Scientific Data Archive at the Environmental Molecular Sciences Laboratory[1]

Daniel R. Adams, David M. Hansen, Kevin G. Walker
Pacific Northwest National Laboratory
PO Box 999 MSIN: K7-28
Richland, WA 99352
dan,dm_hansen,kg_walker@pnl.gov
tel +1-509-375-6579
fax +1-509-375-3641

John D. Gash
Lawrence Livermore National Laboratory
PO Box 808 MS L-103
Livermore, CA 94550
gash@pilgrim.llnl.gov
tel +1-510-422-0708
fax +1-510-423-8274

Abstract: The U.S. Department of Energy's Pacific Northwest National Laboratory (PNNL) has developed a Scientific Data Management system (SDM) that is an integral component of the DOE's Environmental Molecular Sciences Laboratory (EMSL). EMSL is a national scientific user facility that is supported by the DOE Office of Biological and Environmental Research. It is designed to focus Environmental Molecular Science on the puzzles and challenges of environmental restoration. The EMSL provides several major facilities to begin addressing these challenges, including the Molecular Sciences Computing Facility, a laser surface dynamics laboratory, a high field nuclear magnetic resonance laboratory, and a mass spectrometry laboratory. Each of these component laboratories will generate vast amounts of data that must be reliably stored and made available on demand for decades.

The SDM system addresses this challenge by providing a hierarchical storage management system to store data files, as well as a database that captures information about those files (i.e., metadata). The SDM system is a distributed client-server application that runs in a heterogeneous hardware, software, and networking environment. It provides EMSL scientists with 20 terabytes of near-line archival storage together with a metadata database describing the contents of the archive and the context within which archived files exist. This makes it possible to locate information based on what the data is about, not just the name someone gave to a file. The SDM system has been in use at the EMSL since July, 1997.

1. Introduction

EMSL is a collaborative research facility for investigating fundamental molecular processes relevant to the processing, storage, and remediation of hazardous waste and associated health effects at DOE sites. EMSL scientists are engaged in research ranging from *ab initio* molecular orbital calculations on supercomputers (such as the 512-node IBM SP machine) to molecular spectroscopy using one of a kind, ultra high frequency nuclear magnetic resonance instruments. All of these activities are focused on developing new science that can transform the environmental management and restoration processes at DOE nuclear facilities as well as other hazardous environments around the world.

The projected cost of environmental restoration at nuclear facilities is tremendous. In an article on environmental remediation at hazardous waste sites, *Science* magazine reported that "[the] projected costs of remediation of hazardous waste sites continue to mount and now range in the neighborhood of a trillion dollars or more, not counting legal fees. Even with such expenditures, it is unlikely that most sites will be restored to pristine conditions."[2] These estimates are based on current technologies and on reasonable extrapolations and extensions of these technologies within the framework of our present understanding of the underlying sciences. Consequently, new science that supports the development of new cleanup technologies is essential if hazardous waste sites are to be restored to benign states.

In doing this we expect scientists from institutions worldwide to generate several terabytes of data annually, much of which is suitable and appropriate for long term storage and retrieval. The formats of the data files that make up this research data will be as diverse as the scientists who create it. We anticipate binary data streams from proprietary data acquisition systems, word processing files of publications, results from standard numerical models, as well as one-of-a-kind files from custom programs written by visiting scientists and post-doctoral researchers. Some of the experimental data will be from experiments and measurements that are difficult or impossible to reproduce at future dates. Similarly, computational chemists will be storing results from long running (e.g., weeks) supercomputer jobs. Traditional techniques for tracking and managing this data – laboratory notebooks and collections of Zip disks – will simply not be sufficient for the task at hand. Furthermore, we need to make it possible for researchers to locate information that may be germane to current research years after that work was archived.

In order to satisfy these requirements, EMSL has procured a commercial hierarchical storage management system from EMASS and is developing a substantial set of software tools to manage interactions with the data archive. A key feature of the SDM software is the capture and maintenance of metadata about files that are archived. Whenever possible we have automated the process of extracting metadata from files as they are archived. SDM manages this metadata in an object-oriented database that is a component of the SDM software. The EMSL SDM system preserves the appearance of a standard Unix®-like file system, enabling users to locate and retrieve files based on this structure. The real long-term value SDM brings to the archive, however, is the ability to locate files based on their content (as captured in the metadata) and to quickly identify and explore related files. This content based search capability over terabytes of data is the most distinctive feature of the SDM system.

