SNIA ESF | ETHERNET STORAGE

File vs. Block vs. Object Storage

Part of the SNIA ESF “Great Debate” Webcast Series

April 17, 2018

Today's Presenters

John Kim
SNIA ESF Chair
Mellanox

Mark Carlson
Co-Chair SNIA
Technical Council
Toshiba

Saqib Jang
Chelsio Communications

Alex McDonald
SNIA ESF Vice Chair
NetApp

SNIA-At-A-Glance

SNIA-At-A-Glance

170
industry leading organizations

2,500
active contributing members

50,000
IT end users & storage pros worldwide

Learn more: **snia.org/technical** **@SNIA**

SNIA Legal Notice

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

Agenda

- Data Access by block, file, object
  - Differences in access, sharing and workloads
- Block Storage
- File Storage
- Object Storage
- Is one better than the other? Challenge Topics
- Summary

Different Access Methods

- How does the application want to access the data?
  - All at once or piece by piece?
  - Sequentially or randomly?
- What type of data is it?
  - Database, text, video/audio, photo
  - Static or fixed?

Data Sharing

SNIA ESF | ETHERNET STORAGE

- Does the data need to be shared?
  - Shared by the application vs. shared by the storage
  - Shared reading vs. shared writing
  - Narrow or broad sharing?
- Security and access controls
  - Applied at what level?

The Type of Storage

- Storage Design Can Affect Access Choice
  - Media: tape, disk, flash, PM
  - Storage controller performance
- Connectivity can affect choice.
  - Local vs. networked
  - Fibre Channel, Ethernet, SAS, SATA, PCIe, etc.

Saqib Jang

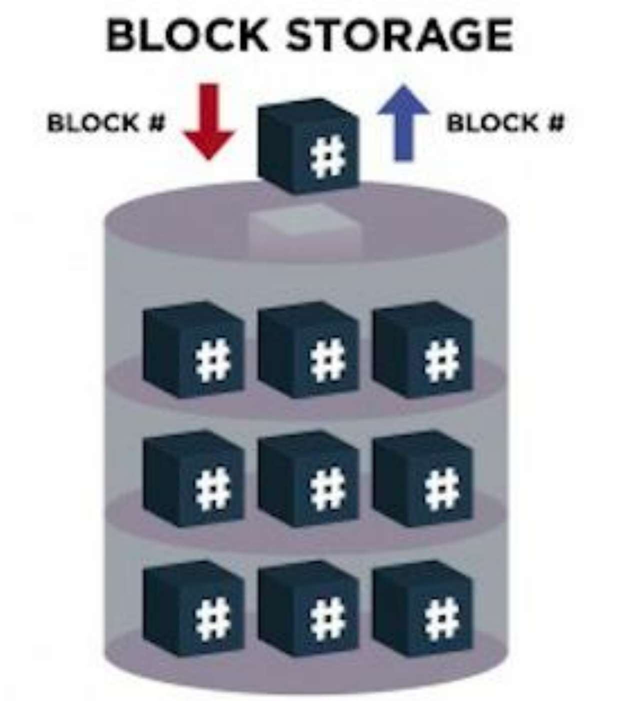
BLOCK STORAGE

Block Storage Use Cases/ Workloads

- Ideal for performance-intensive primary storage
- Use Cases
  - Structured database storage for OLTP and BI
  - Virtual volumes
  - Applications using server-side processing (e.g. Java or PHP)
- Workloads
  - High Change Content
  - Random R/W
  - ”Bursty” IO

