Knowledge-based Grids

Reagan W. Moore
San Diego Supercomputer Center
La Jolla, CA 92093-0505
moore@sdsc.edu
Telephone: +1 858 534-5073
FAX: +1 858 534-5152

Abstract

Grids tie together distributed storage systems and execution platforms into globally accessible resources. Data Grids provide collection management and global namespaces for organizing data objects that reside within a grid. Knowledge-based grids provide concept spaces for discovering relevant data objects. In a knowledge-based grid, data objects are discovered by mapping from scientific domain concepts, to the attributes used within a data grid collection, to the digital objects residing in an archival storage system. These concepts will be explored in the context of large-scale storage in the web, and illustrated based on infrastructure under development at the San Diego Supercomputer Center.

1. Introduction

Grids are middleware that tie together storage systems and execution platforms that reside in multiple autonomous administration domains. The middleware provides a common infrastructure for creating global properties that span separate heterogeneous, independent resources. The common infrastructure is used to support a single sign-on authentication environment, uniform job submission mechanisms, uniform naming conventions for grid resources, and uniform scheduling systems.[1] Grids therefore serve as interoperability mechanisms for turning remote, heterogeneous resources into a globally accessible system.

Data grids build a data management infrastructure on top of grid environments to support access to potentially billions of digital objects. A major capability of data grids is support for data discovery. While it is possible to uniquely name a billion objects, discovering a relevant object is difficult without a context to associate with the object. The context provides the ancillary information that describes the object origin (provenance information), the processes used to create the object (procedural information), the object type (structural information), and the object relevance (functional properties for the given scientific domain). The information is expressed as attributes that are organized into a collection and stored in an information repository.[2] Discovery of objects can then be accomplished by querying the information repository.

Since collections are typically organized around discipline-specific topics, data grids must support access to a large number of information repositories. Digital object discovery then involves identifying the relevant collection, specifying desired values for the collection-specific attributes, issuing a query against the collection, and finally retrieving the resulting digital object. This process requires knowledge on the part of the user to correctly identify which collection to use. Disciplines use ontologies to provide structure to information within their field of study. Ontologies can be used to define relationships between separate collections, and thus serve as an information discovery support layer.

The discovery process also requires knowledge to correctly interpret the meaning of the collection attributes.

This is equivalent to understanding the semantics associated with a collection. When multiple collections are accessed, the user must understand the relationships between the semantics used by each collection. Thus management of very large namespaces through the creation of collections inherently depends upon a representation of knowledge and semantic relationships within each scientific discipline. Knowledge-based grids provide the relationship management.

To minimize the required infrastructure development, we use the simplest possible definition for data objects, information, and knowledge. In this approach, data objects are bit streams. Information corresponds to any tagged data element. Knowledge corresponds to any relationship maintained between two pieces of information. These representations for data, information, and knowledge make it possible to build independent middleware infrastructure layers to manage bit streams, tagged attributes, and attribute relationships.

There are many examples of systems that integrate relationship management with information and data management:

- Logical / semantic relationships are maintained by digital library cross-walks to manage semantic interoperability between two collections.
- Temporal / procedural relationships are maintained by workflow systems that track the processing of documents.
- Spatial / structural relationships are maintained by graphical information systems for co-registering images.
- Functional relationships are expressed in scientific algorithms used for feature extraction.

A knowledge-based grid provides mechanisms to manage all of these types of relationships in a distributed environment, building upon the information management provided by data grids and the interoperability mechanisms provided by the global grid. The resulting infrastructure provides capabilities that exceed those of traditional hierarchical storage managers. Knowledge-based grids support access to multiple levels of storage distributed across separately administered sites, while providing sophisticated data discovery mechanisms. One can envision implementing a web-based hierarchical storage manager through the integration of data grids and knowledge management systems.

Data discovery within grids is designed to differentiate between the concepts used by researchers to describe their domain, and the attributes used to identify data objects.[3] Researchers work in concept spaces, in which terms are defined that represent their view of reality. The terms are typically based on physical abstractions and capabilities. Digital objects are images of reality that are captured as output from simulations or measurements from remote instruments. Researchers would like to interact with a grid environment in terms of their domain concepts. This implies that data discovery within grid environments should manage the mapping from the researcher domain concepts to the collection attributes used in the data grid. This can be achieved by using systems such as the ISO 13250 Topic Map standard.[4] Topic Maps manage typed relationships between topics (concepts) and associations between topics and collection attributes.

