SAND99-0890C

Architecture for Teraflop Visualization

Arthurine R. Breckenridge
Rena A. Haynes
Sandia National Laboratories' Interaction Laboratory
P.O. Box 5800, MS 0318
Albuquerque, NM 87185

Abstract

Sandia Laboratories' computational scientists are addressing a very important question: How do we get insight from the human combined with the computer-generated information? The answer inevitably leads to using scientific visualization. Going one technology leap further is teraflop visualization, where the computing model and interactive graphics are an integral whole to provide computing for insight. In order to implement our teraflop visualization architecture, all hardware installed or software coded will be based on open modules and dynamic extensibility principles. We will illustrate these concepts with examples in our three main research areas: 1) authoring content (the computer), 2) enhancing precision and resolution (the human), and 3) adding behaviors (the physics).

Keywords: computing for insight, teraflops, massively parallel, stereoscopic projection, haptics

1. Historical Background

A multi-platform, dynamically extensible 2D+ environment, the World Wide Web, is evolving. Lessons from the Web's development have contributed the success to open modules and dynamic extensibility. To understand these principles, let's take a look at the historical background of networking and Internet information access.

In the network, TCP/IP packets emerged to provide data in modular units with easy, open format. Pushing and pulling packets over "all components must be up" routes was a major success. However, due to the large learning curve to use such a network, its usage was limited only to highly trained computer specialist. Finally, the Internet really became of age when the network topologies could be dynamically determined and extended. Within two decades, networking was available to the masses.

The Internet information evolution had an elite group of individuals who knew how to use gopher, ftp, and other networking protocols to access information on the Internet. HTTP [Berners-Lee94] emerged to provide data in modular units with easy open, common access methods. It decisively displaced almost all challengers. Tim Berners-Lee showed us that clean breaks between functions allow dynamic loading and unloading of a task into a piece of software as easily as a new information content. Within a few years after the implementation of open modules and dynamic extensible services (irrespective of location), ubiquitous Internet information access was available to the masses.

Traditionally, visualization programmers have developed overarching, monolithic software by adding features to codes containing 10,000+ lines. The usefulness of this approach was questioned with the speed of innovation happening with information access on the Internet. There are certainly aspects of this Internet revolution worth repeating. In the following sections,

DISCLAIMER

we will give a brief overview about an effort in progress to apply these world-changing concepts to 3D+ environments and how Sandia Lab's Interaction Laboratory plans to contribute.

2. Overview of Teraflop Visualization Architecture

Motivation: The Intel ASCI red teraflop [Gorham99] machine provides a unique opportunity for computing for insight. This machine is, simplistically, a collection of personnel computer processors. One could estimate that it would take almost 10 years to compute the same problem on a single PC that the teraflop machine can do in less than a day [(4096 processors X 18 hours)/24 hours in the day X 365 days in the year]. This machine has proven to be one of the more technically successful efforts in massively parallel computing by bounding or eliminating non-scalable components. Using SGI Onyx machines for post-processing data has been identified as a non-scalable component in the modeling and simulation process.

Many system architectures have derived their names from self-organizing phenomena such as hives of bees, swarms of piranha, tornadoes, etc. The teraflop architecture is better understood in rough analogy to two plants.

- **Kudzu** (named after the plant that grows aggressively) hardware concepts [Greenberg97] are based on participation in ACSI efforts in massively parallel, high-performance computing. The hardware must be designed from open modules that enable the machine to adapt to change, to grow and to be pruned.
- **Bamboo** (named after another plant that grows aggressively) software concepts [Watsen98] are based on participation in W3D (formerly VRML) working groups. Taking a cue from Eric Raymond's paper, "The Cathedral and the Bazaar" [Raymond97] and the remarkable success of the HTTP, a facility for rapid development while promoting transparent code reuse via community wide exchange is being built.

