

## **Data Grid Implementations**

**Reagan W. Moore (San Diego Supercomputer Center)**  
**Igor Terekhov (Fermi National Accelerator Laboratory)**  
**Ann Chervenak (Information Sciences Institute)**  
**Scott Studham (Pacific Northwest Laboratory)**  
**Chip Watson (Jefferson Laboratory)**  
**Heinz Stockinger (CERN)**

**January 3, 2002**

The Global Grid Forum is promoting the development of standards for the implementation of data grids. One of the challenges is defining the set of functionalities that are needed in common across all of the data grid, data collection, digital library, and persistent archive projects that are already in place. A second challenge is to understand the set of capabilities that may be required by future applications. To answer both questions, a comparison has been made between the Storage Resource Broker (SRB) data grid from the San Diego Supercomputer Center, the GDMP data replication tool (a project in common between the European DataGrid and the Particle Physics Data Grid, augmented with an additional product of the European DataGrid for storing and retrieving meta-data in relational databases called Spitfire), the Scientific Data Management (SDM) data grid from Pacific Northwest Laboratory, the Globus data grid, the Sequential Access using Metadata (SAM) data grid from Fermi National Accelerator Laboratory, and the JASMine data grid from Jefferson National Laboratory. These systems have evolved as the result of input by user communities that are currently managing data over heterogeneous, distributed storage resources.

The comparison is an attempt at understanding what the data grid architecture must support to meet existing application requirements. What is most striking is that common user requirements are emerging across all of the data grids. Each data grid implements a logical name space that supports the construction of a uniform naming convention across multiple storage systems. Each data grid is adding attributes to the name space to support additional functionality.

Given a consensus on the set of capabilities, a data handling system can then be characterized by the attributes needed to define each capability. Similarly the range of APIs and protocols that are in active use define interface requirements from the user community. It is interesting to note that multiple types of interfaces are used for interacting with remote data, from I/O library interfaces, to shell commands, Java interfaces, web interfaces, etc.

Terms used in the list include:

- Registration corresponds to adding objects to the logical name space, creating a logical name and storing a pointer to the file name used on the storage system
- Attributes represent information that is managed for each object that is registered into the logical name space

- Folders are equivalent to directories in a file system, but are used to organize objects in the logical name space
- Soft links represent the cross registration of a single physical data object into multiple folders 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 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 file systems.
- A container is an aggregation of multiple data files into a single file
- Curation control corresponds to the administration tasks associated with creating and managing a logical collection
- 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.
- 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.
- Load balancing for a logical name space consists of distributing digital objects across multiple storage systems
- 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 completion at end of single write corresponds to synchronous data writes
- 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
- Federated server architecture refers to the ability of distributed servers to talk among themselves without having to communicate through the initiating client.
- 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
- Bulk metadata load is the ability to import attribute values for multiple objects registered within the logical name space from a single input file.
- 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
- DataCutter is the data filtering service developed by Joel Saltz at the Ohio State University, which is executed directly on a remote storage system.

In the following table, areas where an implementation is planned for a capability are marked with “P”. Areas where information has not been received are left blank. The

capabilities have been organized into eleven categories, with up to twenty different capabilities per category. Over three-quarters of the capabilities have been implemented in at least two of the data grids.

Capability	GDMP & Spitfire	Globus	JASMine	SAM	SDM	SRB
<b>Logical name space</b>	Yes	Yes	Yes	Yes	Yes	Yes
Logical name space independence from physical name space	Yes	P	Yes	Yes	Yes	Yes
Hierarchical logical folders	P	P	Yes	No	Yes	Yes
Unix node operations on logical name space – directory manipulations to rename, delete, create and remove files and folders	No	No	Yes	Yes	No	Yes
Recursive operations on logical name space directories for both store and retrieval	No	No	Yes	No	No	Yes
Deletion of entities from logical name space	Yes	Yes	Yes	Yes	Yes	Yes
Delete by setting a deletion attribute	No	No	No	Yes	Yes	Yes
Delete by removing attribute	Yes	Yes	Yes	Yes	Yes	Yes
Soft links between objects in logical folders so that a single file can be listed in multiple folders	No	P	P	Yes	Yes	Yes
Data referenced by catalog owned by a user ID	Yes	Yes	Yes	Yes	Yes	Yes
Data referenced by catalog owned by a Collection ID	No	P	No	Yes	Yes	Yes
Registration of files as objects in logical name space	Yes	Yes	Yes	Yes	Yes	Yes
Registration of databases as objects in logical name space			No	Yes	No	Yes
Registration of database blobs as objects in logical name space	No	No	No	Yes	No	Yes
Registration of URLs as objects in logical name space			No	No	No	Yes
<b>Logical name space Attributes</b>	Yes	Yes	Yes	Yes	Yes	Yes
<b>System level attributes</b>	Yes	Yes	Yes	Yes	Yes	Yes
Replica attributes for storage location, local file name	Yes	Yes	Yes	Yes	No	Yes
Replica attributes for type of storage system	No	No	No	Yes	No	Yes
User access control lists for data in logical name space			P	No	Yes	Yes
Group access control lists for data in logical name space			P	Yes	Yes	Yes
Access control lists for logical name space attributes, to control who can see, add, and change metadata	No	No	No	Yes	Yes	Yes
Extended set of user roles (administrator, collection curator, annotator, user)				Yes	Yes	Yes
Access control lists for resources		Yes		Yes	No	Yes
I/O access pattern					No	Yes
Version attribute	No	No	No	Yes	No	Yes
Audit trails for updates and/or accesses	No	No	Yes	Yes	Yes	Yes
<b>User level attributes</b>	Yes	No	Yes	Yes	Yes	Yes
User defined attributes for data entities	Yes	No	Yes	Yes	Yes	Yes
User defined attributes for collections				Yes		Yes
Annotation attributes	Yes	No			Yes	Yes
User profiles to describe storage usage history			No		No	P

<b>Discipline specific attributes</b>	Yes	No	Yes		Yes	Yes
Dublin Core attributes	No	No	No		No	Yes
Template based metadata extraction for catalog attributes						Yes
External catalog accessible for additional attributes	No	Yes	No		No	Yes
Physics tags				Yes		Yes
<b>Attribute manipulation</b>	Yes	Yes	Yes	Yes	Yes	Yes
Query interface to discover files by attributes				Yes	Yes	Yes
Automated attribute generation for size, time stamp	Yes	Yes		Yes	Yes	Yes
User managed synchronous attribute update	Yes	Yes	Yes	Yes	Yes	Yes
Asynchronous annotation of objects in logical name space	Yes	No	Yes	Yes	Yes	Yes
Export of attributes as XML file or python				Yes		Yes
Bulk asynchronous load of attributes	Yes	No	No	Yes	No	Yes
Bulk asynchronous load of attributes from XML or python file	No	No	No	Yes	No	Yes
<b>Data manipulation</b>	Yes	Yes	Yes	Yes	Yes	Yes
Synchronous creation of replicas with associated metadata creation	Yes	Yes	Yes	Yes	No	Yes
Asynchronous creation of replicas				Yes	No	Yes
Registration of user owned data as a replica of an existing object in the logical name space	No	No	No	No	No	Yes
Load balancing across physical resources				Yes	Yes	Yes
Containers for aggregating small files	No	No	No	No	No	Yes
Logical aggregation of objects for storage on tape	No	No	Yes	Yes	No	No
Container locking on writes	No	No	No	N/A	No	Yes
Updates to containers by appending data	No	No	No		No	Yes
Replication of containers	No	No	No		No	Yes
Container replica invalidation of updates	No	No	No		No	Yes
Staging of containers from archive to disk	No	No	No		No	Yes
Synchronization of containers to archives	No	No	No		No	Yes
Multiple disk caches for containers	No	No	No		No	Yes
Multiple archives for storing containers	No	No	No		No	Yes
Logical container names that represent multiple physical containers	No	No	No		No	Yes
<b>Data Access</b>	Yes	Yes	Yes	Yes	Yes	Yes
Parallel I/O support	Yes	Yes	Yes	Yes	No	Yes
Parallel I/O on get/put commands	Yes	Yes	Yes	Yes	No	Yes
Parallel I/O on partial file reads/writes	Yes	Yes	No	No	No	No
Transmission status checking at the file level	Yes	Yes	Yes	Yes	No	Yes
Transmission block tagging to support transmission restart after interruption	No	Yes	Yes	No	No	No
Transmission restart after interruption at application level	Yes	Yes	Yes	Yes	Yes	Yes
Storage completion at end of single write	Yes	Yes	Yes	Yes	Yes	Yes
Replication completion when write to k of n physical resources	No	No	No		No	Yes
Standard error messages from storage systems, network, and data handling system	Yes	Yes	Yes	Yes	Yes	Yes