How Block Storage is Organized

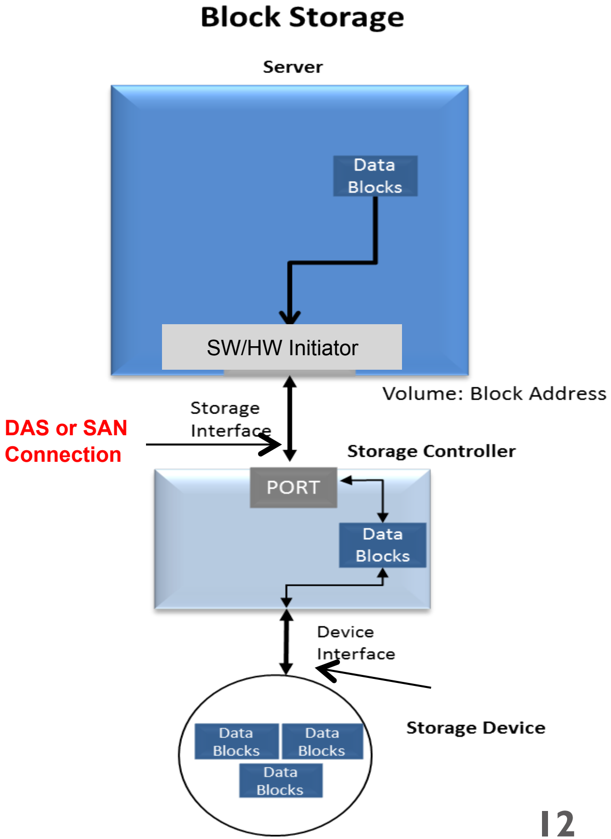
- Data is typically stored on device in fixed-sized blocks (e.g. 512 Bytes)
- Data is stored without any higher-level metadata e.g. for data format, type or ownership
- Accessed by operating system as mounted drive volume
- Applications/file systems decide how blocks are accessed, combined, and modified



How Block Storage is Accessed

- Application writes data block
- Block goes to SW/HW Initiator and over DAS* or SAN** connections
  - DAS: SATA, SAS, FC, NVMe™
  - Ethernet SAN: iSCSI, NVMe-oF
  - Fibre Channel SAN: FCP, NVMe-oF
- Storage controller receives block
- Data written to device as data block

