Storage Resource Broker Global Data Grids

Reagan W. Moore
San Diego Supercomputer Center
moore@sdsc.edu

Arcot Rajasekar
San Diego Supercomputer Center
sekar@sdsc.edu

Michael Wan
San Diego Supercomputer Center
wan@sdsc.edu

Abstract[1]

International collaborations now manage globally distributed data collections based on the data and trust virtualization mechanisms provided by the Storage Resource Broker data grid. We examine the key requirements that have emerged from four production environments and illustrate how the desired capabilities have been implemented in the SRB.

1. Introduction.

Globally distributed data reside in a complex heterogeneous environment characterized by multiple institutions, multiple administrative domains, multiple types of storage systems, multiple types of storage access protocols, multiple types of networks, and multiple types of access clients. The data at each site are typically managed by local site administrators, using different name spaces for identification of both users and files, and are accessed by user accounts local to a particular storage system. Sharing of data requires either the movement of files into an anonymous FTP access cache, or the creation of a user account and the explicit granting of access permissions. The discovery of relevant files, the organization of the files and the access of the files require extensive interactions with the local site administrators.

A global data grid enables the creation of a shared collection that spans multiple sites [13]. Data grids automate administration of distributed files, removing the need for local site administrators to interact with remote users of the shared collection [26]. The properties of the shared collection are stored in a metadata catalog. For each registered file, the data grid manages ownership, access controls, system properties such as creation time and file size, replicas, versions, checksums, and descriptive metadata. Access to the shared data is done through the data grid, ensuring that the collection properties are updated consistently. Operations on the data that affect the collection properties are automatically tracked by the data grid. Sufficient information is retained by the data grid for synchronous or asynchronous update of the metadata.

The challenges of managing distributed data fall into four main categories:

- Control (user, group, site, community)
- Interactivity (fast access to remote data)
- Availability (fault tolerance)
- Preservation (mitigating risk of data loss)

We will examine the particular requirements that four exemplar global data grids have made for these categories, how the requirements have been met by the San Diego Supercomputer Center (SDSC) Storage Resource Broker (SRB) [25], and how generic data management

[1] The Storage Resource Broker at the San Diego Supercomputer Center. Applications of the SRB technology were done by SDSC staff Wayne Schroeder, Richard Marciano, George Kremenek, Sheau-Yen Chen, Charles Cowart, Lucas Gilbert, Bing Zhu, Arun Jagatheesan, Leesa Brieger, Sifang Lu, and Antoine de Torcy. This work was supported in part by the NSF SCI0438741 (NARA supplement), the NSF National Science Digital Library/UCAR Subaward S02-36645, the DOE Particle Physics Data Grid, the NSF National Virtual Observatory, the NIH Bio-medical Informatics Research Network, and the NSF Southern California Earthquake Center. 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 National Science Foundation, the National Archives and Records Administration, or the U.S. government.

infrastructure is able to support data grids, digital libraries, and persistent archives [22].

1. Data Grid Concepts

The Storage Resource Broker data grid is based on the idea that properties of a shared collection can be managed independently of the storage systems where the data reside [3,9]. The data grid becomes the owner of the shared collection and is the software through which all access to the shared collection are controlled. Data grids implement shared collections through the concepts of data virtualization, trust virtualization, and federation. These concepts are also the key requirements for successful management of globally distributed data.

Data virtualization [12] is achieved by having the data grid assume ownership of the name spaces used to describe the distributed data, by implementing a standard set of operations for interacting with storage systems, and by implementing a standard set of operations for supporting access mechanisms. Access thus goes through two levels of indirection, once from the chosen access client to the standard operations supported by the data grid, and again to the operations required by the storage repository. Data grids effectively map from the client access protocols to the storage repository protocols.

The logical name spaces that data grids manage are:

- Storage resource name space
- File (digital entity) name space
- Distinguished user name space
- Descriptive attribute names
- Consistency constraints

The data grid maintains the mapping from the logical file name space to the physical file name used at the remote storage system. Note that the logical file name space can be used to register any named string such as data streams, URLs, file system directories, SQL commands, and database tables. Whenever an operation is performed upon a digital entity (such as creation of a replica) the data grid stores the resulting state information (in this case the location and physical file name of the replica) in the metadata catalog.

