Reagan W. Moore, SDSC
May 4, 2005

Operations for Access, Management, and Transport at Remote Sites

Status of This Memo: Informational

This memo provides information to the Grid community specifying the types of operations needed for access to remote data. Distribution is unlimited.

Copyright Notice

Copyright © Global Grid Forum (2005). All Rights Reserved.

Abstract

Remote data access can be viewed as simply the “get” or “put” of files. However, based upon experience with Data Grids, digital libraries, and persistent archives that access remote data, additional remote operations have been identified. This paper quantifies five types of operations that are performed at remote storage repositories, and illustrates each of the types of operations from examples based upon production systems. The transport of data often involves the coordinated execution of the specified remote operations, and thus the operations can be viewed as extensions to standard data transport. This report does not summarize types of operations across all existing data management systems. Instead, an attempt is made to point out operations that have proven useful in data management.

Contents

Abstract 1
1. Transport Operations on Remote Data 2
1.1 Categories of extended transport operations 3
1.2 GGF Research Groups 3
2. Operations 4
2.1 Byte level access 4
2.2 Latency management mechanisms 4
2.3 Remote Execution of Functions 6
2.4 Protocol conversion 7
2.5 Administrative tasks 8
3. Implementations 8
4. Summary 10
5. Author Information 10
6. Acknowledgements 11
7. Glossary 11
7.1 Data Grid terms for a logical name space 11
7.2 Data Grid terms for a storage repository abstraction 12
7.3 Data Grid terms for an information repository abstraction 12
7.4 Data Grid terms for a distributed resilient scalable architecture 13
7.5 Data Grid terms for a virtual Data Grid 13
8. Intellectual Property Statement 14
9. Full Copyright Notice 14
10. References 14

1. Transport Operations on Remote Data

Access to remote data involves aspects of data management, data manipulation, and data transport. From the perspective of differentiated grid services, one would like to be able to implement each type of data operation as a separate service, and then apply the services sequentially. The services would generate a dataset or file, and would then transport the file using GridFTP. We examine the conjecture that remote data operations may need to be combined with the file transport mechanisms to improve performance.

This informational paper describes five types of remote operations, and examines the impact of combining remote operations with data transport. Each type of operation is illustrated with examples from production systems. Based upon experiences with Data Grids, digital libraries, and persistent archives, access to remote data in file systems can involve additional coordinated operations beyond those of the “get” or “put” of files [13]. Access to other types of storage repositories such as databases also leads to the requirement for additional types of remote operations. In particular, the Data Access and Integration Services Working Group of the Global Grid Forum is developing standards recommendations for data access and integration based on the Open Grid Services Architecture. While the DAIS specification depends on transport mechanisms for delivering data to clients and third parties, the DAIS group has not defined extended transport operations to improve service performance.

The key concept behind the integration of remote operations with transport is the observation that support for remote data access requires the installation of a server at each storage repository that is accessed [12]. The server manages the transport protocol, converts from the transport protocol to the access protocol required by the local storage system, coordinates authenticity information, maps from the Unix identifier under which the data is stored to the Unix identifier of the requestor, applies access controls lists, etc. In addition to these access management operations, the remote server is also responsible for packaging the data before movement. This can be as simple as aggregation into streams of buffered data that are then transported. But in production data management systems, the server can also be required to do additional manipulations upon the data. The additional data manipulations constitute extended transport operations that can be implemented in the server that manages data movement from remote storage systems.

The extended transport operations identified in this document are supported by production systems [40, 20, 25], or are being developed in research Data Grids. To provide usage examples, nineteen data management system implementations are discussed, covering Data Grids for the sharing of data across administrative domains, digital libraries for publishing data and supporting web services, and persistent archives for managing technology evolution. The extended transport operations needed by these three types of data management systems have many common components. The common components can be organized into five basic categories of extended transport operations.

The objective of this document is to:

- Identify types of operations executed during remote data access
- Organize the extended transport operations into categories
- Survey the range of extended transport operations currently in use on representative production Data Grids

This report does not summarize types of operations across all existing data management systems. Instead, an attempt is made to point out operations that have proven useful in end-to-end data management applications.

1.1 Categories of extended transport operations

Five basic categories of extended transport operations are collectively required by Data Grids, digital libraries, and persistent archives.

1. Byte level access
2. Latency management mechanisms
3. Remote execution of functions
4. Heterogeneous system access
5. Administrative tasks

Byte level access transport operations correspond to the standard operations supported by Unix file systems. Latency management transport operations are typically operations that facilitate bulk data and bulk metadata access and update. The remote execution of functions provides the ability to process data directly at the remote storage system. Transport operations related to access of heterogeneous systems typically involve protocol conversion and data repackaging. Administrative tasks involve access to information catalogs to either control the transfer or manage the name space under which the digital entities are referenced. Each of these categories of extended transport operations is examined in more detail in section 2.

