More efficient Lustre-HSM

Cory Spitz – Cray Inc.
LAD developer’s summit
9/22/2016
Anticipating the need for efficient Lustre-HSM

● The use case: Robinhood processing ChangeLogs

● The problem: scaling is a concern
  ● More and faster metadata is coming (multi-slot last_rcvd & DNE)
  ● We desire a single RH instance to keep coherence

● After processing a ChangeLog entry, RH needs to query for file size, ownership, striping, etc.

● These queries are not batched
  ● Adds latency
  ● Provides an additional metadata servicing burden
  ● These requests compete for QoS
  ● Results in higher server lock volume

● Result: Robinhood use-case needs help scaling
Potential solution

- Can we extend ChangeLogs to provide the information?
- MDS already knows striping
- Size and mtime is a challenge
- What else should we consider?
- We’ve tried SoM in the past and abandoned it
- Could we make another (similar) attempt?
Lazy SoM or stat implementation

- Store size in xattr on MDS. Store mtime too?
- Don’t worry about correctness (aka open files)
- Potentially not retrieved for stat(), only for ChangeLog
  - Not for open + create until close
- Client driven or OSS driven?
  - Could client send KMS to the MDS upon close()?
  - Could OSS send object size to the MDS upon object close?
Is this feasible?

Or, what are the gotchas?