High Speed, Wide Area Distributed Computing for Scientific Imaging

William E. Johnston[1], *Van L. Jacobson,*
Stewart C. Loken, David W. Robertson, and Brian L. Tierney

Lawrence Berkeley Laboratory
Berkeley, CA 94720

Abstract

We present a scenario for a fully distributed computing environment in which computing, storage, and I/O elements are configured on demand into "virtual systems" that are optimal for the solution of a particular problem. We also present an experiment that illustrates some of the elements and issues of this scenario. The goal of this work is to make the most powerful computing systems those that are logically assembled from network based components, and to make those systems available independent of the geographic location of the constituent elements.

Introduction: Advances in software paradigms, computing systems, and communications bandwidth over the next few years will help enable an information analysis environment in which scientists have uniform and unimpeded access to computing and data resources regardless of their geographic location. This environment will provide a "just in time" approach to assembling the resources needed to solve specific instances of problems in computational simulation, data acquisition, data analysis, and archiving. It will allow us to design optimal architectures for the solution of specific problems, and then, by using network based resources, to logically assemble and use the required elements only for the time during which they are needed. These resources will consist of (1) computing elements (workstations, parallel and vector processors, and specialized processors for encryption, compression, and graphics rendering), (2) data handling elements (large, high speed data "buffers", and distributed mass storage systems), (3) graphics/image display user front end systems, and (4) the software systems to easily interconnect these elements. Not only will this allow powerful capabilities to be brought to bear on large problems, it will also allow access to these capabilities by a much wider community of people than is presently possible. This environment will be enabled through software and hardware architecture advances expected over the next several years, including: an order of magnitude increase in workstation I/O and memory bandwidth; the routine incorporation of co-processors for special tasks (e.g. video compression); the emerging collaboration between the computing and telecommunications industries for high bandwidth networking; hardware and software improvements permitting multiple heterogeneous computing systems to be easily configured into cooperating elements that form virtual systems; easy access to massive unique data archives enabled through advances in data management and mass storage systems, and; user interface paradigms evolved to allow non-computer specialists to easily assemble the above elements into effective tools to attack scientific problems.

The Research Imaging Paradigm: Configurable systems are essential to many scientific endeavors. For example, the research imaging environment is characterized by three elements that are typically geographically dispersed: the imaging device, and its associated control system; computational and data storage elements necessary for processing speed, large memory, high speed data buffers, etc., to capture and interpret the GBytes of data from the imaging device; local workstations for user control of the operation of the imaging device, and the display of the resulting images and visualizations. Advanced scientific imaging will

---

[1]For further information please contact Bill Johnston, Bld 50B-2239, Berkeley, CA, 94720, johnston@george.lbl.gov (tel: 510-486-5014, fax: 510-486-6363). This work is supported by the Director, Office of Energy Research, Office of the Scientific Computing Staff, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098. Any opinions are solely those of the authors and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory, or the U.S. Department of Energy. Trademarks are acknowledged by †.

frequently involve separation of these components by their very nature (large, expensive, immobile, etc.).

Another problem in research imaging is that the algorithms for analysis, and therefore the optimal computing environment needed to implement them, may not be well understood. A configurable computing environment allows rapid and economical changes in systems during the learning phase.

A Case Study: The Lawrence Berkeley Laboratory (LBL) Information and Computing Sciences and Research Medicine Divisions have collaborated with the Pittsburgh Supercomputer Center (PSC) to demonstrate the possibilities for such a distributed environment. The prototype application is the interactive visualization of large 3D scalar fields (voxel data sets) by using a combination of heterogeneous supercomputers and low cost workstations for display and control.

This application is designed to test the limits of the recently improved network bandwidth and interprocess communications, and to identify bottlenecks that remain in the way of achieving real-time distributed visualization of large 3D data sets.

The computational part of the application is partitioned into two pieces, one optimal for a massively parallel architecture, and one optimal for a vector processor. The first part is run on a Thinking Machines† CM-2, and the second on a Cray†, Y-MP. These systems are located at PSC, and communicate with each other over a HIPPI, 800 Mbits/sec communications channel, while the remote workstations are connected to PSC via the usual variety of local, regional, and wide area networks (WAN) (e.g. NSFNet and DoE's ESNet).