1.2 GGF Research Groups

Multiple groups within the Global Grid Forum are addressing the issue of defining the sets of extended operations that should be performed on remote storage resources:

- Persistent Archive Research Group is defining preservation operations, such as checksums, digital signatures, migration, all of which can be invoked at the remote storage system as a component of the transport operation.
- Open Grid Services Architecture - Data Access and Integration Services Working Group (OGSA-DAI) is defining a set of operations that can be applied across both databases and file systems, as Data Set Services. This includes manipulation of records in databases, and formatting of query results before transport.
- Grid File System Birds of a Feather session is defining operations on logical name spaces, and is mapping these operations to actions performed at remote file systems.
- Data Format Description Language Working Group is defining operations on digital entities, which may be executed as remote processes invoked through remote data access mechanisms

It is possible to build an environment in which all manipulations of data are performed with the result stored at the remote storage site, and then a separate request is issued to transmit the result [6,9,10]. This assumes space is available at the remote storage system to save the intermediate results. When dealing with very large data sets, this may be impractical, with partial transmission of results interspersed with the process of generating the intermediate data. At the other extreme, when dealing with small data sets, it may be much faster to make a bulk request to the remote storage system, rather than multiple individual requests. Each request incurs latency, whether in the wide area network, or within the storage repository itself. By making a single request for multiple small data sets, the performance can be substantially improved.

For large data sets, such that the size is greater than the product of the access latency and the bandwidth, the additional messages that are transmitted when data formatting commands are sent separately from data transport commands, do not impact the performance. However, when the data set size is smaller than the bandwidth*delay product (that is, the product of the bandwidth and the access latency), performance can be degraded by arbitrarily large factors. All of the extended transport operations are intended to improve the performance of data access and data manipulation over wide area networks. The extended operations either reduce the number

of messages that must be sent to achieve a result, or migrate operations to the location where they can be performed optimally.

The multiple Global Grid Forum research groups are effectively defining the set of extended transport operations that improve the performance of their particular service.

2. Operations

The types of operations that will be considered are focused on file and aggregated file level operations. For each of the major categories of extended transport operations, we provide explicit examples of the capabilities that are in use in production data management systems.

2.1 Byte level access

The traditional Unix file system operations include:

- creat(), open(), close(), unlink()
- read(), write(), seek(), sync()
- stat(), fstat(), chmod()
- mkdir(), rmdir(), opendir(), closedir(), readdir()
- chown(), chdir()

Additional file system operations are being developed in Data Grids that provide directory manipulation:

- rewinddir – reset directory handle to the first entry in a directory
- seekdir – set position for next read of a directory
- telldir – get current seek pointer on a directory handle
- scandir – scan a directory and return the list of files specified by a comparison function

The ability to apply these operations directly at the remote storage system is one of the design goals of the Grid File System Research Group. In particular, the ability to read and write is needed for partial file reads at the remote site. Partial file reads make it possible to retrieve a subset of a file, especially important when dealing with very large files. The ability to seek is needed by paging systems for visualization (such as the San Diego Supercomputer Center 3D Visualization Toolkit). The ability to synchronize (sync) is needed when manipulating containers of files, and staging files from archives to disk.

The ability to list and modify the remote directory structure is needed when manipulating remote collections that contain millions of files. The performance of the remote storage system depends upon the number of physical files within a directory. While one can map all logical names to a single physical directory, the performance of physical file systems improves when the logical names are mapped to physical names distributed across multiple physical directories.

The remote server that implements the Unix file system operations can be the same server that is also used to support the transmission of the results of the operations over the wide area networks.

2.2 Latency management mechanisms

Explicit latency management mechanisms are used to manage and manipulate large numbers of files stored at remote sites. The operations involve some form of aggregation, whether of data into containers, metadata into XML files, or I/O commands into execution of remote processes. The operations also may invoke a mechanism that will improve future operations, such as staging of files onto faster media. The following latency management functions are in production use in Data Grids:

- Bulk registration of files
- Bulk data load
- Bulk data unload
- Aggregation into a container
- Extraction from a container
- Staging, required by the Hierarchical Resource Manager
- Status, required by the Hierarchical Resource Manager

Each of these operations requires the execution of a process at the remote storage system, which is invoked simultaneously with the transport request. Again the maximum performance improvement is seen when dealing with small files.

Registration is the process of recursively processing a physical file system directory and creating corresponding logical file records in the Data Grid metadata catalog [18]. The information recorded in each logical file record can include the physical file name, length, registration date, owner, physical source location, etc. The information also includes administrative metadata that is needed to map from the logical file name to the physical file name. The physical file system directory structure can be replicated within the logical name space, and the logical file name can be set equal to the physical file name. This makes it possible to register an existing directory structure into the logical name space using a similar organization of the files. The user of the system can also choose to register the physical files into an entirely different organizational structure in the logical name space, using logical file names that are different from the original physical file names.

