Presenting Author:
Jason R. Lee
JRLee@lbl.gov
MS: 50B-2239
1 Cyclotron Rd.
Berkeley, CA. 94720 USA
510-486-7342

Brian L Tierney
BLTierney@lbl.gov
MS: 50B-2239
1 Cyclotron Rd.
Berkeley, CA. 94720 USA
510-486-7381

William E. Johnston
WEJohnston@lbl.gov
MS: 50B-2239
1 Cyclotron Rd.
Berkeley, CA. 94720 USA
510-486-5014

Data Intensive Distributed Computing: A Medical Application Example

Jason Lee, Brian Tierney, William Johnston
Lawrence Berkeley National Laboratory[1]
1 Cyclotron Rd.
MS: 50B-2239
Berkeley, CA 94720
{jrlee, bltierney, wejohnston}@lbl.gov

Abstract. Modern scientific computing involves organizing, moving, visualizing, and analyzing massive amounts of data from around the world, as well as employing large-scale computation. The distributed systems that solve large-scale problems will always involve aggregating and scheduling many resources. Data must be located and staged, cache and network capacity must be available at the same time as computing capacity, etc. Every aspect of such a system is dynamic: locating and scheduling resources, adapting running application systems to availability and congestion in the middleware and infrastructure, responding to human interaction, etc. The technologies, the middleware services, and the architectures that are used to build useful high-speed, wide area distributed systems, constitute the field of data intensive computing. This paper explores some of the history and future directions of that field, and describes a specific medical application example.

1.0 Introduction

The advent of shared, widely available, high-speed networks is providing the potential for new approaches to the collection, storage, and analysis of large data-objects. Two examples of large data-object environments, that despite the very different application areas have much in common, are health care imaging information systems and atomic particle accelerator detector data systems.

Health care information, especially high-volume image data used for diagnostic purposes - e.g. X-ray CT, MRI, and cardio-angiography - are increasingly collected at

1. The work described in this paper is supported by the U. S. Dept. of Energy, Office of Energy Research, Office of Computational and Technology Research, Mathematical, Information, and Computational Sciences and ERLTT Divisions under contract DE-AC03-76SF00098 with the University of California, and by DARPA, Information Technology Office. This report number LBNL-42690.

tertiary (centralized) facilities, and may now be routinely stored and used at locations other than the point of collection. The importance of distributed storage is that a hospital (or any other instrumentation environment) may not be the best environment in which to maintain a large-scale digital storage system, and an affordable, easily accessible, high-bandwidth network can provide location independence for such storage. The importance of remote end-user access is that the health care professionals at the referring facility (frequently remote from the tertiary imaging facility) will have ready access to not only the image analyst's reports, but the original image data itself.

This general scenario extends to other fields as well. In particular, the same basic infrastructure is required for remote access to large-scale scientific and analytical instruments, both for data handling and for direct, remote-user operation. See [8].

In this paper we describe and illustrate a set of concepts that are contributing to a generalized, high performance, distributed information infrastructure, especially as it concerns the types of large data-objects generated in the scientific and medical environments. We will describe the general issues, architecture, and some system components that are currently in use to support distributed large data-objects. We describe a health care information system that has been built, and is in prototype operation.

1.1 An Overall Model for Data-Intensive Computing

The concept of a high-speed distributed cache as a common element for all of the sources and sinks of data involved in high-performance data systems has proven very successful in several application areas, including the automated processing and cataloguing of real-time instrument data and the staging of data from an Massive Storage System (MSS) for high data-rate applications.

For the various data sources and sinks, the cache, which is itself a complex and widely distributed system, provides:

- a standardized approach for high data-rate interfaces;
- an “impedance” matching function (e.g., between the coarse-grained nature of parallel tape drives in the tertiary storage system and the fine-grained access of hundreds of applications);
- flexible management of on-line storage resources to support initial caching of data, processing, and interfacing to tertiary storage;
- a unit of high-speed, on-line storage that is large compared to the available disks of the computing environments, and very large (e.g., hundreds of gigabytes) compared to any single disk.