Trust virtualization is the management of authentication, authorization, and audit trails independently of the remote storage systems. Data grids authenticate users independently of the remote storage system. Typical authentication systems include the Grid Security Infrastructure based on public key certificates, challenge-response mechanisms that authenticate based on a shared secret, and ticket mechanisms that provide access to the holder of the ticket. The data grid typically owns the data that is stored at a remote site. Users are identified by a unique name, membership in a group, and membership in a data grid (shared collection).

An access control is a constraint that is established between two of the name spaces. Thus data grids can manage access constraints on storage systems, files, and metadata. The access controls are assigned relative to specified roles such as the ability to execute data grid administrative commands, the ability to add users within a group, the ability to change metadata, the ability to turn on audit trails, the ability to read data, and the ability to change data.

The Storage Resource Broker data grid has implemented each of the above capabilities as generic distributed data management infrastructure [10]. Applications of the SRB include support for shared collections, publication of data in digital libraries, and preservation of data in persistent archives. In each application, appropriate user interfaces are ported on top of the SRB data grid. Typical uses of the environment are:

- Ability to use a preferred modern access mechanism for data stored in a legacy storage repository.
- Provision of a location-independent name space for files. As the file is moved, the logical file name does not change.

- Support for descriptive metadata to support browsing and discovery
- Ability to organize distributed data into a logical collection hierarchy

1. Globally Distributed Data

When data grid technology is used in a global environment, the architecture must implement appropriate control, interactivity, availability and preservation mechanisms. We observe that the requirements differ substantially across the multiple projects using the Storage Resource Broker, and that hierarchical mechanisms are needed for each architectural area.

1.1 Control:

We observe that communities may span multiple sites or institutions, with each site managing their own data grid. Within an institutional data grid, there may be multiple collections managed by separate projects. Within a project there may be multiple groups with different permissions on sub-collections. An individual therefore has an identity that has to be tracked at the personal level, the group level, and the data grid level. An individual is both a member of possibly multiple groups, and a member of a specific home data grid, or zone. The observed hierarchy is:

- Community – federation of data grids
- Site – a single data grid
- Project – a group of users
- Person – unique identifier

At each level of the hierarchy, there is the desire to exert independent controls. Thus a community may desire sharing of logical name spaces between federated data grids, a single site may restrict sharing to a particular collection, a project may impose access controls based on identify within a group of users, and an individual may chose to keep her own data private.

In particular, there is a desire to keep all authentication information about a user within their home grid when accessing federated data grids. A version of the Shibboleth approach is used to accomplish this. When a user name is registered into a second data grid (say zone B), the home data grid (say zone A) of the user is also registered. When that user accesses the second data grid, the location of the home data grid is identified, and an authentication request is issued to the home data grid. The zone B data grid can choose to restrict permissions of the persons that are registered from zone A. This restriction is automatically enforced for persons who have special privileges. A person who is a data grid administrator in zone A can only manage information about users from zone A within zone B.

The automation of the management of access controls is strongly desired when collection sizes approach a million files. “Sticky bits” are used to specify whether to apply automatically the access controls of the parent collection when files are added.

1.1 Interactivity:

We observe that users expect interactive response even for remote data. The data grid must therefore provide latency management mechanisms that mitigate delays imposed by:

- Finite speed of light
- Interactions with storage repositories
- Interactions with the metadata catalog

The generic approach is to minimize the amount of data that is moved for a given operation, minimize the number of messages that are required to complete the operation, and maximize the transfer rate through use of parallel streams. The mechanisms that are used are bulk operations for aggregating small files before transport, use of parallel I/O streams for moving large files, use of containers for minimizing interactions with tape archives, use of remote procedures or filters to minimize the amount of data that is transported, bulk operations for registering files into the metadata catalog, and import and export of metadata as XML files.

The challenge is the provision of these mechanisms for storage systems that reside behind firewalls. The data grid can be restricted to use a single port for interacting with a storage system. Operations that would normally use multiple ports (such as parallel I/O streams or parallel metadata registration streams) must be initiated from within the firewall. This means that each remote operation needs two versions; a server-initiated version and a client-initiated version. Each version of the protocol for managing the data transmission requires a different sequence of control messages.

