Exascale Distributed File Systems

MSST 2010
Brent Welch
May 4, 2010

Key Ideas

- Storage is Hard
  - Never fail, ever scale, wire-speed goals
  - Built from low-cost, flakey hardware
- Fault handling is the key to building large systems
  - Performance comes naturally if you can scale up
- Panasas layers its parallel file system on top of its distributed system platform
- Some Ideas about more sophisticated error handling

Background

- Large scale parallel file systems
  - Lustre – research and academia
  - PVFS – research and academia
  - GPFS – research and commercial
  - Panasas – commercial and research
- Largest Panasas single storage cluster in production
  - 2 PB, 60 GB/sec
  - 1000 storage blades, two disk drives and 1GE each
  - 100 manager blades
  - 100 blade chassis, integrated UPS and 10GE switch
  - LANL RoadRunner

Stability vs. Performance

- You get what you pay for
  - Commercial deployments demand reliability and manageability
- It is easier to add performance optimizations on top of a stable platform, than it is to stabilize an unstable (but fast) platform
  - We know – we've been fast and unstable
  - LANL didn't care so much
  - Intel/Disney/Boeing/Citadel cared a lot
  - Intel probably has more practical computing power dedicated to a single application (chip tape out) than most super computers
- Don't worry – competition will drive down prices

Error Recovery in File Systems

- 80% of code is about failure recovery
  - First class error recovery logic, diagnostics, etc
  - (untested) error paths, with peer review as first line of defense
  - Massive test suites, which are tricky to write
- Panasas cluster manager vs. file system meta data mgr
  - Distributed system platform clearly factored from file system
  - PanFS metadata manager is “just another service”
  - Panasas cluster manager manages services and failures

Panasas Distributed System Platform

- Distributed File System layered on top of robust quorum-based, out-of-band Cluster Management protocols
  - PTP (Paxos) voting and a replicated configuration database
- Responsibilities of the platform
  - Tracks hardware and software components
  - Activates services, triggers fail over
  - Admits new hardware and decides if it is dead
  - Handles power up, power down, reboot, upgrade, etc
  - Monitors hardware faults, over temp, AC power etc.
- The platform doesn’t know much about file systems
  - And certainly doesn’t participate directly in FS operations

Panasas Distributed System Platform

- Decide – Control – Monitor
  - Commit tentative decision via a PTP (Paxos) transaction
  - Control distributed system elements (services or blades)
  - Conclude operation with a final PTP transaction
  - Monitor and re-evaluate as necessary (periodic “sweepers”)
- Cluster Manager evolution of Blade States
  - Started with a simple [Online, Not Responding, Dead] states
  - Now: Booting, Self-Test, Off-Version, Low-Battery, Upgrading, Online, Offline, Software Failed, Hardware Failed, Factory Mode, Unavailable (and why)

Error Handling Semantics

- We need new responses to errors
- RAID will handle disk failures, and we’ll be at M+N redundancy
  - But RAID will fail
    - so many controllers, some will die and their fault handling won’t actually work
- Network will have redundant paths
  - But the Network will fail
    - too many switches and cables, and the fault handling won’t actually work
- The File System software will have to deal with it

Some Ideas

- Always On Availability Model
  - Any "+N" fault model generally turns off completely if there are >N failures
  - Techniques like declustering spread out fault domains and yield graceful degradation like "99.5%" availability of the data
- Write steering around failures
  - New data can avoid dead spots in the storage system
- Background addition of more resilience
  - Additional copies, or archival/remote copies can be spawned in the background and fetched to compensate for dead spots

