Demonstrations of a Large Data Object Architecture using Medical Data at Supercomputing '96

Brian L. Tierney (bltierney@lbl.gov), Mary R. Thompson (mrthompson@lbl.gov)
William E. Johnston (wejohnston@lbl.gov), Jason Lee, Gary Hoo
Imaging and Distributed Computing Group
Lawrence Berkeley National Laboratory, Berkeley, CA 94720
http://www-itg.lbl.gov/DPSS

1.0 Introduction

We have developed a highly distributed architecture for collecting, storing, analyzing, searching and accessing, large data-objects. These objects - typically the result of a single operational cycle of an instrument, and of sizes from tens of MBytes to tens of Gbytes - are the staple of modern analytical systems. Examples include data from high energy particle accelerators and detector systems, images and related data from large electron microscopes, high brilliance X-ray sources and various medical imaging devices.

At Supercomputing `96 we will demonstrate a health care imaging application that uses the Distributed-Parallel Storage System (DPSS) and an image cataloguing system (ImgLib) to facilitate collection, storage, analysis, and delivery of medical x-ray fluoroscopy video. In this sample application, DPSS technology is used to collect and play back angiography video sequences and the ImgLib is used to automatically catalogue the images and to create compressed versions of the video.

2.0 DPSS Architecture

The Distributed-Parallel Storage System (DPSS) is a scalable, high-performance, distributed-parallel data storage system developed in the ARPA-funded MAGIC Testbed. A collection of wide area distributed disk servers operate in parallel to provide logical block level access to large data sets. Architecturally the DPSS is a network striped disk array, but is unusual in that its implementation allows applications complete freedom to determine optimal data layout, replication and/or coding redundancy strategy, security policy, and dynamic reconfiguration. The system is implemented by user level software that runs on UNIX workstations. Both the architecture and the implementation are intended to provide for easy and economical scalability. This approach has yielded a data source that scales economically to very high speed, and supports system research on networks and distributed computing.

This work is supported by the DARPA Information Technology Office (http://www.ito.darpa.mil/ResearchAreas.html) and Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences (MICS) Division, of the U. S. Department of Energy under Contract No. DE-AC03-76SF00098

The DPSS design is based on the use of multiple low-cost, medium-speed disk servers which use the network to aggregate server data streams. To achieve high performance we exploit many levels of parallelism, including that available at the level of the disks, controllers, processors / memory banks, servers, and the network (See Figure *1: DPSS Architecture*). Proper data placement strategy is also key to exploiting system parallelism.

The Distributed Parallel Storage Server (DPSS) is essentially a "block" server that is distributed across a wide area network to supply data to applications located anywhere in the network. (See Figure *2: Parallel Data and Server Architecture Approach to the Distributed-Parallel Server System*). There is no inherent organization to the blocks, and in particular, they would never be organized sequentially on a single disk. The data organization is determined by the application as a function of data type and access patterns, and is implemented during the data load process. The usual goal of the data organization is that data is 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, enabling the DPSS to perform as a high-speed image server.