Since a purpose of the data grid is to minimize the need for arcane knowledge on the part of the user (such as whether a storage system is behind a firewall), versions of the data grid commands are needed that can gracefully handle network components including:

- Firewalls
- Load levelers that redirect requests
- Private virtual networks that redirect IP addresses

A second aspect of interactivity is support for caching both data and metadata locally. For files this implies the need to support replicas on local disk. A consequence is that the data grid needs to be able to check for disk space availability when creating a replica, and be able to impose disk space quotas when creating files. Interactive response is improved by first checking whether the data is available at the IP address of the requesting client, then whether the data is on disk elsewhere, and then whether the data is on tape.

When storing data on clustered storage, interactivity is improved through load-leveling of file writes across the multiple storage systems. This implies the ability to assert collective operations on the multiple physical storage systems that are represented by a logical resource name. The required collective operations also include automated replication, and fault tolerant replication in which copies are made on a subset of the physical resources.

A slightly more sophisticated need is support for compound resources, the association of a disk cache with a tape archive, such that a request for data causes the file to be migrated to the disk cache before manipulation. The data grid must then manage the synchronization of files between the disk cache and the tape archive.

Interactive access to the metadata catalog is the most difficult requirement. Despite optimization efforts, users that access a metadata catalog across intercontinental distances notice a delay compared to local users of the database. Two approaches are pursued in data grids to improve access: support for master-slave catalogs, and support for federation of independent data grids. In both cases, a metadata catalog can be created and managed that resides at the remote users' institution.

The master-slave catalog typically assumes all writes will be done to the master catalog, but reads will be done from the slave catalog. This requires mechanisms to synchronize the slave catalog on updates to the master catalog. An advantage is that the master and slave catalogs can be implemented using different database technology.

Data grid federation enables the separation of the governance of the replicated data and metadata from the original data grid [21]. One can choose to synchronize logical name spaces between two data grids, or to manage peer-to-peer environments in which each data grid can only access public data in a peer data grid. Examples of data grid federation include:

- Master-slave federation. A slave data grid (say zone B) registers the locations of the zone B storage resources into the master data grid (zone A), giving the master data grid permission to write on the storage systems. The logical file names from zone A are registered into zone B, and the metadata from zone A is replicated into zone B. User names are also registered from zone A into zone B. Note that the users' identity still maintains a description of their home zone. The user can then access the metadata in zone B and improve their interactivity if the slave catalog is nearby.

- Preservation federation. A second data grid is established with independent logical names. Both data and metadata are read out of the first data grid through a staging mechanism, and then written into the preservation data grid. This ensures that a copy of the data exists along with the required metadata in an independent data grid.
- Central archive federation. This is similar to the preservation federation, but the synchronization of data and metadata is automated. The contents of multiple remote peer data grids are aggregated into the central archive data grid.

Depending upon the type of federation that is established between the data grids, a community can ensure that a local copy of the data is available at each institution.

The management of the control of globally distributed data now incorporates sophisticated federation schemes between data grids. Within an institution, a data grid (zone A) may be federated with a preservation data grid (zone B). At the same time, zone A may also be federated with a third zone. It is possible to build federations that mimic the behaviour of URLs. One can access the first data grid, observe which additional data grids are federated for which one has an established identity, and then access the next data grid in a chain. The chains can be open or closed, with the access from the last data grid returning the user to her home data grid.

1.1 Availability:

We observe a desire across multiple projects to ensure that if a storage resource becomes inaccessible, or if a metadata catalog is off line, or if a network link goes down, there remains a way to access the shared collection. The mechanisms used to ensure availability are similar to the mechanisms used to improve interactivity: replication and federation.

The SRB data grid automatically redirects queries that time out (due to unavailability of resources) to sites where a replica exists. Thus data that are replicated can achieve higher availability. A related management issue is the synchronization of replicas when updates are done. The SRB data grid manages both dirty bits (identifying which copy has been changed) and synchronization of the replicas with the changed copy. In practice, the user communities want not only synchronized replicas, but also time-stamped backups and versions of files.

