SOS10
Data Intensive High Performance Computing – Challenges for the Future

Harriet Coverston
Distinguished Engineer
Sun Microsystems
March 9, 2006

Object-based Storage (1 of 5)

- 2. Object Storage has for a long time been the "love-kid" of this community. As time passes and the component economics change, how should Object Storage evolve?
- OBS is a protocol layer that is independent of the underlying storage hardware
  - OBS can take advantage of commodity economics
- OBS should be adopted and naturally evolve
  - OBS is a standard and this is a good thing
  - OBS avoids the high cost of specialized proprietary storage servers

Object-based Storage (2 of 5)

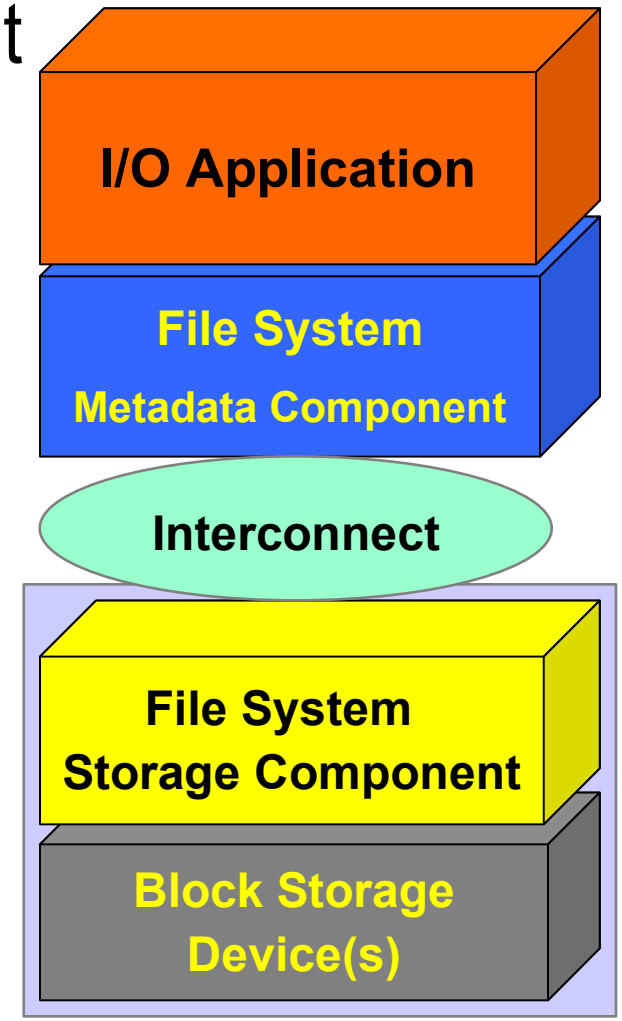
- Should OBS evolve to a completely programmable device?
- Yes, object storage devices with data aware storage can run with the software downloaded onto them.
- Data aware intelligent storage can manipulate objects
  - Object search
  - Data mining
  - Data manipulation

Object-based Storage (3 of 5)

- Should OBS evolve to a partly programmable device and then what?
- A commodity device with data aware storage can be extremely useful.
- Dumb block storage cannot possibly do what we want to do
  - Block storage only knows about sectors
- We need intelligence in the storage
  - Object management
  - Attribute interpretation

Object-based Storage (4 of 5)

- File System Metadata Component
  - > Hierarchy Management
  - > Naming
  - > User Access Control
  - > Data Properties (QoS Attributes)
- File System Storage Component
  - > Space Management
  - > Storage allocation for data entities
  - > Attribute Interpretation



Object-based Storage (5 of 5)

- Can OBS satify other markets?
- IPODs, PDAs, etc., are already “intelligent”, market-specific “object-based” storage devices
- FLASH card can be considered object-based storage because the consumer is only concerned about their pictures, not the file system or mount points or sectors, etc.
- Why not put specialized ASICs into our intelligent storage
  - How about an ASIC that would do a lookup?

Programming Models (1 of 3)

- 3. Programming models have a very basic support for storage: the best example is, that with all the widespread usage of search programming, ingest of data is still completely tied to precisely defined data set. Is this good enough? How can we bridge the gap between search and data usage by programs?
- Should it be possible to store data in a well-defined format on disk (e.g. in a tree structure, like XML) and retrieve/update the data using operations on the tree?

