualcasting end-points increases following the steps (7)–(9), the SAGE Bridge nodes that were initially assigned to an application become overloaded. To sustain the desired throughput, the SAGE Bridge dynamically allocates additional nodes for the application. The implementation of dynamic allocation of SAGE Bridge nodes is one of the most important pieces of future work in this research. In the current implementation, users specify a SAGE Bridge node for each application.



(a) Old application launch procedure. (b) Application launch procedure of visualcasting.

Fig. 5. Without and with VisualCasting application launch procedures.

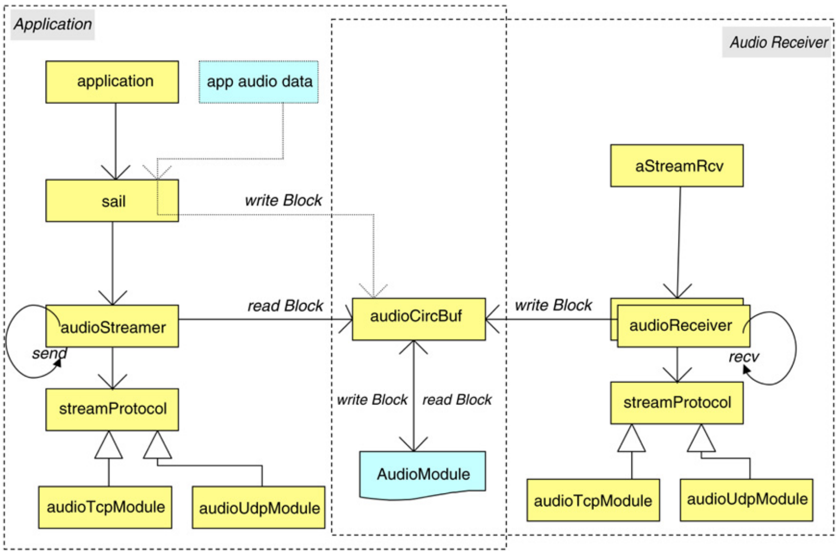


Fig. 6. Audio system is SAGE.

3.2. SAGE bridge architecture

SAGE Bridge includes the stream receiving part of a SAGE Display and the stream sending part of SAIL. These two parts are connected by circular buffers of image data. A SAGE Bridge node can simultaneously serve multiple application instances. The set of end-points is different for each instance. When a new application connects to the SAGE Bridge, an application instance object is created inside the SAGE Bridge. The object manages the list of end-points (FSManagers) subscribing the Visualcasting session of the application. When an end-point subscribes or unsubscribes, the object updates the list dynamically. The application instance object includes a SAGE Receiver, a circular buffer and SAGE streamers. The SAGE Receiver reads image data from network and writes them into the circular buffer. If the circular buffer becomes full, the SAGE Receiver is blocked until an image buffer in the circular buffer becomes writable. The circular buffer has buffer indices associated with each SAGE streamer to check if an image buffer is read by all SAGE streamers or not. Once all SAGE streamers read an image buffer, it becomes writable. Whenever a new end-point joins the Visualcasting session of an application, a SAGE streamer is created to support that end-point. SAGE streamers read image data from the circular buffer and take care of the image partitioning and streaming to display nodes and the dynamic pixel stream reconfiguration. This architecture allows independent window operations or dynamic pixel stream reconfiguration for each tiled display, and dynamic routing of image data from rendering clusters to display clusters. Since each SAGE streamer runs on a separate thread, image streams to each end-point are concurrent, and the stream reconfiguration of a SAGE streamer is totally independent from other SAGE streamers.

4. Audio streaming

Audio streaming is added to SAGE to support distributed collaboration and to enable audio application such as movie player and 4K-animation playback. The design goal is to treat sound data in SAGE in a similar fashion to pixel data. Sound buffers are captured at various sources (microphones, high-definition video capture card, audio files, or application-provided buffers), then streamed in an uncompressed form to minimize latency, and finally sent to a sound card for playback.

Another decision is to keep the system simple, flexible, and portable. The current system enables audio streaming in any SAGE application using audio capture or playback of attached sound files (multi-channel uncompressed WAV files). Simple synchronization is provided between audio and video streams. The audio streams can be sent to a sound card or written back to files. Fig. 6 shows the various steps from an application to audio playback from an audio receiver process. An audio buffer is simply defined by a sample type (float 32-bit, integer 24-bit, 16-bit or 8-bit), a sampling rate (44.1 kHz or 48 kHz usually), a number of channels (for mono, stereo, or multi-channel audio), and a number of audio frames per buffer block. Using a series of threads and circular buffers to handle the data flow, we are trying to minimize audio latency, critical for streaming applications.