In distributed environments, state information about data is inherently inconsistent. Thus a data grid provides mechanisms to assert a consistency property, such as validation of the integrity of files by verifying checksums, synchronization of replicas, synchronization of files in external caches with files in the shared collection, or synchronization of files in the shared collection into an external cache. The management of consistency mechanisms in a high availability environment requires that critical operations check for consistency before execution. The decision for whether to embed the checks within the data grid or to control the checks through an administrative interface depends upon the particular project.

1.1 Preservation:

The mitigation of risk against data loss is one of the benefits provided by data grids. The mechanisms for replication and federation ensure that multiple copies of data and metadata can be managed. For the preservation community, the concept of a "deep archive" is used to ensure that data will not be compromised or lost [11]. Traditionally, a "deep archive" uses an "air gap" to ensure no access by malicious users. Data is written to transportable media that is then carried to the storage system for import. With data grids, it is possible to build logical "air gaps" in which the existence of the deep archive is not observable from the external world, but manual import of data is not required.

A deep archive federates an archive data grid (zone A) with a staging data grid (zone B). An archivist from zone A is registered into zone B. A second archivist from zone B is registered into a data grid that resides in the external world. Data can then be pulled by the zone B archivist through a firewall into the staging data grid through client-initated I/O. Since no person in the external data grid has an identity registered into the staging data grid, they cannot access any data in the staging data grid. The archivist in zone A can pull data from the staging data grid into the deep archive. Again, no user in the staging data grid has an identity in the deep archive, and therefore cannot access any data within the deep archive. The result is that the deep archive has no presence in the external world, while data can still be pulled over networks under appropriate archivist control.

1. Production Global Data Grids

The Storage Resource Broker data grid is used in production at both the San Diego Supercomputer Center and by external institutions for the management of distributed shared collections. Within SDSC, the applications of the technology include real-time sensor data management, shared collections, digital libraries, and persistent archives. Table 1 lists the current projects at SDSC that have assembled shared collections. The noteworthy aspects are that the number of files in a shared collection is typically between 2 and 10 million, the size of the shared collection is now typically 10-100 Terabytes, and the number of persons with access control permissions for building the shared collections is on the order of 50 collaborators.

Date	5/17/2002		6/30/2004			3/13/2006		
Project	GBs of data stored	1000's of files	GBs of data stored	1000's of files	Users with ACLs	GBs of data stored	1000's of files	Users with ACLs
Data Grid								
NSF / NVO [18]	17,800	5,139	51,380	8,690	80	81,854	11,734	100
NSF / NPACI [16]	1,972	1,083	17,578	4,694	380	34,560	7,235	380
Hayden	6,800	41	7,201	113	178	8,013	161	227
Pzone	438	31	812	47	49	20,274	11,690	68
NSF / LDAS [7]	239	1	4,562	16	66	106,727	138	67
NSF / JCSG [6]	514	77	4,317	563	47	16,679	1,742	55
NSF / ENZO [5]			80,354	685	2,962	197,390	4,118	3,267
NIH / BIRN [4]			5,416	3,366	148	13,622	14,408	348
Digital Library								
NSF / LTER [8]	158	3	233	6	35	257	41	36
NSF / Portal	33	5	1,745	48	384	2,620	53	460
NIH / AfCS [1]	27	4	462	49	21	733	94	21
NSF / SIO [24]	19	1	1,734	601	27	2,594	1,118	27
NSF / SCEC [23]			15,246	1,737	52	166,125	3,458	73
Persistent Archive								
NARA [14]	7	2	63	81	58	2,703	1,906	58
NSF / NSDL [17]			2,785	20,054	119	5,291	50,586	136
UCSD Libraries			127	202	29	190	208	29
NHPRC / PAT [20]						1,337	516	28
TOTAL	28 TB	6 mil	194 TB	40 mil	4,635	660 TB	109 mil	5,380

Table 1. Shared collections at SDSC

Worldwide, the amount of data stored in SRB collections exceeds a petabyte. We provide a brief characterization of how the multiple capabilities discussed in Section 3 are used in production environments for the Bio-medical Informatics Research Network (BIRN), the BaBar high-energy physics experiment, the National Optical Astronomy Observatory (NOAO) data management environment, and the Worldwide Universities Network (WUNgrid).

1.1 BIRN

Control: Single data grid. One user community, use audit trails and access controls for data and metadata

Interactivity: Plans for Master-Slave across the coasts

Availability: Replicas and staged copies in local/AFS file systems.