*DAS=Direct-Attached Storage
** SAN=Storage Area Networks

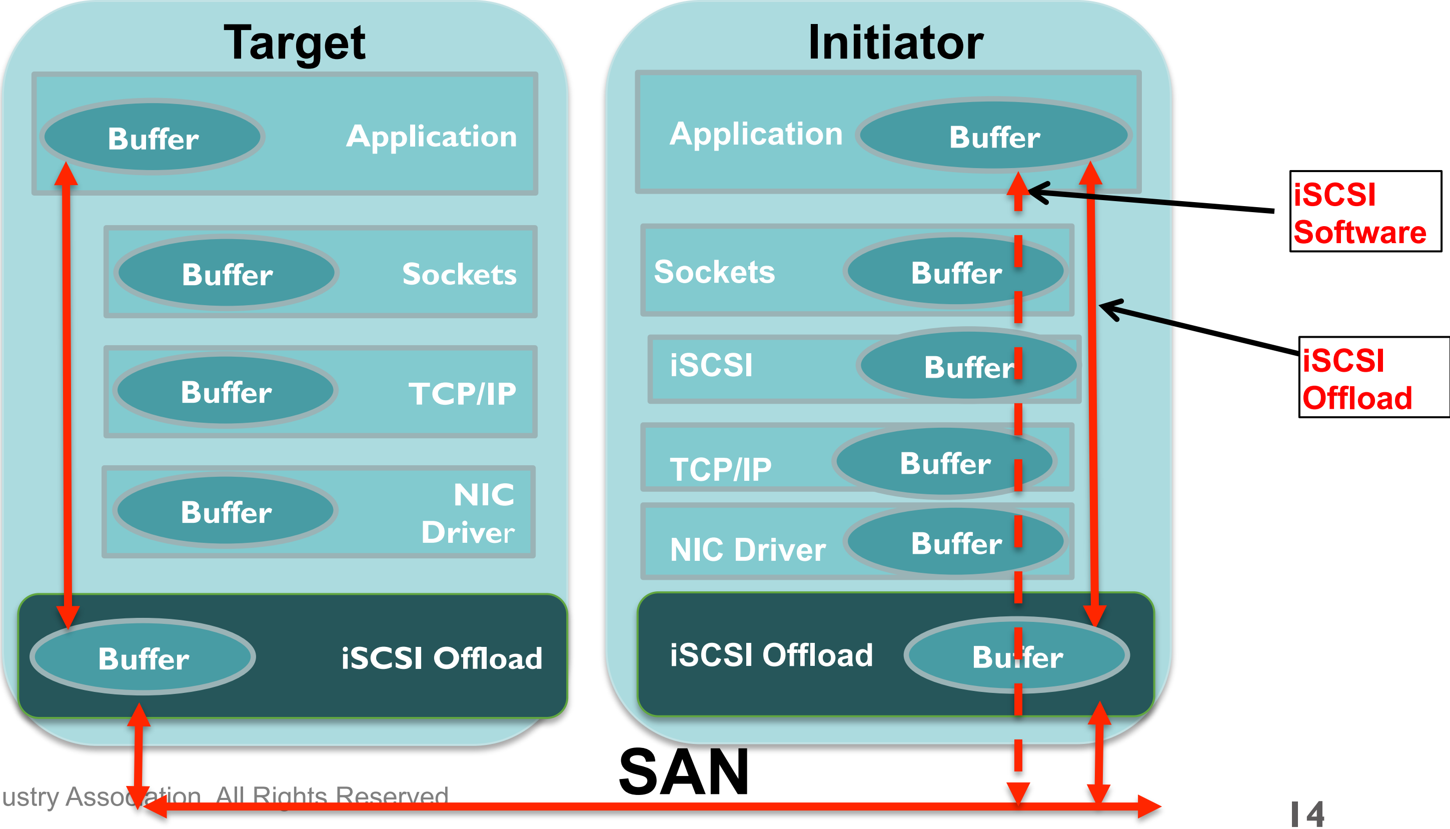


Block Interface Comparison

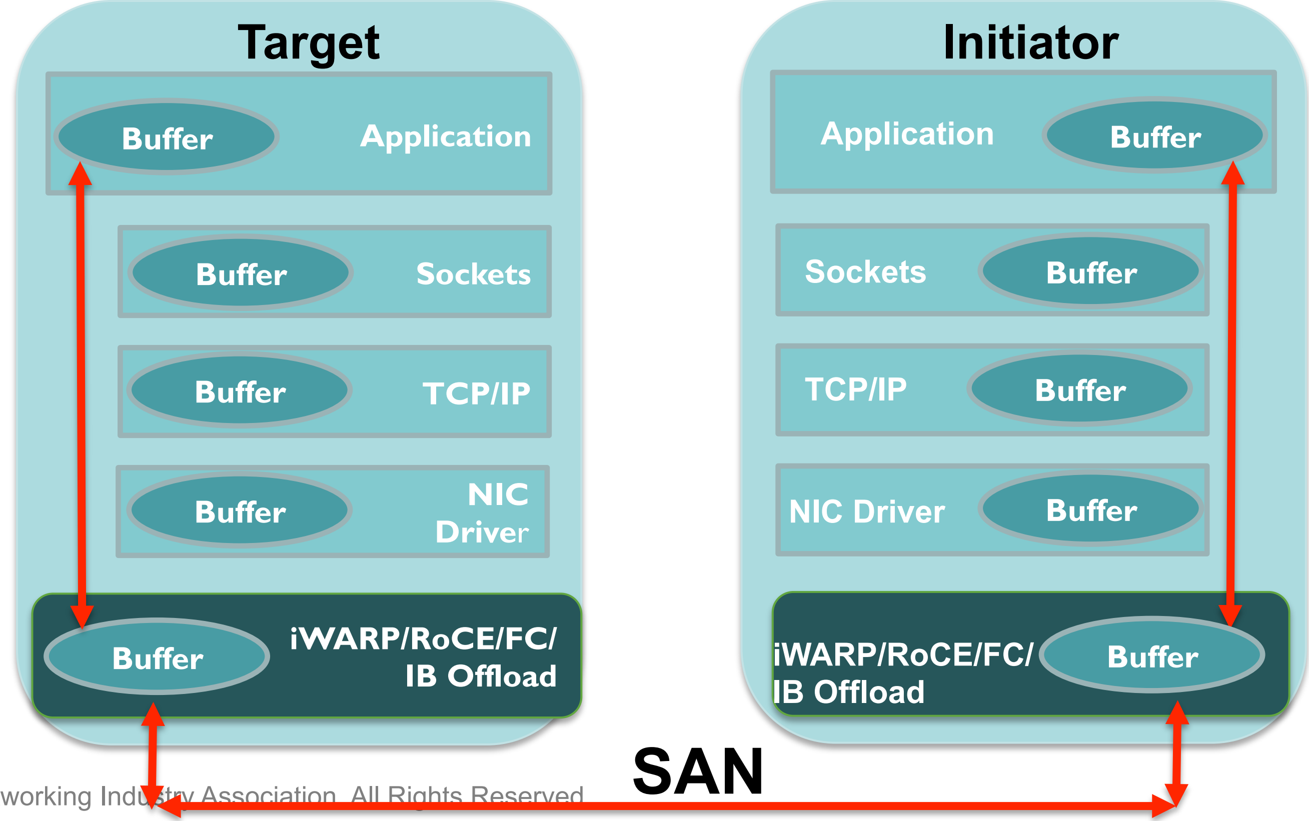
Interface (Protocol)	Deployment Scale	Interface Type	Maximum Transfer Rate (Lanes)
SATA	DAS	On-board	6 GB/s
SAS (SCSI)	DAS	On-board	12 GB/s
Thunderbolt	DAS	On-board	40 Gb/s
NVMe	DAS	On-board	16 GB/s (PCIe 3.0 x16)
Fibre Channel (FCP/NVMe-oF)	DAS/SAN/WAN (FCIP)	HBA	32 Gb/s (1)
Ethernet (iSCSI/iSER/NVMe-oF)	DAS/SAN/WAN	NIC & Offload Adapter	100 Gb/s
InfiniBand (SRP/iSER/NVMe-oF)	SAN	HCA	100 Gb/s

