



Object Storage at Sun

Harriet Coverston
Distinguished Engineer
Sun Microsystems
May 10, 2006



Agenda

- Why do we need a new paradigm?
- What are the new set of data storage problems?
- Why the block interface doesn't work anymore?
- How do we get a new storage architecture?
- Where do we go from here?

Why We Need a New Paradigm

- Convergence of new data storage needs
- The industry can't figure out how to solve them
 - > Despite the hype
- The problem is NOT performance
- It's the convergence of multiple, related needs
 - 1) More systems accessing more data
 - 2) All information stored and managed in digital format
 - 3) 24x7 access to the data
 - 4) Regulatory requirements

1: Storage Consolidation

- Customers need heterogeneous data sharing
 - > Scale computational resources
 - > Data consolidation
 - > Backup, HSM
- All storage needs to be administered
 - > Monitor growth and excess capacity
 - > Back up & archived
 - > Monitor faults and replace failures
- Storage as a shared resource on a network
 - > This was the move to Fibre Channel SANs in the 90's
 - > Added the complexity of managing another network
 - > Offset by savings in managing pooled storage

2: Content-Based Data Management

- Too much data
 - > It can't all fit on spinning rust anymore
 - > Can't back it all up anymore
- But not all data is equal
 - > Some is critical, some can be recreated
 - > Needs to go on the right class of storage
- And, the value of the information changes over time
 - > The data has to move around as its value changes

3: Continuous Access to Data

- Redundancy (RAID)
- Disaster tolerance
 - > Mirrored to remote locations
 - > Offsite archive
- Also mirrored for geographic locality
- A global namespace

4: Regulatory Requirements

- Access must be authenticated
- Track where all copies are
- Track who read the information
- Track who updated it and keep revisions
 - > Prevent unauthorized (or any) modification
- Guarantee retention times

Convergence Creates the Perfect Storm

Consolidation on
the Storage Network

Classes of Storage
based on Content



Continuous Access
to Information

Compliance with
Data laws

And Performance Still Matters

- But it's all about Service Levels
- Consistently slow may be better than sometimes fast
 - > Guaranteed 1-second response instead of average 0.5 seconds with 60-second worst case
- How to guarantee service level
 - > To multiple data clients
 - > Clients may have differing service level needs for the same data
 - > For example, streaming video vs. backup
 - > Based on the data being accessed, not where it resides
- And remember, the Service Level changes over time

Storage Trends

- Storage has moved from direct-attached to shared
- Storage and networking fabrics are converging
 - > iSCSI, InfiniBand, ...
- Storage is becoming more intelligent
 - > Storage is becoming more autonomous
 - > Self-aware, self-managing, self-configuring
- Standards are emerging for richer interfaces and protocols

Business Constraints

- Millions of \$/£/€ invested in Fibre Channel infrastructure
- Data center managers are very slow to change
- So, need a simple (even stealth) upgrade

Today's Architecture Problem

Compute Grid



Raw blocks,
Formatted for 80's drives



Block Storage Servers



Where the data lives.
All data grouping and
properties stripped away
by block protocol

Where the knowledge of the data is:

- Data properties,
- Availability, performance requirements
- Grouping of data
- Access rights
- Revision history, etc.

Tomorrow's Architecture Solution

Compute Grid



Grouped data with:

- Class of Service Requirements
- Security keys, encryption, etc.
- Content info, keywords, etc.
- Creation, revision dates, etc.
- Retention time requirements,
- And anything else we can think of...

Object-based Storage Solves Problems

- Enables scaling to much larger environments
- Allows further consolidation
 - > Migrate data to the right class of storage...in the device!
 - > As the value of the information changes
- Authenticate access to information from compute grid
- Keep an access log
- Track where every copy is located
- Sharing data with object-based file systems solves many problems
 - > Security, concurrency, locking...