The model for data intensive computing, shown in Figure 1, includes the following:

- each application uses a standard high data-rate interface to a large, high-speed, application-oriented cache that provides semi-persistent, named datasets / objects;

- data sources deposit data in a distributed cache, and consumers take data from the cache, usually writing processed data back to the cache when the consumers are intermediate processing operations;
- metadata is typically recorded in a cataloguing system as data enters the cache, or after intermediate processing;
- a tertiary storage system manager typically migrates data to and from the cache. The cache can thus serve as a moving window on the object/dataset, since, depending on the size of the cache relative to the objects of interest, only part of the object data may be loaded in the cache - though the full objection definition is present: that is, the cache is a moving window for the off-line object/data set;

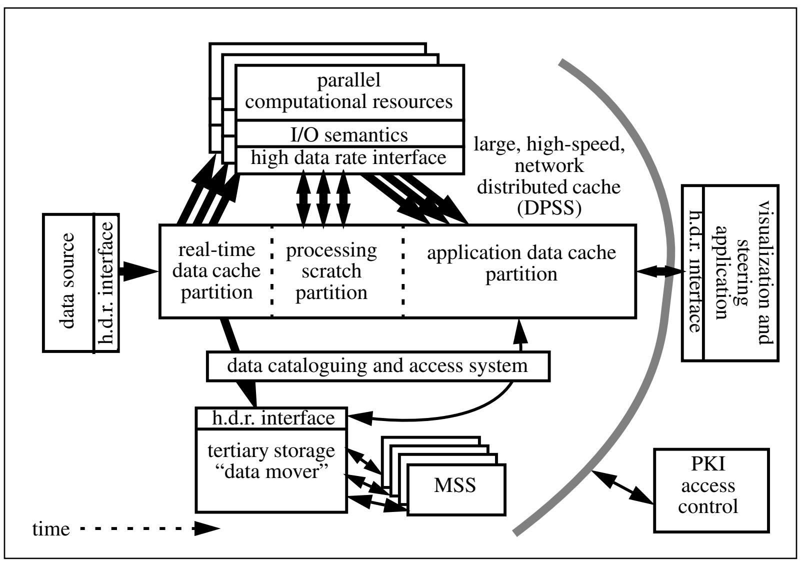

Figure 1 Model for Data-Intensive Computing

- the native cache access interface is at the logical block level, but client-side libraries implement various access I/O semantics - e.g., Unix I/O (upon request available data is returned; requests for data in the dataset, but not yet migrated to cache, cause the application-level read to block or be signaled);

1.2 The Distributed Parallel Storage Server

A key aspect of this data intensive computing environment has turned out to be a high-speed, distributed cache. LBNL designed and implemented the Distributed-Parallel Storage System (DPSS)[1] as part of the MAGIC [6] project, and as part of the U.S. Department of Energy's high-speed distributed computing program. This technology has been quite successful in providing an economical, high-performance, widely distributed, and highly scalable architecture for caching large amounts of data that can potentially be used by many different users. The DPSS serves several roles in high-performance, data-intensive computing environments. This application-oriented cache provides a standard high data rate interface for high-speed access by data sources, processing resources, mass storage systems, and user interface elements. It provides the functionality of a single very large, random access, block-oriented I/O device (i.e., a “virtual disk”) with very high capacity (we anticipate a terabyte sized system for high-energy physics data) and serves to isolate the application from tertiary storage systems and instrument data sources. Many large data sets may be logically present in the cache by virtue of the block index maps being loaded even if the data is not yet available. Data blocks are declustered (dispersed in such a way that as many system elements as possible can operate simultaneously to satisfy a given request) across both disks and servers. This strategy allows a large collection of disks to seek in parallel, and all servers to send the resulting data to the application in parallel (see Figure 2). In this way processing can begin as soon as the first data blocks are generated by an instrument or migrated from tertiary storage.