Block Storage I/O Path – iSCSI

- iSCSI Software
  - Software-based Protocol Processing
- iSCSI Offload
  - Protocol Bypass
  - RDMA



Block Storage I/O Path – NVMe-oF



Block Storage Security

- iSCSI
  - CHAP authentication is available in all iSCSI implementations
  - IPsec is available to secure the communication channel
  - VLANs enable logical isolation of storage and data traffic
    - Large iSCSI SANs may be physically isolated from LANs for optimal storage QoS
- FCP
  - WWN-based access controls for limiting access to storage
    - Includes switch zoning and LUN masking in storage
  - Authentication and in-flight encryption for trusted in-band management and trusted storage networks
  - Switches configured with least amount of access and interconnections restricted
  - FC SAN deployment is always physically isolated from LANs

Managing Block Storage – SNIA Swordfish™ Spec

- Block storage devices (represented by Volumes) provide their capacity to external applications through block-based protocols
- Standard APIs used for management of resources providing access to block storage
- Block storage management functions include
  - Add volume, Allocate volume, Expand storage volume, Review volume metrics
  - Create storage group, Create storage pool
  - Create class of service, Discover class of service, Get capacity by class of service, Find storage service
  - Create line of service, List supported line of service options

Alex McDonald

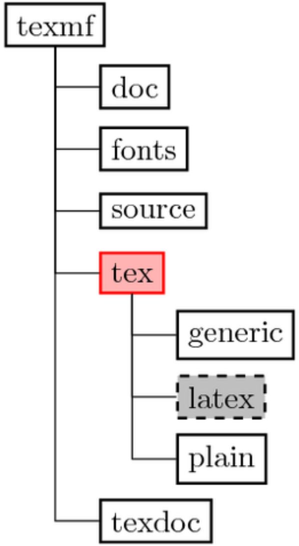
FILE STORAGE

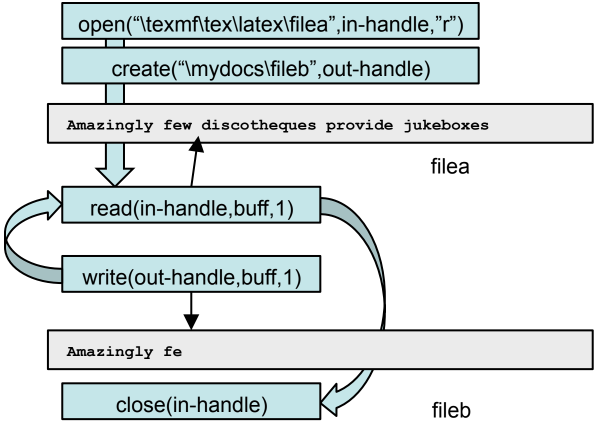
File Storage; Systems & Access

- A little bit of history
  - Filing cabinets for paper based documents
  - File systems are a similar construct; collections of documents
- Use cases
  - Document sharing
  - Clustered databases
  - Big data
  - Media and entertainment
  - Technical computing or HPC
  - Foundation for application independence
    - Provide consistent set of APIs

Files Application View

- Characteristic of a file
  - Files have *names* and are *byte addressable*
  - Randomly accessible
  - Named, with IO operations through a *file handle*
- Organized into *named directories* which are themselves structured files
- Several effective layers (but may be blended)
  - Logical, virtual & physical file systems
  - Network access layer





open("\texmf\tex\latex\filea",in-handle,"r")

create("\mydocs\fileb",out-handle)

`Amazingly few discotheques provide jukeboxes`

filea

read(in-handle,buff,1)

write(out-handle,buff,1)

`Amazingly fe`

close(in-handle)

fileb

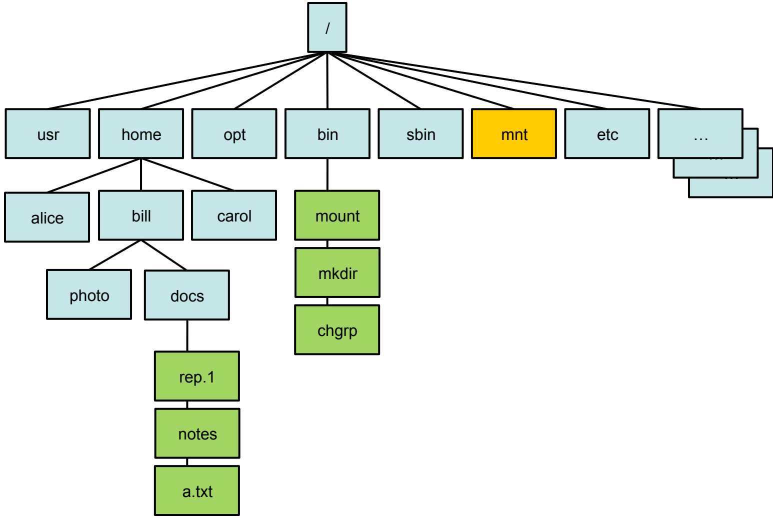
Files Application View

- POSIX or close to POSIX-like support in all major Oses
  - Operations
    - Opening and closing
    - Reading, updating and writing
    - Creating, renaming and deleting
- Same system calls used allows source code portability
- Applications can treat files as
  - Streams or objects; a set of unstructured data
  - Structured data; sets of discrete units contained in a file
  - Block; set of randomly accessible blocks
  - …

POSIX™ defines a standard operating system interface and environment … to support applications portability at the source code level
http://pubs.opengroup.org/onlinepubs/9699919799/