This paper will outline how knowledge management and information management can be used to improve the ability to build collections of data objects and to discover data objects within large-scale storage systems in the web. The discovery environment provides mechanisms to query objects at the data level through feature extraction, at the information level through attribute-based queries, and at the knowledge-level through domain concepts. Just as data management infrastructure is intended to provide access without having to know the data object names used in storage systems, knowledge access infrastructure is intended to provide access without having to know the explicit metadata attribute names used to organize the data gird collections.

2. Collection-based Data Grids

Data-grids can be characterized as collection-based repositories. Instead of accessing data objects by a known path name, data objects are identified by a query on collection attributes. The SDSC Storage Resource Broker (SRB) and Metadata CATalog (MCAT) combine a data handling system with a collection to automate data discovery and retrieval.[5] The SRB/MCAT system is shown in Figure 1.

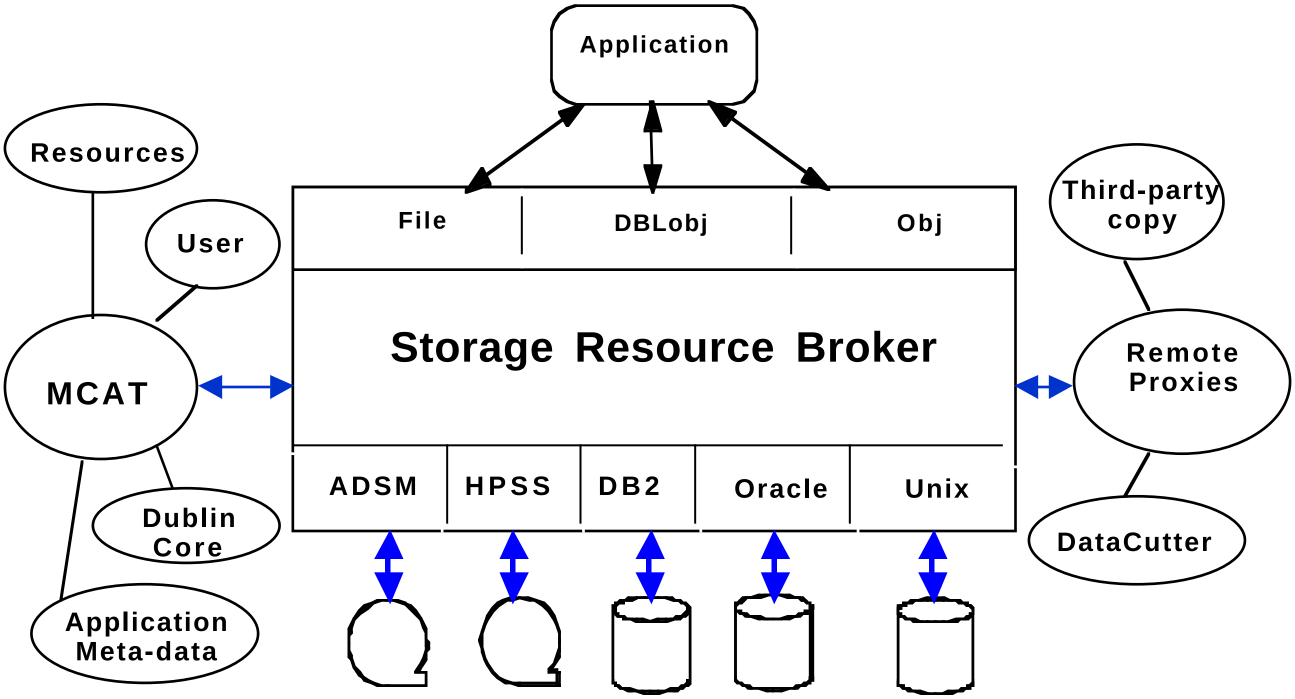


Figure 1. Storage Resource Broker Components

The Storage Resource Broker is implemented as client-server middleware. A SRB server is installed at each site to translate from the local storage access protocol to the Unix semantics provided by the SRB client API. New types of storage systems are supported by writing new drivers for the SRB server. Supported storage systems include archives (High Performance Storage System-HPSS, UniTree, Data Migration Facility, ADSM), file systems (Unix, Linux, Windows NT, Mac OS X), and databases (Oracle, DB2, MySQL). All collection attributes are stored in a database that is managed by the MCAT system. The MCAT system automatically generates the SQL that is used to resolve queries based on the attributes specified in the SRB client discovery API.

