An Open Preservation and Archiving Architecture

James Simon
Chief Technologist, Principal Engineer
Director, North American Systems Practice
Sun Microsystems
What Information is Important?

- In the long run, these questions might not seem so silly

Austin Onion – Nov 2008
Assumptions that drive design

Large archives must support scalable modular, extensible, manageable systems for storage

The components, hardware and software, will change

The primary assumption: over time, systems, software and people will be replaced and DATA will be preserved!
Model for Storage Implementations

- Is there a model that goes beyond backup?

- Will an abstract view of the storage be the future model, will it help?
Multiple archive models should be supported. In a single data center, geo separated data centers. Federated independent archives and independent service based archives.

Model supports LOCKSS, CLOCKSS, BL model, as well as traditional ILS models.
Future Architectures May support Clouds...
Top tier acts as a Services layer - protocol(s), API(s), and Web Services interface(s). Archive applications (e.g. Fedora) connect through the Services layer, which act as a proxy to Storage abstraction. Services provided include, store/retrieve, sloshing object Meta Data services, Xml Schema search, admin services, audit/reporting, federation, IDM, etc.

Protocol / API (object / OID / Query) / Web Services

Out to Fedora / Archive Application

House Keeping
Data, MetatData
XML Schema Tables
Custom store/retrieve Agents

Archive Services Layer
Management, Policy, Access
Proxy for Store Services

Archive Services Layer
Management, Policy, Access
Proxy for Store Services

Storage "Cell" 1
MySQL 1, Store (eg Thor, 4450, etc.)
Attached to Tape

Storage "Cell" 2
MySQL 2
attached to Tape

Storage "Cell" N
MySQL N

Local / Remote
Database's hold Objects (including Metadata) or house keeping about unstructured objects.
Backup to direct attach Tape, could use SAN or VTL.
Backup could work remotely.
- Proxy /services layer handles all access to archive and is replicable and scalable, supports geo separation.
- Supports Network Protocol, API and Web services. Fixed set of base service classes and is modular.
- Supports scaling across multiple machines at all points in architecture.

- Archive holds Objects. Output is OID, hash, Meta Data, Object.. Descriptive Metadata (rights, schema, etc.) is linked and searchable. OID search can return Descriptive Metadata and XML Schema Metadata.
- Support for multiple XML Schema and parallel search. Support for multiple Schemas is straight forward.
- Location Metadata is simple and archivable and replicable. If you lose Metadata, it is recoverable from Tape, Network or Disk. Descriptive Metadata is archivable, and recoverable.
- Sloshing/storage balancing is achieved by adding a new storage node instance and moving Objects and syncing Location Metadata.
- Data integrity is kept by underlying file system, ZFS or QFS as well as continuous check sum checkers that are inherent or could be custom built.

- N store/ N-Copy multiple copies is supported. Objects be stored in data center, and across geographic boundaries. With multiple copies, N check and N restore can be implemented to support “voting” among copies.

- Federation with security can be supported, multiple repository model. Lots of copies supported.
- Storage beans (custom agents) as extension to retrieval/store proxy can solve customization and transformation problems existing in Honeycomb. Support tiered rights model and audit, so archive is secure. Can scale because server tier can scale.

Architecture can support migration to new hardware via object sloshing and migration services. Architecture can support other archivers, such as XAM, via custom storage extensions.

Storage nodes built as “Cells” Cell consists of single instance of MySql db with associated storage. For example our ST 4450 makes a good Cell as it holds 48TB and has enough processing power to host MySql instance.
Block level view of services

Management

Storage services components

Services components

Enterprise instance of Archiver Storage abstraction

Cells work with our JBod’s, Thor, ZFS based storage, not dependent on specific HW.

CPU/Store ratio configured for SMASS use.
Object A stored across sites.

4 archive's federated, showing a unique object stored as 3 instances. Two copies on Archive 1 and one copy on Archive 4.
Summary:
Architecture can support Most if not all of the St 5800 features and more.
Recommended servers, 5240’s and ST 4450 offer high level of integration and scalability.
Resiliency is supported through:
- File system protection (e.g. ZFS) copy on write and checksum, raid Z, snapshot, etc.
- N Copy - ability to create multiple Object copies and check them against each other.
- MySQL (or other) DB used to manage storage abstraction MD.
Storage Nodes (Cell(s)), manageable and restorable. Also replaceable as new technology or improvements are available.
- Geographic separation capable through Location Tables, use of protocols and API's
- Back up and snapshot available at all levels and easily restored
Scale is supported through:
- Multiple independent Data Base instances
- Sloshing/storage node balancing through the adding of new local or remote node instances with sloshing engine
Parallel Schema search is supported

- Server load balancing at each tier
- each tier
Other: DRM, access control, Storage Beans (customer retrieval agents), multiple XML schema’s, metadata search
Thanks!

James Simon
Sun Microsystems

james.simon@sun.com