CUG 2008 HELSINKI • MAY 5–8, 2008
CROSSING THE BOUNDARIES

Stabilizing Lustre at Scale on the Cray XT

Nicholas Henke
Cray Inc.

CRAY
THE SUPERCOMPUTER COMPANY

Introduction

- Lustre as a system resource
- The XT is unique
  - Compute core count & Lustre clients
  - Storage density
- How does Lustre affect system stability?
  - System downtime or job loss

Lustre Connections and Timeouts

- LNET
  - NID to NID peer connections
- Lustre
  - UUID
  - Client to service connections
  - Multiple ride on single LNET connection
- LNET Timeouts
  - provides for hardware health and progress
- Lustre Timeouts
  - Service responsiveness
  - Trigger for resend and failover
  - obd_timeout of 300s
    - Other timeouts determined from this value

Scaling Challenges Overview

- Configuration
  - Storage density and OSTs
  - OST per OSS ratio
    - Only place to change load balance
- Compute node count
  - Lustre clients on each
- Example configuration
  - 320 TB in 160 OSTs on 20 OSS
  - 9,500 sockets, 19,000 compute cores
    - 19,000 lustre clients on CVN

Lustre RPC Load and Processing

- Simpleio
  - Open, write, read, close to unique file per process
- Mount timeouts
  - 152,000 connections per OSS, 3.04M total connections
  - 15 second Lustre timeout
  - Requires OSS processing rate of 10,133 RPCs/sec
- Timeout adjustment
  - 300s for connect, 600s for I/O
  - Observed processing rate was ~700 RPCs/sec
    - 214s to process connect

RPC Processing, cont'd

- UUID searches
  - Check done for each new connection to prevent duplicates
  - Scans linked-list, compares each UUID
  - Each OST performs 180,500,000 comparisons for 19,000 client job
    - 1,444,000,000 comparisons per OSS with 8 OSTs
- Hash table
  - 128 keys
  - Down to 11,017 comparisons per OST
    - 88,134 per OSS

Configuration Implications

- Compared to similar size system
  - 23,000 clients, 288 TB on 144 OSTs on 72 TB
- 8 OSTs per OSS was the tipping point
  - Effectively trebling processing load
- System was reconfigured to help relieve problem
  - 8 TB max LUNs and OSTs
  - Used 4TB, dropped OST count to 80
    - 4 OSTs per OSS
- Failover doubles the ratio
  - 16 OSTs per OSS would drastically reduce failover success

2nd Order Effects

- Heavy I/O jobs show Portals timeouts from Lustre
  - UUID fixes not in place yet, just tuned timeouts
- Portals traces show no problems
- Lustre not getting LND credits
  - Messages stuck in Portals -> LND queue
  - Lustre threads taking all of CPU
- Lustre threads yield to scheduler
- Add timestamp to LND messages
  - Prevent false subsystem errors

Interactions with XT Subsystems

- RCA Heartbeat
  - Lustre threads stealing all cycles
  - RCA thread not able to schedule reliably
  - Bump priority of RCA thread
    - Add “tickles” in hot Lustre and kernel paths
- Console messages
  - Lustre is chatty on errors
  - Load is similar to RPC generation
  - Flooded CRMS network, lost control of machine
- Future protection
  - CRMS flood protection
  - Lustre message rate limiting & quieting

On the Horizon

- Learning from experiences
  - “Thinking in Lustre”
- Lustre: CVN to CNL
  - CVN
    - No locks
    - Client per core
  - CNL
    - Locks and their timeouts
    - Client per node
- Adaptive Timeouts
  - Neat feature, but need to verify it scales

Lustre Isn't So Bad…

- …It is what we do to it!
- The late Richard Gisselquist had it right from the beginning

"Lustre Is the Name of a Software Product

For Jim who thought I couldn't complete this
without mentioning General Custer

Called up in the lustre muster
For duty as a lustre buster,
He began.

Taking aim at lustre cluster,
He unleashed the lustre thruster
He tested.

Quieting the lustre bluster
Tarnishing the lustre luster,
He found fault."