Data Management as a Cluster Middleware Centerpiece

Jose Zero, David McNab, William Sawyer, Samson Cheung
Halcyon Systems, Inc.
1219 Folsom St
San Francisco CA 94103
Tel +1-415-255-8673, Fax +1-415-255-8673
e-mail: zero@halcyonsystems.com

Daniel Duffy
Computer Sciences Corporation
NASA NCCS, Goddard Space Flight Center
Greenbelt MD 20771
Tel +1-301-286-8830
e-mail: Daniel.Q.Duffy@gsfc.nasa.gov

Richard Rood, Phil Webster, Nancy Palm, Ellen Salmon, Tom Schardt
NASA NCCS, Goddard Space Flight Center
Greenbelt MD 20771
Tel: +1-301-614-6155, Fax: +1-301-286-1777
e-mail: Richard.B.Rood.1@gsfc.nasa.gov

Abstract
Through earth and space modeling and the ongoing launches of satellites to gather data, NASA has become one of the largest producers of data in the world. These large data sets necessitated the creation of a Data Management System (DMS) to assist both the users and the administrators of the data. Halcyon Systems Inc. was contracted by the NASA Center for Computational Sciences (NCCS) to produce a Data Management System. The prototype of the DMS was produced by Halcyon Systems Inc. (Halcyon) for the Global Modeling and Assimilation Office (GMAO). The system, which was implemented and deployed within a relatively short period of time, has proven to be highly reliable and deployable. Following the prototype deployment, Halcyon was contacted by the NCCS to produce a production DMS version for their user community. The system is composed of several existing open source or government-sponsored components such as the San Diego Supercomputer Center's (SDSC) Storage Resource Broker (SRB), the Distributed Oceanographic Data System (DODS), and other components. Since Data Management is one of the foremost problems in cluster computing, the final package not only extends its capabilities as a Data Management System, but also to a cluster management system. This Cluster/Data Management System (CDMS) can be envisioned as the integration of existing packages.

1. Introduction

In the last twelve years, Commercial Off-the-Shelf (COTS)-based cluster computing has become the main source of supercomputing providers. From the revolution of the first viable microprocessors that lead the way to replacing vector supercomputers, to passing through new network technologies and arriving at the efficient porting of scientific code, the road to cluster

computing was paved with problems seemingly impossible to resolve. In a sense, the battle was won; but the war is still being fought.

Many aspects of computing have changed so radically that situations from the past seem unbelievably irrelevant today. Up until 1999, computing centers spent an immense amount of time in lengthy negotiations with vendors in an effort to obtain "build-able operating system codes". Today, they can directly download them from the web.

Still, in the midst of a new era with the power of COTS microprocessors, there are many challenges. Despite networks with low latency and high bandwidth and build-able operating systems and the availability of a myriad of open source packages, cluster computing is, at best, a difficult task that fails to replace the panacea days of Cray Research Inc.'s delivery of a C90 supercomputer.

The Data Management System (DMS) attempts to fill the void of middleware that both supercomputing centers and their users need in order to easily manage and use the diverse technology of cluster computers. The DMS is composed of several existing open source or government-sponsored components, such as the San Diego Supercomputing Center's Storage Resource Broker (SRB), the Distributed Oceanographic Data System (DODS), and others. Since data management is one of the major concerns in High Performance Computing (HPC), the final DMS package not only serves as a data management system for very high end computing, but it can easily be extended to a complete cluster management system.

Many areas of science that base their results on computing resources have different ratios of Mega-Flops per byte of data ingested and/or produced. Meteorology is a science that ingests and produces voluminous amounts of data. It is not a coincidence that the same branch of science that produced the word "computer" is now leading the core issues of cluster computing.