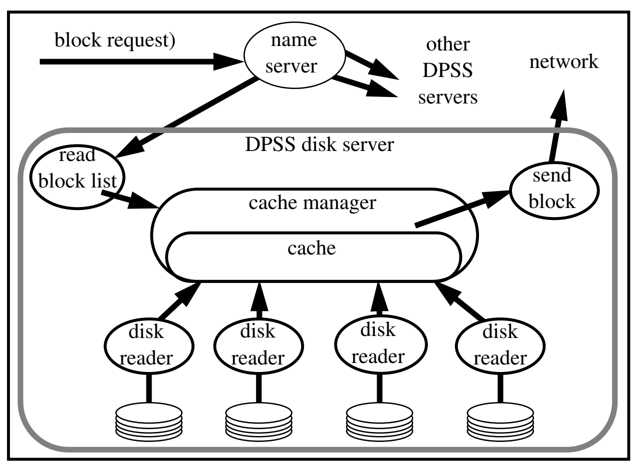

Figure 1: DPSS Architecture

3.0 Health Care Imaging Application

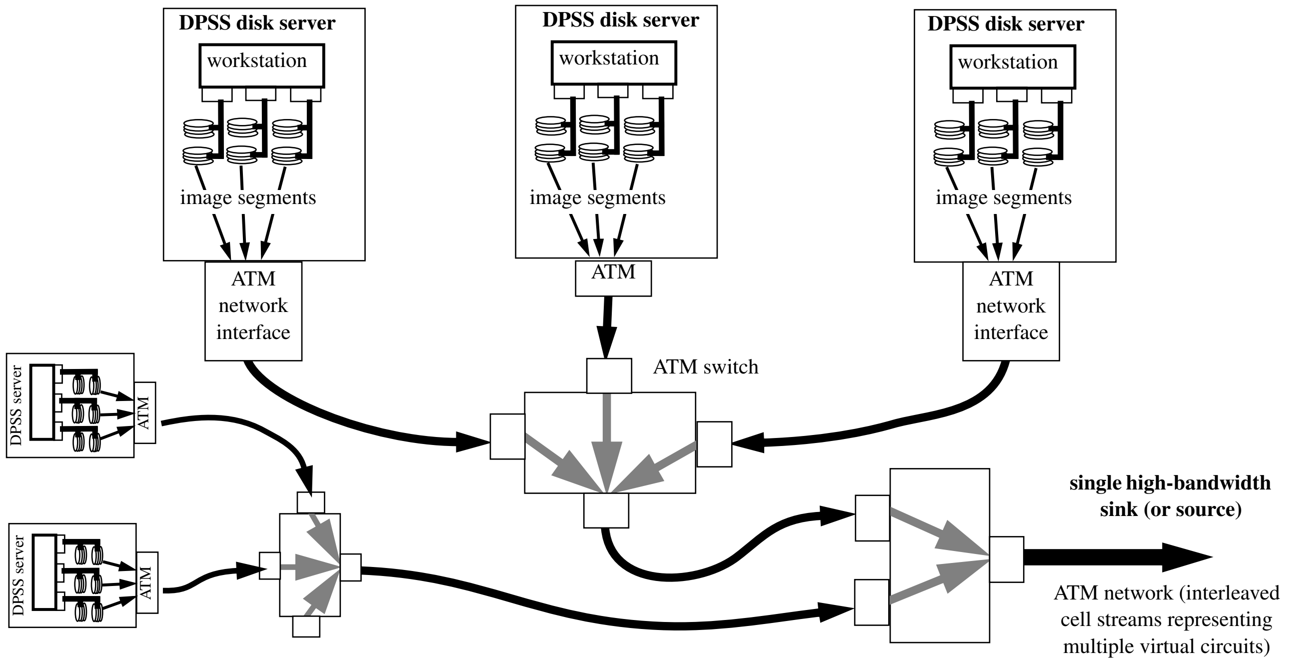

Figure 2: Parallel Data and Server Architecture Approach to the Distributed-Parallel Server System

In this sample application, DPSS technology is used to collect and play back angiography video sequences. Procedures used to restore coronary blood flow, though clinically effective, are expensive. To reduce the cost of such procedures, medical care providers are beginning to concentrate these services in a few high-volume tertiary care centers. Patients are usually referred to these centers by cardiologists at their primary care facilities; the centers then must communicate the results back to the local cardiologists as soon as possible after the procedure.

The advantages of providing specialized services at distant tertiary centers are significantly reduced if the medical information obtained during the procedure is not delivered rapidly and accurately to the treating physician in the patient's home facility. The delivery systems currently used to transfer patient information between facilities include interoffice mail, U.S. Mail, fax machine, telephone, and courier.

Using an ATM network and a high-speed image file server, video sequences can be collected from the angiography imaging systems. These sequences are sent through an ATM network to storage (DPSS) and analysis (ImgLib) systems, as well as directly to the clinic sites. Thus, data can be collected and stored for later use, data can be delivered live from the imaging device to remote clinics in near real-time, or these data flows can all be done simultaneously. Whether the DPSS servers are local or distributed around the network is entirely a function of the optimal logistics. There are arguments in regional healthcare information systems for centralized storage facilities, even though the architecture is that of a distributed system. (See Figure *3: DPSS-based Medical Imaging Application*).

3.1 Network Bandwidth Requirements

Cine-angiography network data storage is an example of an application constrained by the use of a public data network. Data is collected and stored locally during a short recording period. It is subsequently transmitted to network-based storage (the DPSS) for retrieval and review at other locations.

In typical angiography laboratory use, 1.5 gigabytes of data is collected in about four minutes, then there are about fifteen minutes in which to off-load the local storage before the next collection period. 1.5 GB in 15 minutes gives about 14 Mbits/sec of user-level data on the network ($14 \times 1.4 \approx 19$ Mbits/sec at the ATM level). 19 Mbits/sec is about 13% of an OC-3 network, and our *initial* estimate of the maximum reasonable use of a shared network. (The network is Pacific Bell’s SF Bay Area ATM network - an ATM OC-3 network which has 30-40 attached sites.)