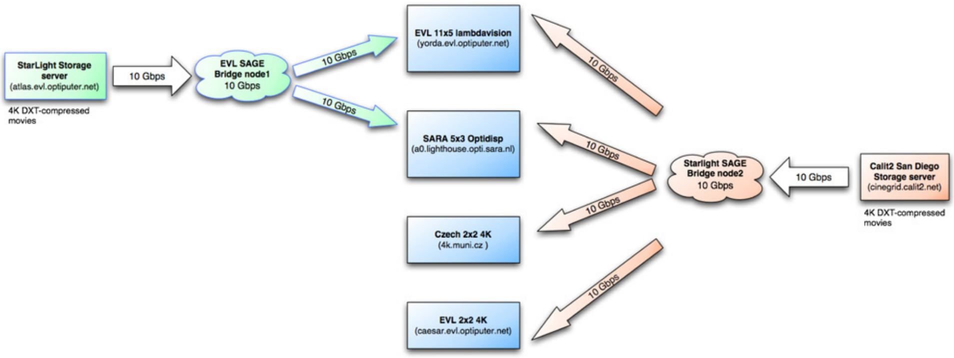


Fig. 7. GLIF testbed for international VisualCasting.

The current SAGE applications using the audio API are the HD video conferencing application (live capture from microphone) and the animation playback tool (audio file associated with each animation).

5. Applications and experiments

In this section, we present some applications and experiments showing the benefits of SAGE and Visualcasting to support distributed collaborative work. We present two implementations of high-definition video streaming with synchronized audio, and a high-resolution animation playback tool.

5.1. Applications

- HDS (i.e. High Definition Streaming): this application streams full-resolution high-definition video and audio from a pro-sumer video HDMI capture card. Using such a card (Blackmagic Intensity HDMI, PCI-express 1 ×, $350), it's possible to capture low-latency audio and video from HD cameras using a HDMI port (such cameras by Sony, Canon, Panasonic, ...). The captured video frames in YUV422 format are passed directly to SAGE for streaming. SAGE supports natively and efficiently such a pixel format using a simple graphics shader implemented on the display side. The total bandwidth can be controlled by the application frame rate and usually kept under 1 Gbps without any significant quality loss.
- iHD1500 is the broadcast-quality HD streaming solution developed by the University of Washington/Research Channel. It provides very high quality audio and video and some multi-point capabilities, for a bandwidth of 1.5 Gbps and a much higher price range than HDS. Recently the same group ported their application to the SAGE environment, giving them many more display options. The iHD1500 video stream is decoded by a 'bridge' machine that converts each frame into a SAGE buffer. The audio stream is kept unchanged and plays synchronously with the video. This association of SAGE and iHD1500 provides the best of both environment in terms of flexibility and quality.
- The last application considered is 4K DXT-compressed animation playback. Scientific animations are an essential tool for education and dissemination of results. To preserve all the intricate details of advanced simulations and rendering (such as climate or molecular simulations), the highest resolution is needed, at minimum HD or better 4K. To achieve interactive speed, real-time compression techniques play a key role. Even in this day of exponential increase of networking bandwidth, compression is required to alleviate the next bottlenecks in modern computer systems: storage speed and memory bandwidth. We use software DXT compression for high-resolution content at an interactive speed. DXT compression allows to stream full HD video over a gigabit connection and 4K streaming without the need for a high-end storage system or an expensive codec. Frame decoding is handled by most of modern GPUs.

5.1.1. GLIF experiment with Calit2, EVL, SARA and Brno with 4K visualcasting

Several groups of the GLVF decided to use the resources available to GLIF members to prove the viability of the visualcasting idea. Storage nodes were setup at Calit2/UCSD and at the Starlight facility in Chicago. The goal was to distribute and share the various 4K content between display nodes in Chicago, Amsterdam and Brno. 10-gigabit GLIF networks interconnect all the sites. We used our 28-node LambdaVision cluster [19] – a tiled display built from 55 LCD screens with a total resolution of 100 megapixels. Each cluster node has dual AMD 64 bit 2.4 GHz processors, an Nvidia Quadro3000 graphics card, 4 GB of main memory, and 1-gigabit Ethernet (GigE) network interface fully connected to each other through a gigabit local network switch. VisualCasting used some high-performance PCs equipped by two dual-core AMD 64 bit 2.6 GHz processors and a 10-gigabit Myricom network card. All the networks and interfaces used were jumbo-enabled to enable high-bandwidth over long distances. We used a similar machine in Brno, Czech Republic, as a display node, and finally a 5 × 3 tilde display at SARA, Amsterdam. The testbed and experiments are summarized in Fig. 7, where 4K streams are sent from Chicago and San Diego simultaneously to four distinct tiled displays around the world. Fig. 8 shows the four displays showing the same animation synchronized, every site showing the same animation frame exactly at the same moment.