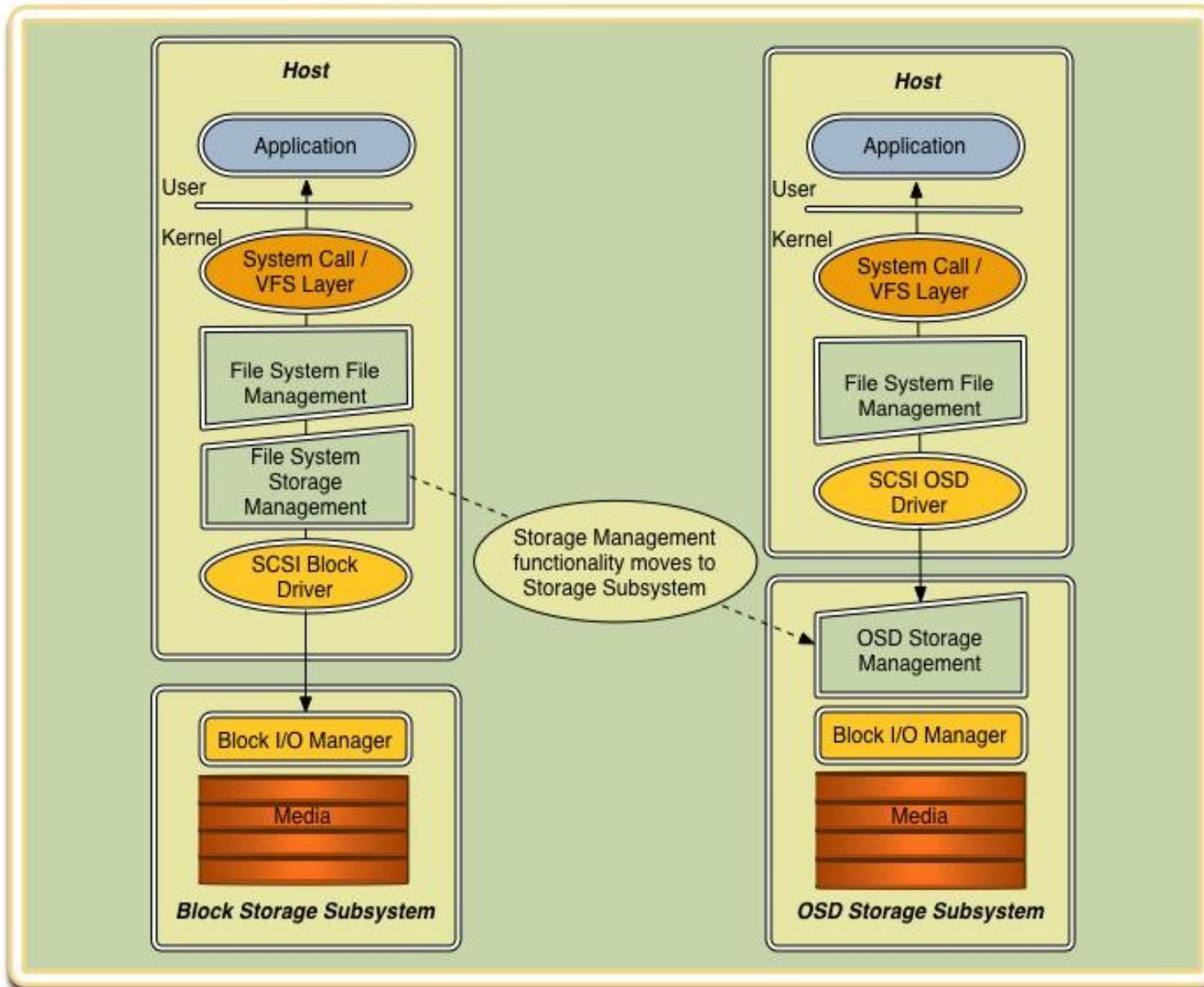
Advantages of Object-Based Storage

- Secure
 - > Devices authenticate client requests
 - > Policy defined independently of devices
- Scalable
 - > Clients communicate directly with devices
 - > Devices directly handle client requests
 - > Metadata server involvement only needed at open
- Improves performance through object knowledge
 - > Service levels attached to data
 - > Better caching, pre-fetching, etc.
- *But existing storage devices cannot be ignored...*