Preservation: None. All files are on disk. Local copies are in each lab under their own administration.

This NIH funded project [4] promotes the sharing of data between researchers located at 17 research hospitals and academic institutions in the United States. The institutions are distributed between the East and West coasts, and span four time zones. The BIRN project needed extended control requirements to meet HIPAA patient confidentiality guidelines. These included the ability to restrict data location to a specified institution, access controls on files, access controls on metadata, end-to-end encryption, and audit trails of usage.

A single data grid was installed that spanned the 17 member institutions. In order to improve interactivity, the ability to federate SRB data grids was developed. However, to ensure that a master catalog retained control of all registered data, an approach based on master/slave metadata catalogs within a single data grid was developed. The choice of community control mechanisms in this case dictated that a single data grid was more appropriate than federated independent data grids.

The BIRN data grid currently manages over 13 Terabytes of data and over 14 million files.

1.2 BaBar

Control: Two data grids with two user communities. Stage data into the second data grid

Interactivity: Separate metadata catalog in each data grid

Availability: Replication of data into the Lyon data grid

Preservation: Two copies of data (SLAC and Lyon).

This DOE funded project [2,19] manages experimental data taken at the Stanford Linear Accelerator, and distributes the data to team members in multiple countries. In particular, observational data is sent from Stanford to the Institut National de Physique Nuclaire et de Physique des Particules (In2P3) in Lyon, France.

Originally the project used a single SRB data grid to manage storage resources in both Lyon and Stanford. Now two independent data grids are used. Data are staged from Stanford to a cache at Lyon, and then loaded into the Lyon data grid. The system has moved over 170 Terabytes of data from Stanford to Lyon, sustaining a Terabyte of data movement per day. With higher capacity networks, the data rate will grow in 2006 to 5 Terabytes of data per day.

1.3 NOAO

Control: Five data grids with one user community. Controls managed by each data grid.

Interactivity: Each major site runs its own data grid, insuring interactive local access.

Availability: Ability to forward requests to the data grid that is up.

Preservation: Use of one data grid for preservation.

The National Optical Astronomy Observatory [15] transports images taken from telescopes at Cerro Tololo Inter-American Observatory in Chile to NOAO headquarters in La Serena, Chile and Tucson, Arizona and to the National Center for Supercomputing Applications (NCSA). The project originally required the ability to manage a single logical name space for files that were distributed across multiple storage repositories through use of a location independent data management system.

During production in 2004, the project used a single data grid with a metadata catalog in Tucson and replicated data between storage repositories in La Serena and NCSA. Over 400,000 images were transported to the US.

To minimize the single point-of-failure inherent in a single data grid and to ensure high availability or the images, the project migrated to a modified replicated data zone model. Five independent data grids were implemented. A file transfer client was developed to manage transfers conducted with the SRB parallel I/O copy command. The client validated md5 checksums to verify successful transfer.

1.4 WUNgrid

Control:	Five data grids with separate user communities. Cross registration of user name spaces to support access in remote data grid.
Interactivity:	Use of local data grid for interactivity.
Availability:	Applications that distribute analysis tasks to data grids that are up.
Preservation:	No specific separate preservation data grid.

The Worldwide Universities Network [24] promotes academic research collaborations and the creation of shared collections in support of education. The initiative started in the United Kingdom and now also involves institutions in Europe and the United States. Each participating site installs a SRB data grid to manage local collections. Federation of data grids is then used to replicate the data for access at other institutions.

The original environment assembled five data grids located at San Diego Supercomputer Center, National Center for Supercomputing Applications, University of Bergen, University of Manchester, and University of Southampton. The name spaces for each data grid were synchronized with each of the other data grids in a star arrangement. One could traverse from one data grid to the next one, eventually returning to the home data grid.

This style of federation does not scale. Instead a hub and spoke federation is being considered, in which each institution data grid federates with the hub data grid. One can still traverse the data grids to data grids at other institutions, but the administrative load required for synchronization is minimized.

The academic research initiatives include collaborations on medieval gardens, bio-informatics, distributed data processing, and music.