5.1.2. SC07 with 4K visualcasting and iHD1500 streaming

During the SuperComputing 2007 conference (SC'07) in Reno, Nevada, we demonstrated again the visualcasting capabilities of SAGE by streaming several new 4K animations from Amsterdam to both the SARA and EVL booths synchronously, as shown in Fig. 9. Fig. 10 is a testimony of the iHD1500 work with SAGE, where Prof. Larry Smarr was discussing with a Professor Ginger Armbrust in Seattle, WA, about her study of Puget Sound. Both endpoints run SAGE and the iHD1500 software handles the HD video conferencing.

6. Conclusion

In this paper, we showed that SAGE Bridge could scalably support real-time distribution of extremely high-resolution images. High-definition audio/video streams and 4K scientific animations were successfully distributed to several locations synchronously using high-speed networks. Visualcasting implemented by SAGE

Fig. 8. Sharing a 4K animation across several countries.

Fig. 9. SARA scientists in front their display at SC07 while streaming several 4K animations from Amsterdam to the show-floor in Reno, NV.

Fig. 10. Prof. Larry Smarr (left) engaging a live discussion from the Research Channel booth (Reno, NV).

Bridge is essential for high-resolution collaborative visualization environment. Visualcasting allows an upgraded collaboration experience by allowing scientists to experience the same content across sites and to engage into successful scientific collaboration.

Acknowledgements

The Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago specializes in the design and development of high-resolution visualization and virtual-reality display systems, collaboration software for use on multi-gigabit networks, and advanced networking infrastructure. These projects are made possible by major funding from the National Science Foundation (NSF), awards CNS-0115809, CNS-0224306, CNS-0420477, SCI-9980480, SCI-0229642, SCI-9730202, SCI-0123399, ANI-0129527 and EAR-0218918, as well as the NSF Information Technology Research (ITR) cooperative agreement (SCI-0225642) to the University of California San Diego (UCSD) for "The OptIPuter" and the NSF Partnerships for Advanced Computational Infrastructure (PACI) cooperative agreement (SCI-9619019) to the National Computational Science Alliance. EVL also receives funding from the State of Illinois, General Motors Research, the Office of Naval Research on behalf of the Technology Research, Education, and Commercialization Center (TRECC), and Pacific Interface Inc. on behalf of NTT Optical Network Systems Laboratory in Japan. The GeoWall, GeoWall2, Personal GeoWall2 (PG2), and LambdaVision are trademarks of the Board of Trustees of the University of Illinois.

References

[1] Han Chen, Yuqun Chen, Adam Finkelstein, Thomas Funkhouser, Kai Li, Zhiyan Liu, Rudrajit Samanta, Grant Wallace, Data distribution strategies for high-resolution displays, Computers & Graphics 25 (5) (2001) 811–818.
[2] J. Leigh, et al., The global lambda visualization facility: An international ultra-high-definition wide-area visualization collaboratory, Future Generation Computer Systems 22 (2006).
[3] L.L. Smarr, A.A. Chien, T. DeFanti, J. Leigh, P.M. Papadopoulos, The OptIPuter, Communications of the ACM 46 (11) (2003) 58–67.
[4] J. Leigh, et al. An experimental OptIPuter architecture for data-Intensive collaborative visualization, in: Third Workshop on Advanced Collaborative Environments, 2003.
[5] L. Renambot, et al. SAGE: The scalable adaptive graphics environment, in: Proc. WACE 2004, Sept 23–24, 2004.
[6] B. Jeong, L. Renambot, R. Jagodic, R. Singh, J. Aguilera, A. Johnson, J. Leigh, High-performance dynamic graphics streaming for scalable adaptive graphics environment, in: Proceedings of SC06, November 2006.
[7] K.A. Perrine, D.R. Jones, W.R. Wiley, Parallel graphics and interactivity with the scaleable graphics engine, in: Proceedings of ACM/IEEE Conference on Supercomputing, 2001.
[8] J.T. Klosowski, P. Kirchner, J. Valuyeva, G. Abram, C. Morris, R. Wolfe, T. Jackman, Deep view: High-resolution reality, IEEE Computer Graphics and Applications 22 (3) (2002) 12–15.