Bulk registration corresponds to packaging the file system metadata before transmission over the network, and then bulk import of the metadata into the logical name space catalog. Standards have been developed in the digital library community for the organization of the metadata. The Metadata Encoding Transmission Standard (METS) is used to aggregate metadata for bulk movement [19]. The registration process implements one aspect of consistency management for associating administrative metadata with logical file names [20]. The METS schema is encoded in XML [49], as a standard syntax for annotating metadata. The encoding in XML and organization into a METS schema take place at the remote storage system before transmission of the file system metadata over the wide area network, and can be done by the same server as used for data transmission.

Data load is the process of registration of the physical file into the logical name space, and the import of the file onto a storage system under the control of the Data Grid. Thus both metadata registration and data movement are needed. When dealing with small files, it is much faster to aggregate the small files before transmission. This can be done by explicit use of containers, physical aggregations of data that are managed by the Data Grid. The files are written into the container before transmission, and the container is stored as an entity within the Data Grid. Small files can also be aggregated for transport without storing the aggregation into the Data Grid. When the files reach the remote storage system, they are stored as independent files. The fastest mechanism in practice for dealing with small files is the explicit aggregation of the files into containers that are then managed by the Data Grid.

Data unload is the export of files from the Data Grid and their movement to the requesting application. Again, when dealing with small files, it is faster to move containers of data. The application then needs an index into the container for the extraction of individual files. In this case, data transport is the coordinated movement of the container and the associated XML file that defines the bitfile offsets of the multiple files stored within the container.

When containers are created, the operations that load the files into the container are performed at the remote storage repository. Similarly, when files are accessed, they may be extracted form

the container, again at the remote storage system, with just the individual file transmitted to the requestor.

Staging and status operations are needed for interactions with resource managers that reorder data access requests. A staging command is issued to request the movement of a file from an archive onto a disk cache. A status command is issued to check whether the staging request has been completed.

2.3 Remote Execution of Functions

Commercial file system providers are examining the ability to support the execution of functions within file systems under the name of object oriented storage [35]. The idea is that the file system can support operations at the object level rather than the block level. Object level manipulations would be implemented through execution of defined functions on the files. This concept is already supported by database vendors.

The central idea behind remote execution of functions is that low complexity operations (defined as a sufficiently small number of operations per byte moved) should always be performed at the remote storage repository to decrease the total time for access and manipulation. Conversely, high complexity operations (a sufficiently large number of operations per byte moved) should always be performed at the most powerful computer that is available. The exact conversion point depends upon the type of data movement that is being supported (streaming, pipelining), the load on the systems, the amount of data reduction that could be achieved, the complexity of the transport mechanism, the ratio of the execution rate and the product of the transmission bandwidth and the operation complexity (rate / (bandwidth*complexity)), and the relative execution rates. Object oriented access takes advantage of the ability of object oriented storage systems to perform appropriate processing steps directly on the remote storage repository.

Example operations that are can be executed more efficiently on the remote storage system include:

- Metadata extraction from files – this typically extracts a few hundred bytes from a file. Unless the file only contains metadata, the operation is best performed at the remote storage repository.
- Extraction of a file from a container – if the file size is much smaller than the container size, the extraction should be done at the remote storage repository.
- Validation of a digital signature – If the local access bandwidth is greater than the wide area network bandwidth, the validation should be done at the remote storage repository.
- Data subsetting – this is similar to reading data out of a container, and should be done at the remote storage system when the data subset is small [4].
- Data filtering – this is similar to data subsetting, but also consists of decisions that are made during the filtering process [4]. When the result set is small, the process is better done at the remote storage system.
- Server initiated parallel I/O streams – the decision on the number of I/O streams to use when sending data in parallel over a wide area network depends strongly on the number of independent resources from which the data can be accessed. This is typically only known by the remote storage system. When data filtering is involved, the transport decisions and filtering results have to be coordinated. In the case of access to very large files that are filtered and then streamed to another storage system, the filtering is an active part of the process and controls the number of I/O streams.
- Checksum checking – on storage it can be worth checking that data was transmitted correctly, and on transmission, it may be worth checking that data has not been corrupted while being stored [11].
- Encryption as a property of the data file. For biomedical data, all data transmissions of personal data must be encrypted. The encryption process must be invoked at the remote storage system on every file transfer.

- Compression as a property of the data file. The decision to compress typically depends upon the bandwidth of the final network leg. The compression takes place at the remote storage repository to guarantee that the network over which the data is sent can handle the load.

