FIO

CERN IT Department

CERN Lustre Evaluation

Arne Wiebalck

HEPiX Spring Meeting
Umeå, 27th May 2009

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Agenda

- Motivation and methodology
- Areas of investigation
- Initial findings
- Current thoughts
- Conclusion

Why a Lustre evaluation?

CERN IT Department

Lustre:

- Performance
- Scalability
- HEPiX FSWG

CERN:

- Home directories & projects on AFS
- CASTOR as HSM
- xrootd plus CASTOR for analysis

Lustre: An opportunity for consolidation?

Does Lustre meet the requirements?

Can Lustre be operated?

CERN - IT Department
CH-1211 Genève 23
Switzerland
www.cern.ch/it

Project Plan

- ✓ Assemble a list of points to look at
  - ✓ Understand requirements of the use cases
  - ✓ Manageability
- ✓ Gather information
  - ✓ Lustre training
  - ✓ Attend LUG
  - ➢ Exchange experiences with other sites
- ➢ Get hands-on experience
  - ➢ set up test instances
  - ➢ familiarize with certain functionality
  - ➢ integrate with CERN's tools for fabric management
- Document the findings

Use case requirements

CERN IT Department

Home directories/ Project space

- Strong authentication
- Backup
- Wide-area access
- Small files
- Availability
- ACLs & Quota

HSM

- Generic interface, CASTOR/TSM
- Scalable
- Support for random access

Analysis space

- Low latency access (disk-based)
- O(1000) open/ sec
- Several GB/s aggregate bandwidth
- ~10% Backup
- Mountable plus xrootd access
- ACLs & quota

Areas of investigation

- Strong authentication
- Backup
- Fault-tolerance
- Special performance & Optimization
- HSM interface
- Life cycle management (LCM) & tools

Strong authentication

- v1.8: no Kerberos
- v2.0: early adaptation, v2.x full Kerberos
- LUG: „robust“ and „usable“
- implementation not (yet) complete
- PSC early adopter, WAN setup

Backup

- Meta data (MDT)
  - × tar of extended attributes (EA)
  - ➢ LVM snapshots

  - ? Orphan files, ghost files

- User data (OST)
  - × name space crawling or tar of EA
  - ✓ versions before 2.0: e2scan
  - ➢ with version 2.0: changelogs

Fault-tolerance

- Lustre comes with support for failover
  - active/active for OSSs
  - active/passive for MDS
  - Used at other sites
- Technology
  - Shared storage: FC, DRBD, iSCSI
  - Heartbeat

Special performance

- Small files
  - No client-side caching, 1MB transfer unit
  - How bad is it?
  - Recent patches
  - Recent improvements (OSS read caches)
- Understand tuning options
  - FSWG is looking into this as well

HSM

- CEA (J.-C. Lafoucrière, A. Degrémont)
  - interface design
  - implementation for HPSS
  - Sun SAM-QFS
  - FNAL (Enstore)
  - beta version available later this year

Life cycle management

- Adding capacity (add an OSS)
  - Possible, but quiesce clients (?!)
  - Allocation policy, coordinated filling
- Removing capacity (remove an OSS)
  - no user-transparent data migration
- MDT (re-)sizing
- Managing Quota
- Develop procedures / tools
  - Lustre upgrades
  - System upgrades (kernel)

Current thoughts (1/2)

- Client-server coupling
  - recovery
- Too powerful users
  - striping, pools
- (Missing) Support for life cycle management
  - Data migration
  - Version compatibility
  - Upgrades & down times
- Moving targets on the roadmap

Current thoughts (2/2)

- New versions bring some helpful features
  - v1.8: VBR, adaptive timeouts, OSS read cache
  - v2.0: Kerberos, change logs, Clustered MDS
- Increased emphasis on Quality Assurance
- Milestones-based pre-releases
- Responsiveness of Lustre team
- Interest in non-performance features by other sites

Summary

- Lustre is an interesting file system
- Investigation on manageability is missing
- Goal: report at autumn HEPiX
- Input is welcome, share your experiences!