Distributed large data-object management architecture

William Johnston, Jin Guojun, Jason Lee, Mary Thompson, Brian Tierney

Imaging and Distributed Computing Group,
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory, Berkeley CA 94720

ABSTRACT

We are exploring the use of highly distributed computing and storage architectures to provide all aspects of collecting, storing, analyzing, and accessing large data-objects. These data-objects can be anywhere from tens of MBytes to tens of GBytes in size. Examples are: single large images from electron microscopes, video images such as cardio-angiography, sets of related images such as MRI data and images and numerical data such as the output from a particle accelerator experiment.The sources of such data objects are often remote from the users of the data and from available large-scale storage and computation systems. Our Large Data-object Management system provides network interface between the object sources, the data management system and the user of the data. As the data is being stored, a cataloguing system automatically creates and stores condensed versions of the data, textual metadata and pointers to the original data. The catalogue system provides a Web based graphical interface to the data. The user is able the view the low-resolution data with a standard internet connection and Web browser, or if high-resolution is required can use a high-speed connection and special application programs to view the high-resolution original data.

Keywords: large data-objects, distributed architectures, digital images, Web-based libraries

Introduction

We are evolving a strategy for using high-speed wide area networks and high capacity, high performance network storage systems for the management of data from on-line instruments and imaging systems. The high-level goal is to dramatically increase our ability to organize, search, and provide high performance and location independent access to "large data-objects" (LDO). These objects - typically the result of a single operational cycle of a scientific instrument (or supercomputer run), and of sizes from tens of MBytes to tens of GBytes - are the staple of modern analytical systems. It is further the case that many of the instrumentation systems that generate such data-objects are used by a diverse and geographically distributed community - examples from the scientific community include physics and nuclear science, high energy particle accelerators and detector systems; large electron microscopes; ultra-high brilliance X-ray sources, etc., and there are similarly complex imaging and instrumentation systems in the health care community. Such dispersed user communities require location independent access to this data.

In any scenario where data is generated in large volumes, and especially in a distributed environment where the people generating the data are geographically separated from the people cataloguing and using the data, there are several important issues:

- automatic generation of at least minimal metadata;
- cataloguing of the data and the metadata as the data is received (or as close to real time as possible);
- transparent management of cache and tertiary storage systems;
- facilitation of co-operative research by allowing specified users at local and remote sites immediate access to the data;
- incorporation of the data into other databases or documents.

We have several examples of such a system currently in use at Lawrence Berkeley National Laboratory. This paper describes the architecture of an LDO system which contains about 1000 hi-resolution scanning-electron microscope images and about 500 historical photos with related text information. This system is a simplified version of the general case in that the original data is loaded statically to tertiary storage and does not use the large high-speed data cache. The cataloguing, search functions and tertiary storage management of this system follow the LDO model.

Distributed Large Data-Object Management

Our model for large data-objects includes flexibility of definition, location independent access, high performance access, and persistence in naming, location of reference, and in time. These characteristics are provided through an architecture that combines the addressing, searching, association, and "method" based access features of the Web, together with a high speed network cache system for receiving high-speed data streams, and making large data sets readily available.

The general structure of one of these "objects" includes persistent pointers (URLs), a generalized object definition, and search mechanisms. The object definition is manifest as a Web document that contains pointers to metadata, data component locations, access methods, derived information and representations, and search methods.

The basic elements of our distributed, large data-object architecture include:

- data collection and the instrument-network interface
- on-line cache storage that is distributed throughout the network and transparently managed by the system
- near-line tertiary storage for high volume archives, transparently managed by the system
- processing elements - also distributed throughout the network - for various sorts of data analysis and derived data generation
- data management that provides for the automatic cataloguing and metadata generation that produces the large data-object
- data access interfaces, including application-oriented interfaces
- user access to all relevant aspects of the data (application, data, metadata, data management)
- flexible mechanisms for providing various searching strategies
- transparent security that provides access control for the data and all of the system components based on the resource / data - owner's policies

These elements all need to be provided with flexible, location-independent interfaces so that they can be freely (transparently) moved around the network as required for operational or other logistical convenience.

The steps in entering an object into a large-data object library are:

- The data-objects are entered in the high-speed cache storage and subsequently archived in a tertiary storage system.
- Metadata is generated by analyzing the object, encoding information forwarded by the object generator, or by associating other information with the object. Typically there is a "tagged-field" text file for description, small "thumbnail" images for image data, and compressed versions of video data.
- All of the information related to the data-object is combined into (or otherwise made available as) a Web document-based description that defines the large data-object: this includes an index of all related data and metadata for the data-object.