The hardware and software conceptual designs are simple and intertwined. Open modules are the basic building blocks to provide hardware resources (CPU, memory, displays, networks, graphics acceleration) and software resources (object, components, inter-module dependencies). The definition of an open module is intended to be as non-specific as possible in order to speed innovation. Dynamic extensible services are to be constructed from independent modules.

3. Related Work

Related Work on Hardware Architectures: Until recently, work in the area of parallel visualization hardware has been dominated by SGI except for the University of North Carolina's Pixel Plane projects [Olano98]. Related scalability work generally discusses small-, medium-, or even large-scale MPP systems and the problems they address in scaling CPUs or processor activities. Large-scale visualization research has primarily focused on techniques for reducing the data rather than scaling up visualization hardware [Smith99]. Other research addresses work in overcoming limitations in networking, memory, and disk storage.

Related Work on Software Architectures: Scientific visualization has focused on the developments of virtual environments such as SNL's EIGEN/VR [Breckenridge98], CAVERNSoft [Cruz-Neira92], and other efforts. However, these efforts tend to be monolithic architectures, difficult to include other research, and/or available on only a few platforms. Our first re-design task to utilize parallel hardware was to compare our teraflop visualization requirements with current software frameworks. We hoped to avoid using only object-oriented development since its reliance on very fine-grained components (objects) would force developers to work at too low of a level. Currently, the industry embraces three major de-facto component interoperability standards: Microsoft's COM (including DCOM) for OLE (object linking and embedding) and ActiveX components, JavaBeans for Java components, and CORBA (common object request broker architecture) for C/C++. Each framework had major contributions and disappointments (not heterogeneous, performance with OpenGL, heavy weight installations, Internet restrictions through firewalls, excessive amount of network traffic while delaying latency critical interactions).

Shown in Figure 1 is another approach proposed in the VRTP (Virtual Reality Transport Protocol) white paper [Brutzman97]. The underlying kernel is called Bamboo. This modular, extensible, dynamic dependency prototype is multi-organization open source effort. Its goal is achieved by modeling the WWW browser plug-in metaphor for 3D+ simulation. The Bamboo *root* supports a basic callback tree. Each function is developed under modular units with lightweight open, common syntax methods. Bamboo then extends this metaphor by adding inter-module dependencies. A Bamboo dependency would be called a *shoot* (i.e., teraflop visualization module). Bamboo code implements platform independence by using OS function calls through ACE wrappers and OpenGL, C++, and STL (standard template library). Sandia's Interaction Laboratory has chosen to participate in the Bamboo open coding effort.

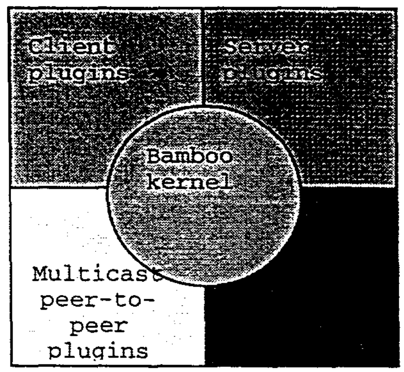

Figure 1.VRTP components

4. Authoring Content

Hardware Modules: The ASCI Red Machine and other ultracomputers provide a unique opportunity to author content. Computer speeds, traditionally measured in MIPS (millions of instructions per second), are now measured in "TeraOp" rates on today's ultracomputers. These rates are achieved by combining processing speed-ups (Moore's Law) with massively parallel technology. Speed-ups in graphics cards have also occurred (primarily due to Moore's Law). AGP graphics output (number of triangles/sec) is 10 times better than hardware-accelerated performance at the 1996 level. Despite these speed-ups, hardware resources will always be in shortage and need to be used parsimoniously.