Transitioning from Blocks to Objects

- Investment in existing storage must be leveraged
- Object Storage Servers provide a bridge
 - > Object interface to hosts
 - > Block interface to storage
 - > Can exist alongside “native” object storage devices
 - > Can provide a gradual transition with coexistence of blocks and objects
- New file systems must be developed
 - > Parallel file systems can take full advantage of object storage

Block vs. Object Architecture



Object Storage Advantages

- Space allocation performed by storage devices
 - > Block management limits scaling of parallel file systems today
 - > Speed of allocation scales with capacity increases
- Knowledge of the data available to device
 - > Reduces data fragmentation
 - > Attributes can be stored with the data
 - > Key to quality-of-service (QoS) policies
- Automatic Data Migration (HSM) performed by devices
 - > Object archiving & staging scales up with capacity increases
- Security enforced by storage devices
 - > Clients need not be trusted!

Objects Are The Future ...

- Decouple the physical storage technology from applications and file systems
 - > A paradigm shift for file systems
 - > Storage devices become peers of compute nodes
 - > Storage devices can be hybrids of disks, tapes, optical, DRAM, flash memory, etc.
- Object-based storage enables a platform for new innovations in storage
 - > Underlying storage technologies can evolve independently of the data that they store and the protocols that access them

Sun's Strategy for Recording, Capturing, and Feeding Data to the Grid Today

- NFS/NAS
- Shared SAM-QFS (SAN)
- Parallel 3rd Party Partnership with Lustre on Linux
 - > POSIX Compliant
 - > SAMBA exportable
 - > NFS exportable

Introduction to Sun's StorEdge QFS

- Near linear scalability for large sequential I/O
- Optional metadata separated from data
- Aggregation of striped devices
- Data consolidation with SAN (shared) file system
 - > Delivers a performance edge over NFS for large files
 - > Takes advantage of the SAN with multiple data paths in contrast to NFS where all data is transferred OTW
 - > Example application – Oracle RAC with SunCluster
 - > Solaris (SPARC and X64), and Linux clients
 - > SANergy client support (Windows, AIX, IRIX, etc.)
- Integration with SAM for continuous backup