2. Our Work

The SDM system is a distributed client-server application that runs in a heterogeneous hardware, software, and networking environment. Supported client hardware ranges from desktop PC's running Windows 95 to UNIX workstations to an IBM SP supercomputer. It has been in production use at the EMSL since July, 1997. The SDM software is responsible for file transfer, metadata storage and search, and access control.

The SDM system is strictly file-oriented; data is stored and retrieved in units of whole files. We do not support retrieving segments or cross-sections of large datasets from the archive (i.e., no processing of the archived data is performed by the archive system). Metadata must be provided for each file that is archived at the time the file is archived. The SDM software automates much of the metadata extraction task in order to minimize the impact of this requirement on the users. Archived files cannot be directly accessed via ftp or as a network mounted file system (i.e., the archive file system is not available to users for direct

manipulation); files can only be accessed via the SDM software. This makes it possible to ensure that metadata is provided for each file, and it precludes the use of the archive as a large, low-cost scratch disk. It also facilitates the implementation of access control on archived files.

The SDM software enforces access control restrictions that may be placed on archived files by the owners of the data. The SDM software is built using the Open Group's Distributed Computing Environment (DCE) software and thus has reliable and trustworthy Kerberos-based authentication credentials for users. SDM uses DCE Access Control Lists (ACL) and a reference monitor implementation to restrict connections to the servers. However, SDM does not use DCE ACLs to restrict access to archived files once a user has been authenticated to use the SDM software. Instead it maintains and enforces file and directory ACLs within the database component of SDM. The owners of archived data may restrict who has access to files and directories as well as constraining who may view the metadata associated with their files.

While it is possible to use DCE to encrypt the data in files as they are transferred to and from the archive, we have chosen not to do that. Also, in support of monitoring and controlling access to archived data, owners may request e-mail notification from the archive when specific access events occur (such as an attempt by an unauthorized user to retrieve a file).

2.1 Hardware Environment

The data archive is an EMASS hierarchical storage management system using their FileServ Hierarchical Storage Management (HSM) software product running on twoSilicon Graphics, Inc. (SGI) Challenge XL 10000 servers. The secondary storage is contained in an EMASS AML/E robot with a single quadro-tower and 4800 slots for IBM 3590 media. We presently have the tape robot loaded with 2000 tapes for a storage capacity of 20 TB. Each SGI server has 1 GB of RAM, 191 GB of RAID 3 disk cache, 32 GB of local disk, 2 OC-3 ATM interfaces, 2 HiPPI interfaces, and controls four IBM 3590 tape drives in the EMASS robot. This is illustrated in Figure 1.

The ATM network provides the backbone of the EMSL network. Individual workstations typically connect to the network via switched Ethernet connections and the Ethernet traffic is carried over ATM using LAN Emulation. The HiPPI interfaces are part of a private network which presently includes the archive, a 512 node IBM SP supercomputer (which has eight HiPPI interfaces) and SGI graphics and visualization workstations. All network traffic uses the TCP/IP protocols. When fully loaded the archive is capable of transferring data from network to disk at a sustained rate greater than 124 Mbytes/sec. Once the RAID disk caches are full, the tape drives become the bottleneck in throughput and the sustainable data transfer rate drops to 52 Mbytes/sec.

The SDM archive is accessed by a variety of computers. These include a variety of SGI, Sun, and IBM Unix workstations as well as numerous PCs running Windows 95 and Windows NT. A few of these machines are directly connected to the ATM backbone via OC-3 interfaces but the majority use standard 10 Mbit/sec Ethernet interfaces.