1. Summary

Application requirements for managing globally distributed data can be organized based on the categories of control, interactivity, availability, and preservation. These resulting capabilities can be mapped into five functional areas related to latency management, trust virtualization, data virtualization, collection management, and federation management. These functional areas form the basis for implementing a data grid architecture. For each functional area, data grids manage logical name spaces, the set of operations that are supported on the name spaces, and the associated state information that must be updated consistently.

The capabilities provided by the Storage Resource Broker data grid are summarized in Tables 2 and 3, organized by the five functional areas. The chart lists representative capabilities, and does not include administrative functions and some of the extensions developed for specific research projects.

Latency management focuses on the minimization of the number of messages and of the amount of data sent over wide area networks to improve interactivity and scalability. Trust virtualization focuses on the management of authentication and authorization independently of the remote storage systems. Data virtualization focuses on the management of object characteristics (including naming) independently of the remote storage systems. Collection management focuses on the organization of the name spaces. Federation management focuses on the exchange of name spaces and files between data grids, making it possible to build hierarchies of globally distributed data.

Global data grids based on the Storage Resource Broker are being used as production systems for the management of distributed data collections. The example applications illustrated the use of data grids in support of data sharing environments, data transport environments, and data preservation environments. Each of the example applications built their preferred access mechanisms on top of the SRB data grid technology for their particular data management application. A noteworthy aspect is the migration of most of the projects to data grid federations to minimize single point of failure, improve interactivity, and improve availability.

Data grid federations are emerging as the preferred mechanism for implementing the control, interactivity, availability, and preservation desired in globally distributed shared collections.

	Logical naming	Standard operations	State information
Trust Virtualization	Logical user names	Add or delete user	User:Group:Zone
		GSI authentication	Certificate authority location
		Challenge-response authentication	Encrypted user password
		Issue ticket-based authentication	Time to live and number of allowed accesses
	User roles	List user roles	Curate, audit, annotate, read, write, group administration, superuser, public
		Set access control by role for user	Access controls on users
	Group names	Set access control by role for group	Access controls on groups
		Set access control on metadata for user	Access controls on metadata
		Set access control on resource for user	Access controls on resources
		Turn on audit trails	Audit trails
		Enable client-based encryption	Encryption key
		Resolve error number	System log of all accesses
Data Virtualization	Logical entity names	Define SRB physical file name structure	SRB physical file pathname structure
		Load a file into SRB collection (Sput)	Physical location where SRB stores file
		Unload a file from a SRB collection (Sget)	
	Shadow links	Register existence of external file	Location of external file
		Register existence of external directory	Location of external directory
	Logical container names	Create container	Physical file in which data is aggregated
		Create checksum	Checksum
		Verify checksum	
		Synchronize replicas	Dirty bit for writes
		Synchronize remote files with SRB files	
		Synchronize SRB files with remote files	
		Synchronize SRB files between two SRB collections	
		Posix I/O - partial read and write	Replica location
		Delete file	
		Recursive directory registration	
		Register a file as a replica of existing file	Owner, size
		Create version	Version number
		Create backup	Backup time
		Lock a file	Lock status
		Register SQL command	Data type
		Issue a registered SQL command	
		Create and issue a Datascope query	
		Register URL	

Table 2. Storage Resource Broker logical name spaces, global data manipulation operations, and global state information for the functional areas of trust virtualization and data virtualization.

	Logical naming	Standard operations	State information
Latency Management	Logical resource names	Load leveling	Quotas on storage and usage of storage
		Fault tolerant replication	Replication state
	Compound resources	File staging	Names for file system cache
		Automated access control setting	Sticky bits to inherit access controls of parent collection
		Client and server initiated parallel I/O on access	Creation time, update time
		Client and server initiated bulk file registration	
		Client and server initiated remote procedures	Location in SRB of remote procedures
		Client and server initiated bulk metadata load	
		Bulk delete - trash can	Deletion flag
		Automated checksum verification on load	
		Third party transfer	
		Store files in a logical container	
Collection Managment	Descriptive metadata	Extensible metadata	Descriptive metadata for SRB file
	Collection hierarchy	Create/delete subcollection	Parent collection identity
		Create collection metadata	Descriptive metadata for SRB collection
		Extensible schema	Table structure of metadata
		Create soft link between two logical files	Soft link
		Import of XML files	
		Export of XML and HTML files	
		Remote template-based metadata extraction	Location in SRB of templates
		Synchronize slave catalog with master catalog	Location of slave catalog
		Queries on descriptive and state information	
Federation Management	Distinguished zone names	Access zone authority to register zone name	Zone name and port number
	Zone authority name	User authentication by home zone	
		Cross-registration of resources between zones	
		Synchronization of user names between zones	
		Synchronization of file names between zones	
		Synchronization of metadata between zones	

Table 3. Storage Resource Broker logical name spaces, global data manipulation operations, and global state information for the functional areas of latency management, collection management and federation management.

