Object Drives: A New Architectural Partitioning

Mark Carlson/Toshiba

SNIA Legal Notice

- The material contained in this tutorial 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 Education Committee.
- 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.

Abstract

Object Drives: A New Architectural Partitioning

A number of scale out storage solutions, as part of open source and other projects, are architected to scale out by incrementally adding and removing storage nodes. Example projects include:

- Hadoop's HDFS
- Ceph
- Swift (OpenStack object storage)

The typical storage node architecture includes inexpensive enclosures with IP networking, CPU, