Fujitsu Contributions to Lustre*

High Performance Data Division

Oleg Drokin

April 16, 2013

* Other names and brands may be claimed as the property of others.

Network jitter: Doing away with pings

- On large systems pings are expensive:
  - Clients * targets pings every obd_timeout/4 interval (default 25 sec)
- Main purposes of pinging:
  - Lets clients detect restarted/recovering servers in reasonable time
  - Proactively weeds out unreachable/dead clients
- With Imperative Recovery we’ve got #1 covered
- Many existing systems already know about dead clients from cluster management tools
  - Lustre provides a way for those systems to tell it about dead clients for immediate eviction
- Now servers have a way to tell clients to avoid idle pinging

LNet routes hashtable

- It was noticed that LNet stores routing entries in a linked list
- As number of routes increases on large systems, iterating the list becomes more and more expensive
- Hash table is a pretty natural solution to this problem

Limiting OS jitter – ldlm poold

- On FS with 2000 OSTs `ldlm_poold` was using 2ms of cpu every second on every client
  - Investigations revealed it was walking a linked list of all ldlm namespaces (one per connected service) every second to update lock stats
- The lock statistics on empty namespaces do not change
  - So no need to walk empty namespaces at all
- An updating action is performed every 10 seconds on clients
  - So no need to wake up every second, just see how much time left till next action and sleep this much
- A lot of the calculations don’t need to be periodic and could be predicted, making `ldlm_poold` pointless (TBD)

SPARC* architecture support

- SPARC architecture is big-endian
  - Fujitsu performed a full Lustre* source audit for endianness issues and contributed the results back to the community
- SPARC Linux has “different” error numbers (Solaris* compatible)
  - This highlights a bigger problem of assuming the error numbers being compatible on different nodes in network which is not true.
  - Fujitsu came with an errno translation table solution that it contributed back to community
    - Intel is working on integrating this solution into 2.x releases
- Fujitsu also contributed access to a SPARC system test cluster

Memory usage improvements

- `/proc` statistics on clients tends to use a lot of RAM
  - Esp. if you have thousands of targets connected, it could use hundreds of megabytes
- Fujitsu developed and contributed a way to disable such statistic tracking
  - Being adopted by Intel for inclusion into Lustre 2.x

More fine-grained control of striping

- Current Lustre striping of “starting at X, Y wide” is not always adequate
- Fujitsu developed and contributed code to allow very-fine-grained stripe allocation on per-OST basis
  - This is currently being adopted by Intel into inclusion into Lustre 2.x
- Additionally, assumption about contiguous OST numbering is also removed which would allow for flexible OST-numbering schemes

intel®