Robinhood initial scan improvement

Sergey Cherementsev
cherementsev@gmail.com

Robinhood

The Robinhood Policy Engine is an open-source tool that assists in the management of large file systems.

- Works on a lustre client
- Retrieves metadata(changelog data) from the FS(Lustre)
- Stores metadata in a DataBase (keeps an updated copy)
- Provides mechanism to query metadata from the database, rather than from the FS directly

Robinhood initial scan

Robinhood performs following steps on initial scan:

- **readdir** for each dir in the System
- **Stat** and **getstripe** for each file
- **Fgetattr** - only if the feature to collect id data from ea is enabled
- **path2fid** for each file

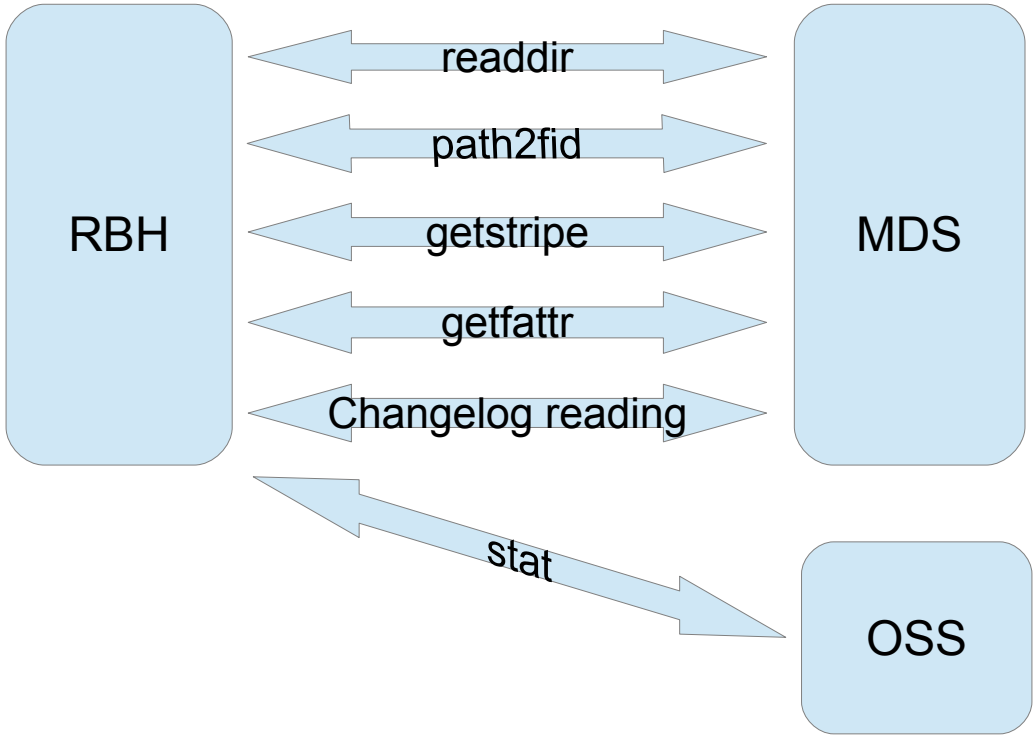
Initial filesystem scan is needed only when / if the relational copy becomes out of sync OR on initial scan.

Robinhood Changelogs reading

To keep the **DB up-to-date** RBH daemon **reads changelogs** from the MDS. It performs following actions for each changelog record:

- **Getfattr** (for hsm only)
- **Stat**
- **Getstripe**
- **Fid2path** (only when local DB doesn't match the FID) - very rare

RBH dataflow diagram



Changelogs

The changelogs feature records events that change on the file system namespace or file metadata. For example:

2 02MKDIR 4298396676 0x0 t=[0x200000405:0x15f9:0x0] p=[0x13:0x15e5a7a3:0x0] pics
3 01CREAT 4298402264 0x0 t=[0x200000405:0x15fa:0x0] p=[0x200000405:0x15f9:0x0] chloe.jpg
4 06UNLNK 4298404466 0x0 t=[0x200000405:0x15fa:0x0] p=[0x200000405:0x15f9:0x0] chloe.jpg

To save disk space changelog records contain FIDs instead of the full path name of the file.

Problems

Now on FS with millions of files RobinHood **initial scan** has to perform millions **path2fid, stat, fgetattr(optionally)** and **getstripe** requiring **a lot of time(days**). Also it produces excess network traffic and load on server.

- path2fid is used to convert paths to FIDs as changelogs records stores only FID
- Stat and getstripe give file's size and stripe location

Changelogs reading phase also produces extra RPCs. Fid2path, getstripe, getfattr and stat are called for all new changelog records from retrieved changelog.

