Lustre Tricks You Probably Didn't Know

Andreas Dilger
Principal Engineer, Lustre Group
Program Agenda

- Power lfs find Usage
- Tools for Pools
- `filefrag` Reveals Fragmentation
- Interesting Tunables
- Recovery Tools
Power *lfs find* usage

*lfs_migrate* script

- Useful for rebalancing OST space usage
- Simple migration script *lfs_migrate*
- At most basic is simple copy + rename operation

```bash
#!/bin/bash
while read F ; do
    cp -a "$F" "$F.tmp" &&
    mv -v "$F.tmp" "$F"
done
```

- NOT currently safe for open or in-use files
- Version in manual, better version in bug 22481
- Hopefully will be in 1.8.4, and/or 2.0
Power *lfs find* usage

**OST rebalancing**

- OST000[2,4] too full, within the last couple of days

```bash
client$ lfs find /myth -type f -mtime -2 -size +2G
    -obd myth-OST0002 -obd myth-OST0004 | lfs_migrate
```

- OST000{5,6} are new/less full, move files TO them

```bash
$ lfs find /myth -mtime +90 -size +20G -name "*.iso"
    ! -obd myth-OST0005 ! -obd myth-OST0006 | lfs_migrate
```
Tools for Pools

Basic Commands

• Create a dedicated OST pool for classes of files

mgs# lctl pool_new myth.audio
mgs# lctl pool_add myth.audio OST0004
mgs# lctl pool_new myth.video
mgs# lctl pool_add myth.video OST000[0-3]
client$ lctl pool_list myth.video
Pool: myth.video
myth-OST0000_UUID
myth-OST0001_UUID
myth-OST0002_UUID
myth-OST0003_UUID
Tools for Pools

Other Commands

• Many Lustre tools have pools support

client$ lfs df -p myth.audio

<table>
<thead>
<tr>
<th>UUID</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Available</th>
<th>Use%</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>myth-MDT0000</td>
<td>9174328</td>
<td>178572</td>
<td>8471468</td>
<td>1%</td>
<td>/myth[MDT:0]</td>
</tr>
<tr>
<td>myth-OST0004</td>
<td>721984264</td>
<td>653299296</td>
<td>68684904</td>
<td>90%</td>
<td>/myth[OST:4]</td>
</tr>
</tbody>
</table>

Summary: 721984264 653299296 68684904 90% /myth

client$ lfs find -p myth.audio -uid ...

• Returns files created in myth.audio, not just OST0004
filefrag Reveals Fragmentation

- Can see layout of objects on OSTs

```bash
client$ filefrag -v tv/kids/foo.mpg

File size of tv/kids/foo.mpg is 1015406592 (991608 blocks of 1024 bytes)

ext: device_logical:  physical_offset:  length:  dev:
  0:  0..  2047:  45056..  47103:  2048: 0000:
  1:  2048..  4095:  38912..  40959:  2048: 0000:
  ...
  14:  0..  735:  78300068..  78300803:  736: 0004:
  15:  736.. 1023:  70399296..  70399583:  288: 0004:
  ...

tv/kids/foo.mpg: 49 extents found
Interesting Tunables

get_param and set_param

- Direct /proc access is discouraged
  - Use lctl get_param and lctl set_param for portability
  - Names map directly onto /proc/{sys,fs}/-{lustre,lnet}/pathname

$ lctl get_param version  #/proc/sys/lustre/version
version=
lustre: 1.8.2.53
kernel: 47
build: 1.8.2.53-CHANGED-2.6.16.46-0.15

- Avoids need for scripts to set multiple parameters

client$ lctl set_param osc.*.max_dirty_mb=32
osc.myth-OST0000-osc.max_dirty_mb=32
osc.myth-OST0001-osc.max_dirty_mb=32
Interesting Tunables

How to List Tunables without *ls /proc*

