Lustre HSM Project

Aurélien Degrémont
aurelien.degremont@cea.fr

- Introduction
- Requirements
- Architecture
- Status
Introduction (1/3)

- **Two powerful components**

- Size from some TB to few PB
  - Fast and parallel

- Size from lots of TB to many PB
  - Slow data accesses

Lustre filesystem

[Diagram showing Clients, MDS, OSS, and External HSM connections]
Introduction (2/3)

- Interesting cooperations

- A fast cache for a HSM

- Very wide disk space for a Lustre filesystem

- Backup a Lustre filesystem
Requirements
An HSM extension for Lustre

- To inter operate with existing storage systems
- No strong binding with external storage
  - Basic copy-in, copy-out must work with a simple user space tool

Provide basic features

- Lustre will act like a cache
- Cache miss, archive, purge, transparency
- Can be used as backup
Requirements (2/2)

- **All files are always visible in the file system, but a file can reside:**
  - On primary storage (Lustre)
  - On the backend storage
  - On both

- **Metadata (size, …) are always up-to-date**
  - Add migration status information

- **Scalable and parallel**
  - Lustre HSM must have a small impact on Lustre performances
  - Target is to impact Lustre performances only when data are not in Lustre (time to bring back data when a cache miss occurs)
Architecture: How doing it?

- **Features needed**
  - Copy data to HSM
  - Purge file in Lustre
  - Bring them back

- **Introduce new components in Lustre infrastructure**
  - Agents
  - Space Manager
  - Coordinator
Architecture: Copy it!

- **Agent**
  - a service used to move data, to cancel such movement and to remove external storage files

- **Archiving tool**
  - spawns by Agents on specific client nodes
  - interface between Lustre and the HSM
  - knows how to communicate with a specific HSM
Architecture: Purge them

- **Space Manager**
  - watches filesystem disk space usage
  - pre-migrates not recently used data
  - when space is lacking, purges data from files already copied in the HSM
**Architecture: Bring them back**

- **Initiating**
  - Cache misses are detected on MDT and OST

- **Coordinator**
  - Centralizes migration requests
  - Dispatches them on agents
Architecture: Sum up

- OSS
- Client
- Coordinator
- Archiving tool
- External HSM
- MDS
- Space Manager
- I/O
- triggers requests
- watches
- initiates requests
- dispatches migrations
- Agent
- Archiving tool
- transfer data

External HSM
Zoom on: External elements

- A userspace command able to
  - Copy from posix (Lustre) to HSM
    - Lustre access is made through a hidden path (/mnt/.lustre/FID/...)
  - Copy from HSM to posix (Lustre)
  - Remove a file in HSM
  - Cancel a transfer (optional)
  - Manage data transfer progress

- External HSM
  - Do not know about the Lustre namespace
  - No Lustre knowledge is needed in the HSM

- A reference to HSM object ID and a version number (returned by HSM) is kept in Lustre
Zoom on: Policy

- **Use of pre-migration**
  - Automatic: by *Space Manager*
    - Could be based on size, modification date, ...
  - On demand: by a user/admin tool

- **File system space management is either:**
  - Automatic
    - At OST level
    - At FS level (MDT)
  - On demand: Based on a provided list of files

- **Purge method**
  - Keep start/end of files on disk
  - At OST level (objects)
  - At FS level (all file)
Project status

• Project
  ◆ CEA/SUN collaboration
    ➤ Architecture design made by Lustre designers and CEA
    ➤ High Level Design/Detailed Design/Coding by community (CEA, SUN, ...)

• Development
  ◆ Architecture and High Level Design are done
  ◆ Detailed designs and prototypes are under progress

• Roadmap
  ◆ Target is Lustre 2.0
  ◆ Early/beta code for Summer '08
  ◆ Final version for end of '08
Questions ?