An extension of the concept of remote execution of functions is the automated conversion of the encoding format of the digital entity to a desired encoding format. The Data Format Description Language Research Group is developing mechanisms to characterize the structure of digital entities, the semantic labels that are applied to the structures, and the operations that can be performed upon the structures. The characterizations correspond to digital ontologies that can be applied at the remote storage repository during access [22]. Preservation environments depend upon the ability to migrate digital entities to new encoding formats to ensure the ability to display archived material [22, 23, 41, 42, 43, 45]. The digital ontologies describe the structural relationships present within the digital entity, usually expressed using the Resource Description Framework syntax [37].

2.4 Protocol conversion

File based access, such as that provided by GridFTP, assumes that a dataset can be generated by the remote storage system and then transported using the File Transfer Protocol. A form of extended transport operations occurs when the remote storage system that is being accessed does not provide standard Unix file system operations. In these cases, the data must be manipulated into a suitable form for transport, or the access mechanism must be modified to work with the protocol used by the remote storage system. The following additional types of storage systems are being accessed by production data grids:

- Database blob access – support reading and writing of blobs in databases
- Database metadata access – support aggregation of query results into an XML file before transport
- Object ring buffer access – support queries on the object in the ring buffer, and return only the objects that satisfy the query, while aggregating the objects into a single file.
- Archive access – manage archive access requirements such as server-initiated parallel I/O. In this case the remote storage system determines the optimal number of I/O streams to use.
- Hierarchical Resource Manager access – support staging requests and status requests for the placement of data within the remote storage system.
- Preferred API – support access methods such as Python, Java, C library, Shell command, Open Archives Initiative, Web Services Description Language, Open Grid Services Architecture, http, Dynamic Load Libraries, GridFTP. In this case, the transport mechanism that delivers the data is determined by the access mechanism. At some point in the transfer, the transport system will have to convert from the transport protocols of the remote storage system to the buffering scheme required by the chosen access method.

The DAIS working group has defined a set of services for access to database management systems based on administrative tasks (publish, subscribe, propagate, consume) and operational tasks (createConsumption, alterConsumption, startConsumption, stopConsumption, dropConsumption, publishData, deliverData, deliverEvent, getData). The result sets can often be subjected to Unix file like operations in relation to transport. However, there are still cases where additional operations can be integrated with the data transport:

- Asynchronous application of SQL commands, with results delivered under propagation rules. For large result sets, multiple partial result sets may need to be delivered.
- Asynchronous delivery of results from DAIS patterns executed at a remote site, with transport to an intermediate site for joins across result sets before delivery to a third party.
- Specification of a workflow with interaction between the steps requiring joins across result

sets.

An important aspect of preservation environments is the ability to manage access to multiple types of storage repositories. In particular, when new technology is developed, a persistent archive needs the ability to migrate data from the old technology to the new technology. This will require the ability to interoperate with multiple storage repository protocols while moving data. If third party transport is used, then the protocol conversion takes place entirely within the transport mechanism. In addition, preservation systems use a standard data encoding format, called an Archival Information Package (AIP), to encapsulate preservation metadata with each digital entity. AIPs are defined in the Open Archival Information System standard [33, 34]. On transport, elements of the AIP may need to be updated to reflect the new storage location, during the transport. On third party transport of AIPs, the update will take place within the transport mechanism.

2.5 Administrative tasks

A fruitful area for discussion in future documents is the integration of distributed administrative tasks with transport operations. Examples of such tasks include:

- Data replication – should the transport mechanism make the decision for whether to replicate data? Current schemes use an access history to decide when performance can be improved by replication. If the access history is maintained at the remote site, the transport mechanism can check the frequency of access versus the location of the requestor, and automate the replication of the file to a closer resource.
- Archive/restore functions – should the transport mechanism force the archiving of less frequently used data to make room for the current transfer? The restore function is equivalent to a staging request to a hierarchical resource manager, and may also be implemented within the transport mechanism.
- Data transformation and translation – should the transport mechanism enforce the conversion of binary objects into the binary encoding format used by the receiving operating system? An example is the implementation of support for the External Data Representation Standard (XDR) within the transport protocol [48].
- Data integration in a distributed environment – should the mapping from a logical name space to physical file names occur as part of the transport protocol? An example is the specification of a logical file name for a transport operation, instead of providing a physical file name.
- Data integration between federated name spaces – should the mapping between logical name spaces in multiple Data Grids occur as part of the transport protocol? An example is the automated forwarding of a transport request to the Data Grid that is managing the desired digital entity.
- Generating load average – should the transport mechanism access time dependent host information? An example is the Grid Datafarm which supports generation of load average [11].