The SRB provides the ability to create a logical collection that spans multiple storage systems. Objects that are resident in archives, or remote file systems, or that reside as objects in object-relational databases, can be organized into a single coherent collection with access controls that are managed separately from the local storage systems. A hierarchical folder organization can be applied to the collection that is independent of the local storage system data organization. The objects within the collection are owned by the SRB data handling system, and access control lists are used to protect proprietary data. The SRB expands the set of data resources that are accessible from an application from those data sets owned by the researcher, to all of the data sets in the SRB collections for which access permissions have been granted.

In general, collection-based data grids address issues related to:

- Global namespace
- Location transparency
- Protocol transparency
- Latency hiding

The SRB global namespace is defined relative to a particular collection. The global name is an attribute that can be specified by the data object owner. The local name is constructed using rules specific to the collection. The MCAT collection maintains both the global name and the local file name. Replicas of the data object are automatically given the same global name, but are assigned unique local file names. Replica information is maintained in the MCAT collection catalog. Additional attributes are kept to describe Unix file semantics (file length, file update time, access control lists, etc.), to describe Dublin Core provenance attributes (data set creation information), and to describe discipline specific properties (physical features of the data).

Location and protocol transparency are achieved in the SRB by storing metadata describing the storage location and access protocol along with the local file name and replica number in the MCAT catalog. The SRB data handling system automatically invokes the correct low level driver for accessing a storage system. The SRB data access API uses the same semantics for accessing data in file systems, databases, and archives.

Latency hiding within grids is accomplished by minimizing the number of times the wide-area network latency is incurred. The standard mechanisms for hiding latency are to cache data locally, create replicas at multiple sites within the grid, aggregate data, and aggregate I/O commands. For archival storage systems, latency hiding requires the use of containers for physically aggregating small files before transfer to the archive. The container size is adjusted to the access latency of the archive (size = bandwidth * latency) and is typically 300 Mbytes for the SDSC archive. The MCAT maintains attributes for the location and length of each digital object in the container. Containers provide two benefits: associated objects are retrieved simultaneously, and the impact on the archive namespace is minimized. For example, a 5-million image 2-Micron All Sky Survey is being archived at SDSC in only 147,000 containers.

For minimizing latency when doing remote I/O operations, I/O commands can be aggregated through use of a remote proxy. The invocation of the proxy can be used to execute multiple I/O operations that are needed to define a data subset. The remote proxy mechanism used in the SRB is based upon the DataCutter technology developed at the University of Maryland by Dr. Joel Saltz.[6] Remote data sets can be indexed and manipulated through a single command sent over the wide-area network. The resulting data subset is then accessible through the SRB data access API.

3. Knowledge-based Data Management

A knowledge management environment builds upon collection-based data grids by adding infrastructure to manage domain concepts and to manage relationships between domain concepts and collection attributes.[7] In addition, a knowledge management environment provides ingestion mechanisms that support registration of new digital objects and of the associated information and knowledge. The knowledge environment must also provide access mechanisms that support discovery and retrieval of data, information, and knowledge. The three infrastructure components (ingestion, management, and discovery) are shown in figure 2 as three separate columns. The three columns represent the technologies needed to manage the ingestion process, manage the repository infrastructure, and manage the discovery environment. The three rows represent the separate infrastructure components needed to manage knowledge, information and data.

	Ingestion	Management	Discovery
Knowledge	Concepts Relationships	Knowledge Repository for Relationships	Knowledge or Topic-Based Query
Information	Semantics Attributes	Information Repository	Attribute- based Query
Objects	Fields Containers Folders	Storage (Replicas, Persistent IDs)	Feature-based Query
	Standards	Infrastructure	Languages

Figure 2. Knowledge-based Grid

Ingestion corresponds to the steps of knowledge mining/tagging, information mining/tagging, and data set structure specification that are needed to characterize digital objects. Knowledge mining includes both identification of domain concepts that will be used to reference the collection, and the mapping of the domain concepts to the collection attributes. Information mining consists of tagging the information content identified within data objects, and also of identifying the ancillary information that is associated with the data objects. The information is expressed as attributes stored in a collection. Data ingestion requires the characterization of the internal structure of the data set, such as through use of the Hierarchical Data Format, the aggregation of digital objects into containers to simplify storage management, and the logical organization of the digital objects into a hierarchy of folders.