```
client$ lctl get_param -NF osc.*.*
lctl get_param -NF llite.*.*
llite.myth-ffff88006c8cc000.blocksize
llite.myth-ffff88006c8cc000.checksum_pages=
llite.myth-ffff88006c8cc000.contention_seconds=
llite.myth-ffff88006c8cc000.direct_io_default=
llite.myth-ffff88006c8cc000.dump_page_cache
llite.myth-ffff88006c8cc000.extents_stats="
llite.myth-
    ffff88006c8cc000.extents_stats_per_process=
```
...
Interesting Tunables

Client Import State

client$ lctl get_param osc.*.import
osc.myth-OST0000-osc.import=
import:
    name: myth-OST0000
    target: myth-OST0000_UUID
    current_connection: 192.168.20.1@tcp
    state: FULL
    connect_flags: [write_grant, server_lock, ..., early_lock_cancel, adaptive_timeouts, lru_resize, alt_checksum_algorithm, version_recovery]
    import_flags: [replayable, pingable]
Interesting Tunables

Client Import State, cont.

connection:
  connection_attempts: 69
  generation: 1
  in-progress_invalidations: 0

rpcs:
  inflight: 0
  unregistering: 0
  timeouts: 67
  avg_waittime: 32234 usec

service_estimates:
  services: 1 sec
  network: 1 sec
Interesting Tunables

Client Import State, still cont.

transactions:
  last_replay: 0
  peer_committed: 55834582048
  last_checked: 55834582048

read_data_averages:
  bytes_per_rpc: 829101
  usec_per_rpc: 33809
  MB_per_sec: 24.52

write_data_averages:
  bytes_per_rpc: 605745
  usec_per_rpc: 41162
  MB_per_sec: 14.71
Interesting Tunables
Server Export State

- Per-client brw_stats
  oss$ lctl get_param mds.*.exports.*.brw_stats

- See which client NIDs are connected
  mds$ lctl get_param -NF mds.*.exports.*
  mds.myth-MDT0000.exports.192.168.20.153@tcp/
  mds.myth-MDT0000.exports.192.168.20.159@tcp/
  mds.myth-MDT0000.exports.clear=
Interesting Tunables
Server Export State, cont.

• Map client UUIDs to NIDs
  mds$ lctl get_param mds.*.exports.*.uuid
  mds.myth-MDT0000.exports.192.168.20.153@tcp.uuid=
    31007da1-a19f-6537-15df-8a6cbc6f9342
  mds.myth-MDT0000.exports.192.168.20.159@tcp.uuid=
    99ca0c3f-f91b-8ee6-28c2-891101d95256

• Evict clients by UUID
  mds# lctl set_param mds.*.evict_client=
    31007da1-a19f-6537-15df-8a6cbc6f9342

• Evict clients by NID
  mds# lctl set_param mds.*.evict_client=
    nid:192.168.20.153@tcp
Interesting Tunables

Cache Tuning

• Limit client-side memory usage
  
  client# lctl get_param llite.*.max_cached_mb
  llite.myth-ffffff88006c8cc000.max_cached_mb=1501
  client# lctl set_param llite.*.max_cached_mb=512
  llite.myth-ffffff88006c8cc000.max_cached_mb=512

• Read entire “small” file into cache on first access
  
  client# lctl set_param
    llite.*.max_read_ahead_whole_mb=5.5
  client# lctl set_param
    llite.*.max_read_ahead_per_file_mb=10

• Cache only files < 6MB on OSS, avoid cache thrashing
  
  oss# lctl set_param
    obdfilter.*.readcache_max_filesize=6M
Interesting Tunables

Permanently Storing Tunables

- Permanently store parameter in configuration file
  - Has a slightly different syntax than get_param, set_param
  - Immediately sent to all clients

mgs# lctl conf_param {fsname}.{subsys}.{param}
mgs# lctl conf_param
  myth.obdfilter.readcache_max_filesize=6M
mgs# lctl conf_param
  myth.llite.max_read_ahead_whole_mb=5.5

client# dmesg | tail -1

Lustre: Setting parameter

  myth-client.llite.max_read_ahead_whole_mb

in log myth-client
Recovery Tools
lustre enhanced tar

- Small change to RH/FC tar to backup/restore lustre xattrs
- Accepted upstream into FC13, hopefully RHEL6
- Restores stripe_{count,size,pool} before restoring data

client# lfs getstripe -c *stripe
1stripe: 1
2stripe: 2
3stripe: 3

client# tar czf --xattr * | tar xzvf - -C tmp/
client# lfs getstripe -c *stripe
tmp/1stripe: 1
tmp/2stripe: 2
tmp/3stripe: 3
Recovery Tools

**ll_recover_lost_found_objs**

- `fid` xattr set on every OST object on first client access
- Useful in case of OST corruption, and later `lfsck`
- `e2fsck` moves unreferenced inodes to `lost+found`
- `ll_recover_lost_found_objs` gets object ID from `fid`
- Rebuilds O/0/d{0,31} directory hierarchy, `LAST_ID` file
- OST mount creates most other files (especially `last_rcvd`)

```bash
oss# mount -t ldiskfs /dev/vgmyth/lvmythost0 /mnt
oss# ll_recover_lost_found_objs -d /mnt/lost+found
oss# umount /mnt
```
ORACLE IS THE INFORMATION COMPANY
The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle’s products remains at the sole discretion of Oracle.