A hardware module is simply a module that should work as needed in conjunction with other hardware modules (irrespective of location). Each year, a new phase, or branch, will be added to the Kudzu plant to increase its capability with the latest, cost-effective hardware components. Older, possibly obsolete hardware will be pruned from the plant. In order to realize this grow and prune strategy, the plant must be designed from modules that enable the machine to adapt to change. The architecture diagram would simply be the Internet "cloud". Our environment includes the large-scale MPP Intel Teraflop machine running a custom operating system, a medium-scale cluster of Digital Alpha running Red Hat Linux, and a small-scale Compaq cluster running NT.

Software Modules: A customer will be able to choose not only PC versus workstation but also UNIX and NT. The graphics software should run on any of these combinations and in parallel quantities. OpenGL has emerged to provide simple geometric objects. Bamboo has started to emerge. Bamboo launches from a 2D plug-in and loads 3D plug-in modules. Then, plug-in modules (called Bamboo shoots) will be accessible throughout the Internet making it easy to share work from other research and commercial efforts. Our effort is divided into a handful of distinct and well-defined focus modules. Figure 2 illustrates the abstract logical view of the callback tree for Sandia's teraflop visualization shoot module(s).

The physical graphics termination point, usually the visual display, is the largest single resource that must be managed. In the purest form, an application creates and sends data to be displayed in exactly the format required by the graphics hardware. In reality, data units are massaged, manipulated, stored, retrieved, transmitted, many times before ending up on the computer screen. Often rasterization accelerators can rasterize more triangles than the geometry (done in the CPU) can transform. New techniques such as parallel volume rendering [Silva96] and dynamic isosurface generation [Shen96] have the potential for real-time processing. Visualization can be very compute intensive.

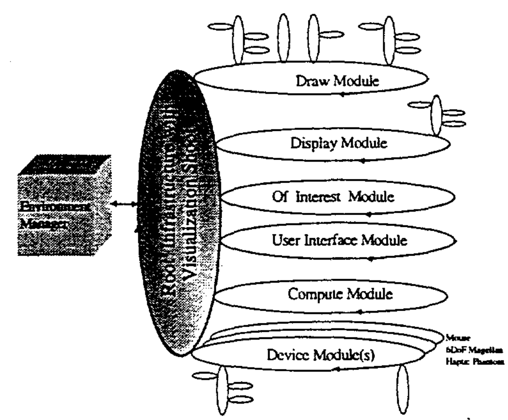

Figure 2. Callback Tree

Compute modules are glued together with a data movement abstraction. Message Passing Interface, MPI, has become the data movement library of choice for applications [Clark94]. Developing functionality via modules allow experimentation with other simple data movement mechanisms that take maximum advantage of fast network hardware such as Myrinet [Seizovic94]. The compute modules will be able to dynamically detect low-level dependencies of the hardware.

Memory in teraflop computing is no longer a homogenous shared component. While simulations that run on small-scale and/or medium-scale configurations may use shared memory, large-scale simulations must decompose their problems across distributed memory. Currently at Sandia, simulation output data are collected in parallel from distributed sources and delivered as a single input stream to visualization [Chen98]. As we couple visualization compute modules to simulation computation, our draw and user interaction modules have been coded to utilize distributed memory resources.

The "of interest" management modules are needed to support visualizing and exploring data distributed based on spatial, temporal, and/or functional attributes. Draw and user interface modules create objects and register for events on the objects. Modules may register for events in remote as well as local domains. A module instance monitors objects for modules within an "of interest" domain and communicates with its interest management peers when an event occurs on remotely registered objects. Global events are broadcast to all participating "of interest" management modules. The end result is a virtual environment for the analysis of the content generated by simulations.

5. Enhancing Precision and Resolution

Hardware Modules: The ASCI Red Machine or other ultracomputers provide a unique opportunity to increase resolution and precision of the simulation. However, the bottleneck of the display hardware becomes very noticeable. For example, much of the detail of a 100M cell+ simulation is often removed in order to visualize the data on a 1280x1024-resolution monitor. Alternative output devices such as CAVE™-like environments or stereoscopic wall projection units have worked on this limitation. Further technical enhancements are needed as well as reducing the cost of these tools. Figure 3 shows an illustrated drawing of our research display facility called the KIVA (no acronym; the modern version of a large chamber in a New Mexico Pueblo village).