Management infrastructure consists of knowledge repositories to manage rules and relationships, information repositories to hold attributes or metadata about the digital objects, and data storage systems to manage the bit streams.

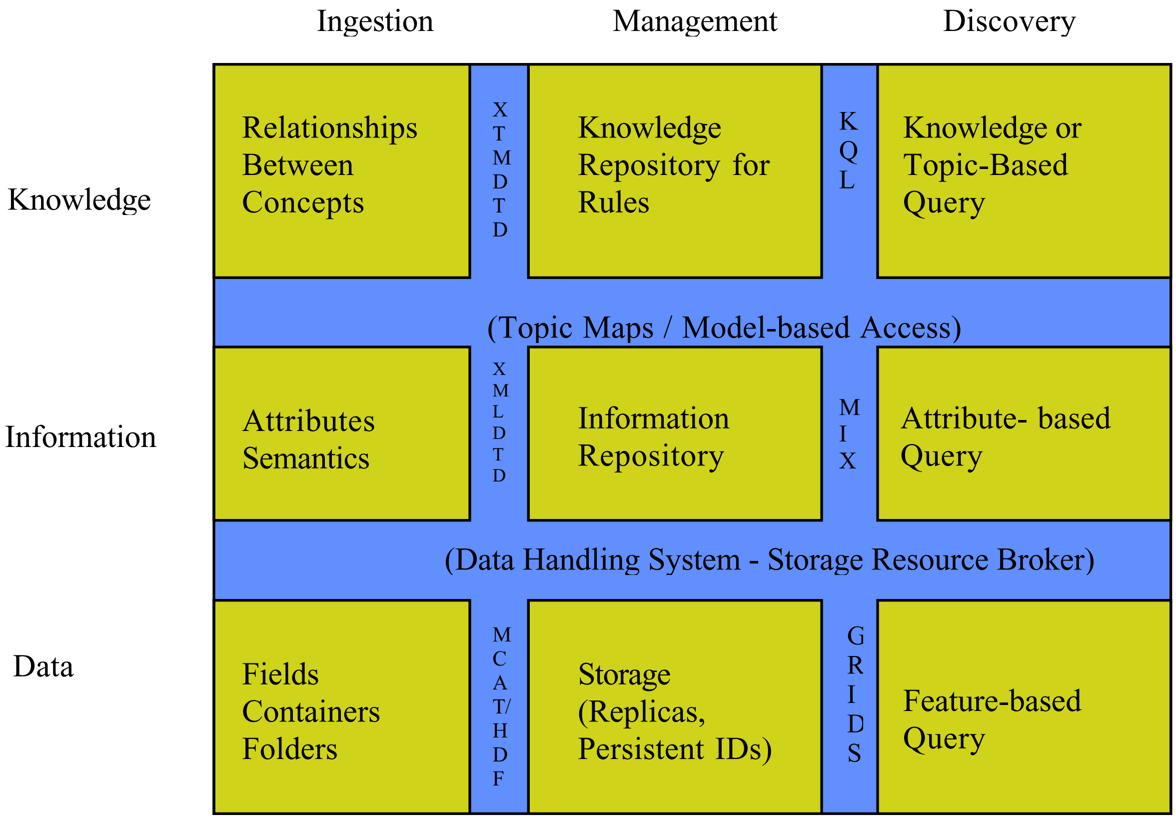
The discovery environment provides mechanisms to query objects at the knowledge-level through domain concepts, at the information level through collection attributes, and at the data level through feature extraction. Just as the information management infrastructure is intended to provide access without having to know data object names, the knowledge access infrastructure is intended to provide access without having to know the names of the explicit metadata attributes used to organize the grid information repositories.

4. Knowledge-based Data Grids

A knowledge-based grid provides the software infrastructure to support interoperability between different implementations of each of the infrastructure components used to build knowledge-based data management systems. As shown in Figure 3, a grid provides the interoperability mechanisms linking multiple implementations of knowledge, information, and data repositories. The interoperability mechanisms have traditionally been different for each type of grid object. Thus the mechanisms used to manage data objects are different from the mechanisms used to manage executable programs. One goal of a “simple” grid architecture will be to use common information and knowledge management mechanisms that support all types of grid objects.

