Lustre/HSM Binding

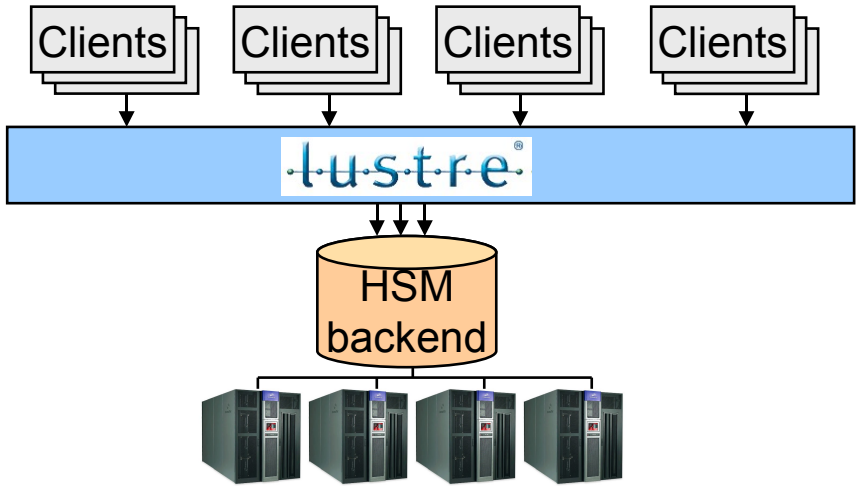
Aurélien Degrémont
aurelien.degremont@cea.fr

Agenda

- Presentation
- Architecture
- Components
- Use cases
- Project status

Presentation (1/2)

- **HSM seamless integration**

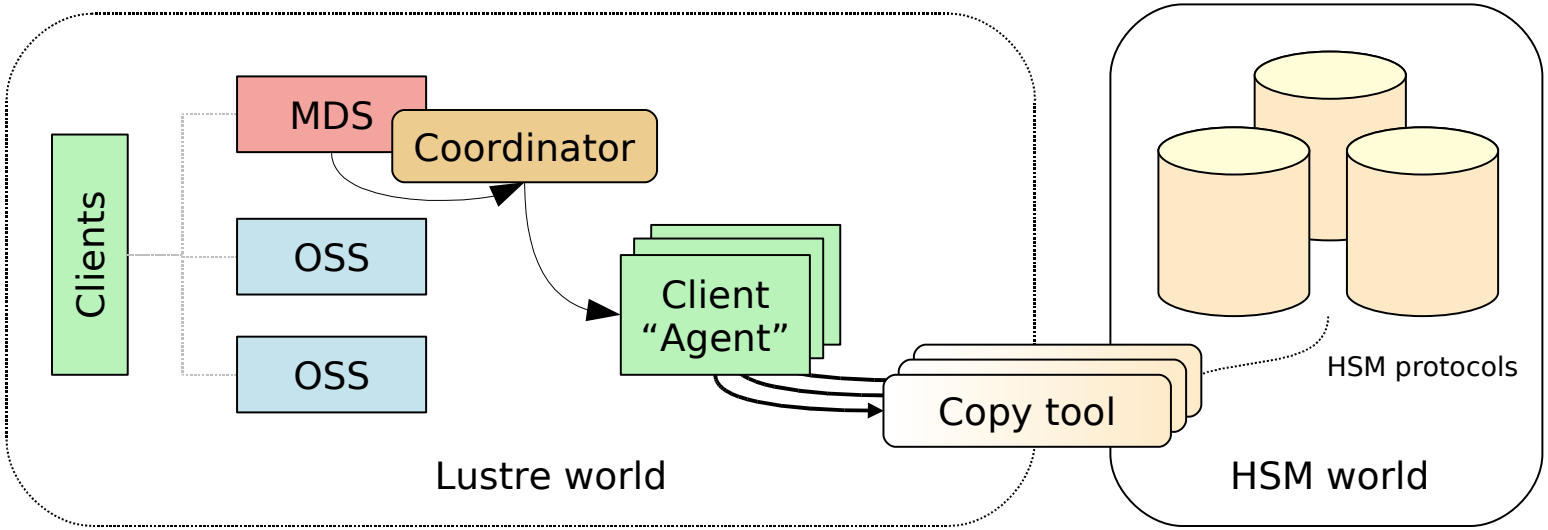


- **Takes the best of each world:**
  - Lustre: high-performance disk cache in front of the HSM
    - parallel cluster filesystem
    - high I/O performance, POSIX access
  - HSM: long-term data storage
    - Manage large number of disks and tapes
    - Huge storage capacity

Presentation (2/2)