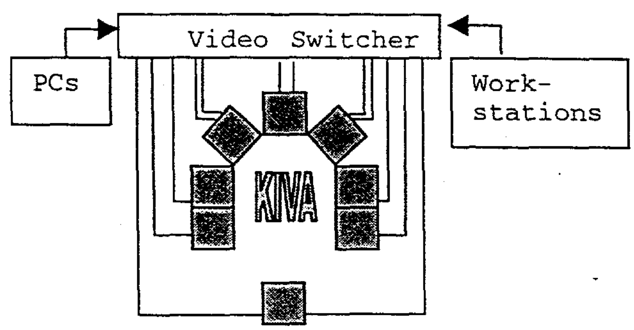

Figure 3. Illustration of KIVA

Specifically, the KIVA as installed in the Interaction Laboratory, is a theater made of eight rear-projected modules. The projector's optics are folded by mirrors onto translucent, no gain, opalescent acrylic screens. Each module is self-enclosed so a light-tight room is not required. High resolution LCD projectors with wide angle lenses throw full-color computer generated fields (1024X768pixel) onto each 6X8 feet screen raised 2 feet from the ground, giving approximately 7,000 linear pixel resolution to the surrounding composite image. One projection unit does not participate in the composite image. This unit is reserved for next-generation testing (autosteroscopic) and current generation needs (video conferencing, a printable electronic white board). All KIVA units are connected to a video switcher to allow output from several computers (Irix SGIs, NT PCs, Linux PCs) and video equipment. Computer controlled spatial sound is provided using four channel audio equipment and Acoustetron II.

Our simulation codes are moving to 3D as quickly as possible. On going developments have led to an increasing demand for 3D displays. Indeed, a broad range of fairly mature 3D equipment is already on the market. The available systems, however, suffer from the drawback that users have to wear special devices to separate the left eye's and the right eye's images. Current approaches for "aided view" include color, polarization, time and location multiplexing. Color rivalry of filtering near-complementary colors and unpleasant aftereffects restrict the use of anaglyphic (color-multiplexing) methods. In the early 1990's, location multiplexed displays such as Binocular Omni-Orientation Monitor (BOOM [Bolas94]), proved expensive and customers did not like the natural surroundings being occluded from sight. Today, our single monitor systems have matured to use time-multiplexed displays with LC-shutter goggles. Time multiplexing is limited for video projection due to the impairing cross talk that results from persistence of CRT phosphors, particularly of the green phosphor. Adding "fast" phosphors lead to noticeable loss of brightness not to mention cost. Our KIVA uses polarization multiplexing which is well suited for video projection. The basic setup consists of two projectors mounted on racks with full 6DOF adjustment as shown in Figure 4. Orthogonal oriented filter sheets cover the projector lenses. The customer wears linear polarizing glasses that are inexpensive and lightweight. The success of using a 3D large projection screen to gain insight into the complex phenomenon coded in the simulations has been demonstrated over and over again.

Software Modules: The KIVA provides an effective environment to implement parallel display objectives. Each Bamboo object knows the number of times it is referenced and has a synchronization primitive, thus enabling thread-safe access. For example, a display module to support multiple viewports can quickly load and callback drawing modules with the object at different positions or level of detail for unprecedented quantitative and qualitative analysis.