Extra Material

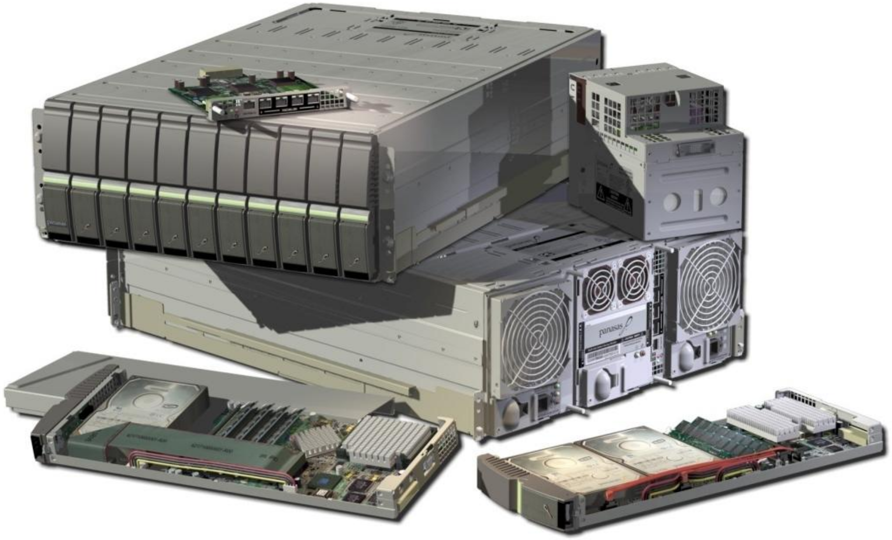
MSST
Brent Welch
April 27, 2010

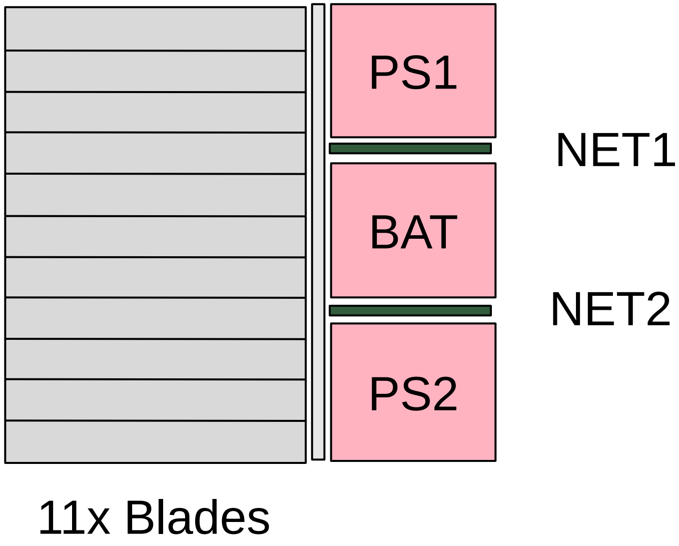
Metadata service resilience

- PanFS metadata managers maintain transaction logs
  - Battery-protected memory, replicated over network to backup
    - Heavy reliance on the cluster of metadata servers
  - Clients are second class citizens
  - OSD are almost completely dumb
    - Maintain an error (i.e., “fence”) bit for each object
- No FSCK – most repairs are online
  - The rest can be deferred indefinitely

A-Series 4th Generation Blade

2002	850 MHz / PC 100	80 GB PATA	
2004	1.2 GHz / PC 100	250 GB SATA	330 MB/sec
2006	1.5 GHz / DDR 400	500 GB SATA	400 MB/sec
2008	10 GE shelf switch	750 GB SATA	600 MB/sec
2009	SSD Hybrid	1000 GB SATA, 32GB SSD	600 MB/sec
2010	**1.67 GHz / DDR3 800**	**2000 GB SATA, (*64GB SSD*)**	**~1 GB/sec**