At this point the original dataset has a comprehensive "description" (object definition), a permanent instance in tertiary storage, and (perhaps) a temporary instance in the network cache.

The user interface to the data-objects consists of:

- A Web interface for searching, browsing, and accessing the metadata, the derived data components, and the primary data component.
- A web interface for adding to or modifying the metadata.
- A Web interface for managing the migration of the data-object into and out of the network cache (or any other cache), as needed by the applications.
- Specific applications that access the primary data component can be launched directly from the Web interface, or can use the Web interface to migrate these components to cache, and access them there.

The Web aspects do not imply strictly interactive, human actions, though this is clearly possible. Access to any of the object components can be directly from a program, as is typically the case during the initial, automatic acquisition, storage, and indexing of the data to create the LDO. A remote client can access the objects via the standard Web httpd protocol.

Together, these elements of a large data-object management architecture have provided effective management for several classes of data-objects. (See [1] and [2].)

Image Library Example

The original application that inspired our LDO architecture is an image library system (ImgLib)[3] that currently contains two separately-curated collections of images. One collection consists of about 1000 grey-scale microscopy images ranging in size from 2Mbytes to 4Mbytes. The other collection consists of about 500 historical photos from the Lawrence Berkeley National Laboratory photo archives. These are both grey-scale and color photos and range in size from 10-50 Mbytes.

In both of these collections the original large images are entered by their owners into the Lab Mass Storage System (MSS), which provides tens of GBytes of disk cache, backed by terrabytes of tape storage managed by tape robots. This system runs IEEE standard Unitree software which provides several kinds of access to files, e.g. FTP and NFS. In a heavily used Mass Storage System such as the one at the Lab, files are typically on tape. It takes about 10-20 minutes to get a file reloaded from an MSS tape to its disk. Then the user can copy the file to his own machine or reference it directly from the MSS disk. Thus referencing large images directly from the MSS tends to be very slow. The ImgLib was originally designed to provide a fast way that users could look though large numbers of such images.

The two ImgLib collections are each divided into smaller subcollections to allow quicker and more focused browsing and to help the collection curator organize the large number of images into more manageable groups. See Figure 1 for an example of the top level collection index. Various parameters such as quality and size of the thumbnails, text fields and default values for these fields and locations of the original images, may be set for each subcollection. A subcollection of objects could represent a compound object where several objects share some common information and most usually need to be accessed as a group

Two basic interfaces are provided: a search interface to search over all the textual information associated with each file (see Figure 2) and a browsing interface that looks at pages of small "thumbnail" images (see Figure 3). The Image Library is implemented by a standard NCSA Web server running a set of cgi scripts. This server, called the ImgLib server, has its own disk on