3. Implementations

We examine nineteen grid environments funded by multiple US federal agencies and educational institutions, and in use in multiple international projects. The grid environments represent multiple scientific disciplines and education projects. The grid environments are used to support applications ranging from data sharing [20-21] to data publication [3,20] to data preservation 22,24]. Despite the differences in motivation, scientific discipline, and governing institution the grid environments were built upon common grid infrastructure. This can be viewed as a success for the concept behind grids: common infrastructure can be used to support application specific requirements. The categories of remote operations presented in this paper are in use in at least one production grid environment.

The implementations are based on the Globus toolkit [10], Condor [7], and the SDSC Storage Resource Broker [2]. The extended transport operations that are used are listed for each project. Note that all of the implementations specify transport operations through use of logical file names.

- National Aeronautics and Space Administration (NASA) Information Power Grid – "traditional" Data Grid [29]. Bulk operations are used to register files into the Grid. Containers are used to package (aggregate) files before loading into an archive. Transport operations are specified through logical file names.
- NASA Advanced Data Grid – Data Grid [26]. Bulk operations are used to register files and metadata.
- NASA Data Management System/Global Modeling and Assimilation Office – Data Grid [27]. Containers are used for interacting with archives. The logical name space is partitioned across multiple physical directories to improve performance.
- NASA Earth Observing Satellite – Data Grid [28]. Read and write operations are supported against a "WORM" file system. This means that all updates cause a new version to be written.
- Department of Energy (DOE) Particle Physics Data Grid (PPDG) / BaBar high energy physics experiment – Data Grid [36]. Bulk operations are used to register files, load files into the Data Grid, and unload files from the Data Grid. A bulk remove operation has been requested to complement the bulk registration operation. Staging and status operations are used to interact with a Hierarchical Storage Manager.
- Japan/High Energy Research Accelerator Program (KEK) – Data Grid [15]. Bulk operations are used to register files, load files, and unload files.
- National Virtual Observatory (NVO)/United States Naval Observatory-B – Data Grid [31, 47]. Registration of files is coordinated with the movement of Grid Bricks. Data is written to a disk cache locally (Grid Brick). The Grid Brick is physically moved to a remote site where bulk registration and bulk load are invoked on the Grid Brick to import the data into the Data Grid.
- National Science Foundation (NSF)/National Partnership for Advanced Computational Infrastructure – Data Grid [32]. Containers are used to minimize the impact on the archive name space for large collections of small files. Remote processes are used for metadata extraction. Data subsetting is done through use of DataCutter remote filters [4]. The seek operation is used to optimize paging of data for a 4D visualization rendering system. Data transfers are invoked using server-initiated parallel I/O to optimize interactions with the HPSS archive. Bulk registration, load and unload are used for collections of small data. Results from queries on databases are aggregated into XML files for transport.
- National Institute of Health/Biomedical Informatics Research Network – Data Grid [5]. Encryption and compression of data are managed at the remote storage system as a property of the logical name space. This ensures privacy of data during transport.
- Library of Congress – Data Grid [17]. Bulk registration is used to import large collections of small files.
- NSF/ Real-time Observatories, Applications, and Data management Network (Roadnet) – Data Grid [38]. Queries are made to object ring buffers to obtain result sets.
- NSF/Joint Center for Structural Genomics – Data Grid [16]. Parallel I/O is used to push experimental data into remote archives, with data aggregated into containers.
- NVO/2-Micron All Sky Survey – digital library [1, 31]]. Containers are used to organize five million images. An image cutout service is implemented as a remote process, executed directly on the remote storage system. A metadata extraction service is run as a remote process, with the metadata parsed from the image and aggregated before transfer.
- NVO/Digital Palomar Observatory Sky Survey – digital library [8]. Bulk registration is used to register the images. An image cutout service is implemented as a remote process, executed directly on the remote storage repository.
- NSF/Southern California Earthquake Center – digital library [39]. Bulk registration of files is used to load simulation output files into the logical name space (1.5 million files

generated in a simulation using 3000 time steps).

- National Archives and Records Administration (NARA) - persistent-archive [46]. Bulk registration, load, and unload are used to access digital entities from web archives. Containers are used to aggregate files before storage in archives. Transport operations are automatically forwarded to the appropriate Data Grid for execution through peer-to-peer federation mechanisms.
- NSF/National Science Digital Library (NSDL) - persistent-archive [30]. Bulk registration, load, and unload are used to import digital entities into an archive. Web browsers are used to access and display the imported data, using http.
- University of California San Diego – persistent archive [44]. Bulk registration, load, and unload are used to import digital entities into an archive.

The GridFTP transport system incorporates extended operations beyond the traditional “get” and “put” of the original FTP mechanism [13]. The extended operations include:

- Partial file access. The read and write operations are supported for reading parts of files and for modifying parts of files.
- Parallel I/O. Data is sent using multiple I/O streams to the requestor.
- Guaranteed data transmission. Data transport is restarted as needed across all interruptions.