One of the legacy items from the previous computing models of the 60's, 70's, 80's, and 90's is the separation of mass storage engines and computing clusters. At this point, it is more efficient to follow the management structure of the computing centers rather than the computing architecture of the systems. COTS mass storage units, with multiple terabytes of attached disks, are just as reliable and economical as the COTS computing nodes. COTS CPU power has grown side-by-side with high bandwidth internal interconnects and new devices like Serial ATA and others that can provide support for multi-terabyte storage on each single unit. At the same time, OS improvements (Linux, etc.) make it possible to support those large file systems.

In a generic scientific computing center, the problem that must be solved is how to manage the vast amount of data that is being produced by multiple users in a variety of formats. And, the added challenge is to do so in a manner that is consistent and that does not consume all of the users' time manipulating such data or all of the computer center's personnel in endless migrations from one system to another and from one accounting report to the next. This holds true across a broad range of actions from software engineering practices, to the production of code, to upgrading OS versions and patches, and includes changes in the systems, in accounting, in system engineering practices, and in the management of the actual scientific data.

Despite the best efforts of computing centers, "dead data" continues to mount up in mass storage vaults. The increasing cost of maintaining the storage, migrating, and in general curating can reach up to 40% of the total budget of a typical computing center. These curation activities (such as changing ownership, deleting, browsing, etc.) add to the burden of data management. Likewise, the proliferation of mass storage vaults is increasingly higher: two copies in situ, a third copy for catastrophic recovery, a copy in the computing engine (scratch) and additional copies wherever users need them (desktops, websites, etc.). This not only drives up costs, but it also undermines the collaboration among different scientists wherein data sharing becomes a limiting factor.

The cost and expertise necessary to deploy a Grid-useable computing node is too high for small computing groups. Groups of ten to twenty computer users typically have one or two system administrators and no system software developers, which makes the start-up cost beyond their reach (both in terms of dollars and expertise). As computing power increases, fewer groups need a true supercomputer platform. A successful Grid should easily deploy smaller nodes and maintain production level.

Finally, the lack of connection between the datasets and the software engineering practices (code version, patches, etc.) and the computing environment (CPU type, number of CPUs, etc.) limits the life of a dataset, its utility, and the scientific verification value.

In this paper we describe an integration effort composed of several existing packages that solves, to a large extent (but not totally), the data management problem for data coming out of a cluster computing environment. As a posteriori result we describe how the data management, essential to the utility of a cluster, becomes a centerpiece for its management. We also propose an ensemble set that can be used as a turn-key engine for a further integration of Cluster/Data Management into a full Grid/Data Management System ("Incoherent"). In this area, Halcyon proposes that Incoherent be an Open Source Project.

2. Basic Requirements for a Data Management System

The following list contains the basic requirements for the DMS.

- Ensure a single point of information wherein data is retrieved/searched. Though there might be many different interfaces, the initial point of contact for each interface should be the same.
- Provide system tools to cap storage costs and select datasets to be expunged.
- Provide methods for minimizing the number of data copies (and conceivably provide a live backup of the data). The copy that is more efficient to fetch should be the one that is accessed.
- Establish a linkage between data, scientific metadata, computing metadata, and configuration management data.
- Provide support for data migration whether it is from computing nodes to local storage (where users are) or from one storage system to another.
- Support plug and play of different visualization tools.

- Avoid multiple, full, or subset copies of datasets in the system by providing a Virtual Local Data capacity (data always feels local), along with the automatic use of local caches and sub-setting on-the-fly.
- Provide robust, easily deployed, grid-compatible security tools.
- Deploy with ease. Most department-type scientific groups do not have the resources to integrate a fully deployed Cluster Software Management and Mass Storage System.

3. Data Management System, Present Components

Halcyon Systems has integrated several packages to work together as a DMS:

- Storage Resource Broker (front-end mass storage, metadata catalog)
- Distributed Oceanographic Data System (transport layer, connection to manipulation and visualization tools)
- Configuration Management Software for all systems involved
- Distributed Oceanographic Data System and GrADS visualization tool

A minimal number of changes were implemented in the SRB software. A build tool and benchmarks were produced for ease of administration. Exit codes were changed to comply with standard UNIX command return codes. The underlying database is Oracle 9i running on Linux.