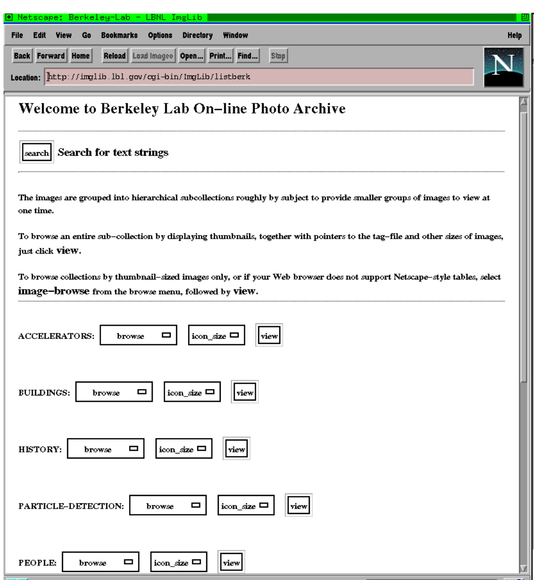

Figure 1. Collection Index

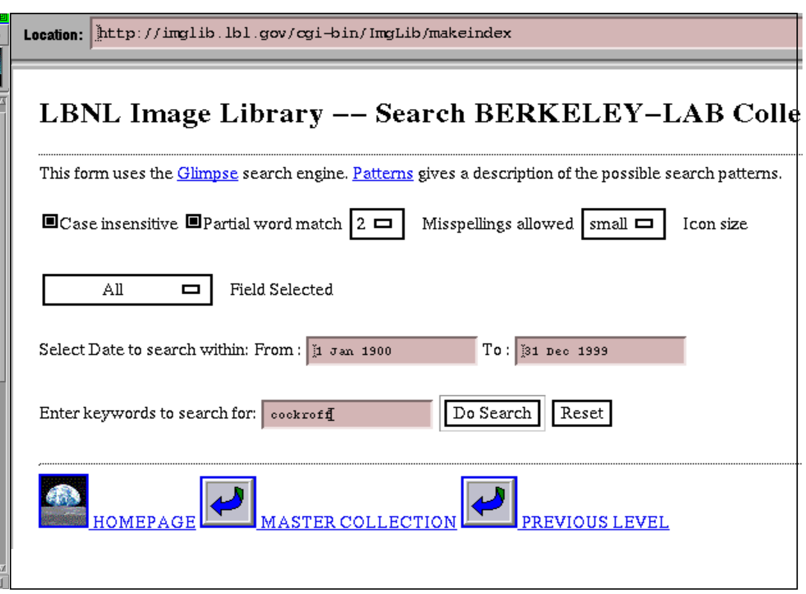

Figure 2. Search Form

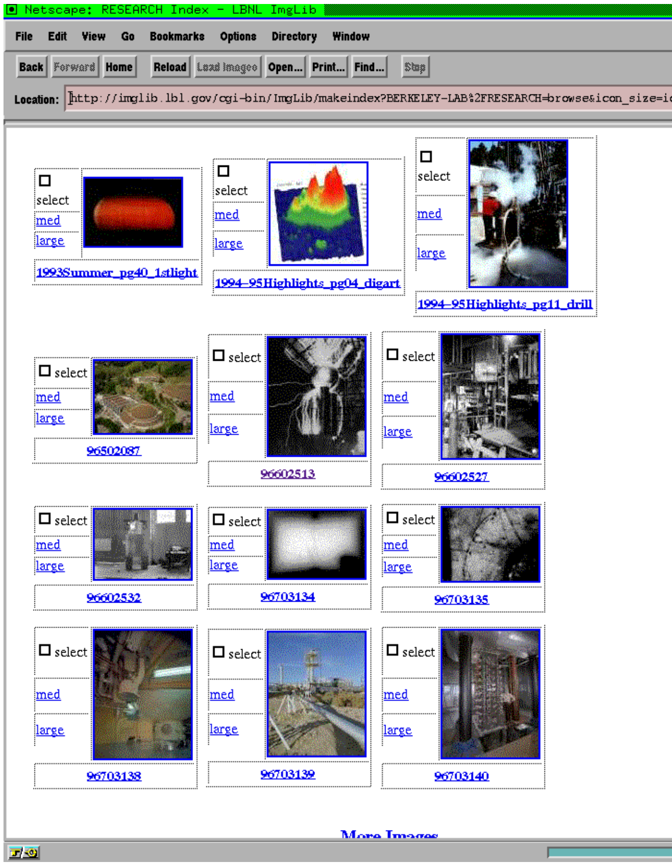

Figure 3. Browse Interface

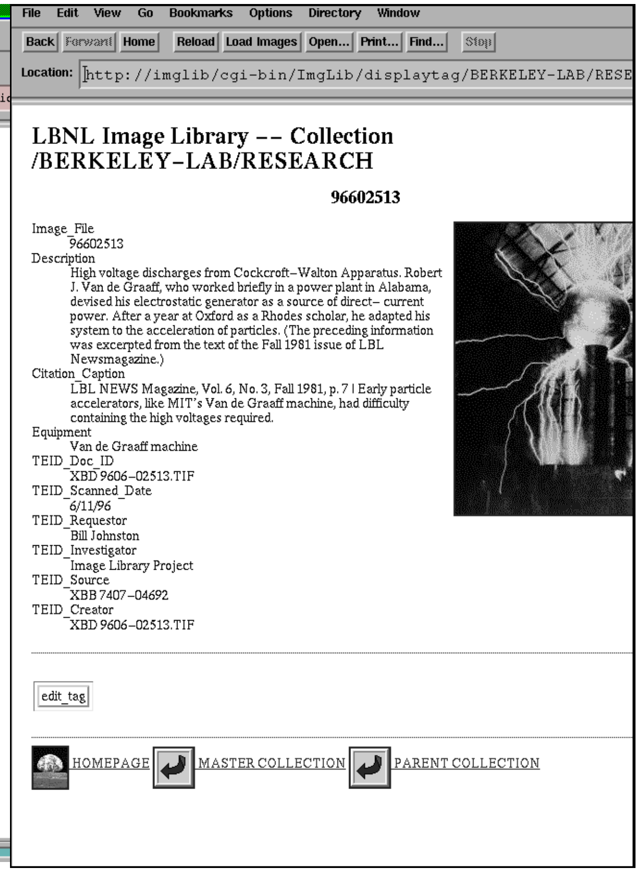

Figure 4. Object Representation

which to store metadata and derived data. The user of this system runs a standard Web browser program, e.g. Netscape, to access the collection.