The GridFTP remote operation mechanisms for partial file access and parallel I/O are used in the Particle Physics Data Grid, the National Virtual Observatory, the Information Power Grid, the Southern California Earthquake Center, and other projects.

4. Summary

Five major categories of extended transport operations have been identified. For each category, an example production Data Grid has been identified, along with the particular extensions that are used. To improve the analysis, comparisons with additional data management systems is warranted to decide whether important extensions have been overlooked. In reality, multiple projects are now facing the challenge of deciding where computations should take place within the grid. The manipulations can take place at the remote site where the data is stored [4], or the data can be transported to the location where it is processed, as in the Grid Physics Network’s [14] use of Chimera. Of interest to the Data Transport Working Group, is an understanding of when transport and remote manipulation need to be combined, with partial transfer of results as the data is generated.

5. Author Information

Reagan W. Moore
San Diego Supercomputer Center (SDSC)
9500 Gilman Drive, MC-0505
La Jolla, CA 92093-0505]
moore@sdsc.edu

6. Acknowledgements

The results presented here were supported by the NSF NPACI ACI-9619020 (NARA supplement), the NSF NSDL/UCAR Subaward S02-36645, the DOE SciDAC/SDM DE-FC02-01ER25486 and DOE Particle Physics Data Grid, the NSF National Virtual Observatory, the NSF Grid Physics Network, and the NASA Information Power Grid. The views and conclusions contained in this document are those of the author 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.

7. Glossary

The terminology used in this report is based upon Data Grid semantics that has been in use within the Global Grid Forum. We define terms related to logical name spaces; storage repository abstractions; information repository abstractions; distributed, resilient, scalable architecture; and virtual Data Grids. References for the terms may be found in papers on the Storage Resource Broker [2-3, 20-24], the Globus Toolkit [9, 10], the original book on Grid Computing [25], research projects such as the Grid Physics Network [14], and a recent book on Grid Computing [21].

7.1 Data Grid terms for a logical name space

A logical name space is a naming convention for labeling digital entities. The logical name space is used to create global, persistent identifiers that are independent of the storage location. Within the logical name space, information consists of semantic tags that are applied to digital entities.

Metadata consists of the semantic tags and the associated tagged data, and is typically managed as attributes in a database. Metadata is called data about data.

Collections organize the metadata attributes that are managed for each digital entity that is registered into the logical name space

Registration corresponds to adding an entry to the logical name space, creating a logical name and storing a pointer to the file name used on the storage system.

The logical name space can be organized as a collection hierarchy, making it possible to associate different metadata attributes with different sets of digital entities within the collection. This is particularly useful for accession, arrangement, and description.

Logical folders within a collection hierarchy represent sub-collections, and are equivalent to directories in a file system, but are used to manage different sets of metadata attributes.

Soft links represent the cross registration of a single physical data object into multiple folders or sub-collections in the logical name space

Shadow links represent pointers to objects owned by individuals. They are used to register individual owned data into the logical name space, without requiring creation of a copy of the object on storage systems managed by the logical name space.

Replicas are copies of a file registered into the logical name space that may be stored on either the same storage system or on different storage systems.

Collection-owned data is the storage of digital entities under a Unix user ID that corresponds to the collection. Access to the data is then restricted to a server running under the collection ID.

User access is accomplished by authentication to the Data Grid, checking of access controls for authorization, and then retrieval of the digital entity by the Data Grid from storage through the collection ID for transmission to the user.

7.2 Data Grid terms for a storage repository abstraction

A storage repository is a storage system that holds digital entities. Examples are file systems, archives, object-relational databases, object-oriented databases, object ring buffers, FTP sites, etc.

A storage repository abstraction is the set of operations that can be performed on a storage repository for the manipulation of data.

A container is an aggregation of multiple digital entities into a single file, while retaining the ability to access and manipulate each digital entity within the container.

Load balancing within a logical name space consists of distributing digital objects across multiple storage systems

Storage completion at the end of a single write corresponds to synchronous data writes into storage.

Third party transfer is the ability of two remote servers to move data directly between themselves, without having to move the data back to the initiating client

Metadata about the I/O access pattern is used to characterize interactions with a digital entity, recording the types of partial file reads, writes, and seeks.

Synchronous updates correspond to finishing both the data manipulations and associated metadata updates before the request is completed.

Asynchronous updates correspond to completion of a request within the data handling system, after the return was given to a command.

Storage Resource Managers control the load on a Hierarchical Resource Manager or disk file system. They rearrange the submitted work load to optimize retrieval from tape, stage data from the HRM to a disk cache, and manage the number of allowed simultaneous I/O requests.