4.0 The LBL Image Library (ImgLib)

In any scenario where data is generated in large volumes and with high throughput, a key issue is the automatic generation of at least minimal metadata, and the cataloguing of the data and the metadata as the data is received. Our approach to this problem (called “ImgLib”) is to use the World Wide Web tools to provide a preliminary cataloguing of incoming data. This is done by extracting associated metadata and converting it into text records, generating auxiliary metadata and derived data, and combining these into Web documents. An ImgLib

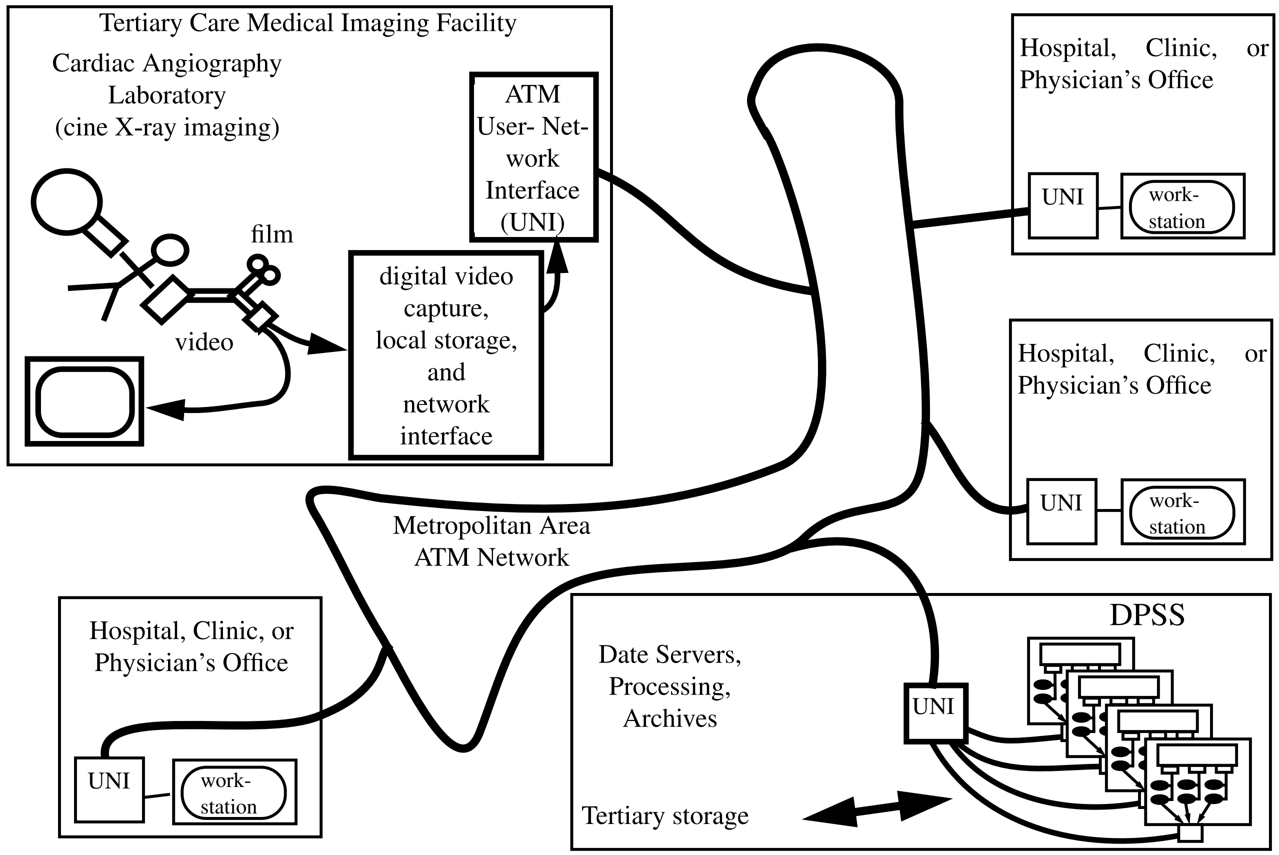

Figure 3: DPSS-based Medical Imaging Application

catalogue "entry" consists of (derived) thumbnail images and textual indexing material, together with pointers to the original images, video, or other data-objects (whose permanent storage is typically off-line). The ImgLib server also directly manages a DPSS-based on-line cache, and manages migration of data from tertiary storage to cache.

5.0 For more information

There is much more information on this project on the WWW. In particularly, see:

http://www-itg.lbl.gov/Kaiser

http://www-itg.lbl.gov/DPSS/

http://imglib.lbl.gov/

http://www.magic.net/