Backup and Recovery for Petascale File Systems

Malcolm Cowe, Solutions Architect

High Performance Data Division

October 2014

LEGAL DISCLAIMERS

Lustre's Key Strengths
Massive Performance at Massive Scale

512PB capacity

TB/sec aggregate I/O

1000s of connected machines

The Administrator’s Conundrum

How do I back up all that data?

Define the Requirements

What form of data protection are you trying to achieve?

- Versioned backup?
- Archive?
- Storage redundancy?
- Online replicas?

Understand the Constraints

What is the backup and/or recovery window?

What is the change set size?

How much bandwidth is available?

- Do not assume that there is an infinite supply of archive storage or bandwidth

Be Selective

Scope the backup and recovery requirements

- Include data management processes, backup window, SLA for recovery

Define critical data

- Put a value on the data and prioritise accordingly
- Determine the minimum data set required to restore functional service
- Decide what data can be ignored (e.g. temporary files)

Establish the recovery window requirements

- Determine the infrastructure required to manage recovery within the SLA

Be Realistic

For example: to back up 10 PB in a 4 hour window

- Requires sustained bandwidth of 700GB/sec (2.5 PB/hour: ~694GB/sec)
- This is unlikely to be feasible in many environments

What is the cost of losing data?

- Value of the data itself, as well as the cost per hour of being "down".

What is the recovery window?

- Can the data be re-generated?

How much can I back up or restore per hour?

Options

Tape

Reliable, scalable, cost-efficient long term data storage medium

- Backup
- Near-line Archive / HSM

Tape is still relevant

- Online storage capacities increasing dramatically
- Tape systems can meet demand to support (archive systems are being measured in 100s of PB)

Backup to "Tape"

LTO-6 can support up to 160MB/sec sustained transfer rate

- Uncompressed capacity is 2.5TB
- Performance and capacity approximately equivalent to a single spinning HDD

160MB/sec ~= 13.8TB/day

- 1PB would take nearly 73 days to backup through a single drive unit plus overheads for robots to change tapes
- Compression improves performance
- What is the expected duration of the backup window?
- Full back up of 1PB of data in a 24 hour window would require 73 drives, delivering 11.7GB/sec aggregate throughput, plus supporting infrastructure

Tape as Archive

Tape is a strong vehicle for large scale archival and long term data retention

- Can provide high capacity near-line storage
- Increasingly used to complement large scale on-line storage
- May be tightly integrated with Lustre (eg. via HSM) or loosely-coupled

Archive != Backup

- Backup: short term, versioned copies of active data for point in time recovery. Best suited for active data sets where loss of data requires prompt recovery
- Archive: long term retention of infrequently used but permanent production data. An indexed library of digital assets

HSM

Hierarchical Storage Management (HSM) is principally a capacity management and archival platform, rather than a backup system

- Tiered storage to manage the balance between high performance and high capacity, long term storage requirements
- Targeted and automated archival of data to long term storage
- Automated, on-demand recall
- Lustre for high performance closest to the application, where it is needed
- Longevity, capacity, retention in the archive
- Lustre for very active data sets, archive for infrequently accessed data

“Tape” Backup Pros and Cons

Pros

- Ubiquitous: used everywhere, mature, well understood
- High capacity, low power footprint
- Media longevity
- Enterprise backup workflows typically include off-site storage for DR

Cons

- Throughput performance does not match Lustre, lengthening backup window
- Recovery window likely to be insufficient on its own to meet SLA for DR

Archive / HSM as “Backup” – Pros and Cons

Pros

- Seamless integration of online and near-line storage, balancing performance and long-term capacity and retention needs
- Single name space to address all data
- On-demand recall, transparent to applications

Cons

- Archives are not a backup solution but may be used in complement
- Versioning is implementation dependent, not always available
- Full restore from Archive has the same constraints as traditional backup

Snapshots

Snapshots provide a means to roll-back storage volumes to a previously known-good version

- Some capacity is reserved in the primary storage to allow versioning (10-20% is typical)
- Common use cases:
  - Temporary view for creating an off-line backup; snapshot is destroyed on completion of backup
  - Persistent, rolling online “backups” of the file system for fast recovery
  - Create a copy of current, “known-good”, state prior to upgrade