7.3 Data Grid terms for an information repository abstraction

An information repository is a software system that is used to manage combinations of semantic tags (attribute names) and the associated attribute data values. Examples are relational databases, XML databases, Lightweight Directory Access Protocol servers, etc.

An information repository abstraction is the set of operations that can be performed on an information repository for the manipulation of a catalog or collection.

Template based metadata extraction applies a set of parsing rules to a document to identify relevant attributes, extracts the attributes, and loads the attribute values into the logical collection.

Bulk metadata load is the ability to import attribute values for multiple objects registered within the logical name space from a single input file.

Curation control corresponds to the administration tasks associated with creating and managing a logical collection

7.4 Data Grid terms for a distributed resilient scalable architecture

Federated server architecture refers to the ability of distributed servers to talk among themselves without having to communicate through the initiating client.

GSI authentication is the use of the Grid Security Infrastructure to authenticate users to the logical name space, and to authenticate servers to other servers within the federated server architecture

Dynamic network tuning consists of adjusting the network transport protocol parameters for each data transmission to change the number of messages in flight before acknowledgements are required (window size) and the size of the system buffer that holds the copy of the messages until the acknowledgement is received.

SDLIP is the Simple Digital Library Interoperability Protocol. It is used to transmit information for the digital library community

7.5 Data Grid terms for a virtual Data Grid

The automation of the execution of processes is managed in virtual Data Grids. References to the result of a process can result in the application of the process, or direct access to the result.

Knowledge corresponds to relationships between attributes, or to relationships that characterize properties of a collection as a whole. Relationships can be cast as inference rules that can be applied to digital entities. An example is the set of structural relationships used to parse metadata from a digital entity in metadata extraction.

The application of processes at remote storage systems is accomplished through systems such as the DataCutter, a data filtering service developed by Joel Saltz at the Ohio State University, which is executed directly on a remote storage system [4].

Transformative migrations correspond to the processing of a digital entity to change its encoding format. The processing steps required to implement the transformative migration can themselves be characterized and archived, and then applied later.

Digital ontologies organize the set of semantic, structural, spatial, temporal, procedural, and functional relationships that are present within a digital entity. The digital ontology specifies the order in which the relationships need to be applied in order to correctly display or manipulate the digital entity.

Derived data products are created by execution of processes under the control of a virtual Data Grid. For persistent archives, derived data products can be data collections or transformative migrations of digital entities to new encoding formats. A data collection can be thought of as a derived data product that results from the application of archival processes to a group of constituent documents.

8. Intellectual Property Statement

The GGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the GGF Secretariat.

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

9. Full Copyright Notice

Copyright (C) Global Grid Forum (2005). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the GGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the GGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the GGF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE GLOBAL GRID FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE."

10. References

1. 2-Micron All Sky Survey (2MASS), http://www.ipac.caltech.edu/2mass/
2. Baru, C., R, Moore, A. Rajasekar, M. Wan, "The SDSC Storage Resource Broker," Proc. CASCON'98 Conference, Nov.30-Dec.3, 198, Toronto, Canada.
3. Baru, C., R. Moore, A. Rajasekar, W. Schroeder, M. Wan, R. Klobuchar, D. Wade, R. Sharpe, J. Terstriep, (198a) "A Data Handling Architecture for a Prototype Federal Application," Sixth Goddard Conference on Mass Storage Systems and Technologies, March, 198.
4. Beynon, M.D., T. Kurc, U. Catalyurek, C. Chang, A. Sussman, and J. Saltz. ``Distributed Processing of Very Large Datasets with DataCutter". Parallel Computing, Vol.27, No.11, pp. 1457–1478, 2001.
5. Biomedical Informatics Research Network, http://nbirn.net/
6. Chan, W.M. and S. E. Rogers. "Chimera Grid Tools Software" Gridpoints - Quarterly Publication of the NAS Division, NASA Ames Research Center, Spring, 2001. http://www.nas.nasa.gov/About/Gridpoints/PDF/gridpoints_spring2001.pdf