In terms of our LDO model, the original image is the primary data component, the five "thumbnail" images which are 37, 100, 150, 200 and 650 pixels wide are the derived data and a "tagged-field" text file is the textual metadata. The thumbnail images are created and stored on the ImgLib server's disk space when the primary data is entered into the Image Library. Some of the text data comes from default values specified for the subcollection, some is forwarded as part of each image entered and some is added by the collection curator after the image is in the library. The location of the original image is also kept as part of the LDO definition. A graphic representation of the LDO object is shown in Figure 3.

The ImgLib provides several styles of browsing interfaces to a subcollection: the user can choose to view either 37 or 100 pixel-wide images, with or without pointers to other information. The user may chose an interface based on the speed of his network connection to the server and his familiarity with the images. The searching interface is implemented by a Web form and allows searching over all the text associated with a collection or over just selected fields of information. This searching is implemented by a slightly modified version of the Glimpse search engine[4] developed at the University of Arizona. The results of a search query are returned as an HTML document that displays the thumbnail images and the matching text. This page has links to the complete object representation of each image that matched the search query.

Curator functions are also provided through web forms. The curator can add and delete images, move images from one subcollection to another and edit object and subcollection metadata. The curator functions are protected by Web passwords to the owner of the collection or users who have been specifically granted access. Read access to the collection itself can also be controlled by the owner. Access can be granted to anyone, to users in specific net domains or to specific users.

The browse, search and curator functions are all performed on the metadata or derived data which resides on the ImgLib storage. Thus the user can quickly move through a large collections of LDOs until he finds the one he wants. When the user wants to access the primary data, ImgLib acts as an intermediary to whatever storage system the object resides on. In the case of ImgLib, it will be the Lab MSS. For other large objects it maybe a network storage cache. Here again there is a simple Web interface through which the user can stage an image from tape to disk on the MSS and then download the image to his local machine to view it. In the case of more complex data objects, the Web browser can be used to launch a special application that knows how to display the object.

The ImgLib server has been in operation at LBNL for about 18 months. The collection of micrographs is used regularly by the research group that created them and well as several colleagues at other sites. Several other groups at the Lab are planning to create similar collections for their own images. The research group has benefited from the ability to quickly search and view the low resolution images, the new, simpler interface to the MSS and the ability of colleagues at remote sits to gain access to the images.

The photo archive collection is still only partially built, but so far it has been used to provide on-line images for several Lab retrospective documents. See “Lawrence and his Laboratory” (http://www.lbl.gov/Science-Articles/Research-Review/Magazine/1981/index.html) and “Berkeley Lab Nobel Laureates” (http://www.lbl.gov/LBL-PID/Nobel-laureates.html) This use has underscored the need for a persistent location reference for the objects. The object must be able to move around within the collection as the collection grows and gets reorganized, but URLs saved in external documents must still remain valid. So far the hardest part of building this collection has to been to gather meaningful information about each picture and input it to as textual metadata. Also the breadth of the image subjects has emphasized the need to be able to reorganize a collection into new and different subcollections as it grows.

The success of these two projects, as well as a third collection of medical video clips, has lead us to believe that a Web based framework can be used to manage systems of wide-area distributed large data-objects. The user side of the interface is provided by any standard Web browser and is thus platform independent and widely available. The server and the client-server interactions and access control is done by a standard HTTP Web server. These interactions can be expanded if necessary by Java applets. All we need to provide is a set of scripts and programs, and possibly applets to deal with the various data types, organization of collections and types of storage. Thus we can focus on providing sets of standard functions customized to a particular type of LDO.

Acknowledgments

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 Division (http://www.er.doe.gov/production/octr/mics), under contract DE-AC03-76SF00098 with the University of California. Contact: wejohnston@lbl.gov, Lawrence Berkeley National Laboratory, mail stop: B50B-2239, Berkeley, CA, 94720, ph: 510-486-5014, fax: 510-486-6363, http://www-itg.lbl.gov). This is report no. LBNL-39613.

References

1. Johnston, W., Jin Guojun, Gary Hoo, Case Larsen, Jason Lee, Brian Tierney, Mary Thompson, “Distributed Environments for Large Data-Objects: The Use of Public ATM Networks for Health Care Imaging Information Systems”, (http://www-itg.lbl.gov/~johnston/APII.1.1.fm.html)
2. M.Thompson, J.Bastacky, W.Johnston “A WWW interface of viewing and searching sets of digital images” *Proceedings of Microscopy and Microanalysis 1996* p.402-403 San Francisco Press, San Francisco 1996
3. Image Library Web site, http://imglib.lbl.gov/
4. Udi Manber, Sun Wu, Burra Gopal, http://glimpse.cs.arizona.edu/