The DODS dispatch script is CGI-Perl. It was modified to make calls to SRB S-utilities to retrieve and cache files from SRB. Once a file has been transferred to local disk, it remains there until either the SRB version is modified or the cache fills and it is the oldest file. The DODS server authenticates as SRB identity "dods", and users who wish to export their files via DODS add read access for that user to the files' access control lists.

The DODS environment does not maintain a separate metadata catalog for managing semantic-based access to the data. There is presently no connection between DODS metadata, which is synthesized from the DODS-served file depending on the data format, and SRB metadata, which is stored in the MCAT associated with the file. MCAT data cannot yet be retrieved through DODS, nor is DODS-style synthesized metadata stored in MCAT.

Configuration Management Software is a set of commands enabling the user/administrator to enter changes in the specifically devoted tables created separately from the SRB tables in the Oracle database.

GrADS is already integrated with DODS; however, future work will have a separate server (GrADS-DODS server or GDS) fully integrated with SRB. In this way, a wider set of data manipulation and computation will be directly accessible to DMS users.

Note on GCMD integration: DMS uses SRB's "user defined metadata" facility to store GCMD-compliant metadata. We have defined site-standard user metadata attributes corresponding to the attributes defined in GCMD; then restricted their values based on GCMD convention. An application level tool replaces the general purpose SRB metadata manipulation client and enforces the conventions.