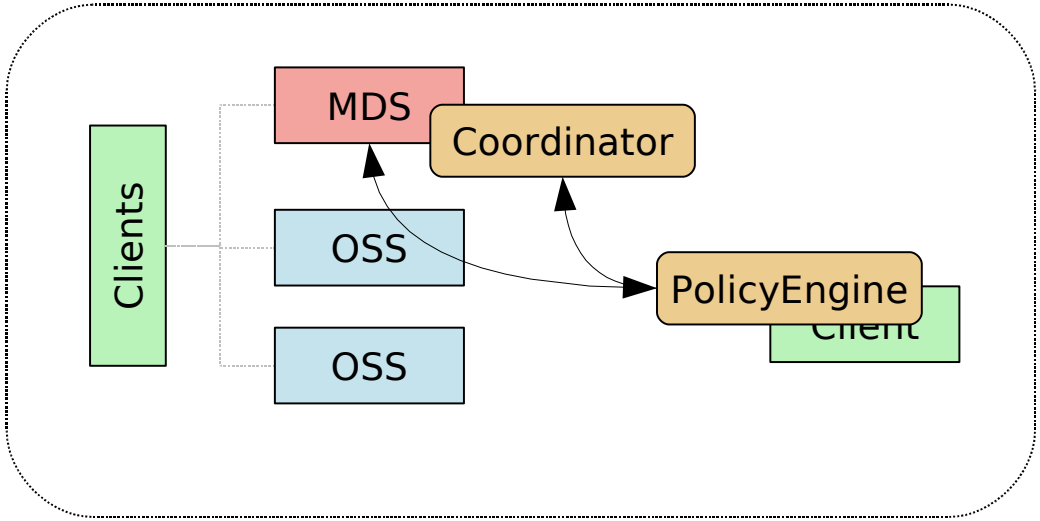
- **Features**
  - Migrate data to the HSM
  - Free disk space when needed
  - Bring back data on cache-miss
  - Policy management (migration, purge, soft rm,…)
  - Import from existing backend
  - Disaster recovery (restore Lustre filesystem from backend)
- **New components**
  - Coordinator
  - Archiving tool (backend specific user-space daemon)
  - Policy Engine (user-space daemon)

Architecture (1/2)



- **New components: Coordinator, Agent and copy tool**
  - The coordinator gathers archiving requests and dispatches them to agents.
  - Agent is a client which runs a copytool which transfers data between Lustre and the HSM.

Architecture (2/2)



- **PolicyEngine manages archive and release policies.**
  - A user-space tool which communicates with the MDT and the coordinator.
  - Watch the filesystem changes.
  - Trigger actions like archive, release and removal in backend.

Component: Copytool

- **It is the interface between Lustre and the HSM.**
- **It reads and writes data between them. It is HSM specific.**
- **It is running on a standard Lustre client (called Agent).**
- **2 of them are already available:**
  - HPSS copytool. (HPSS 7.3+). CEA development which will be freely available to all HPSS sites.
  - Posix copytool. Could be used with any system supporting a posix interface, like SAM/QFS.
- **More supported HSM to come**
  - DMF
  - Enstore

Component: PolicyEngine Robinhood

- **PolicyEngine is the specification**
- **Robinhood is an implementation:**
  - Is originately an user-space daemon for monitoring and purging large filesystems.
  - CEA opensource development: http://robinhood.sf.net
- **Policies:**
  - File class definitions, associated to policies
  - Based on files attributes (path, size, owner, age, xattrs…)
  - Rules can be combined with boolean operators
  - LRU-based migr./purge policies
  - Entries can be white-listed

Robinhood: example of migration policy

- **File classes:**

```
Filesets {
      FileClass small_files {
          definition { tree == "/mnt/lustre/project" and size < 1MB }
          migration_hints = "cos=12" ;
…
      }
}
```

- **Policy definitions:**

```
Migration_Policies {
      ignore { size == 0 or xattr.user.no_copy == 1 }
      ignore { tree == "/mnt/lustre/logs" and name=="*.log" }

      policy migr_small {
                target_fileclass = small_files;
                condition { last_mod > 6h or last_copyout > 1d }
      }
      …
      policy default {
                condition { last_mod > 12h }
                migration_hints = "cos=42";
      }
}
```

Robinhood: example of purge policy

- **Triggers:**

```
Purge_trigger {
  trigger_on = ost_usage;
  high_watermark_pct = 80%;
  low_watermark_pct = 70%;
}
```

- **Policy definitions:**

```
Purge_Policies {
  ignore { size < 1KB }
  ignore { xattr.user.no_release = 1 or owner == "root" }

  policy purge_quickly{
       target_fileclass = classX;
       condition { last_access > 1min }
  }
  …

  policy default {
       condition { last_access > 1h }
  }
}
```

Component: Coordinator

- **MDS thread which "coordinates" HSM-related actions.**
  - Centralizes HSM-related requests.
  - Ignore duplicate request.
  - Control migration flow.
  - Dispatch request to copytools.
  - Requests are saved and replayed if MDT crashes.

Automatic migration

- **Example with a video demo**
  - A Lustre filesystem is bound to an HPSS system
  - New files are created and automatically migrated thanks to a Robinhood aggressive policy.
  - When filesystem usage threshold is reached, Robinhood requests purges for oldest files.

    ☞ Video demo #1

Manual migration and purge

- **Thanks to Robinhood, you can easily manage the files in Lustre.**
- **2 examples in video demo:**
  - It is easy to manually synchronize out of date files in Lustre into your HSM. (by example, before a maintenance)
  - It is easy to empty a specific OST (by example, if it is going to be removed).

☞ Video demo #2

Importing an existing archive: HPSS example

- **It is possible to import an existing archive into Lustre.**
- **The HPSS copytool imports an existing HPSS namespace into Lustre.**
- **The Lustre binding namespace will be put separately.**
- **Files are only a symbolic link to the imported one as long as they are not modified.**
- **Example with a video demo:**
  - A HPSS namespace already exists.
  - It is imported and the Lustre namespace is bound to /lustre-hsm

☞ Video demo #3

Project status

- **Status**
  - Prototype working
  - Coding is finishing
  - Integration tests, debugging, stress tests

Questions ?