[9] G. Humphreys, et al. Chromium: A stream-processing framework for interactive rendering on clusters, in: Proceedings of SIGGRAPH, 2002.
[10] D. Germans, H.J.W. Spoelder, L. Renambot, H.E. Bal, VIRPI: A high-level toolkit for interactive scientific visualization in virtual reality, in: Proceedings of Immersive Projection Technology/Eurographics Virtual Environments Workshop, 2001.
[11] Distributed multi-head X project. http://dmx.sourceforge.net/.
[12] R. Singh, B. Jeong, L. Renambot, A. Johnson, J. Leigh, TeraVision: A distributed, scalable, high resolution graphics streaming system, in: Proceedings of IEEE Cluster, 2004.
[13] L. Childers, T. Disz, R. Olson, M.E. Papka, R. Stevens, T. Udeshi, Access Grid: Immersive group-to-group collaborative visualization, in: Proceedings of Fourth International Immersive Projection Technology Workshop, 2000.
[14] S. Deering, D. Cheriton, Multicast routing in internetworks and extended LANs, ACM SIGCOMM 1988, Stanford, CA, August 1988.
[15] J. Jannotti, D.K. Gifford, K.L. Johnson, M.F. Kaashoek Jr., Overcast: Reliable multicasting with an overlay network, in: Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI), October 2000.
[16] S. Banerjee, B. Bhattacharjee, C. Kommareddy, Scalable application layer multicast, Technical report, UMIACS TR-2002. 2002.
[17] Bram Stolk, Paul Wielinga, Building a 100 Mpixel graphics device for the OptIPuter, Future Generation Computer Systems 22 (2006) 972–975.
[18] B. Jeong, L. Renambot, R. Singh, A. Johnson, J. Leigh, High-performance scalable graphics architecture for high-resolution displays, EVL Technical Document Technical publication 20050824_Jeong, 2005.
[19] LambdaVision. http://www.evl.uic.edu/cavern/lambdavision.
[20] Stephanie Teasley, Lisa Covi, M.S. Krishnan, Judith S. Olson, How does radical collocation help a team succeed? in: CSCW '00: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work, ACM Press, New York, NY, USA, 2000, pp. 339–346.
[21] Stephanie D. Teasley, Lisa A. Covi, M.S. Krishnan, Judith S. Olson, Rapid software development through team collocation, IEEE Transactions on Software Engineering 28 (7) (2002) 671–683.
[22] J. Leigh, M. Brown, Cyber-commons: Merging real and virtual worlds, Communications of the ACM 51 (1) (2007) 82–85.

Luc Renambot received a Ph.D. at the University of Rennes-1 (France) in 2000, conducting research on parallel rendering algorithms for illumination simulation. Then holding a Postdoctoral position at the Free University of Amsterdam, till 2002, he worked on bringing education and scientific visualization to virtual reality environments. Since 2003, he joined EVL/UIC first as a PostDoc and now as Research Assistant Professor, where his research topics include high-resolution displays, computer graphics, parallel computing, and high-speed networking.

Byungil Jeong is a Ph.D. graduate from the Department of Computer Science at the University of Illinois at Chicago (UIC). He now works at Sun Microsystems.

Hyejung Hur is a Ph.D. student in the Department of Computer Science at the University of Illinois at Chicago (UIC).

Andrew Johnson is an Associate Professor in the Department of Computer Science and member of the Electronic Visualization Laboratory at the University of Illinois at Chicago. His research focuses on the development and effective use of advanced visualization displays, including virtual reality displays, auto-stereo displays, high-resolution walls and tables, for scientific discovery and in formal and informal education.

Jason Leigh is an Associate Professor of Computer Science and director of the Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago. Leigh is a co-founder of VRCO, the GeoWall Consortium and the Global Lambda Visualization Facility. Leigh currently leads the visualization and collaboration research on the National Science Foundation's OptIPuter project, and has led EVL's Tele-Immersion research agenda since 1995. His main area of interest is in developing collaboration technologies and techniques for supporting a wide range of applications ranging from the remote exploration of large-scale data, education and entertainment.