7. Condor, http://www.cs.wisc.edu/condor/
8. Digital Palomar Observatory Sky Survey, http://www.astro.caltech.edu/~george/dposs/
9. Foster, I., J. Vockler, M. Wilde, Y. Zhao, “Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation”, Proceedings of the 14th Conference on Scientific and Statistical Database Management, Edinburgh, Scotland, July 2002.
10. Globus – The Globus Toolkit, http://www.globus.org/toolkit/
11. Grid Datafarm - http://datafarm.apgrid.org/
12. Grid Forum Remote Data Access Working Group. http://www.sdsc.edu/GridForum/RemoteData/.
13. GridFTP, a high-performance, secure, reliable data transfer protocol optimized for high-bandwidth wide-area networks, http://www.globus.org/datagrid/gridftp.html
14. GriPhyN – Grid Physics Network project, http://www.griphyn.org/index.php
15. High Energy Accelerator Research Organization, KEK, http://www.kek.jp/intra.html
16. Joint Center for Structural Genomics, http://www.jcsg.org/
17. Library of Congress, National Digital Information Infrastructure and Preservation Program, http://www.digitalpreservation.gov/
18. MCAT - “The Metadata Catalog”, http://www.npaci.edu/DICE/SRB/mcat.html
19. METS – “Metadata Encoding and Transmission Standard”, http://www.loc.gov/standards/mets/
20. Moore, R., A. Rajasekar, “Common Consistency Requirements for Data Grids, Digital Libraries, and Persistent Archives”, Grid Procotol Architecture Research Group draft, Global Grid Forum, April 2003.
21. Moore, R., C. Baru, “Virtualization Services for Data Grids”, Book chapter in "Grid Computing: Making the Global Infrastructure a Reality", John Wiley & Sons Ltd, 2003.
22. Moore, R., “The San Diego Project: Persistent Objects”, Proceedings of the Workshop on XML as a Preservation Language, Urbino, Italy, October 2002.
23. Moore, R., C. Baru, A. Rajasekar, B. Ludascher, R. Marciano, M. Wan, W. Schroeder, and A. Gupta, “Collection-Based Persistent Digital Archives – Parts 1& 2”, D-Lib Magazine, April/March 2000, http://www.dlib.org/
24. Moore, R. (2000a), “Knowledge-based Persistent Archives,” Proceedings of La Conservazione Dei Documenti Informatici Aspetti Organizzativi E Tecnici, in Rome, Italy, October, 2000.
25. 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.
26. NASA/Goddard Space Flight Center Advanced Data Grid (ADG), http://sunset.usc.edu/gsaw/gsaw2003/s7/gasster.pdf
27. NASA Data Management System / Global Modeling and Assimilation Office, http://dao.gsfc.nasa.gov/
28. NASA Earth Observing Satellite, http://eospso.gsfc.nasa.gov/
29. NASA Information Power Grid (IPG) is a high-performance computing and data grid, http://www.ipg.nasa.gov/
30. National Science Digital Library (NSDL), http://www.nsdl.org/
31. National Virtual Observatory (NVO), http://www.us-vo.org/
32. NPACI Data Intensive Computing Environment thrust area, http://www.npaci.edu/DICE/
33. OAIS - Reference Model for an Open Archival Information System (OAIS). Submitted as ISO draft, http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-1.pdf, 1999.
34. OAIS – “Preservation Metadata and the OAIS Information Model”, the OCLC Working Group on Preservation Metadata, June 2002, http://www.oclc.org/research/pmwg/
35. Object Oriented Storage, http://www.itworld.com/itwebcast/oo_storage/
36. Particle Physics Data Grid, http://www.ppdg.net/
37. RDF - Resource Description Framework (RDF). W3C Recommendation, http://www.w3.org/TR/
38. Real-time Observatories, Applications, and Data management Network (RoadNet), http://roadnet.ucsd.edu/
39. Southern California Earthquake Center, http://www.scec.org/

40. SRB - “The Storage Resource Broker Web Page”, http://www.npaci.edu/DICE/SRB/
41. Thibodeau, K., “Building the Archives of the Future: Advances in Preserving Electronic Records at the National Archives and Records Administration”, U.S. National Archives and Records Administration, http://www.dlib.org/dlib/february01/thibodeau/02thibodeau.html
42. Underwood. W. E., “As-Is IDEF0 Activity Model of the Archival Processing of Presidential Textual Records,” TR CSITD 98-1, Information Technology and Telecommunications Laboratory, Georgia Tech Research Institute, December 1, 198.
43. Underwood, W. E., “The InterPARES Preservation Model: A Framework for the Long-Term Preservation of Authentic Electronic Records”. Choices and Strategies for Preservation of the Collective Memory, Toblach/Dobbiaco Italy 25-29 June 2002. To be published in Archivi per la Storia.
44. University of California, San Diego library, http://libraries.ucsd.edu/
45. Upward, F., “Modeling the records continuum as a paradigm shift in record keeping and archiving processes, and beyond - a personal reflection’, Records Management Journal, Vol. 10, No. 3, December 2000.
46. US National Archives and Records Administration, http://www.archives.gov/, also see http://www.sdsc.edu/NARA/
47. United States Naval Observatory (USNO-B) Catalog, http://arxiv.org/abs/astro-ph/0210694
48. XDR – External Data Representation Format, http://www.faqs.org/rfcs/rfc1832.html
49. XML - Extensible Markup Language, http://www.w3.org/XML/