Guiding Principals

• Stability
  > A technology preview is not a product

• Benchmarking
  > Evidence based decision making

• Interoperation
  > It’s not optional

• Execution
  > Deliver the roadmap
Delivering Stability

• Development Process
  > From architectural requirements through to code
  > Release “gate”

• Restructuring / Refactoring
  > CLIO
  > Porting APIs

• Improved Test Facilities
  > New test cluster in Broomfield
  > REP system
  > Test Automation
  > We need YOU!

• Conservative Feature Rollout
  > Not until it’s ready
Interoperation

• We hear you!
  > Site-wide shutdown is unacceptable
  > Cluster-wide shutdown is unacceptable
  > Deployment intractable without “version smear”
  > Different versions may need to interoperate for weeks

• Guarantee
  > Node-by-node upgrade
  > Rolling upgrade path always possible

• Fine Print
  > Arbitrary version interoperation not guaranteed
  > Possible reduced performance on version mismatch
  > Node upgrade order may be prescribed
  > Downgrade may not be supported
Upcoming Releases
And Release Numbering

• 1.6.5 – Imminent
  > Bug Fixes, Minor improvements

• 1.8 – Fall
  > New Features
    – 2.0 Interoperability
    – Recovery Improvements

• 2.0+ – End of Year
  > Major New Features
Adaptive Timeouts

• RPC timeout => server death
  > On a large cluster (10,000s of nodes), extreme server load indistinguishable from death.
  > Site tunables

• Adaptive Timeouts
  > Client adapts timeouts to observed service times
  > Server pre-empts timeouts with “early” replies
  > Eliminate tunables
  > Increase responsiveness
Version Based Recovery
Recovering Uncommitted Client RPCs on Server Restart

• Current recovery
  > All clients replay in original execution order
  > Fixed recovery window – late clients lose
  > Transactions after a “gap” lose

• VBR
  > Recovery transaction checks object version
  > “Gaps” not fatal
  > Clients may reconnect late
  > COS resilience/performance tradeoff
ZFS

• Easier Administration
  > Pooled storage model
  > No volume manager
  > Snapshots

• Immense Capacity
  > 128-bit file system

• End-to-end data integrity
  > Copy-on-write, transactional design
  > Everything checksummed
  > MD block replication
  > RAID-Z/Mirroring
  > Resilvering
Lustre ZFS Performance Today

Considerable improvement is required but it's doable!
Lustre ZFS Performance Today

Considerable improvement is required but it's doable!
Lustre ZFS Performance Today

Throughput as a function of PIOS threads

With Zero Copy (simulated – but it’s doable!)
ZFS rollout

• Initial ZFS release
  > Only for new file systems
  > Client works with both ldiskfs and ZFS servers

• Later ZFS releases
  > Online OSS migration via space management tools
    – Add ZFS OSTs
    – “Empty” ldiskfs OSTs
    – Piecemeal or wholesale
  > Offline MDS migration via conversion utility
  > Online MDS migration still an open issue
    – CMD
    – ldiskfs EOL
Request Visualisation
Request Visualisation
Network Request Scheduler

• Today, requests processed in FIFO order
  > Only as fair as the network
  > Over-reliance on disk elevator

• NRS re-orders RPCs on arrival
  > Enforce fairness
  > Working set == buffered RPCs not # service threads
  > Work with block allocator

• Global NRS to coordinate servers
  > QoS
Simulator

- Discrete Event Simulator
  - Simplicity v. Accuracy
  - 100K + node simulations

- Component Models
  - Client
  - Network
  - Server side request scheduler
  - Backend F/S
  - Disk Elevator
  - Disk