The Application: The goal is the interactive display of large 3D scalar fields (e.g. a high-resolution MRI data set of the human brain). The data set used for the experiment is 256 x 256 x 128 x 1 byte voxels, or 8.4 MBytes of data. "Interactive" here is taken to mean the ability to generate and display images at a rate of at least 5 frames per second as a result of changing the isosurface, region of interest, or viewing parameters for the resulting geometry.

Application Architecture: The application consists of three inter-communicating processes. One process runs on a local workstation, and controls the other two processes via an X-window interface. The CM-2 process reads the voxel data and locates isosurfaces using the *dividing cubes* algorithm [2]. The resulting surface data is sent across a HIPPI channel to a process running on a Cray Y-MP, which does the 3D graphics rendering needed to convert this data into an image. The image is then sent across the network to the local workstation for display as part of the user interface.

Connection Machine Issues: *Dividing cubes* is a data-parallel algorithm and maps well to a SIMD architecture machine like the CM-2. Finding the normal vector for every data point is performed in parallel immediately after the data is loaded into memory. This computation takes only a few seconds, and only needs to be done once, since the CM-2 has sufficient memory to store all normals.

Determining the isosurface can also be done in parallel, and takes about 0.11 seconds. The results of these calculations are scattered across many processors, and must be gathered together before they can be sent to the Cray. This operation takes about 0.6 seconds. After the results are gathered, the data is converted to Cray format and sent to the Cray process over the HIPPI channel.

CM to Cray Communication Issues:

Because of the radical differences in machine architectures, data conversion between the CM-2 and the Y-MP is a non-trivial problem. On the Connection Machine, which uses many parallel bit processors, there is no straightforward relationship between the internal ordering of the data and the standard serial data ordering used on the Cray. Our application must perform a conversion to serial data form for each block of data transferred. This conversion takes approximately 0.06 seconds for a 1.8 MByte block. The CM-2 is connected to the HIPPI channel via the CM I/O bus, a direct parallel connection to the CM-2 supporting data

rates of up to 240 Mbits/sec. The CM-2 does not support Internet protocols, and to simplify the use of the HIPPI channel, PSC has written a library of routines to hide both the networking and data conversion issues from the user [3].

Cray Issues: The main Cray issue is utilizing the limited memory (32 MByte) available to a user at the site in an optimal manner, and vectorizing as many of the calculations as possible. Because there is not enough memory to store incoming x, y, z and normal vectors, the lighting calculations are performed on the CM-2 instead of the Cray. Using the normals to perform the lighting calculation on the CM-2, all that need be sent is a resulting 8-bit intensity value, a 6-fold reduction in data volume.

Using an 8-bit data type brings up a problem in that the Cray is a word-oriented rather than a byte-oriented architecture, and the use of character data types in a loop inhibits vectorization. To circumvent this problem, the data is read off the HIPPI channel in integer format, and the data for two points placed in one Cray word.

Since the lighting calculation has been performed on the CM, the Cray does the projection and hidden-surface removal part of the 3D graphics. The projection calculation vectorizes trivially. Vectorizing the hidden-surface removal is more difficult.

Local Workstation Issues: The main reason for performing the visualization on supercomputers instead of a local workstation is to demonstrate that scientists can use a configurable environment to assemble available computing elements in order to visualize large data sets, displaying the results on an inexpensive color workstation.

For this experiment to be successful it is necessary to send raster images between the graphics process and the image display device (the workstation) fast enough to achieve motion/rotation visual queuing (at least 5-10 frames / second). Using images generated at this rate, this experiment tests the new 45 Mbits/sec WAN's and 100 Mbits/sec (FDDI) LAN's and workstation interfaces.

Another issue arises from the display of images arriving at interactive rates. This demonstration uses an X-Windows server together with a standard toolkit to handle the user interface, and a proprietary direct graphics access library in order to display the images at a fast enough rate. (For displaying images the overhead of the standard X protocol causes a noticeable degradation in speed.)

Network Issues: There are several network issues that limit end-to-end performance in distributed systems. One of these issues is described here, and several others are discussed in [4]. TCP is used in all of our experiments because it is by far the best developed transport protocol. However, with traditional TCP implementations, there is a problem in that as network speeds increase, throughput becomes limited both by the speed-of-light propagation time between the communicating computers, and by peculiarities in the TCP implementation. The standard version of TCP/IP can send at most 64KB of data per round-trip-time and, in practice, the sustained throughput was at most half this theoretical maximum because of the packet acknowledgement scheme. One of us (VJ) has developed TCP/IP extensions for high performance [1], wide area transport. Working closely with Cray Research and Sun Microsystems these extensions were implemented and used for this experiment. These extensions remove the 64KB per round-trip-time limit and allow TCP/IP to run at the full speed of the underlying network, independent of the end-to-end propagation time. These modifications are essential to achieving fast image display over a wide area network.