Figure 4. 6DOF Rack

The main strength of the module approach is to allow a range of displays from the single desktop monitor to the composite image of the KIVA theatre. The hardware appears transparent to the customer. When each module loads, it verifies the immediate dependencies and loads only those modules not already memory resident. Modules to support SGI's Multiple Channel Option for composite images are currently available. Sandia is participating in two efforts to accomplish composite images on parallel machines connected to multiple pipes. A "pipe" in this context refers to the software structures, buffers, and the state of the graphics hardware associated with each single window. First, the successful effort by Princeton University [Blumrich98], allows displaying a composite image across multiple projectors. Enhancements are planned to remove the memory requirement that each projector's pipe must contain the whole image even though a portion of the image is displayed on each projector. Second, in conjunction with SUNY, Sandia has developed TNT_PMESA, a parallel version of MESA (a 3D graphics library that utilizes the OpenGL command syntax and state machines) that can be called by parallel (or non-parallel) surface creation programs [Mitra98]. SGI's open source release of the OpenGL specification will have an enormous impact on development of accelerated 3D OpenGL drivers with direct rendering. We use the term direct rendering to mean the client applications are allowed to directly access the graphics hardware, with an absolute minimum amount of intervention for the operating or windowing system (X11, Windows). Sounds risky. Actually, windows into memory or direct memory access (DMA) hardware are well understood in the parallel computing environments.

6. Adding Behaviors

Hardware Modules: The ASCI Red Machine or other ultracomputers provide a unique opportunity to add behaviors to the simulation. The 2DOF mouse is being upgraded to what we called the "next-generation mouse" defined as an interface that has at least 6DOF or more. Adding an additional 3DOF with force feedback, as shown in Figure 5 is also a "nice touch".

Software Modules: Let's consider the current human/computer interface software, or the lack thereof. To solve problems with the aid of a computer, a customer must be able to easily and accurately communicate to the machine and control what needs to be done. Designing a means to do so (i.e., the human/computer interface) is by no means trivial. Standardization of user interfaces on X11 and Windows have come a long way in changing computing. However, these de-facto standards map into 2D space. Continuing to use 2D widgets reduces the look and feel so any object can never be more than a flat, not intuitive, multiple failure-prone command operation.

Our bamboo shoot supports event handler, user interface, and device modules. Using a token structure, appropriate modules can be included to match the device(s) available to the customer. This structure allows for virtual reality concepts to be consistent across platforms from a desktop with a mouse and keyboard to more elaborate peripherals such as voice recognition and haptic devices. The underlying code is based on applying methods to operations such as rotation and velocity not action/reaction to a mouse button being pushed. The user interface has been implemented to use VRML widgets and OpenGL objects to allow 6DOF behaviors [Anderson98].

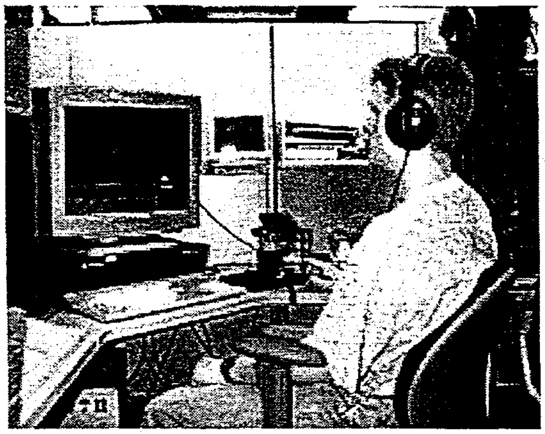

Figure 5. Audio and Haptics Feedback

7. Future Research

The most exciting aspect of parallel modular systems is the new unlimited scope a project may take. In the past, making a change meant recompiling the complete software, which severely limited the size of our undertakings. Bamboo modules allow developers to have lateral conversations through the transparency of the 3D source. Removal of the need for either compiling the results or seeking permission, certification, or registration for anyone else has led to the largest example of parallel development seen to date, and the results will be world changing.

In conclusion, teraflop visualization will be made possible with the steepest adoption curve in history. Computing for insight is happening both faster and better since open modules and dynamic extensibility principles are being applied.

References

Anderson, T. FLIGHT, Feedback, MuLti-dimensional Interface as a General Human-Computer Technology. User and Programmer's Manual, Sandia National Laboratories, 1998.

