FROM RESEARCH TO INDUSTRY

cea

Focus: Lustre/HSM with Robinhood v3

Robinhood User Group 2016

Henri Doreau <henri.doreau@cea.fr>

SEPTEMBER, 19th 2016

www.cea.fr

Overview

Using Robinhood v3 for Lustre/HSM

- Configuration changes between v2/v3 wrt. Lustre/HSM
- Robinhood extended: semantic changes & additional features
- Controlling copytools
- What to expect next

Effort to improve robinhood for Lustre/HSM

V3 development points

- Policy management code was almost entirely rewritten between v2 and v3
- Lustre/HSM implemented as a plugin (shipped by default)
  - Automatically loaded if you **%include "includes/lhsm.inc"**
  - Exposes a status manager and corresponding handlers
- 4 companies involved in LHSM code
  - + sites and vendors who provided us with feedback

Lustre/HSM with Robinhood v3 (1/7)

v2 to v3 configuration changes

- To setup Lustre/HSM policies, include “lhsm” template:
  - `%include “includes/lhsm.inc”`
- It defines the following policies:
  - lhsm_archive: archive files to the backend
  - lhsm_release: release data from Lustre disks
  - lhsm_remove: remove orphan files in backend
- In your old v2 config substitute:
  - s/migration/lhsm_archive/
  - s/purge/lhsm_release/
- Specify triggers for lhsm_archive_trigger, lhsm_remove_trigger. E.g.:

```
# triggers lhsm_archive policy hourly
lhsm_archive_trigger {
    trigger_on = scheduled;
    check_interval = 1h;
}
```

Lustre/HSM with Robinhood v3 (2/7)

About “lhsm_remove” policy

- Policy to remove files from backend after they are deleted from Lustre
- In v2.x, very poor hsm_remove policy:
  - Only criteria: “deferred_remove_delay”
- Now: a policy like others
  - Can specify rules based on original path, owner, size, etc...
  - New time criteria “rm_time”
- Example of lhsm_remove rules:

```
lhsm_remove_rules {
    ignore_fileclass = keep_forever;

    rule rm30d {
        target_fileclass = somedata1;
        target_fileclass = otherdata2;
        condition { rm_time > 30d}
    }
    rule rm90d {
        target_fileclass = keeplonger;
        condition { rm_time > 90d}
    }
 ...
```

Lustre/HSM with Robinhood v3 (3/7)

rbh-undelete

- When an archived file is deleted from Lustre, robinhood keeps a copy of its metadata until it is actually remove from the backend (lhsm_remove policy).
- It can be restored using `rbh-undelete` command
  - Restore a file and its metadata
  - Associate the newly created entry with the one from the backend (external call)
- Copytool-specific configuration
  - Example for POSIX copytool:

```
lhsm_config {
    rebind_cmd = “/usr/sbin/lhsmtool_posix --hsm_root=/mnt/backend
                                          --archive {archive_id}
                                          --rebind {oldfid} {newfid} {fsroot}”;
}
```

Lustre/HSM with Robinhood v3 (4/7)

UUID-based HSM mapping: motivations

- Some copytools (like POSIX) address entries in the archive by Lustre fid

  ➔ Not satisfying, as the archive is supposed to be more durable than the Lustre filesystem.

  - Also, Lustre fid may change: undelete/disaster recovery, MDT migration...

  ➔ Require to “rebind” entries in the backend (rename).

- **LU-6866** suggested that copytools should address entries using a persistent identifier which is not based on Lustre FID: an UUID.
  - When a file is archived this identifier can be attached to Lustre entries as an XATTR.
  - If entry fid changes (undelete, MDT migration), no operation is needed in the archive.

Lustre/HSM with Robinhood v3 (5/7)

UUID-based HSM mapping: support in robinhood (Cray)

- Robinhood configuration indicates how the *uuid* is stored:

```
lhsm_config {
        uuid {
            # where the CT stored the UUID
            xattr = "trusted.lhsm.uuid";
        }
}
```

- When processing a HSM ARCHIVE changelog record (or when scanning), robinhood retrieves this xattr value and saves it to its database.
- When a file is deleted from Lustre, this UUID is kept by robinhood, so the file can be later “undeleted” and has the right binding to the archived copy
  - No operation is required in the archive.

Lustre/HSM with Robinhood v3 (6/7)

Specifying target archive

- “archive_id” is to support several archive backends with Lustre/HSM
- In robinhood v3, “archive_id” is managed as an *action parameter*
  Example:

```
lhsm_archive_rules {
        rule archive30d {
            target_fileclass = data1;
            target_fileclass = data2;

            action_params {
                  archive_id = 3;
            }

            condition { last_mod > 30d }
        }
}
```

- action_params can be specified at multiple levels (default per policy, overridden by fileclass or by policy rule).