Striping support	No	Yes	Yes	No	Yes	P
Thread safe client	Yes			Yes	Yes	Yes
Static network tuning	Yes	Yes	Yes	P	No	Yes
Dynamic network tuning (window and buffer size)	No	Yes	No	P	No	No
GridFtp protocol support	No	Yes	P	P	No	P
TCP/IP custom control protocol	No	No	No	P	No	Yes
Java parallel custom control protocol	No	No	Yes	No	No	No
<b>Multiple Access APIs</b>	Yes	Yes	Yes		Yes	Yes
Remote data access by I/O redirection from Linux or Solaris system I/O calls	Yes	No	No		No	Yes
C I/O library API	Yes	Yes	No		Yes	Yes
C++ I/O library API	Yes	No	No	Yes	Yes	Yes
Command line interface	Yes	Yes	Yes	Yes	Yes	Yes
Java interface	Yes	P	Yes		Yes	Yes
Web service interface	P	No	Yes	Yes	No	Yes
Visual Basic interface	No	No		No	Yes	Yes
XML query interface				No	P	Yes
DLL/API interface for Python					No	Yes
Predicate assertion interface					No	Yes
SDLIP interface					No	P
Windows browser interface	No	No	No		Yes	Yes
<b>Distributed client-server architecture</b>	Yes	Yes	Yes	Yes	Yes	Yes
Federated client server	Yes	Yes	Yes		Yes	Yes
Distributed servers	Yes	Yes	Yes	Yes	Yes	Yes
Distributed storage systems	Yes	Yes	Yes	Yes	Yes	Yes
64-bit name space						Yes
Logical resources that represent multiple physical resources	No	No	No	Yes	No	Yes
Third party transfer from storage system to specified remote destination	Yes	Yes	Yes	No	Yes	Yes
GSI authentication		Yes		P	No	Yes
PKI authentication		Yes		P	No	Yes
Challenge response authentication				Yes	No	Yes
Ticket based access control for number of accesses and restricted time period for access				Partial	P	Yes
<b>Latency Management</b>	Yes	Yes	Yes		Yes	Yes
Streaming	Yes	Yes	Yes		Yes	Yes
Caching	Yes	Yes	Yes	Yes	Yes	Yes
Prefetch (partial file caching)					No	Yes
Containers for data				Yes	No	Yes
Containers for metadata				Yes	No	Yes
Remote I/O proxies for aggregating I/O commands, remote data filtering, metadata extraction	No	Yes	No		No	Yes
Remote Proxies through DataCutter	No	No	No		No	Yes
Remote Proxies through GridFTP	No	Yes	No		No	No
Staging				Yes	Yes	Yes
Status checking				Yes	No	Yes
Replication	Yes	Yes	Yes	Yes	No	Yes

<b>Multiple system support</b>	Yes	Yes	Yes	Yes	Yes	Yes
Storage Resource Manager interface	Yes	No	Yes	Yes	No	Yes
Database interface for reading/writing blobs	No	No	No		No	Yes
Database interface for queries to relational database registered as a data object	Yes	No	No		No	Yes
Database interface for exporting attributes as an XML file	Yes	No	No		No	Yes
Archive interface to at least one archive	Yes	Yes	Yes	Yes	Yes	Yes
Archive interface to HPSS	Yes	Yes	No		No	Yes
Archive interface to DMF	Yes	Yes	No		No	Yes
Archive interface to ADSM	Yes	No	No		No	Yes
Archive interface to Enstore				Yes		No
Archive interface to UniTree	Yes	No	No		No	Yes
Archive interface to JASMine	No	No	Yes		No	No
Archive interface to HPSS for storing file sizes greater than 2 GB	Yes	No	No		No	Yes
Single catalog server, database technology used to replicate catalog	Yes	Yes	Yes		Yes	Yes
Hierarchical distributed catalog	P	P	No	No	No	No
<b>Performance enhancements</b>	Yes	Yes				Yes
Performance for import/export of files greater than 20 files/sec	Yes	Yes			No	Yes
Performance for import/export of files greater than 1100 files/sec	Yes	No			No	Yes
Bulk metadata load				Yes	No	Yes
Pre-spawned processes for data transfers					No	Yes
Database index optimization					Yes	Yes
Database communication tuning					No	Yes
<b>Robustness: Fault tolerance and error handling</b>				Yes		Yes
Automatic fail over to alternate replica when the first copy is unavailable	No			Yes	No	Yes
Automatic retrials in metadata catalogue access	No			Yes		No
Automatic retrials in archive access	No			Yes		P
Data transfer resumption upon system restart (crash and reboot), transparently to the user jobs	Yes			Yes		P
Single-command resumption of very long user jobs upon system restart	Yes			Yes		No
Configurable time-outs on user usages of every resource (to protect against abandoned/misbehaved user jobs)	P			Yes		No
Handling of mass storage system software errors, including protocol non-compliance and other aberrations	No			Yes		Yes
Handling of underlying file transfer tools (such as ftp) errors, including non-protocol compliance	Yes			Yes		Yes
File checksum	Yes	P	P	No	No	Yes