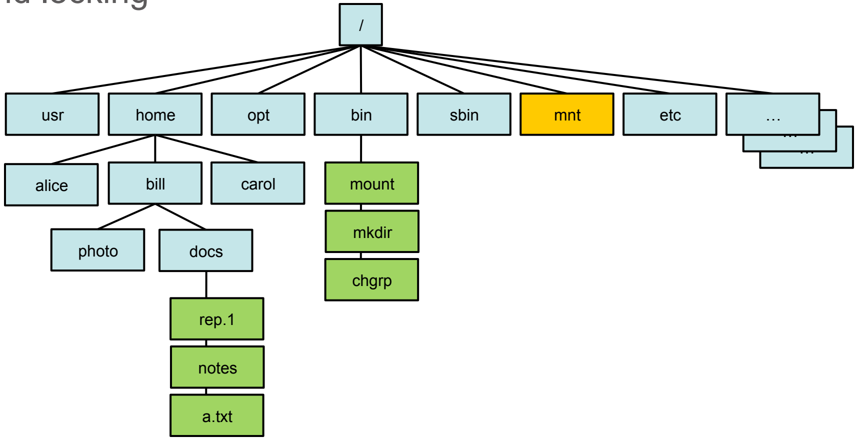
File Systems

- Physical layer built on storage devices
  - Tape, disk, flash, persistent memory…
- Virtual layer built on physical layer
  - Many 100s of file systems:
    - EXT3 EXT4 JFS ZFS GPFS ResierFS and many more
    - Each has different characteristics
  - This layer is typically the mount point or share
- Logical layer brings together virtual levels in a single root
  - Rooted tree of directories and files
  - Names and paths through directory
    - Fully qualified name: /home/bill/docs/a.txt



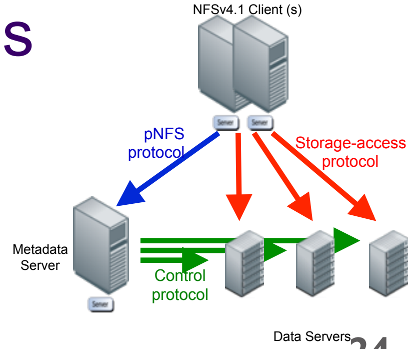
Distributed File Systems

- NFS and SMB
  - *nix and Windows
  - Allow creation, deletion, reading, writing, sharing and locking
  - Supported by all major OSes and hypervisors
  - (typically) No extra client software needed
  - Provide access over networks
- Distributed File Systems
  - Make distributed look exactly like local file system
- Key is transparency
  - Access & location
  - Consistency & concurrency
  - High level of tolerance to failure
  - Heterogeneous, scalable, replicatable, migratable
- Uses RPCs (Remote Procedure Calls) & network protocol
  - Ethernet and TCP/IP
- No knowledge required of underlying structures; nothing “pokes through” to the end user



File System Performance

- Can be very good indeed
  - Competes with iSCSI
    - Suitable for VM datastores, containers
  - Parallelization (for example, pNFS)
  - Abstraction layer is deeper (hence higher latencies than block)
- Space requirements can be significantly less
  - Compression
  - Hole punching or sparse files



Security & Privacy

- NFS & SMB use Kerberos or LDAP
  - “network authentication protocol which works on the basis of 'tickets' to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner” https://en.wikipedia.org/wiki/Kerberos_%28protocol%29

Authentication

Proving to a system that you are who you claim to be

- Name mapping
- Kerberos tickets

Authorization

What a user is allowed to do once in a system

- Group membership
- Access Control Lists (ACLs)

Terms are often confused with each other; not interchangeable

Managing File Storage

- Meta data
  - Attributes & extended attributes
  - Times & dates, size, …
- Many thousands of utilities
  - Archival, backup, restore (NDMP)
  - Compression, deduplication
  - Specific structured file type management
    - For instance: video, sound, documents

Mark Carlson

OBJECT STORAGE

Object Use Cases

- Public, Private and Hybrid Cloud Storage
  - Sync and share from desktop to devices, offload from email, etc.
- Archival Storage
  - Ultimate tier is Object Repository with policies for retention
- Server-less Container Storage
  - Shared state among micro-services
- Analytics and IoT, Machine (Deep) Learning
  - Ingest from edge, dump in a lake, analyze and train, make decisions
- Green Field applications with need for rich metadata capabilities

How is the data organized for Objects?