PS1

NET1

BAT

NET2

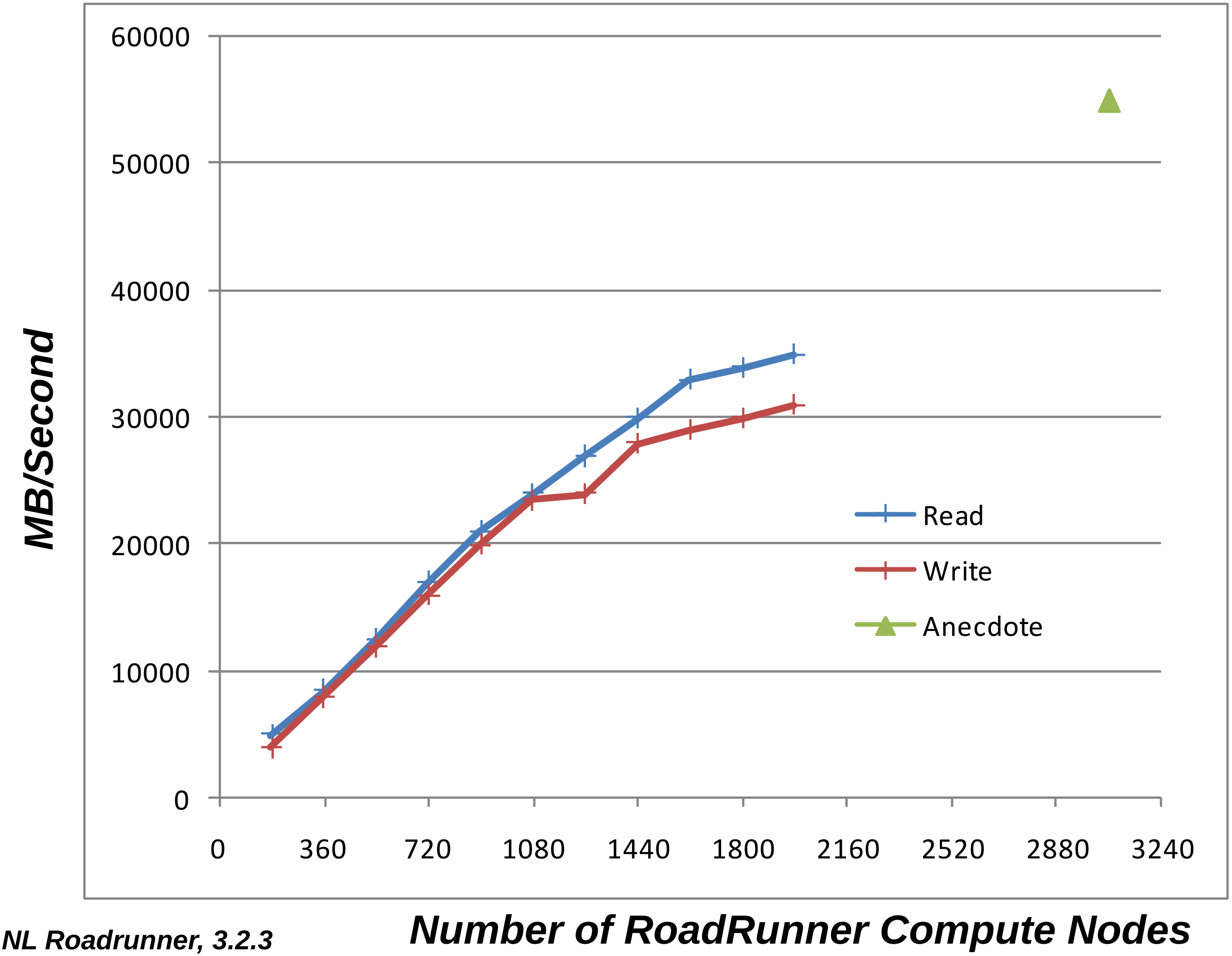
PS2

11x Blades

Panasas Features

- Object RAID (2003-2004)
- NFS w/ multiprotocol file locking (2005)
- Replicated cluster management (2006)
- Declustered, Parallel Object RAID rebuild (2006)
- Metadata Fail Over (2007)
- Snapshots, NFS Fail Over, Tiered Parity (2008)
- Async Mirror, Data Migration (2009)
- Hybrid Blade (2009)
- 64-bit multicore (2010)
- User Group Quota (2010)

Scaling Clients (100 shelves)



"Anecdote" is a 55 GB/sec observation during a full machine checkpoint restart

The read/write results are from early tests when about half the machine was available

LANL Roadrunner, 3.2.3

pNFS Standard Status

IETF approved Internet Drafts in December, 2008

- Editorial review took one year

RFCs for NFSv4.1, pNFS-objects, and pNFS-blocks issued Jan 2010

- RFC 5661 - Network File System (NFS) Version 4 Minor Version 1 Protocol
- RFC 5662 - Network File System (NFS) Version 4 Minor Version 1 External Data Representation Standard (XDR) Description
- RFC 5663 - Parallel NFS (pNFS) Block/Volume Layout
- RFC 5664 - Object-Based Parallel NFS (pNFS) Operations

pNFS Implementation Status

Implementation interoperability continues

- San Jose Connect-a-thon March ’06, February ’07, May ’08, June ’09, **Feb ‘10**
- Ann Arbor NFS Bake-a-thon September ’06, October ’07
- Dallas pNFS inter-op, June ’07, Austin February ’08, Sept ’08, **October ‘09**

Server vendors waiting for Linux client

- Sun, NetApp, EMC, IBM, Panasas, …
- 2.6.30
  - exofs object storage file system (local) and iSCSI/OSDv2
- 2.6.31
  - most of nfsv4.1: sessions, 4.1 as an option, no pnfs yet
- 2.6.32 released
  - Adds server back-channel support.
- 2.6.33 in stabilization
  - More 4.1 bug fixing, still no pNFS option nor server recovery