4. Existing Architecture

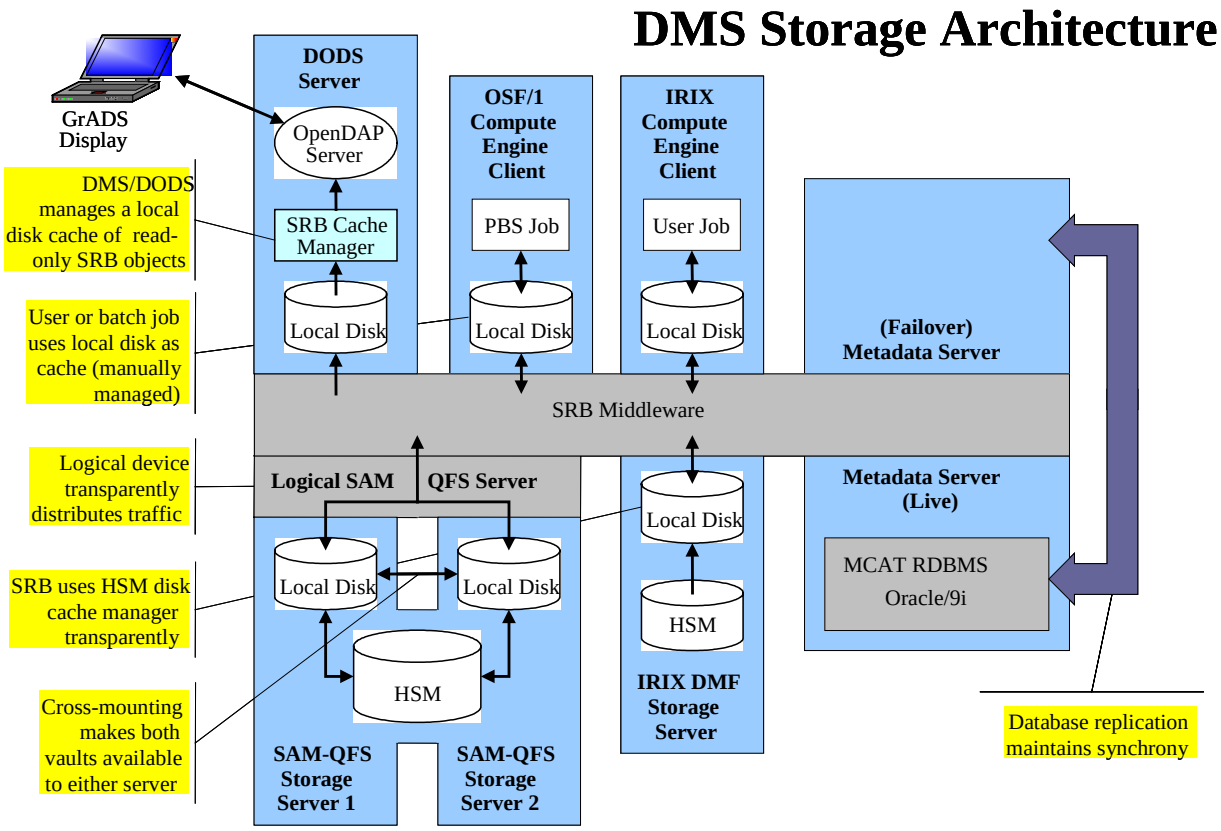

Figure 1: Depicts the existing components of the DMS deployed at the NCCS and their functionality.

5. Requirement Fulfillment

Based on the requirements and the architecture described above, the DMS currently meets the following requirements.

- There should be a single point of information for retrieving/searching the data. Even though there might be many different interfaces, the initial point of contact for each interface should be the same. SRB provides a single point of access for the data.

- The system should provide tools to cap storage costs and select datasets to be expunged. The Data Management System Toolkit provides tools to manage expiration dates for datasets and mechanisms allowing users to preserve selected datasets beyond a given lapse of time (separate description).

- The system should provide ways to minimize the number of data copies and could provide a live backup of the data. The copy that is more efficient to fetch should be the one fetched. DODS/OpenDAP can manage a local cache, network-wise close to the users. Computations, on-the-fly sub-setting can be provided by tools like the GrADS-DODs server (active implementation already on-going).

- Scratch space on the computing platforms can be managed by a short expiration date of a SRB replica of a given dataset.

- Linkage between data, scientific metadata, computing metadata. and configuration management data. SRB flexible metadata schemas provide a linkage between datasets and their scientific content. Metadata schema has been modified to accommodate the format provided by the Global Change Master Directory Software (GCMD), although a fully compatible version of GCMD has not been implemented as yet. Halcyon also has integrated a Configuration Management Software into the DMS that links the system "state" (patches, compilers, etc.) of computing engines with the dataset metadata.

- Provide support for data migration from computing nodes to local storage (where users are) or from one storage system to another: SRB provides bulk transfer from legacy mass storage systems to newer ones; and DODs/OpenDAP can manage local caches as datasets are requested by users. The Halcyon DMS Toolkit provides the following features:

    - file ownership management (user, group, project)
    - file expiration dates management tools
    - **dms acct** uses MCAT interface for accounting reports
    - **dms admin** provides administrative commands
    - **dms meta** provides metadata management, search
    - **dms ingest** stores files with metadata automatically
    - adds concept of file certification. A process through which the users can extend the life of a file beyond expiration dates.

- Provide robust, easily deployed, grid-compatible security tools. SRB's underlying security infrastructure is compatible with the Grid Security Infrastructure (GSI). At the moment, the current DMS deployment is using password encryption, which is more robust than FTP and does not pass clear text passwords. GSI can support tickets (PKI) and Kerberos infrastructure.

- Ease of deployment. Most department-type scientific groups do not have the resources to integrate a fully deployed Cluster Software Management and Mass Storage System. Halcyon is planning to deploy a turn-key server, named Infohedron, to deploy the DMS software in a single box (see next section).

6. Performance

As with all high performance production systems, the risk of *not* utilizing all available network bandwidth can be a significant issue. In tests performed between two single points at NCCS, the following results have proven that the DMS and, particularly, SRB are able to sustain performance levels equivalent to scp transfers without the overhead of CPU consumption due to encrypting and decrypting the data.

