Distributed Storage Management Concepts

Reagan W. Moore
UNC-CH

The problem is posed as the management of storage systems that are geographically distributed, that reside in multiple administrative domains, and that include heterogeneous hardware and software systems. A simple example would be the management of storage that includes a Unix file system, a tape archive, an object storage system, a Windows file system, and a database, each located in a different administrative domain at a different location. Is there a unifying way to manage storage of data?

The generic middleware approach towards managing the above infrastructure is:

- Define the global logical name spaces that will be used to identify resources. A mapping will be required from the global logical name space to the physical naming conventions within each type of storage system.
- Define the operations that are applied on each global logical name space. Examples include manipulation of entities, aggregation of entities, access controls on entities.
- Define the virtualization mechanisms that will enable application of the operations across the multiple hardware and software systems.
- Define management policies that will be enforced on the name spaces (quotas, integrity, chain of custody, authenticity, arrangement)
- Define the operations required to enforce the policies (typically applied across multiple name spaces as a system level operation)
- Define federation mechanisms

This generic approach can be simplified by removing some of the constraints (heterogeneity, use of networks for remote access, multiple administrative domains). If you remove all of these constraints, a Storage Area Network may be sufficient for your use cases.

Let's look at a variety of use cases where access and management of distributed storage systems are required. These include federal agencies, national scale projects, high performance computing sites, institutional repositories, and compute clusters.

- XSEDE – they require high-performance read/write to local disk for generation of petabytes of simulation data per day, and then replicate data across multiple independent data archives
- Broad Institute – they organize genomics data from multiple projects, automate the processing of the data into standard forms, and store results in an archive.

- Wellcome Trust Sanger Institute – they organize genomics data across multiple projects, automate the processing of the data into standard forms, and store results in an archive.
- UNC-CH Genomics data grid – they organize genomics data from multiple projects, automate the processing of the data into standard forms, and store results in an archive.
- NSF Ocean Observatories Initiative – they manage real time sensor data streams, and automate the archiving of a copy of the data at the National Climatic Data Center.
- NSF iPlant Collaborative – they access a wide variety of distributed data repositories (typically databases), extract desired data subsets, and then share analyses of the data.
- National Optical Astronomy Observatory – they archive images taken on telescopes in Chile on storage systems in Arizona and Illinois.
- National Climatic Data Center – they manage data ingestion from multiple external projects, replicate their digital holdings, and store the data across multiple types of storage systems.
- NASA Center for Climate Simulations – they provide access to major digital holdings (MODIS Moderate Resolution Imaging Spectroradiometer 650-Terabyte data set) for access from the Earth Systems Grid through a file system interface
- BaBar High Energy Physics – they replicated 2 Petabytes of data between archival storage systems at SLAC and Lyon, France
- Texas Digital Libraries – they federated storage systems across university libraries in Texas, replicating data into the Texas Advanced Computer Center
- French National Library – they periodically migrate their digital holdings across storage system technologies
- T2K QMUL neutrino experiment – they aggregate small data files before archiving, and distribute data from Japan to London
- Australian Research Collaboration Service – they distribute observational data sets to the university that has the associated experts, and then share data across storage systems
- Sickkids Hospital (Ontario, Canada) – they manage HIPAA data across multiple storage systems
- UK e-Science data grid – they decoupled storage resources from compute resources, and distributed data products generated by each computation into a national storage environment.

The generic operations from the above use cases include:

1. High performance access to disk caches while computing
2. Management of storage space
3. Archiving of data to alternate storage systems
4. Distribution of data to multiple sites
5. Replication of data sets
6. Access to distributed data sets

7. Enforcement of access controls in the distributed environment
8. Automated application of processing steps to data sets
9. Automated management of caches
10. Organization of data into collections that span storage systems

The implementations of the storage management system strongly depends upon the desired performance. Thus item 1 is typically achieved in a tightly coupled system that has very high I/O bandwidth within a local environment, but items 2 through 10 can be implemented using loosely coupled middleware.

The name spaces that are used to implement these operations across distributed storage systems in a loosely coupled environment include:

Name Space	**Operations**	**Virtualization interface**
Users	Authentication, authorization, groups	GSSAPI / PAM
Objects	Partial I/O, move, copy, replicate, share	Posix I/O & staging
Collections	Organization, Browsing	System metadata
State information	Add, update, delete, query	Catalog interface to DBMS
Resources	Load leveling, fault tolerance, grouping	Storage drivers
Policies	Management, administrative, verification	Policy language
Procedures	Basic functions on each name space	Workflows

The choice of which name spaces, operations, and virtualization interfaces to apply when managing distributed storage depends upon the the reason why the digital objects are being stored (manipulation, access, preservation, publication).

Relevant state information attributes may then include:

- File identifier
- File creation time
- File modification time
- File size
- File checksum
- File storage location
- File type
- File retention period
- File disposition
- File version
- Replica number
- Replica location

- Replic creation time
- Replica checksum
- Collection identifier (logical groups of files)
- Storage system identifier
- Storage groups
- Storage quota per user or user group
- User identifier
- User groups
- Access controls
- Collection sticky bit (for inheritance of access controls)
- Audit trail

A more sophisticated system would map from a process to the object that would be created. The process could be represented as a workflow, and invocation would dynamically generate the desired data set. Storage management is becoming linked to the execution of processes as a way to handle massive collections.

NSF is developing the Future Internet Architecture, which embeds policies in the network. This can be viewed as mapping from the virtual name spaces used to manage distributed storage systems, to a virtual network that links the systems. Policies within the network can then control:

- Access by file name
- Access controls
- Data distribution
- Data replication
- Data versioning
- Data caching
- Quotas

An expectation is that within 5-10 years, many of the properties associated with a storage system will be moved into the network for management of distributed storage systems.

Policies are also being embeded in storage controllers. This enables the incorporation of knowledge within each storage device, and the automated processing of data. An example would be the automated extraction of features as each data set is written to storage. The policies control the execution of procedures that search the data for a desired feature, and then create an index to support retrieval based on the identified features.

The above approach has been applied in the integrated Rule Oriented Data System, and is available as open source software at http://irods.diceresearch.org.