Analysis: In this experiment about 500,000 3D points are typically generated to describe one isosurface. This process takes approximately 0.8 seconds using 16,384 processors on the CM-2 (time for surface generation plus time to gather results). It takes 0.06 seconds for the data conversion, 0.1 seconds for the HIPPI transfer, and 0.3 seconds for one Cray Y-MP processor to render the image. Over a 45 Mbits/sec cross-country network, it takes about 0.1 seconds to transfer the resulting $320 \times 320 \times 1$Byte (100 KByte) image to the local workstation. Therefore the total time is around 1.3 seconds.

Changing only the viewpoint on the geometric model representing the isosurface in the scalar

field is faster because the geometry does not need to be recomputed. The total time in this case is around 0.5 seconds. However, the application can be run in "movie" mode, where images are generated for a set of incremental rotations. In this case the CM - 2 and Y - MP are working in parallel, and can generate images of the rotating surface at a rate of about 3 frames per second.

The speed of rotation can be further increased by using a less general hidden - surface removal algorithm (see [4] for more information). Rendering in this case takes 0.03 seconds. Thus in movie mode, network bandwidth becomes the limiting factor, and the maximum speed of rotation is ten frames per second.

We are working on more general methods of distributing the application to ease its usage in a truly heterogeneous "on demand" environment. This work will add workstation clusters to the collection of computing elements that can be configured into virtual systems. Whether traditional supercomputers or workstation clusters are used will be transparent to the user. Our present work focuses on PVM (Parallel Virtual Machine) [5] as a model of handling interactions among heterogeneous supercomputing resources.

Conclusion: We have presented a collection of technologies that, taken together, will provide the possibility for: (1) routinely partitioning problems between heterogeneous supercomputers; (2) doing remote siting of data intensive scientific experiments; (3) providing access to capabilities that could previously only be obtained at a small number of sites due to the size, cost, or experimental nature of the implementation, and; (4) providing new capability enabled by the nature of the networks themselves.

The example application demonstrates that wide - area networks are not necessarily the bottleneck to widely distributed imaging applications. Widely deployed Gbits / sec wide - area networks and the associated interconnecting hardware and software promise to alter the way that many large scale problems are approached. These networks will allow the creation of "network" or "virtual" supercomputers - - computing systems comprised of geographically distributed components communicating with each other at high speeds, and configured on demand into virtual systems that exist only as long as necessary to solve a particular problem, or until a better combination of elements to solve the problem becomes apparent, at which point the virtual system is reconfigured.

Acknowledgements: The authors thank Van Jacobson of LBL for his ongoing support in providing access to his latest TCP / IP protocol implementations; Wendy Huntoon, Jamshid Mahdavi, and Ralph Roskies of PSC for their collaboration with the case study; and Peter Schroder, formerly of Thinking Machines Corporation, Carl Crawford of General Electric Company, and Mark Roos of the LBL Research Medicine Division, for their support and help with this project.

References:

[1] V. Jacobson, R.T. Braden, D.A. Borman, *TCP Extensions for High Performance,* Internet Requests for Comment (RFC) 1323, DDN Network Information Center, Menlo Park, CA., May, 1992.

[2] H.E. Cline, W.E. Lorensen, S. Ludke, C.R. Crawford, and B.C. Teeter, *Two Algorithms for the Three - Dimensional Reconstruction of Tomograms*, Medical Physics, May 1988.

[3] M. Schneider, *Pittsburgh's Not - So - Odd Couple*, Supercomputing Review, August, 1991.

[4] Johnston, W., V. Jacobson, D. Robertson, B. Tierney, S. Loken, *High Performance Computing, High Speed Networks, and Configurable Computing Environments* LBL Report, LBL - 32161, also in "High Performance Computing in Biomedical Research", CRC Press, November, 1992.

[5] A. Beguelin, J.J. Dongarra, G.A. Geist, R. Manchek, and V.S. Sunderam, *Graphical Development Tools for Network - Based Concurrent Supercomputing*, Supercomputing '91 Conference Proceedings, pp. 435-444.