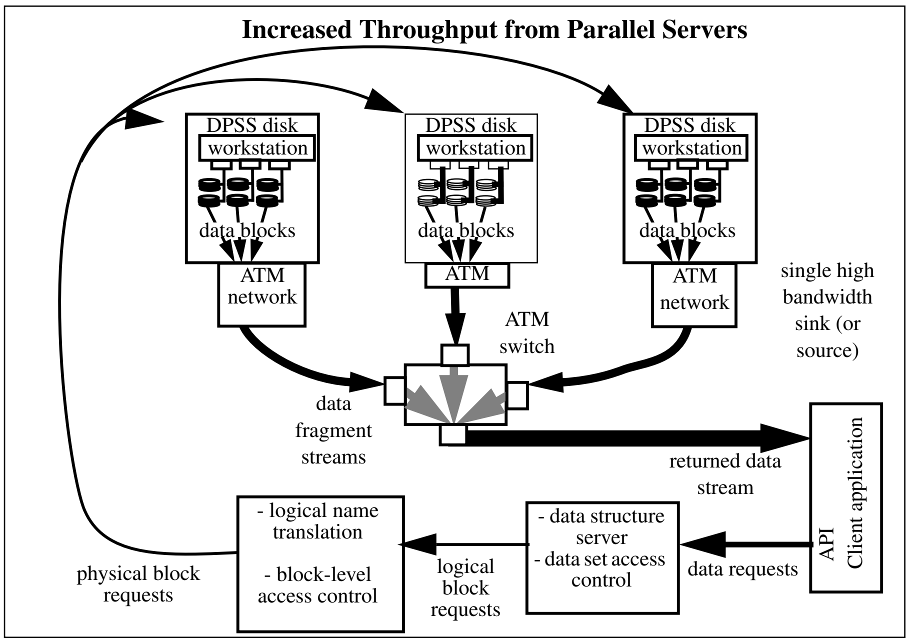

Figure 2 **DPSS Architecture**

The high performance of the DPSS - about 14 megabytes/sec of data delivered to the user application per disk server - is obtained through parallel operation of independent, network-based components. Flexible resource management - dynamically adding and deleting storage elements, partitioning the available storage, etc. - is provided by design, as are high availability and strongly bound security contexts. The scalable nature of the system is provided by many of the same design features that provide the flexible resource management (that in turn provides the capability to aggregate dispersed and independently owned storage resources into a single cache).

The DPSS provides several important and unique capabilities for a data intensive computing environment. It provides application-specific interfaces to an extremely large space of logical blocks; it offers the ability to build large, high-performance storage systems from inexpensive commodity components; and it offers the ability to increase performance by increasing the number of parallel disk servers. Various cache management policies operate on a per-data set basis to provide block aging and replacement.

2.0 A Medical Application Example

BAGNet was an IP over OC-3 (155 Mbit/sec) ATM, metropolitan area network testbed that operated in the San Francisco Bay Area (California) for two years starting in early 1994. The participants included government, academic, and industry computer science and telecommunications R&D groups from fifteen Bay Area organizations. The goal was to develop and deploy the infrastructure needed to support a diverse set of distributed applications in a large-scale, IP-over-ATM network environment.

In BAGNet, there were several specific projects involving subsets of the connected sites. In particular, LBNL, the Kaiser Permanente health care organization, and Philips Palo Alto Research Center collaborated to produce a prototype production, on-line, distributed, high data rate medical imaging system. (Philips and Kaiser were added to BAGNet for this project through the Pacific Bell CalREN program.)

The Kaiser project [8] focused on using high data rate, on-line instrument systems as remote data sources.

When data is generated in large volumes and with high throughput, and especially in a distributed environment where the people generating the data are geographically separated from the people cataloguing or using the data, there are several important considerations for managing instrument generated data:

- automatic generation of at least minimal metadata;
- automatic cataloguing of the data and the metadata as the data is received (or as close to real time as possible);
- transparent management of tertiary storage systems where the original data is archived;
- facilitation of cooperative research by providing specified users at local and remote sites immediate as well as long-term access to the data;
- mechanisms to incorporate the data into other databases or documents.

The WALDO (Wide-area Large-Data-Object) system was developed to provide these capabilities, especially when the data is gathered in real time from a high data rate instrument [8]. WALDO is a digital data archive that is optimized to handle real-time data. It federates textual and URL linked metadata to represent the characteristics of large data sets. Automatic cataloguing of incoming real-time data is accomplished by extracting associated metadata and converting it into text records; by generating auxiliary metadata and derived data; and by combining these into Web-based objects that include persistent references to the original data components (called large data objects, or LDOs) [9]. Tertiary storage management for the data components (i.e., the original datasets) is accomplished by using the remote program execution capability of Web servers to manage the data on mass storage systems. For subsequent use, the data components may be staged to a local disk and then returned as usual via the Web browser, or, as is the case for several of our applications, moved to a high-speed cache for access by specialized applications (e.g., the high-speed video player illustrated in the right-hand part of the right-hand panel in Figure 3). The location of the data components on tertiary storage, how to access them, and other descriptive material are all part of the LDO definition. The creation of object definitions, the inclusion of "standardized" derived-data-objects as part of the metadata, and the use of typed links in the object definition, are intended to provide a general framework for dealing with many different types of data, including, for example, abstract instrument data and multi-component multimedia programs. WALDO was used in the Kaiser project to build a medical application that automatically manages the collection, storage, cataloguing, and playback of video-angiography data[2] collected at a hospital remote from the referring physician.

Using a shared, metropolitan area ATM network and a high-speed distributed data handling system, video sequences are collected from the video-angiography imaging system, then processed, catalogued, stored, and made available to remote users. This permits the data to be made available in near-real time to remote clinics (see Figure 3). The LDO becomes available as soon as the catalogue entry is generated — derived data is added as the processing required to produce it completes. Whether the storage systems are local or distributed around the network is entirely a function of optimizing logistics.