- Objects have a handle that is a URL, or ObjectID (both)
  - Objects may be grouped into flat buckets or hierarchical containers (also with URLs)
- Objects have metadata
  - User metadata, system metadata
- Objects may have versions
  - Underlying infrastructure may be immutable storage
- Storage of Objects may be RAID-less
  - Shard into smaller pieces, erasure code protection and then distribute

Object Access

- No need for a “mount” operation
- Access any Object from any endpoint modulo security
  - Resolve the location from the URL
- RESTful interfaces scale out better with load balancers
- Objects in a Hybrid cloud can move back and forth between private and public infrastructure transparently
- Object store can utilize an underlying file system, or can organize the data itself

Key Value

- Application level Key Value can be thought of as an object interface
  - Access not by URL, but by using a Key in a specific store
- A Key Value drive interface can also underlay the Object or KV store
  - Key based on the ObjectID or hash of the value perhaps
  - Each shard is a Key/Value across multiple drives
  - More efficient (less costly) interface to PCIe SSDs

Performance

- Performance (and Availability) of Objects is AT SCALE
  - The interface allows for scale out implementations that also provide redundancy
  - Data is in multiple places and the result can be created from the fastest responses
  - Globally unique handles allow response from the least loaded node
- May not be the highest throughput or lowest latency for single machine applications with attached drives

Using Objects

- Access via standard Internet networking infrastructure
  - DNS, HTTP, TCP/IP, Ethernet (CE not required)
- Broad support from nearly any programming environment
- More intuitive (click to view)
  - Javascript and HTML brings the access client into your browser
- Any device, server or IoT sensor can access worldwide

Security and Privacy

- Rich Object metadata may be used to provide additional service levels
  - Share Objects with only those you choose
  - Grant temporary (time based) access
  - Audit trails help after an intrusion, monitor compliance
- Data management is interoperable and enhanced
  - Requirements for services over the lifetime of the object
- Privacy is defined by the Object owner

Managing Objects

- Once you put your data into an Object how is it managed?
- Each Object has “knobs” that control the underlying data services and how they treat this Object
  - Data System Metadata is used for this purpose
  - How performant, available, protected is the Object right now?
- SREs and Admins can use these knobs to optimize the treatment of this data at this point in its lifecycle

Should I use block, file or object storage?

COMPARE AND CONTRAST

Workloads

- Do databases prefer block or file storage? Why?
- Which is best for video storage?
- Where should I store VMs and containers?
- How do you know which is best for your application?
  - Do some apps support *only* block, or file, or object?

Comparing Performance

SNIA ESF | ETHERNET STORAGE

- Is one faster than the other? Does it matter?
- Can block, file and object all use flash storage?
- Do they run on different-speed networks?
- What about distributed/scale-out performance?

Comparing Manageability

- Which solution is best for managing…
  - Big vs. small chunks of data
  - Shared vs. non-shared data
  - Very large volumes of data
- Meta-analysis on metadata
  - Does Block storage lack useful metadata?
  - How do file and object metadata differ?

Sharing & Security?

SNIA ESF | ETHERNET STORAGE

- Which solution has the best sharing?
  - Most granular vs. most scalable
  - Most secure vs. easiest
- Which solution has the best security?
  - For protecting against external threats
  - For controlling levels of internal app/employee access
- How do security needs influence sharing?

Summary

- Almost all storage accessed via block, file or object
- They provide differing ways to access and manage data
- Not a question of which one is “better” but which one is the best fit for your application and workload

More Webcasts

- Next Live Webcast: Everything You Wanted To Know About Storage But Were Too Proud To Ask - Part Aqua: Storage Controllers
  - May 15, 2018, 10:00 am PT
  - Register at: https://www.brighttalk.com/webcast/663/312607
- On-Demand “Everything You Wanted To Know About Storage But Were Too Proud To Ask” Series
  - https://www.snia.org/forums/esf/knowledge/webcasts-topics
- SNIA resources on File, Block and Object
  - Evolution of iSCSI: https://www.brighttalk.com/webcast/663/197361
  - Comparing iSCSI and NVMe-oF blog: http://sniaesfblog.org/?p=647
  - What is NFS Webcast: https://www.brighttalk.com/webcast/663/191035
  - Object Storage 101 Webcast: https://www.brighttalk.com/webcast/663/110683

After This Webcast

SNIA ESF | ETHERNET STORAGE

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Ethernet Storage Forum (ESF) website and available on-demand at www.snia.org/forums/esf/knowledge/webcasts
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-ESF blog: sniaesfblog.org
- Follow us on Twitter @SNIAESF

Thank You