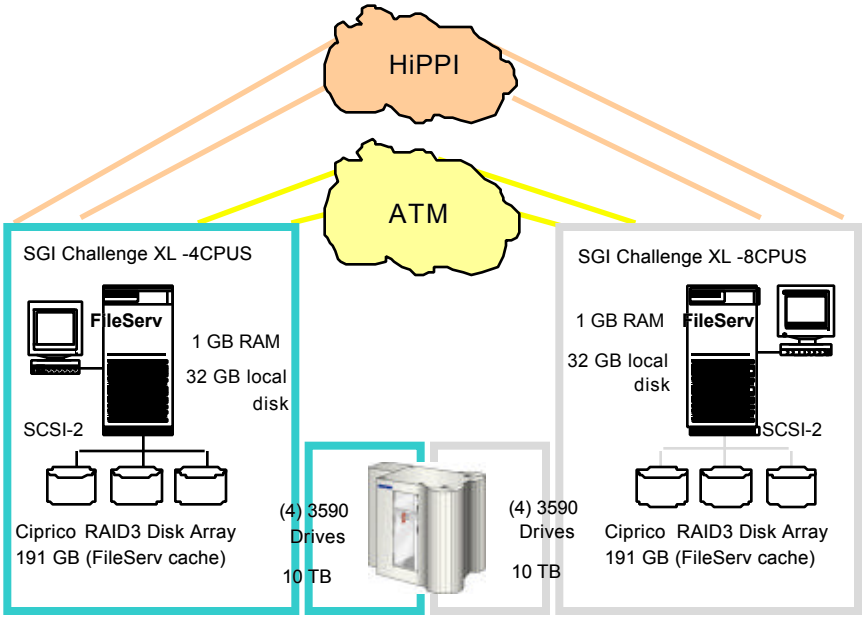

Figure 1. SDM Archive Hardware Environment

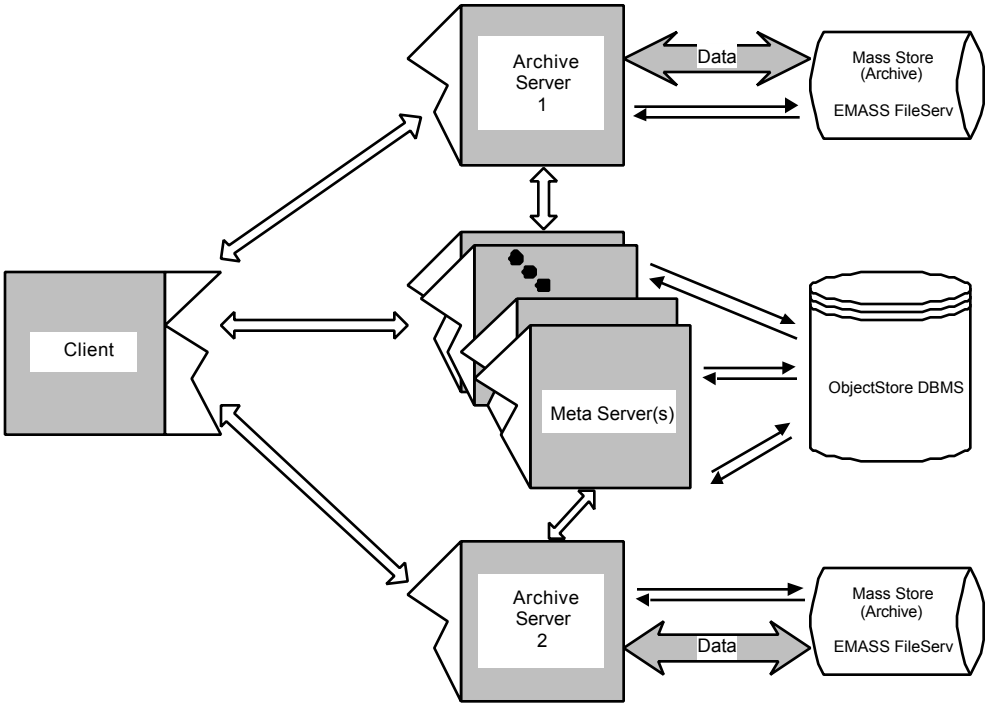

Figure 2. SDM Software Architecture

2.2 SDM Software Architecture

The SDM software is a client-server system built using DCE remote procedure calls. It has three major components: the client, Meta Server, and Archive Servers (see Figure 2). Each client workstation runs a copy of the SDM client that handles authentication of the user to the SDM system (through DCE) and processes user commands such as directory listings, storage, and retrieval.

When an SDM client is started, it makes a connection to the SDM Meta Server (actually one of possibly many Meta Servers, for reasons explained later). The Meta Server lies at the heart of the SDM software in that it provides most of the functionality of the system.

The Meta Server includes the client interface to an ObjectStore object oriented database management system manufactured by Object Design Inc. (ODI). The Meta Server mediates queries and interactions between the SDM client and the database.

