Directions and Issues for High Data Rate Wide Area Network Environments

William Johnston, Jason Lee, Brian Tierney, and Craig Tull, LBNL
Dave Millsom, SLAC

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. In the past six months, as part of work intended to provide remote high speed access to very large scale tertiary storage systems, we have conducted a set of high-speed, network based, data intensive computing experiments between Lawrence Berkeley National Lab (LBNL) and the Stanford Linear Accelerator (SLAC) facility. These experiments demonstrated the feasibility of very high bandwidth, application-to-application data communication: a sustained 57 megabytes/second of data were communicated from a storage system cache at LBNL to an application running on a computer at SLAC.

In principle, this capability could provide remote access to storage systems at rates equal to, or better than, local access.

However, the distributed systems - including access to tertiary storage systems - that solve large-scale problems will always involve a collection of supporting services. Data must be located and staged, cache and network capacity must be available at the same time as computing capacity, etc. Distributed services provide these capabilities, and account for the fact that every aspect of such wide area systems are 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, which has evolved over the past ten years to the point where it represents a potentially significant scientific resource - as demonstrated by the LBNL - SLAC high data rate experiments.

However, much work remains to be done in order to provide the environment and services to enable routine use of these high-speed environments. This paper explores some of the history, current state, and future directions of wide area, high data rate distributed systems.

1. Introduction

As a precursor to routine remote high-speed access to large-scale mass storage systems (MSS), a recent set of experiments were conducted between Lawrence Berkeley National Laboratory (LBNL) in Berkeley, Calif., and the Stanford Linear Accelerator (SLAC) in Palo Alto, Calif. The National Transparent Optical Network testbed (NTON - see [7]) provides eight 2.4 gigabit/sec data channels around the San Francisco Bay, of which four are usually used for OC-48 SONET. For this experiment, the network configuration involved four to six ATM switches and a Sun Enterprise-4000 SMP as a data receiver at SLAC, all with OC-12 (622 Mbit/sec) network interfaces, and four smaller systems at LBNL configured as distributed caches and serving as data

sources. 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 fairly impressive experiment is the result of a ten-year evolution of computing and networking technology, involving advances in platform and interface technologies, monitoring and management approaches, and parallel distributed software architectures and algorithms.

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 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

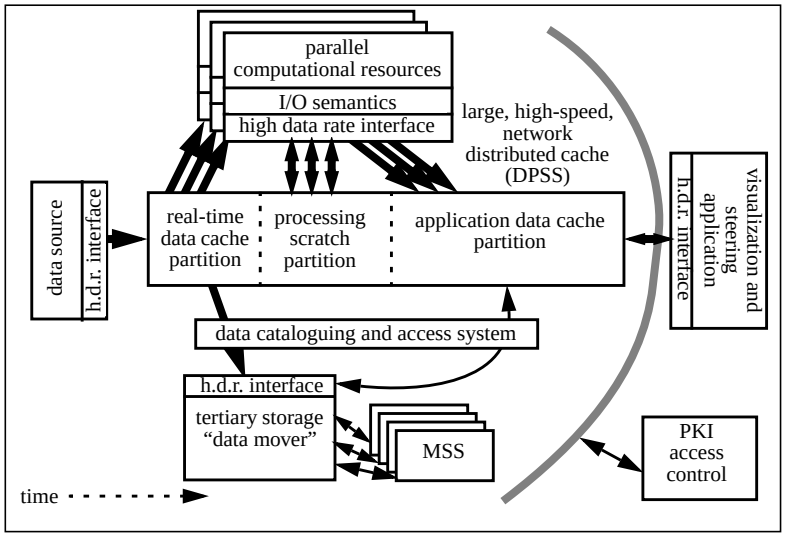


Figure 1 Model for Data-Intensive Computing

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;

- 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);

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 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. In this way processing can begin as soon as the first data blocks are generated by an instrument or migrated from tertiary storage.

The DPSS provides several important and unique capabilities for HENP 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.

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 LBNL-SLAC-NTON High Data-Rate Experiments

We conducted a series of experiments in high-speed, wide area distributed data processing that represent 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 [4].)

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 NTON network testbed that connects LBNL and SLAC provided a five-switch, 100-km, OC-12 path (and could be configured as a 2000 km, OC-12, path). All experiments were application-to-application, using TCP transport.