6. References

1. AFCS – Alliance for Cell Signaling, http://www.afcs.org
2. Babar – http://www.slac.stanford.edu/BFROOT/
3. Baru, C., R, Moore, A. Rajasekar, M. Wan, "The SDSC Storage Resource Broker," Proc. CASCON'98 Conference, Nov.30-Dec.3, 1998, Toronto, Canada.
4. BIRN – The Biomedical Informatics Research Network, http://www.nbirn.net.
5. ENZO – Cosmological simulation code, http://cosmos.ucsd.edu/enzo/
6. JCSG – Joint Center for Structural Genomics, http://www.jcsg.org/
7. LDAS – NASA Land Data Assimilation System, http://ldas.gsfc.nasa.gov/.
8. LTER, US Long Term Ecological Research network, http://lternet.edu/
9. Moore, R., M. Wan, A. Rajasekar, "Storage Resource Broker: Generic Software Infrastructure for Managing Globally Distributed Data", Proceedings of IEEE Conference on Globally Distributed Data, Sardinia, Italy, June 28, 2005.
10. Moore, R., A. Rajasekar, M. Wan, "Data Grids, Digital Libraries and Persistent Archives: An Integrated Approach to Publishing, Sharing and Archiving Data", Special Issue of the Proceedings of the IEEE on Grid Computing, Vol. 93, No.3, pp. 578-588, March 2005.
11. Moore, R., "Preservation Environments," NASA / IEEE MSST2004, Twelfth NASA Goddard / Twenty-First IEEE Conference on Mass Storage Systems and Technologies, April 2004
12. Moore, R., C. Baru, "Virtualization Services for Data Grids", Book chapter in "Grid Computing: Making the Global Infrastructure a Reality", pp. 409-436, John Wiley & Sons Ltd, 2003.
13. Moore, R., C. Baru, A. Rajasekar, R. Marciano, M. Wan: Data Intensive Computing, In ``The Grid: Blueprint for a New Computing Infrastructure'', eds. I. Foster and C. Kesselman. Morgan Kaufmann, San Francisco, 1999.
14. NARA Persistent Archives project, http://www.sdsc.edu/NARA/
15. NOAO – National Optical Astronomy Observatory, http:// www.noao.edu/
16. NPACI Data Intensive Computing Environment thrust area, http://www.npaci.edu/DICE/
17. NSDL – National Science Digital Library, http://www.nsdl.org/
18. NVO – National Virtual Observatory, http://www.us-vo.org/
19. Particle Physics Data Grid, http://www.ppdg.net/
20. PAT – Persistent Archives Testbed, http://www.sdsc.edu/PAT/
21. Rajasekar, A., M. Wan, R. Moore, W. Schroeder, "Data Grid Federation", PDPTA 2004 - Special Session on New Trends in Distributed Data Access, June 2004.
22. Rajasekar, A., Michael Wan, Reagan Moore, Arun Jagatheesan, George Kremenek, "Real Experiences with Data Grids - Case studies in using the SRB", International Symposium on High-Performance Computer Architecture, Kyushu, Japan, December, 2002.
23. SCEC – Southern California Earthquake Center community digital library, http://www.sdsc.edu/SCEC/
24. SIO Explorer Digital Library Project to provide education and research material from oceanographic voyages in collaboration with NSDL, http://nsdl.sdsc.edu/.
25. SRB - "The Storage Resource Broker Web Page", http://www.sdsc.edu/srb/
26. Stockinger, H.,O. Rana, R. Moore, A. Merzky, "Data Management for Grid Environments," European High Performance Computing and Networks Conference, Amsterdam, Holland, June 2001.
27. WUN – Worldwide Universities Network, http://wungrid.org/