In the Kaiser project, cardio-angiography data was collected directly from a Philips scanner by a computer system in the San Francisco Kaiser hospital Cardiac Catheterization Laboratory. This system is, in turn, attached to an ATM network provided by the NTON and BAGNet testbeds. When the data collection for a patient is complete (about once every 20–40 minutes), 500–1000 megabytes of digital video data is sent across the ATM network to LBNL (in Berkeley) and stored first on the DPSS distributed cache (described above), and then the WALDO object definitions are generated and made available to physicians in other Kaiser hospitals via BAGNet. Auxiliary processing and archiving to one or more mass storage systems proceeds independently. This process goes on 8–10 hours a day.

---

2. Cardio-angiography imaging involves a two plane, X-ray video imaging system that produces from several to tens of minutes of digital video sequences for each patient study for each patient session. The digital video is organized as tens of data-objects, each of which are of the order of 100 megabytes.

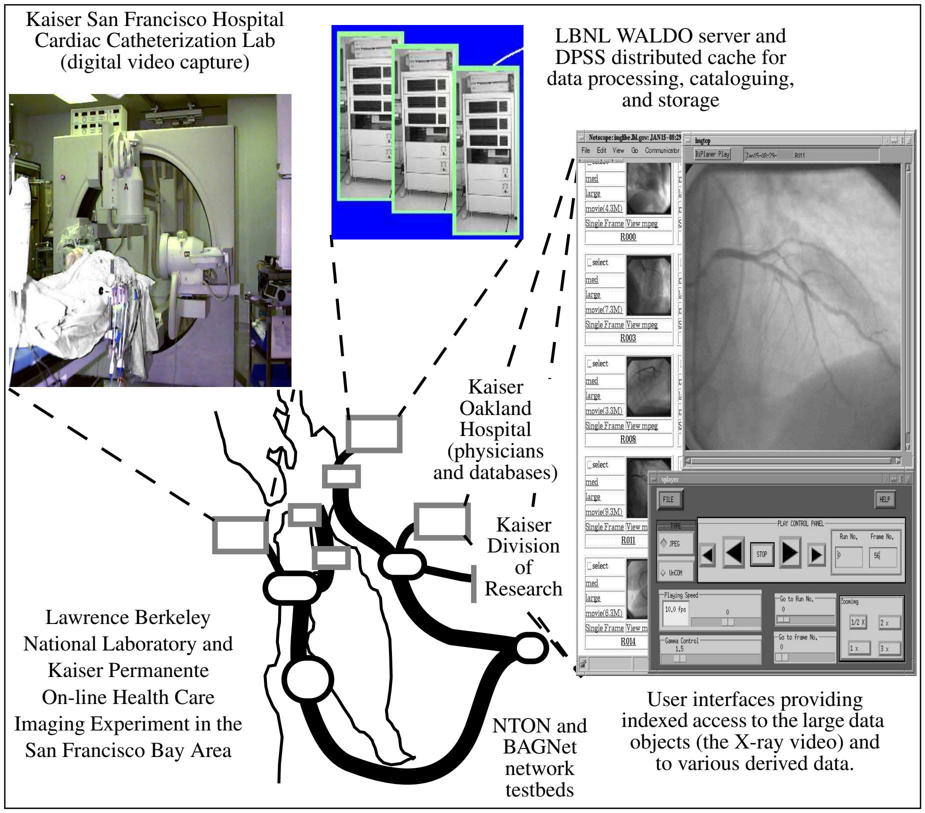

Figure 3 Capture of Digitized Angiograms

3.0 Distributed System Performance Monitoring

A central issue for using high-speed networks and widely distributed systems as the foundation of a large data-object management strategy is the performance of the system components, the transport / OS software, and the underlying network. Problems in any of these regimes will hurt a data intensive computing strategy, but such problems can usually be corrected if they can be isolated and characterized. A significant part of our work with high-speed distributed systems in MAGIC has been developing a monitoring methodology and tools to locate and characterize bottlenecks.

There are virtually no behavioral aspects of high-speed, wide area IP-over-ATM networks that can be taken for granted, even in end-to-end ATM networks. By “network” we mean the end-to-end data path from the transport API through the host network protocol (TCP/IP) software, the host network adaptors and their device drivers, the many different kinds of ATM switches and physical links, up through the corresponding software stack on the receiver. Further, the behavior of different elements at similar places in the network architecture can be quite different because they are implemented in different ways. The combination of these aspects can lead to complex and unpredictable network behavior.