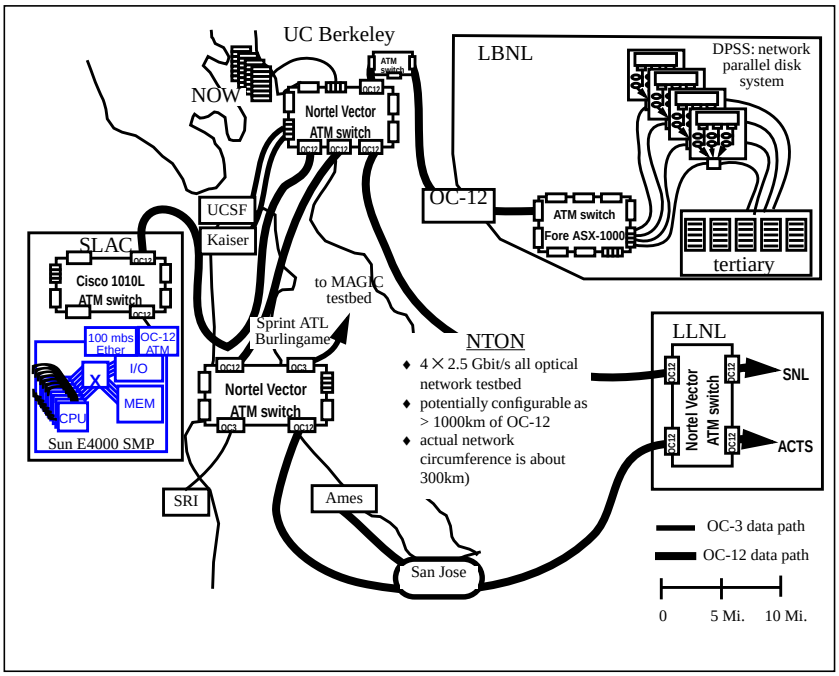


Figure 2 High-Speed Distributed Processing Experiment

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.

In addition to the advances in architecture and software, the success of this experiment is due to the “third generation” platform architectures like the Sun UltraSPARC (e.g. the Enterprise 4000) that provide gigabyte/sec memory bandwidth and OC-12 ATM interfaces that actually pass data through at the full rate.

2. China Clipper: What Comes Next?

The China Clipper project[1], a collaboration between LBNL, SLAC, and Argonne National Lab that builds on the infrastructure of the Energy Sciences network[], has as its high level goals designing and implementing a collection of independent but architecturally consistent service components. This is intended to enhance the ability of a variety of applications and systems to construct and use distributed, high-performance infrastructure. Such middleware will support high-speed access to, and integrated views of, multiple data archives; resource discovery and

1. Like Pan American Airway’s historic China Clipper that made the first trans-Pacific airmail flights from San Francisco to Honolulu and Manila, and its companion “flying boats” at the beginning of large-scale airline service, the Clipper Project anticipates the future in both flexibility and performance.

automated brokering; comprehensive real-time monitoring and performance trend analysis of the networked subsystems, including the storage, computing, and middleware components; and flexible and distributed management of access control and policy enforcement for multi-administrative domain resources.

Adaptability is an important aspect of distributed environments. Critical subsystems and components (e.g., network caches) must be capable of dynamic reconfiguration in a manner that is transparent to the application. Applications will also have to make use of performance trend information from the distributed components, and dynamically optimize their behavior.