For managing data objects within a grid, the infrastructure components that are being developed at the San Diego Supercomputer Center are shown in Figure 3. Between the columns “Ingestion” and “Management”, standards are used to define consistent tagging mechanisms for knowledge (eXtensible Markup Language Topic Map Document Type Definition or XTM DTD), for information (XML DTD[8]), and for data organization (logical folders and physical containers). Between the columns “Management” and “Discovery”, standard query languages are used for knowledge-based access (logic or rule manipulation languages), attribute-based access (SDSC Extensible Metadata CATalog SQL generator or UCSD Mediation of Information using XML mediator), and feature-based access (application of procedures within a computational grid). Between the “knowledge” and “information “ environments, a standard representation is used to map from concepts to attributes, such as topic maps or model-based access systems. Between “information” and “object” environments, the SDSC Storage Resource Broker data handling system is used to map from attributes to storage locations.

The grid middleware constitutes the software needed to implement the interfaces (blue regions) between the grid resource components. The goal is to be able to implement common middleware across all grid resource types. Note that in this view, the traditional grid components occur mainly in support of feature-based query in the lower –right triangular section of the figure. Also note that much of the infrastructure development done by the digital library community occurs in the upper-left triangular section of the figure. A knowledge-based grid provides a way to integrate digital library and grid technologies.



	Ingestion		Management		Discovery
Knowledge	Relationships Between Concepts	XTMDTD	Knowledge Repository for Rules	KQL	Knowledge or Topic-Based Query
			(Topic Maps / Model-based Access)		
Information	Attributes Semantics	XMLDTD	Information Repository	MIX	Attribute- based Query
			(Data Handling System - Storage Resource Broker)		
Data	Fields Containers Folders	MCAT/HDF	Storage (Replicas, Persistent IDs)	GRIDS	Feature-based Query

Figure 3. Knowledge-based Grid for Data Objects

One way to accomplish the goal of knowledge-based access is to use the ISO 13250 Topic Maps standard to maintain mappings between domain concepts and the attribute names used in the data collection schema. It is very interesting to note that relationships are also implicit between each of the nine areas defined in Figure 3. The relationships either define rules that can be applied to data collections, or quantify associations that can be made between collection elements. Examples are:

- Relationships that quantify rules:
    - Rules for defining grid attributes
    - Rules for organizing attributes into a schema
    - Rules for feature extraction
    - Rules governing object creation
- Relationships that quantify associations:
    - Organization of concepts into topic maps
    - Ontology mapping between concept maps
    - Mapping of concepts to collection attributes
    - Mapping of concepts to feature extraction rules
    - Mapping between attributes and data fields (semantics)
    - Semantic mapping between collections
    - Mapping between attributes and storage
    - Mapping between attributes and features
    - Clustering of data into containers

Of interest is the emerging need for knowledge management as well as information management and data management when accessing data objects in grids. When we look at collections, we see multiple interfaces where knowledge is required to be able to adequately describe relationships inherent within the grid.

- domain knowledge (relationships between domain concepts)
- structural knowledge (interpretation of field structures within a data object and the topology associated with data objects)
- procedural knowledge (workflow creation steps for digital objects)
- functional knowledge (algorithms used to extract features).

5. Applications of Knowledge-based Grids

A knowledge-based grid generalizes many of the capabilities of hierarchical storage managers, by extending storage management into a distributed system that spans arbitrary storage resources interconnected by the web. By adding data, information, and knowledge management capabilities to local storage systems, it is possible to assemble a distributed storage system. The use of grid interoperability mechanisms also make it possible to build persistent archives, that reliably store data while the underlying software and hardware technologies evolve.[10]

5.1 Network Interconnected Distributed Archives

By combining storage systems that have different media costs, different access latencies, and different I/O performance, it is possible to use the data management capabilities of a knowledge-based data grid to create a hierarchical data management system. From this perspective, a knowledge-based data grid is a hierarchical storage management system that uses autonomous storage systems to implement multiple caching levels. Through use of explicitly managed replicas, it is possible to extend archives with distributed data caches, without impacting the design of a local hierarchical storage management system. The result is a storage system that can target data caching to a local site, while managing archived data in a remote storage system.