RSnapshot is a wrapper around rsync that provides pseudo-snapshot

- http://rsnapshot.org

Snapshots – Pros and Cons

Pros

- Online versioning of data held on primary storage
- Fast recall of previous version
- No additional infrastructure required (only reservation of additional capacity)

Cons

- Does not increase hardware redundancy or provide DR capability
- Negative performance impact when used with Linux LVM
- Coordination of distributed storage resources for consistent snapshot
- Process for mounting snapshots tricky in Lustre today

Duplication / Replication

2 or more copies of critical data, stored redundantly on independent hardware

- Primary and secondary storage typically employ equivalent technology
- Synchronous or asynchronous
- Copies may be local or remote
- Often used in support of Disaster Recovery

Block-level Replicas

Mirroring of the storage data blocks across independent devices / LUNs

- Provides additional protection against hardware failure
- Mirrors may be local (same site) or remote (for multi-site DR)
- Can be provided in storage hardware or in conjunction with software

iSCSI, SRP, DRBD are device-independent options

Vendor-specific solutions also available

DRBD / iSCSI / SRP

iSCSI and SRP are general-purpose protocols

- Standards-based presentation of block devices to networks
- No native support for replication – use additional software layer (LVM or MD-RAID)
- Synchronous only? Performance impact when running multi-site?

DRBD is specifically designed to provide block device mirroring

- Supports both synchronous and asynchronous transfers
- Local and remote mirroring
- More commonly associated with low-cost HA solutions for databases
- Commercial support available

Block Replication – Pros and Cons

Pros

- Block-for-block copy of all storage targets, providing automatic replication of all data held on the file system
- IP-based storage network straightforward to implement, support
- Complete hardware redundancy providing additional failure protection

Cons

- Doubles storage costs; physical LUNs must also be fault tolerant (RAID61, 101)
  - May also increase networking costs
- Synchronous replication adds latency, as writes must complete on both copies before returning
- For multi-site DR, failover from primary to [remote] secondary copy is complex
- Not a backup: no data versioning

File Level Replication

Opportunity to provide scalable online, “versionable” backup

- Asynchronous copy of files from primary to independent secondary target
  - Primary and secondary file systems may be in mutually isolated locations
  - Secondary may allocate a subset of available storage for receipt of copies (remainder can be used for other purposes)
- Replica on secondary serves two functions:
  - Failover site for disaster recovery
  - Hot backup of critical data for fast recovery of files
- During failover, the secondary file system becomes the new primary
- Requires capability to synchronise in the other direction for recovery

RSync – Original poster-child for efficient replication

Why not just use RSync?

- Versatile application, mature and has served sysadmins well for a long time, but has limited scalability
  - Single process [per data mover], single client throughput limits transfer of large files
  - Must walk file system tree to build file list
  - Workarounds exist to address number of files but not file size

lustre_rsync serves a similar purpose, but is not directly related

- Consumes changelogs; does not require tree walk if primary and secondary are initially identical
- Not a parallel application
- Fewer options compared to rsync

Strong concept, but how to implement?

Approach

- Build file list
  - Walk the tree once for initial sync (e.g. lfs find)
  - Identify files that have changed for subsequent iterations (Lustre Changelogs)
- Split file list into evenly sized chunks for distribution across data movers
- Copy files from primary to secondary, creating backups of existing files as required
  - Parallelise the copying of large files across multiple nodes
- Submit replication tasks to job scheduler, monitor for failures and re-run as required
  - May also be driven by a policy engine

Problem: Finding a scalable, parallel copy

Profusion and confusion of options for copying files in parallel

- BBCP, dcp, fpart, gridftp, bittorrent, FDT, UDR/udt, Unison, pcp, mutils
- Commercial suites
- Limited options for versioned replication (cf. rsync --link-dest)
- Limited support for Lustre attributes

File Replication – Pros and Cons

Pros

- Files held in online backup, easily accessed, fast time to recover
- Supports DR
- Secondary copy can be held on any POSIX target

Cons

- RSync does not provide scalable performance, alternatives rare
- Existing parallel copy tools may not support versioning, incremental copies
- Asynchronous. No real-time options

Summary

Understand your data

- Identify critical data sets (active data, archive data, scratch)
  - Number of files?
  - File size? (min, max, average)
  - Rate of change?

Identify the organisational requirements

- Recovery window / SLA?
- DR?
- Backup or Archive? Both?

Recovery is key