Berners-Lee, T. and D.Connolly. HyperText Markup Language Specification –2.0, 28 November 1994. ftp://ds.internic.net/internet-drafts/draft-ietf-html-spec-00.txt

Matthias Blumrich, Richard Alpert, Yuqun Chen, Douglas Clark, Stefanos Damianakis, Cezary Dubnicki, Edward Felten, Liviu Iftode, Kai Li, Margaret Martonosi, and Richard Shillner. Design Choices in the SHRIMP System: An Empirical Study. Proc.of the ACM/IEEE 25th

International Symposium on Computer Architecture, Spain, June 1998, pp. 330-341.

Bolas, M.T., E. R. Lorimer, I. E. McDowall, R. X. Mead. Proliferation of Counterbalanced, CRT-based Stereoscopic Displays for Virtual Environment Viewing and Control. Proc. SPIE 2177, 1994, pp. 325-334.

Breckenridge, A. Teraflop Visualizaton. Virtual Environment '98 Conference, Stutgart, Germany, June 16-18, 1998.

Brutzman, D., M.Zyda, et al. Virtual Reality Transfer Protocol (VRTP) Design Rationale. Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises (WET ICE), MIT, Cambridge MA. http://www.stl.nps.navy.mil/~brutzman/vrtp_design.ps.

Chen, P., M. Christon, J. Sturtevent, P. Heermann, PDS/PIO: Lightweight Libraries for Collective Parallel I/O. Proc. SC 98, IEEE Computer Society Press, Orlando, FL, November 1998.

Clark, L., I. Glendinning, R. Hempel. The MPI Message Passing Interface Standard. Technical report, Edinburgh Parallel Computing Centre, The University of Edinburgh, 1994.

Cruz-Neira, C., D. J. Sandin, T. A. DeFanti, R. Henyon, and J. C. Hart. The CAVE, Audio Visual Experience Automatic Virtual Environment. Communications of the ACM, June 1992.

Gorham, E. Accelerated Strategic Computing Initiative. http://www.sandia.gov/ASCI, 1999.

Greenberg, D. S., R. Brightwell, L. Fisk, A. Maccabe, R. Riesen. A System Architecture for High-End Computing. Proc. of SC97, IEEE Computer Society Press, San Jose, CA, November 1997.

Mitra T. and T. Chiueh. Implementation and Performance Evaluation of Parallel Mesa Library. IEEE International Conference on Parallel and Distributed Systems (ICPADS), December 1998. See http://www.cs.sandia.gov/VIS/pmesa.html

Olano, T. Marc. A Programmable Pipeline for Graphics Hardware. Doctoral Dissertation, University of North Carolina, 1998.

Raymond, E. Cathedral and the Bazaar. http://www.tuxedo.org/~esr/writings/cathedral-bazaar.

Seizovic, J. Pipeline Synchronization. Prec. of the International Symposium on Advanced Research in Asynchronization Circuits and Systems, IEEE Computer Science Press, 1994.

Shen, H., C. Hansen, Y. Livnat, C. Johnson. Isosurfacing in Span Space with Utmost Efficiency (ISSUE). Visualization '96, San Francisco, CA.

Silvia, C. A. Z. Kaufman, C. Pavlakos, PVR: High Performance Volume Rendering. IEEE Computational Science and Engineering, Winter 1996, pp. 18-28.

Smith, P. H., et al. Department of Energy Data and Visualization Corridors Principal Investigators' Meeting, Pasadena, CA., March 1999.

Watsen, K., M. Zyda. Bamboo-- A Portable System for Dynamically Extensible, Real-time, Networked Virtual Environments. 1998 IEEE Virtual Reality Annual International Symposium (VRAIS'98), Atlanta, GA.

This work was supported by Sandia National Laboratories, a multi-program laboratory operated by Sandia Corporation (a Lockheed Martin company) for the United States Department of Energy under contract DE-AC04-94AL85000.