The NCCS implementation is built around a pair of redundant Linux-based SRB MCAT servers running Oracle/9i to provide database services. These DMS servers are identically configured two-CPU Xeon systems with 4 GBytes of RAM and SCSI RAID disk arrays. One machine, the primary, is the active server. The second is a hot backup that can be brought into production within two hours should a catastrophic failure disable the first, losing at most thirty minutes worth of MCAT transactions—although in the vast majority of situations the RAID arrays prevent this type of serious failure and no transactions will be lost.

DMS/SRB I/O bandwidth was measured between two hosts, "halem", a Compaq Tru64 compute cluster acting as SRB client, and "dirac", a Solaris9-based SAM-QFS storage server. The tests reported here used a single node of halem and a single node of dirac interconnected by Gigabit Ethernet. Thirty-two transfer threads ran simultaneously—although test results indicated that the performance changed little from eight to sixty-four nodes. These bandwidth tests were designed to demonstrate that DMS/SRB is capable of supporting the near-term projected storage load for NCCS, which was estimated at 2 TBytes per day with a ratio of three writes to one read—i.e., 1.5 TB write traffic and 0.5 TB read traffic per day. The average file at NCCS is 40 MBytes in size, and it was calculated that in order to meet the daily write requirement it would be necessary to complete the transfer of 1600 files in an hour. Although only one third this number of files had to be transferred within an hour to meet the read test requirements, for convenience the tests ran with the same group of 1600.

A significant part of the file transfer time is due to MCAT overhead independent of the file size, so the aggregate throughput increases significantly as the file size increases. For these tests, no NCCS-specific network optimization—for instance adjustment of network buffer sizes—took place.

TEST	ELAPSED m.	MB/s	TB/day
write	30.5 - 33.3	32 - 35	2.6 – 2.9
read	17.6 – 32.2	33 – 60	2.7 – 5.0

1600 40MB files, 32 threads, halem → dirac
requirement: 1 hr. or less, 2TB day (3:1 W:R)

NOTE: single client system to single server system;
no optimization to NCCS network

As the table demonstrates, DMS/SRB was easily able to meet the requirements even without optimization. The daily performance numbers were extrapolated from the 1600-file test performance.

The second group of tests measured MCAT transaction performance and were intended to demonstrate that DMS can support the expected number of file metadata operations per day. For the tests, it was estimated that each file would have 15 associated metadata attribute-value pairs, and similarly to the bandwidth tests a group of 1600 canonical 40 MByte files was used.

Metadata insertions and deletions were tested, as well as simple queries—display of the metadata attributes associated with a particular file. 50,000 insertions and deletions were required each day, as well as 10,000 searches.

DMS Performance: Metadata

TEST	ELAPSED m.	TRANS/s	TRANS/day
insert	43.5 – 48.6	8.2 – 9.2	711K – 795K
query	2.9 – 3.1	129 – 140	11.2M – 12.1M
delete	42.4 – 45.3	8.8 – 9.4	770K – 815K

1600 40MB files, 32 threads, halem → dirac
requirement: 50K inserts/day, 10K search/day

Even more so than with the bandwidth tests, the DMS/SRB easily exceeded the requirements.

7. Infohedron System Architecture

Presently, the DMS system is built on a Linux and Oracle 9i platform with limited redundancy (manual switchover), which covers the minimal needs of a production system. The cost of upgrading to a replicated database is largely due to the cost of an Oracle replicating database. Halcyon is testing the deployment of a Postgres-based, underlying database. In this area, Halcyon has been using an SRB 2.1 server while advancing to Postgres version 7.4. This decision has been based on the large customer base of Postgres – which allows it to mature faster – and the smaller customer base of SRB, which implies a slower maturation process of the software to arrive at the production level required by the NCCS environment. With an upgrade to SRB 3.0, the process would close the compatibility of Infohedron platforms by distributing the metadata catalog and, thereby, form a federated DMS.