One requirement of the SDM system is that the users see a single file system, without regard to the actual hardware configuration of the archive. In our case, the EMASS hardware provides two distinct Unix file systems, so the Meta Server – by way of the associated ObjectStore database – maps files into a single, unified file system. Users store files in their directories and the Meta Server allocates files to a specific server based on criteria such as balancing utilization of each SGI server.

While the ObjectStore server is multithreaded and can handle multiple concurrent queries, the same cannot be said for the ObjectStore client that is part of the Meta Server. Because we were unwilling to single-thread all interactions with the database in the Meta Server portion of the SDM software, we instead implemented multiple Meta Servers as multiple heavyweight processes. Each of these Meta Servers can safely connect to and interact with the ObjectStore database, thus allowing us to circumvent the single-threaded nature of the client. We use the DCE Cell Directory Service to transparently allocate client connections to Meta Servers. Unfortunately, even this was not sufficient to resolve all of our problems with multithreaded applications.

Even though we implemented multiple heavyweight Meta Servers, each server is still a DCE application and is thus intrinsically multithreaded. Initially, the Meta Servers and ObjectStore database were hosted on the more powerful of the SGI Challenge servers. However the ObjectStore client for SGI was not able to work within a DCE application due to conflicts in handling Unix signals. ODI provided us with a patched client for SGI that worked around these issues but we eventually moved the Meta Servers and ObjectStore database from the SGI to a Sun Solaris platform where DCE is better supported and accommodated.

An SDM Archive Server is running on each of the SGI servers. The Archive Servers are responsible for actually reading and writing to the archive file systems. Metadata is streamed to the Meta Server and data is streamed from(to) the SDM client to(from) an Archive Server by using DCE pipes. The Archive Server handles all interactions with the FileServ software through Application Programming Interface calls or separate executables for operations such as staging files from tape to disk for subsequent retrieval.

Because archive operations involve processes running on (typically) three different machines, we needed some sort of distributed transaction manager to keep the SDM database, archive file systems, and current operations synchronized. Commercial products such as Encina were either not available on all the platforms we support or their costs were prohibitive. The SDM software manages distributed transactions through a “contract”

mechanism. A contract represents an operation and its associated components (files) and the status of the operation on each component. We take advantage of the atomicity property of the database to maintain current state and employ a conservative, fail-safe, approach to errors.

For example, when the SDM client requests to store a directory hierarchy to the archive, it contacts the Meta Server with this request and the Meta Server creates a contract in the database with information regarding the files to be stored. The Meta Server sends the contract to one of the Archive Servers, informs the client of the contract identifier, and directs the client to the Archive Server to actually transfer the data. When the client contacts the Archive Server it presents the contract identifier as part of the initial negotiation. The Archive Server verifies the contract with the Meta Server and begins transferring the first file. After each file is transferred to the archive, the Archive Server sends a status update to the Meta Server that reports on the status of the transfer of that particular file. The Meta Server notifies the database and, on successful completion of the transfer, the database makes a permanent entry for the file and marks that portion of the contract as having been completed.

Thus, if there is a network or hardware failure during a file transfer the database will never contain an entry for a file that was not fully and successfully transferred. Because the contracts are stored in the database, a partially completed contract will not be lost if there is a hardware failure. The SDM software can identify partially completed contracts and roll back any file operations that might have been in progress at the time of the failure. Throughout the design and implementation of the contracts, the central principle was to make sure that we never asserted that a file was successfully stored in the archive when in fact it was not.

2.3 Metadata

The metadata that is captured in the SDM database is the key to keeping the archived data accessible and useful by researchers[3]. We have partitioned the metadata into three categories: file oriented, content oriented, and context oriented. File oriented metadata contains information about the file system objects in the archive such as:

- name of file as seen by users
- name of file in the archive
- original owner
- current owner
- size
- date stored
- date of last access
- access history
- access restrictions
- notification expectations.

Content oriented metadata is typically keyword=value pairs that have been extracted from files as they are stored into the archive. Examples of this type of metadata would be:

Comment	=	This spectral data was gathered by the light of the silvery moon using dew drops on gossamer threads.
Operator	=	Dave Dataman
Calibration date	=	1/21/98
Pressure	=	3 torr
Output Energy	=	3 mJoule
Peak power	=	2 terawatts
Instrument	=	12 tesla ICR

Finally, context metadata is information about how a given file or directory is related to others. For example a directory might contain data files from a certain experiment. If these data had been subjected to analysis and quality control checks and the results of these computations were also stored in the archive, then a user with access to the data would be able to establish a contextual link between two archive objects (i.e., files or directories). Later, if the results of the analysis are used as part of a refereed publication, that publication can also be archived and a link established between it and the analysis data. A user who accessed any one of these components (possibly as a result of searching for files related to dew drops) would also be led toward the other parts of the chain. Other relationships are possible and the establishment of relationships is bounded only by the willingness of users to document the relationships or of software to discover and establish them.