Grid technologies greatly simplify management of data within distributed environments. The SDSC SRB relies upon the Grid Security Infrastructure to support inter-realm authentication. The Generic Security Service API (GSSAPPI) is used to map from the GSI inter-realm authentication mechanism to the local authentication environment.[9] Since the digital objects within the SDSC knowledge-based grid are owned by the data handling system, the only user IDS that must be established on the remote storage system are those under which the SRB data handling system is run. All user accesses are authorized after checking with the access-control lists maintained within the MCAT metadata catalog. The result is a system that can be extended to operate in a wide-area environment with a minimum amount of operational coordination required between the sites. Only the SRB user accounts and the amount of storage space used by the SRB need to be negotiated. Everything else is managed by the knowledge-based data grid.

Knowledge-based data grids provide a mechanism to extend archival storage systems into wide-area networks, with disaster backup copies transparently managed across multiple autonomous sites. The optimization of the data storage organization is a high priority concern of archives. Knowledge-based grids provide a way to exploit the replication of data for disaster recovery, to minimize data retrieval times. A second disaster back-up copy can be stored using an orthogonal structure to provide enhanced access to the data. An example is the LIGO experiment, which is storing data in time order (all data acquired every two seconds), versus storing data by I/O channel (all data acquired by an I/O channel for the duration of the experiment). The first ordering scheme requires that every data set be read to generate a single comprehensive time series. Similar orthogonal mechanisms for storing data images are used in remote sensing archives. Either data is stored as received by the sensor (temporally ordered), or data is reordered to provide all data sets for a particular region (spatially ordered). In knowledge-based grids, it is possible to manage data stored within a back-up archive using an orthogonal organization structure, and use retrieval mechanisms that access the optimally ordered version of the data.

Knowledge-based grids make it possible to access data stored in archives through the same data access API as data stored on disk caches. The ability to access archived data on the same basis as cached data eliminates the need to provide a separate data backup system. Knowledge-based data grids treat archived data as the original data, with replicas accessible from local disk caches. By adding versions numbers as an attribute of the archived data, original copies of digital objects can be maintained within the same system that is used to provide disaster recovery.

5.2 Persistent Archives

Every archival storage system has to manage the evolution of technology, in which the contents of the archive are migrated from old technology to new systems. Examples are migration of data objects onto new media when the shelf life of the old media expires, and integration of new versions of hardware data storage peripherals. In practice, hierarchical storage systems add new device drivers to support the new peripherals. But a major challenge still exists when an archived collection must be migrated to new software. Because knowledge-based data grids manage access to storage systems instead of to data storage peripherals, the creation of a persistent archive is greatly simplified. Knowledge-based data grids add the drivers required to talk to the new storage system software, and transparently manage replicas between the old and new software systems. From the user’s perspective, a knowledge-based data grid provides the same interface to data across all updates to the underlying software systems. The interoperability mechanisms used by data grids to tie together heterogeneous systems in space, work equally well at supporting multiple versions of storage systems over time.

The second capability needed for a persistent archive is the management of the context that is associated with archived data objects. Without a description of the object origin, the processes used to create the object, the object type, and the object relevance, digital objects will become unusable over time. A persistent archive needs to manage the collection attributes that will make it possible to discover a relevant data object at an arbitrary time in the future. Knowledge-based data grids provide the information discovery mechanisms and the process management systems that make it possible to migrate collections forward in time. The ability to manage a collection independently of the choice of information repository or database, makes it possible to migrate a collection to new database technology. By adding new drivers for the new versions of the storage and information repositories, grids make it possible to both store and discover digital objects on new technology. . Persistent archives are realized through use of grid technology that supports interactions with heterogeneous storage systems and information repositories.

6. Summary:

Knowledge-based data grids provide management systems for data, information, and knowledge. A data grid extends the capabilities of local archival storage systems by adding support for information discovery, by integrating replica management across data caches, and by handling heterogeneity in data access protocols. The interoperability mechanisms of data grids make it possible to build an archive that spans distributed storage resources, make it feasible to integrate digital libraries into the common storage infrastructure, and make it possible to construct persistent digital archives that manage the evolution of technology. The interoperability issues required for interoperation between multiple data grids are the same as the interoperability issues required to migrate grids forward in time onto new technology. The technology that is used to build a collection-based data grid makes it possible to extend archival storage systems into infrastructure that supports large-scale storage in the web.