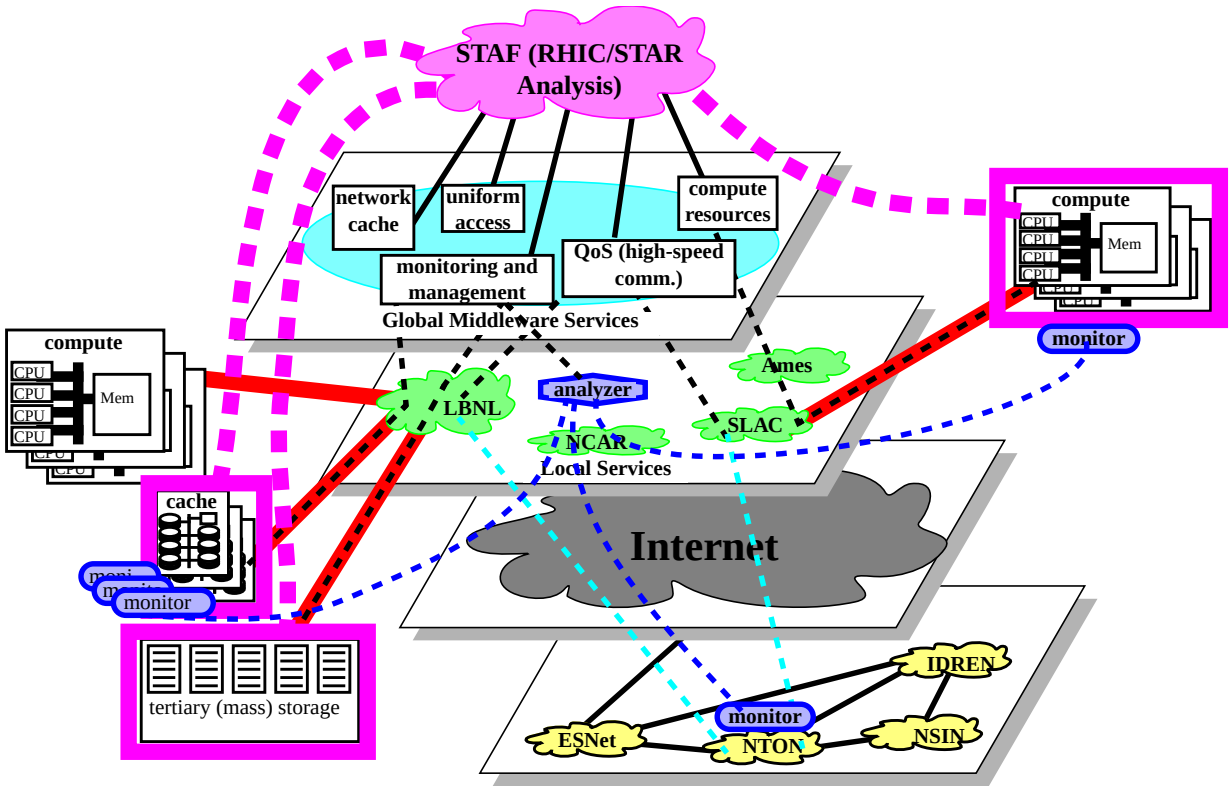
The challenge addressed by the Clipper project is how to accelerate routine use of applications that:

- require substantial computing resources
- generate and/or consume high rate and high volume data flows
- involve human interaction
- require aggregating many dispersed resources to establish an operating environment:
  - multiple data archives
  - distributed computing capacity
  - distributed cache capacity
  - “guaranteed” network capacity
- operate in widely dispersed environments.

Our general approach to addressing this challenge involves a combination of architecture, network functionality, middleware, and their integration into prototype applications. This project will develop network and middleware capabilities and prototype applications that provide and demonstrate environments for routine creation of robust, high data rate, secure distributed systems. Project objectives include:

- high-speed network connectivity that offers schedulable quality of service by effectively providing some level of bandwidth reservation among the elements of distributed systems;
- data management architectures that provide federated views of and high-performance “external” access to multiple archival mass storage systems;
- distributed, high-speed caches that provide applications with very high-performance access to data that is collected from on-line scientific instruments, staged from mass storage systems, or is in various stages of analysis, regardless of the physical location of the data or computing elements;
- resource monitoring to support problem diagnosis and infrastructure performance experiments, and to provide performance trend indicators to adaptive applications;
- active management of distributed elements to provide fault-tolerant operation of distributed system components and software;
- a security infrastructure that enforces the use and scheduling agreements among the services that are required by an application, and that also provides strong access control;

Clipper is envisioned not so much as a “system” but rather as a coordinated collection of services that may be flexibly employed by a variety of applications (or other middleware) to build on-demand, large-scale, high-performance, wide area, problem-solving environments.



The Clipper project and its integration with, and support of, systems like Globus ([5], [3]), WALDO[8], DPSS[10], Netlogger[9], and SRB [6], will enable the next generation of configurable, distributed, high-performance, data-intensive systems; computational steering; and integrated instrument and computational simulation.

3. References

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

[2] ESNet, “The Energy Sciences Network,” www.es.net. (“ESnet provides global networking for the DOE research and development mission. We are a leader in internet design and innovation providing a major piece of the U.S. Internet backbone.”)

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

[4] 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. (Available at http://www-itg.lbl.gov/STAR)

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

[6] Moore, R., et al, “Massive Data Analysis Systems,” San Diego Supercomputer Center. See http://www.sdsc.edu/MDAS

[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] 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.

[10] 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.)