In planning for the full deployment of Infohedron, Halcyon has included the GrADS-DODS server to fulfill the needs of NCCS major customers, such as the Global Modeling and Assimilation Office (GMAO), as well as the following packages (to make it useful to a wider audience of customers).

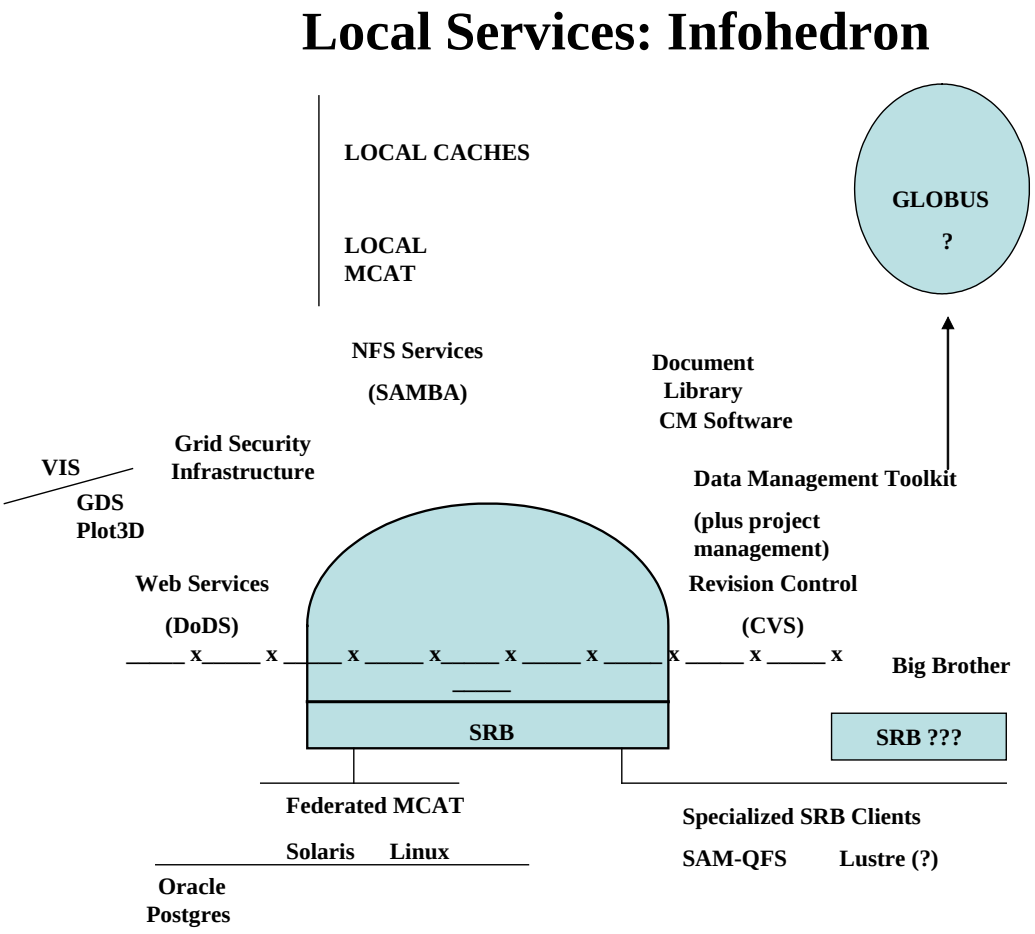

Figure 2: Depicts the turn-key option with typical services needed by a scientific group to adhere to a Grid-like infrastructure. The seemingly chaotic disposition of the packages is intended to depict large variations in needs from group-to-group. The question marks indicate uncertainties in the configuration of groups or the possibility of replacing them with other packages.

8. DMS as Cluster Management

By managing the accounting in the cluster and providing Virtual Locality for Data, DMS can provide full utilization of the cluster and the local caches co-located with the users and the scratch space of the computing cluster itself. By containing the software engineering information and the computing configuration management, DMS is able to provide data integrity and reproducibility.