Introduction to Sun's StorEdge SAM-FS

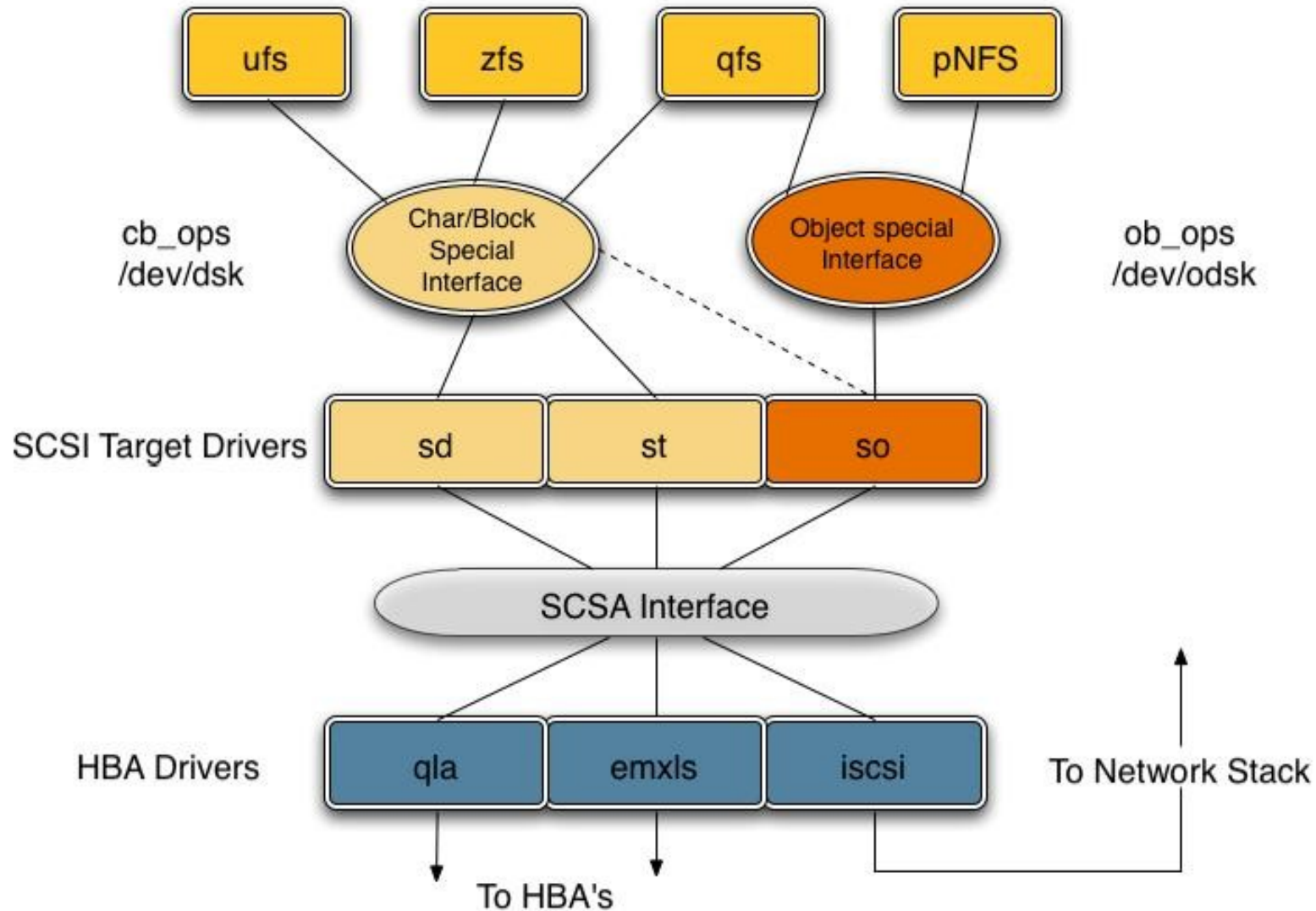
- Keeps all data available, but not on high cost storage
- On-demand, transparent file retrieval
- Continuous backup – no waiting until midnight
- Quick disaster recovery for business continuance
- Policy based automatic data migration
 - > Media can be disk, tape or optical
 - > Local or remote copies
 - > Media format is open, not proprietary – tar format
 - > Files can be recovered with or without SAM – our media format is open, NOT proprietary
- Facility for migration from other HSMs

Sun's Object Storage Roadmap

- Implementing the T10 OSD standard
- February 2006: Proof-of-concept using Shared SAM-QFS
 - > First demonstration of object functionality
 - > Seagate supplied Fibre Channel OSD drive
 - > Dave.B.Anderson@seagate.com
 - > Emulex supplied the FC adapter
 - > Jean-Yves.Chevallier@emulex.com
- Deliver in OpenSolaris the OSD device driver and Solaris stack (SCSA) changes this year
 - > Large SCSI Command Descriptor Block support
 - > Bidirectional SCSI support

opensolaris

Solaris Storage Driver Stack



Object Storage QFS and pNFS

- Implement the T10 OSD standard
- Support objects in Shared QFS
- Support heterogeneous data sharing with Object Storage pNFS
 - > Participate in the IETF standards process
 - > Demonstrate interoperability this year
- Plan to support standard OSD devices & object storage servers

Sun's Parallel File System Today

- Shared SAM-QFS
 - > SAN file system
 - > Block-based
 - > Single metadata server
 - > Integrated archiving and data management
 - > Solaris and Linux support
 - > SANergy support for other operating systems

Sun's Parallel File System Tomorrow

- Shared SAM-QFS
 - > SAN file system
 - > Support for iSCSI allows use of networking infrastructure
 - > Object-based
 - > T10 compatible
 - > Multiple metadata servers
 - > Improves scalability
 - > Storage pools for managing classes of data
 - > Policy-based automatic migration and data protection
 - > Solaris and Linux support
 - > Object-based pNFS support for other operating systems
 - > Standards-based

In Summary

- Need to solve the convergence of:
 - > Consolidation of storage resources
 - > Information content management
 - > Continuous 24x7 access to data
 - > Compliance with information laws
- Need to go beyond 1970's technology in the application-to-storage and file system-to-storage interfaces
- Need technology from the neutral research community & standards bodies
- Need a cross-industry effort



harriet.coverston@sun.com