We have built performance and operation monitoring into the storage system and several applications, and have designed tools and methodologies to characterize the distributed operation of the system at many levels. As requests and data enter and leave all parts of the user-level system, synchronized timestamps are logged using a common logging format. At the same time, various operating system and network parameters may be logged in the same format. This is accomplished by the Netlogger monitoring system [10], which has been used to analyze several network-generated problems that showed up in the distributed applications [11].

4.0 Other DPSS Applications

We have conducted a set of high-speed, network based, data intensive computing experiments between Lawrence Berkeley National Laboratory (LBNL) in Berkeley, Calif., and the Stanford Linear Accelerator (SLAC) in Palo Alto, Calif. The results of this experiment were that a sustained 57 megabytes/sec of data were delivered from datasets in the distributed cache to the remote application memory, ready for analysis algorithms to commence operation. This experiment represents an example of our data intensive computing model in operation.

The prototype application was the STAR analysis system that analyzes data from high energy physics experiments. (See [3].) A four-server DPSS located at LBNL was used as a prototype front end for a high-speed mass storage system. A 4-CPU Sun E-4000 located at SLAC was a prototype for a physics data analysis computing cluster, as shown in Figure 2. The National Transparent Optical Network testbed (NTON - see [7]) connects LBNL and SLAC and provided a five-switch, 100-km, OC-12 ATM path. All experiments were application-to-application, using TCP transport.

Multiple instances of the STAR analysis code read data from the DPSS at LBNL and moved that data into the memory of the STAF application where it was available to the analysis algorithms. This experiment resulted in a sustained data transfer rate of 57 MBytes/sec from DPSS cache to application memory. This is the equivalent of about 4.5 TeraBytes / day. The goal of the experiment was to demonstrate that high-speed mass storage systems could use distributed caches to make data available to the systems running the analysis codes. The experiment was successful, and the next steps will involve completing the mechanisms for optimizing the MSS staging patterns and completing the DPSS interface to the bit file movers that interface to the MSS tape drives.

5.0 Conclusions

We believe this architecture, and its integration with systems like Globus ([5], [2]), will enable the next generation of configurable, distributed, high-performance, data-intensive systems; computational steering; and integrated instrument and computational simulation. We also believe a high performance network cache system such as the DPSS will be an important component to these “computational grids” [5] and “metasystems”[4].

6.0 References

[1] DPSS, “The Distributed Parallel Storage System,” http://www-didc.lbl.gov/DPSS/

[2] Globus, “The Globus Project,” http://www.globus.org/

[3] Greiman, W., W. E. Johnston, C. McParland, D. Olson, B. Tierney, C. Tull, “High-Speed Distributed Data Handling for HENP,” Computing in High Energy Physics, April, 1997. Berlin, Germany. http://www-itg.lbl.gov/STAR/

[4] Grimshaw, A., A. Ferrari, G. Lindahl, K. Holcomb, “Metasystems”, Communications of the ACM, November, 1998, Volume 41, no 11

[5] Foster, I., C. Kesselman, eds., “The Grid: Blueprint for a New Computing Infrastructure,” Morgan Kaufmann, publisher. August, 1998.

[6] B. Fuller and I. Richer “The MAGIC Project: From Vision to Reality,” IEEE Network, May, 1996, Vol. 10, no. 3. http://www.magic.net/

[7] NTON, “National Transparent Optical Network Consortium.” See http://www.ntonc.org/.

[8] Johnston, W., G. Jin, C. Larsen, J. Lee, G. Hoo, M. Thompson, B. Tierney, J. Terdiman, “Real-Time Generation and Cataloguing of Large Data-Objects in Widely Distributed Environments,” International Journal of Digital Libraries - Special Issue on “Digital Libraries in Medicine”. November, 1997. (Available at http://www-itg.lbl.gov/WALDO/)

[9] Thompson, M., W. Johnston, J. Guojun, J. Lee, B. Tierney, and J. F. Terdiman, “Distributed health care imaging information systems,” PACS Design and Evaluation: Engineering and Clinical Issues, SPIE Medical Imaging 1997. (Available at http://www-itg.lbl.gov/Kaiser.IMG)

[10] Tierney, B., W. Johnston, B. Crowley, G. Hoo, C. Brooks, D. Gunter, “The NetLogger Methodology for High Performance Distributed Systems Performance Analysis,” Seventh IEEE International Symposium on High Performance Distributed Computing, Chicago, Ill., July 28-31, 1998. Available at http://www-itg.lbl.gov/DPSS/papers.html.

[11] Tierney, B., W. Johnston, J. Lee, and G. Hoo, “Performance Analysis in High-Speed Wide Area ATM Networks: Top-to-bottom end-to-end Monitoring,” IEEE Networking, May 1996. (Available at http://www-itg.lbl.gov/DPSS/papers.)