Programming Models (2 of 3)

- Should it be possible to insert data into the middle of a file (or to add “tags” to a file format that supports them, like JPEG)?
- Changing the artist name in a MP3 file requires rewriting the entire file
  - Rewriting the entire file gets expensive for large files
- Is there value to update files without rewriting them?
  - e.g. Replace a video clip in a 1TB file full of video clips

Programming Models (3 of 3)

- Data inserts would require rethinking the whole notion of block storage, but this is what object-based storage enables
- Object-based storage supports more data centric programming models

Specialized Storage (1 of 2)

- 4. As the storage usage volume moves away from computing, do we need to look again beyond commodity, to some forms of storage that are "specialized for HPC"? Should this be an industry effort? (Is there enough market?) Or should this be a government(s) sponsored effort?
- Commodity storage devices have well known savings in cost and development cycles
- Commodity devices are not as reliable and require more intervention than enterprise devices

Specialized Storage (2 of 2)

- Commodity parts work for graphics subsystems, processors, and memory
  - Used to be highly specialized and expensive
  - Cost goes down by orders of magnitude with a mass market
- Commodity disk drives are here and we are seeing them take over enterprise-class disk drive applications
  - Raid 5 & 6, N-levels of mirroring, fail-in-place
- Need new storage architectures which assume components will fail, but the entire storage system will have enterprise-class reliability

System Research

- 1. For the next several years, assuming an increasing demand for storage capacity and speed, but no change in the storage device availability and structure, what should be the areas for system research ? Assuming that the "preservation" trend continues and its challenges can be met by a new function in storage then how could the HPC community benefit from it?

The Office of Science Data-Management Challenge

Report from the DOE Workshop 2004

"I'm spending nearly all my time, finding, processing, organizing, and moving data – and it's going to get much worse"

http://www.sc.doe.gov/ascr/Final-report-v26.pdf

Challenge: Finding Data

- Finding any one file among trillions of files
- Finding anything in the petabytes of data
  - Data mining
- Multi-dimensional addressing
  - Indexing, dynamic translation

Challenge: Sharing Data

- Need to share data in heterogeneous environments
  - Object-based storage for pNFS
- Need global access
- Data & compute resources are often not co-located

Challenge: Managing Data

- Exponential data growth
  - Petabytes of online data
  - Exabytes of nearline data
- Increasing operational costs
  - $1 in storage costs $8 dollars to manage (Gartner, 2003)
- 90% of data is dormant
  - May never be accessed once older than 30 days
- Off-site disaster recovery is a necessity
- Data persists through hardware generations
  - Servers, storage, transports come and go

Current Best Practices

- Classical HSMs have only one disk cache
- Data is migrated to/from the archives (disk, tape, optical) to this one disk cache
- The one disk cache feeds the compute processors
- Many parallel file systems do not have HSM
- The file system interface (DMAPI) is a single point and it does not scale
- Think out of the JBOD!

Storage Pools

- Proposal for a new storage architecture – we need to start thinking about storage not as just one end point of like drives, but as different storage pools
- Each pool has different bandwidth, latency, and reliability attributes
  - DRAM, Enterprise disks, SATA, MAID, tape, etc.
- Ubiquitous support for multiple pools in one file system
  - Site chooses pools to match application requirements
- Each pool can supply data to the compute clusters
  - The best storage pool is selected for each request based on QoS requirements and current system state

User Data Access

- Seamless application access to data
  - User does not have to retrieve/backup data to/from the compute file systems
  - User is freed up to do science
- Data Security and integrity
  - Policy based encrypted storage with key management
  - End-to-end check data
- Support semantic information to find data sets

Automated Data Management

- Automatically copy/move the data between the pools of storage according to access patterns and policy to increase overall system performance
- Data Preservation
  - > Automated policy based management
  - > Multiple copies for data protection
  - > Remote pools for disaster recovery and sharing
  - > Version support

Solve Data Complexities

- Scaling data management
  - > Learn from parallel applications – perform data movement in parallel
  - > Object Archives horizontally scale
  - > Use 3rd party copy
- Migrating data through technology refresh cycles
  - > Standardize data formats

http://www.nitrd.gov/subcommittee/hec/workshop/20050816_storage/breakout/2b.pdf

Data Management Realities Today Cleveland Clinic: Womb to Tomb

- Acquire over 2 Tbyte/week of new data
- Maintain data for life of patient
  - > No convenient way to purge
  - > Legal needs 5 years, some 7, some 21
- Timely deterministic access for all data

"You have to provide an end-to-end solution, not just more hardware or disks or spindles. That's why Sun's approach with the multi-tiered management capabilities of Sun StorEdge SAM-FS and QFS software is the core of our digital imaging storage strategy."

- Dr. Robert Cecil, Network Director of Radiology and Cardiology, Cleveland Clinic Foundation

Failure Is Not an Option

SOS10
Data Intensive High Performance Computing – Challenges for the Future

harriet.coverston@sun.com

Emerging Storage Hierarchy