Lustre/HSM with Robinhood v3 (7/7)

Passing custom parameters to copytools

- Other “action parameters” are passed to the copytool
  - Like: `lfs hsm_archive --data “…”`
- Format is copytool-specific:
  - HPSS copytool: CSV
  - Cray copytool: JSON

- Example for HPSS copytool (http://lustrehpss.sourceforge.net/):

Configuration:

```
action_params {
        cos=3;
        stripe_size=16MB;
}
```

String passed to HPSS copytool:

```
“cos=3,stripe_size=16MB”
```

Generic copytool

Use anything as a backend for Lustre/HSM

- More and more copytools being developed
- Must share a lot of common code for Lustre/HSM-specific logic
  - Register/Deregister copytool to the MDT
  - Receive/Parse commands
  - Issue data transfer ← backend-specific
  - Action begin/end (llapi_hsm_action_*)
- Solution: separate lustre/hsm code and data transfer
  - Fork a sub-command for the transfer
  - Pass it an open file descriptor and fid

Generic copytool

Introducing new companion tool: lhsmtool_cmd

- Allow binding any backend archive to Lustre
  - All you need is to provide a command-line to put/get data
- Ease experiments with new archive systems
- Ease integration of features (gpg encryption, MD expansion)
- Configuration driven: one command per HSM action

```
[commands]
archive=dd if=/proc/self/{fd} of=/mnt/nfsarchive/{fid} bs=1M
restore=dd if=/mnt/nfsarchive/{fid} of=/proc/self/{fd} bs=1M
```

- Less than 1k LoC, distributed in “robinhood-tools” RPM
- Actually used in production (2PB+ to google drive @Stanford)

Future plans

Lustre/HSM work in progress

On Robinhood side

- Lustre/HSM workload leveling (rate limiter)
- HSM request batching

On Lustre side

- New Kernel-Userland communications (LU-7659)
  - Optimize changelog streaming and HSM actions flow
- HSM requests QoS (LU-8324)
  - Prioritize restores over archive, typically

Conclusion

V3 is now available: where do we go?

- Sites
  - Deploy it
  - Imagine new policies, innovative use cases
  - Share your experience (good stories, bug reports...)
- Vendors
  - Explore the plugin architecture
  - Increase added value with new plugins

Thanks for your attention!

Questions?

Commissariat à l'énergie atomique et aux énergies alternatives
CEA / DAM Ile-de-France | Bruyères-le-Châtel - 91297 Arpajon Cedex
T. +33 (0)1 69 26 40 00

DAM Île-de-France

Backup: sample output (1/2)

Lustre-specific attributes

```
> rbh-report -e src/robinhood/rbh_find.c | grep lhsm

lhsm.status    :       released
lhsm.archive_id:       1
lhsm.no_release:       no
lhsm.no_archive:       no
lhsm.last_archive:     2016/09/13 13:28:39
lhsm.last_restore:     2016/09/15 08:35:17
```

Find files with a given HSM status

```
> rbh-find -status lhsm:released -lsstatus

[0x200000400:0x1a5dd:0x0] file 4672  lhsm:released /mnt/lustre/robinhood-3.0/README
[0x200000400:0x1a1df:0x0] file 27171 lhsm:released /mnt/lustre/robinhood-3.0/Makefile
...
```

Backup: sample output (2/2)

List undelete candidates in a directory

```
> rbh-undelete -L /mnt/lustre/robinhood-3.0

rm_time,  id, type, user, group, size, last_mod, lhsm.status, path
2016/09/15 13:45:03, [0x200000400:0x1a678:0x0], file, root, root, 26.26 KB,
2016/09/09 16:08:20, released, /mnt/lustre/robinhood-3.0/src/robinhood/rbh_du.c
```

Undelete a file

```
> rbh-undelete  -R /mnt/lustre/robinhood-3.0/src/robinhood/rbh_du.c

lhsm | Rebinding [0x200000400:0x1a678:0x0] to [0x200000400:0x1a9a5:0x0] in archive
lhsm | Executing rebind command: /usr/sbin/lhsmtool_posix --hsm_root=/tmp/backend
--archive 1 --rebind 0x200000400:0x1a678:0x0 0x200000400:0x1a9a5:0x0 /mnt/lustre
Restoring '/mnt/lustre/robinhood-3.0/src/robinhood/rbh_du.c'...  restore OK (file)
```