Somewhere between the content and context metadata is the notion of annotative metadata. Users may also place the electronic equivalent of "yellow sticky notes" on archive objects. These are simply text records that are associated with archive objects. An example of this would be a warning that the calibration of an instrument used in an archived experiment is suspect. To promote collaboration, a user is usually allowed to annotate any file he can read. However, to guard against abuse, every user is not necessarily granted permission to read annotations placed on files. Thus one user could annotate another's results and say that they are totally bogus, but everyone else who viewed those files would not see the accusation.

Due to the variety of users and academic disciplines represented, we do not maintain a rigid ontology of keywords. Similarly, because the types and formats of the files are not constrained, we have adopted a very promiscuous approach to handling metadata. Specifically, we take whatever the users say is a keyword and track it as such. Software that is logically part of the SDM client attempts to recognize file formats and automatically extract keyword=value pairs from the files as they are streamed to the archive. This approach definitely has significant drawbacks – such as spelling errors, namespace collisions, redundancy – however it appears to be the most workable solution for accommodating the variety and variability we expect in the data. The database schema we use for tracking the file system objects and their metadata is illustrated in Figure 3.

We have attempted to turn this promiscuous metadata approach into an asset by enhancing techniques available for searching the metadata. In particular we "documentize" the metadata wherever possible – turning it into text streams. Throughout the SDM database, where we have textual fields, we build searchable indices using text indexing and search tools made byVerity. Users can query the archive with keyword searches similar to those used for web searches and these searches look through the metadata in finding matches.

3. Related Work

Lawrence Livermore National Laboratory's Intelligent Archive[4,5] (IA) project is addressing similar problems although their approach differs from ours in the association of metadata with the archived data. IA maintains metadata in a database much as our SDM system does, however the acquisition of the metadata is not directly tied to the archiving of files, neither is the submission of metadata a mandatory requirement for use of their archive. Their approach is similar in that they have distinct applications that attempt to automate the extraction of metadata from user files.

The San Diego Supercomputing Center's Massive Data Analysis Systems[6,7] (MDAS) is another project with related but much broader goals. Similar to SDM they are coupling descriptive metadata with files to facilitate the location and retrieval of data based on content as opposed to file system semantics. The MDAS project scope goes beyond this to address the issues of making the managed data self-describing and coupling computational methods with the data and metadata.