Homogeneous, easily deployable security infrastructure couples with federated metadata catalogs enabling the Grid. Migration of data and underlying data movements can be controlled in a small environment automatically and in a larger environment with the aid of user indirect manipulation (SRB replication process). Finally user control of data sharing and user quotas (SRB 3.0) can enable cluster sharing, producing a CDMS.

9. Future Directions

Though many of the components described in this paper already exist, and their integration is relatively simple, the production level will be arduous to achieve. Halcyon provides a rigorous system engineering background to test, document and deploy all components. While the effort is sizeable, it has the potential to move progressively toward deployment of a large grid by doing the hardest work first – incorporating legacy data into a Data Management System and then enlarging the DMS into a wider set of services service like CDMS.

Parallel transfers of datasets over separate rails support is provided by SRB. However, it has not been tested under production on DMS.

GSI infrastructure has not been deployed at NCCS. The level of Software Systems support has not yet been determined.

Grid wise accounting has not yet been defined under CDMS.

The Earth System Modeling Framework (http://www.esmf.ucar.edu/) is in the process of formulating an I/O interface. The DMS project will provide a library to interact directly with DMS. If proper network support is provided, an application running in a computer cluster could directly deposit files into mass storage systems. In this way, a consolidation of high performance file-systems would provide savings, as well as avoid the usual double I/O process of depositing files in a local parallel file-system and then transporting them to mass storage.

Integration of the DMS with Lustre: Luster is a distributed file-system designed to provide high performance and excellent scalability for cluster computers. The resulting system would combine the simplicity, portability, and rich interfaces of DMS with the high performance and scalability of Lustre, effectively extending DMS to efficiently support data-intensive cluster-based supercomputing.

Lustre is designed to serve clusters with tens of thousands of nodes, manage petabytes of storage, and achieve bandwidths of hundreds of GBs/sec with state of the art security and management infrastructure. It is currently being developed with strong funding from the Department of Energy and corporate sponsors.

Experimentation with more integration between SRB and the underlying Hierarchical Storage Systems could lead to a more efficient sub-setting by extracting only necessary parts of the files to be sub-set directly from tape (no full file recalling). This is similar to the ECMWF MARS Archive.

In conclusion we propose a two-tier approach: Firstly, convert the typical mass storage/computing cluster architecture most computing centers have to a service rich Cluster/Data Management System Architecture as, for example, the one described in this paper. Secondly, produce a brick-like engine that can take care of most requirements of the diverse, medium- to small-size groups. These bricks would provide local data caches and direct connection to software trees, as well as many other services targeted to the individual groups.

In this manner local idiosyncrasies can be accommodated while maintaining a homogeneous systems engineering throughout a Computing Grid.

The further development of this project would be a breakthrough in data-intensive supercomputing, alleviating a persistent performance bottleneck by enabling efficient analysis and visualization of massive, distributed datasets. By exploiting dataset layout metadata to provide direct access to the relevant portions of the data, it is possible to avoid the performance limiting serialization traditionally imposed by requiring transfer of the entire dataset through a non-parallel mass storage system.

References

[1] Rajasekar, A., M. Wan, R. Moore, "mySRB and SRB, Components of a Data Grid", 11th High Performance Distributed Computing conference, Edinburgh, Scotland, July 2002.

[2] Arcot Rajasekar, Michael Wan, Reagan Moore, George Kremenek, Tom Guptil, "Data Grids, Collections, and Grid Bricks", Proceedings of the 20th IEEE Symposium on Mass Storage Systems and Eleventh Goddard Conference on Mass Storage Systems and Technologies, San Diego, April 2003.

[3] http://www.unidata.ucar.edu/packages/dods

[4] http://www.esmf.ucar.edu

[5] http://gcmd.gsfc.nasa.gov

[6] http://www.globus.org

[7] http://www.escience-grid.org.uk

[8] http://www.nas.nasa.gov/About/IPG/ipg.html

[9] http://www.globalgridforum.org