7. Acknowledgements

There is a strong synergy between the development of data grids, digital libraries, and persistent archives. All of these research areas require the ability to manage knowledge, information, and data objects. What has become apparent, is that even though the requirements driving the infrastructure development within each community are different, a uniform architecture is emerging that meets all of the requirements.

Multiple projects are driving the development of the SDSC data handling architecture, including:

- National Science Foundation – Digital Library Initiative, Phase 2.
- National Science Foundation – National Science, Math, and Engineering Technology Education Digital Library
- National Science Foundation – National Partnership for Advanced Computational Infrastructure data grid for neuroscience brain image federation
- NASA – Information Power Grid for distributed data processing
- Department of Energy – Accelerated Strategic Computing Initiative Data Visualization Corridor for remote data visualization
- Department of Energy – Particle Physics Data Grid object replication
- National Library of Medicine – Digital Embryo Project data grid for image processing and image storage
- National Archives and Records Administration – Collection-based persistent archive

An iterative technology development cycle links all of the projects. An original DARPA project developed the data handling capabilities as part of the Distributed Object Computation Testbed. The NASA IPG project integrated the data handling technology with computational grid technology (common security environments). The NSF NPACI project integrated information management with data handling to support federation of digital libraries. The ASCI PPDG then applied the technology to support replica management across heterogeneous systems. And the NARA project applied the technology to manage migration of collections across evolving infrastructure technology.

This research has been sponsored by the National Archives and Records Administration and Advanced Research Projects Agency/ITO, "Intelligent Metacomputing Testbed", ARPA Order No. D570, issued by ESC/ENS under Contract #F19628-96-C-0020, and by the Data Intensive Computing thrust area of the National Science Foundation project

ASC 96-19020 “National Partnership for Advanced Computational Infrastructure.”

This paper presents ideas and concepts developed by the San Diego Supercomputer Center Data Intensive Computing Environment group[11], including Chaitanya Baru, Arcot Rajasekar, Michael Wan, Bertram Ludaescher, Amarnath Gupta, Richard Marciano, Ilya Zaslavsky, George Kremenek, and Bing Zhu. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

References:

1. Foster, I., Kesselman, C., “The Grid: Blueprint for a New Computing Infrastructure,” Morgan Kaufmann, San Francisco, 1999.
2. Moore, R., C. Baru, P. Bourne, M. Ellisman, S. Karin, A. Rajasekar, S. Young, “Information Based Computing,” Proceedings of the Workshop on Research Directions for the Next Generation Internet, May, 1997.
3. Baru, C., V. Chu, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, and P. Velikhov. XML-Based Information Mediation for Digital Libraries. In ACM Conference on Digital Libraries, Berkeley, CA, 1999. Exhibition program.
4. ISO/IEC FCD 13250 – Topic Maps, 1999. http://www.ornl.gov/sgml/sc34/doocument/0058.htm.
5. Baru, C., R, Moore, A. Rajasekar, M. Wan,"The SDSC Storage Resource Broker,” Proc. CASCON'98 Conference, Nov.30-Dec.3, 1998, Toronto, Canada.
6. M. D. Beynon, R. Ferreira, T. Kurc, A. Sussman, and J. Saltz. “DataCutter: Middleware for filtering very large scientific datasets on archival storage systems.” In Proceedings of the 2000 Mass Storage Systems Conference, College Park, MD, March 2000. IEEE Computer Society Press. To appear.
7. Ludäscher, B., A. Gupta, M. E. Martone. “Model-Based Information Integration in a Neuroscience Mediator System ,” demonstration track, 26th Intl. Conference on Very Large Databases (VLDB) , Cairo, Egypt, September, 2000.
8. Extensible Markup Language (XML). http://www.w3.org/XML/, 1998.
9. R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, V. Welch. “Design and Deployment of a National-Scale Authentication Infrastructure.” IEEE Computer, 33(12):60-66, 2000.
10. Moore, R., C. Baru, A. Rajasekar, B. Ludäscher, R. Marciano, M. Wan, W. Schroeder, and A. Gupta, “Collection-Based Persistent Digital Archives,” D-Lib Magazine, 6(3,4), 2000, http://www.dlib.org/
11. NPACI Data Intensive Computing Environment thurst area, http://www.npaci.edu/DICE/