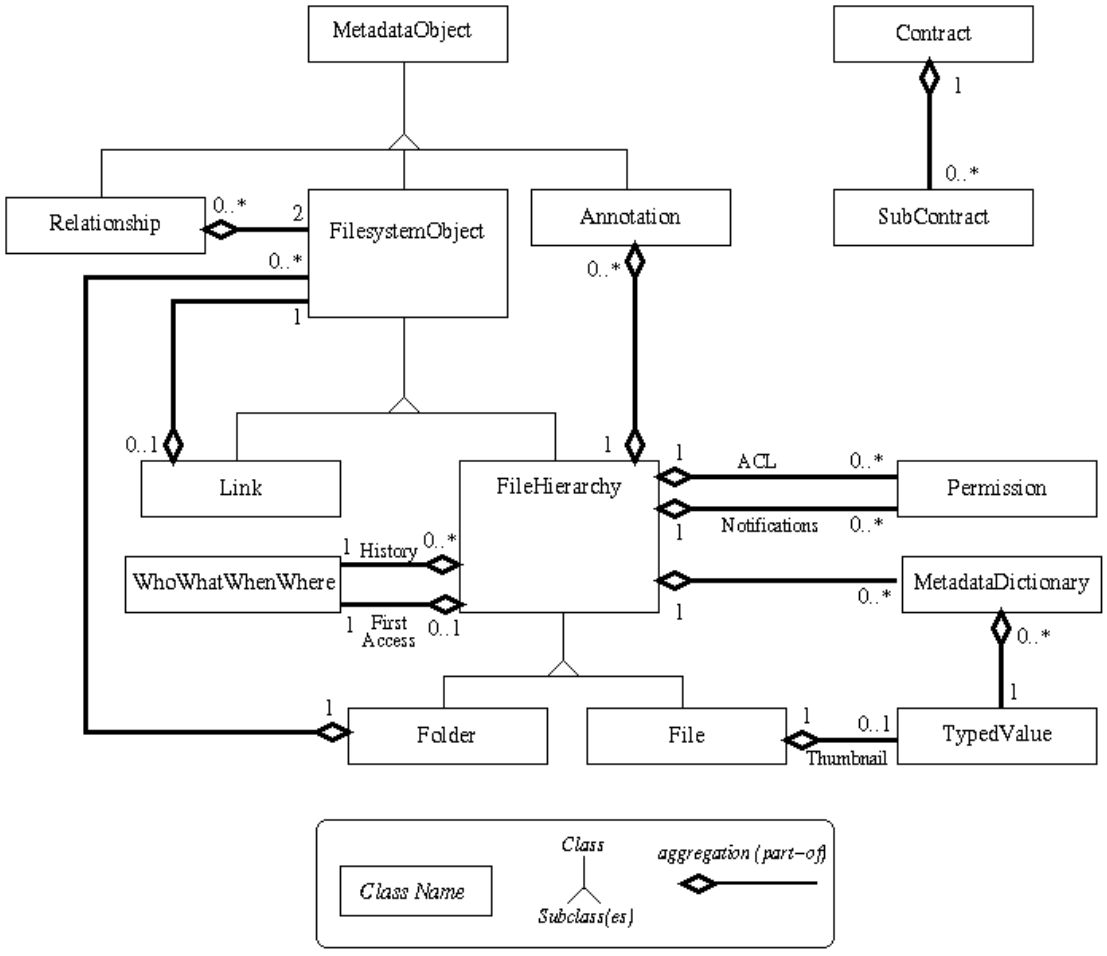

Figure 3. SDM Database Schema

4. Future Work

The SDM system has only recently been deployed so we expect to gather quite a few suggestions from our users as they begin to take advantage of the system. In the near term we will be focusing on operational issues such as increasing the number of file types from which metadata can be automatically extracted, enhancing the tools for performing metadata searches and queries, improving data transfer rates, and enhancing graphical interfaces to the SDM archive. We also have ongoing research activities to develop techniques for identifying and exploiting relationships between distinct archive objects and developing tools to exploit these relationships.

5. Conclusions

We have developed a data archive which couples metadata (file-oriented, content, and contextual) with files as they are archived. The software couples DCE client-server programs with an object oriented database. It insulates users from needing detailed knowledge of the archive structure in order to use the system and it captures and maintains metadata about every file that is archived. Using our Scientific Data Management software it is possible to locate relevant information in the archive without having to know who created or owns the information or what file naming conventions were used when the data was archived. By using annotative metadata and establishing relationships between elements of the archive, the system offers a capability for capturing information about the context in which data exists in addition to simply documenting the contents of the archive.

---

[1] This work was funded by the Environmental Molecular Sciences Laboratory construction project at Pacific Northwest National Laboratory (PNNL). PNNL is a multiprogram national laboratory operated by the Battelle Memorial Institute for the U.S. Department of Energy under contract DE-AC06-76RLO 1830.

[2] "Remediation of Hazardous Waste Sites", *Science*, Volume 255, number 5047, page 901, February 21, 1992

[3] David M. Hansen and Daniel R. Adams. *A Database Approach to Data Archive Management*. In Proceedings of the First IEEE Metadata Conference, Silver Spring, MD, April 1996. IEEE Computer Society Mass Storage Systems and Technology Technical Committee, IEEE Computer Society Press.

[4] http://www.llnl.gov/ia/

[5] Springmeyer, R., Nancy Werner, and Jeffery Long, *Mining Scientific Data Archives Through Metadata Generation.* In Proceedings of the First IEEE Metadata Conference, Silver Spring, Maryland, April 1996. IEEE Computer Society Mass Storage Systems and Technology Technical Committee, IEEE Computer Society Press.

[6] http://www.sdsc.edu/MDAS/

[7] C. Baru, R. Frost, J. Lopez, R. Marciano, R. Moore, A. Rajasekar, M. Wan; *Metadata Design for a Massive Data Analysis System.* In Proceedings of CASCON '96 (11/96)