Solution

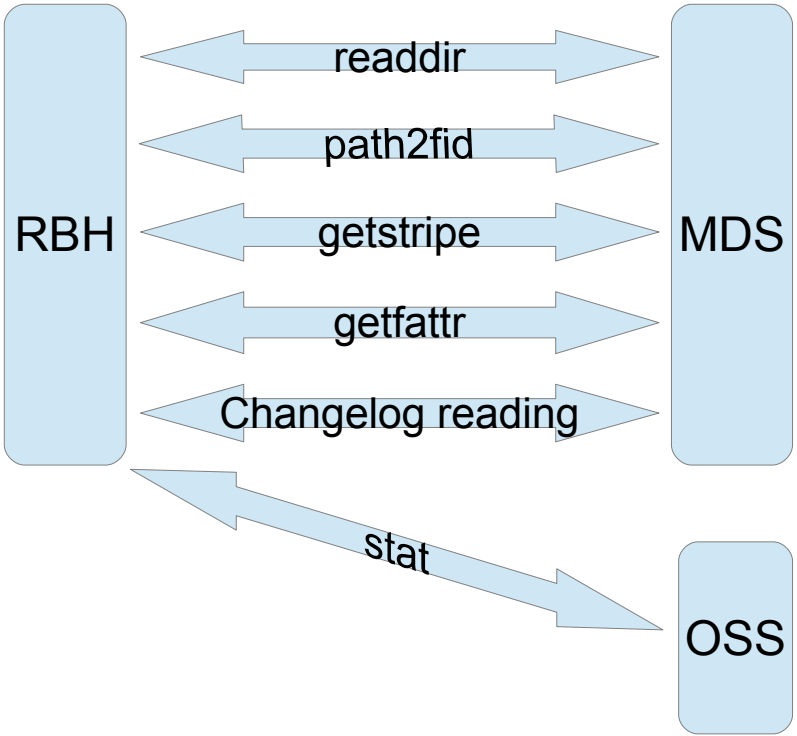
On initial scan phase MDT **goes through all inodes** and **makes** several **records** for each inode. RBH reads changelog from the MDT and populates own DB.

Introduce **new changelog record types** that may store additional attributes. One for inode(index object) attributes one for non-lustre specific XATTRS and one per each lustre-specific XATTR.

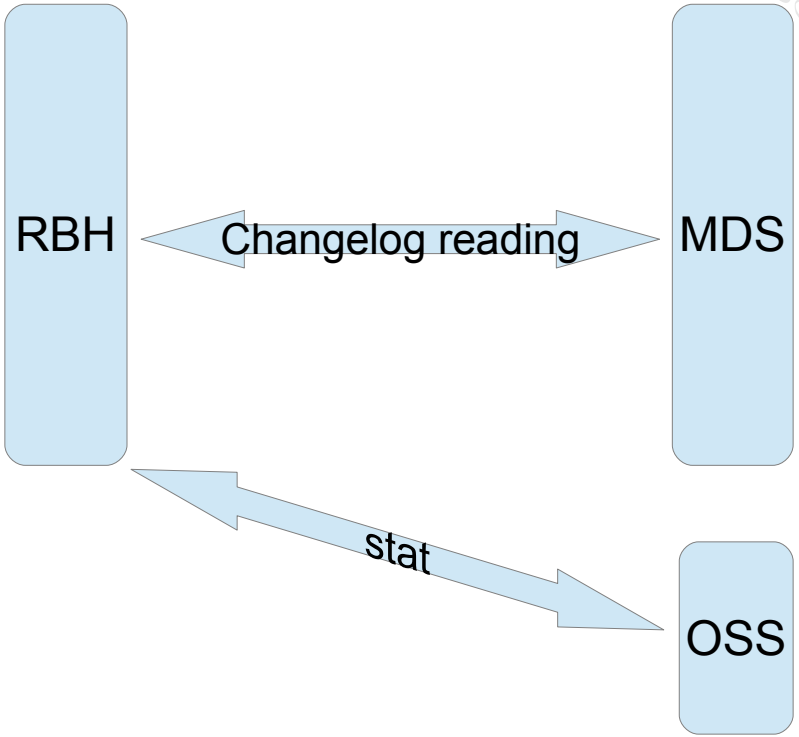
Initial scan populates data the same changelog as standard on-going operations.

- no need to do path2fid. Having PFID, FID and FNAME - full path names could be reconstructed based on this data
- no need to make extra requests to MDS(stat, fgetattr) - extended changelog records have already had that
- no new API - use default changelog reading API

Before



After



Solution details

Initial scan can be running in parallel with on-going operations.

RBH doesn't wait the end of inodes scanning and begins reading and parsing changelogs. As DB population is not finished RBH may face the FID that hasn't a record with full path name in DB. In such case RBH can call fid2path or delay this record handling.

Initial scan adds the following records for each inode:

CL_INODE_ATTR (fid, inode attributes)
CL_XATTR_BODY (fid, xattrname, xattrbody)
CL_XATTR_LUSTRE_*(**layout, hsm** …) - lustre specific EA

Initial scan adds the following records for each directory inode:

CL_CREAT (pfid, fid, name) for each direntry

On-going operations reuse new log records

Example of on-going operations:

- Creat operation: CL_CREAT, CL_INODE_ATTR , CL_XATTR_LAYOUT
- Setattr operation: CL_INODE_ATTR
- Setxattr operation: CL_XATTR

Llog size estimation after initial scan

- Xattrs maximum occupies **8KB**
- New changelog record header + file name + inode attr + lustre specific EA is about **1KB**

Ext4 supported maximum inodes number is **4GB**.

Thus additional **maximum changelog file size** is **36TB** (9Kb * 4GB).

With default striping (stripecnt = 1) xattrs need about 400 bytes. Thus without any extra non-lustre attributes maximum additional changelog size is about **1.6 TB**.

That shouldn’t be a problem as usually MDT’s disk space is used only for 10%.

If it is a problem we can use bytes per inode parameter to cover these 9Kb.

Possible improvements

Changelogs size optimisation - **filter** of not lustre specific **XATTRS by name**.
We do not necessary need all the EAs in the changelog. One of the needs is the HSM fid. Also RH may get a policy to handle some EA data.

Thanks for your attention !

Questions ?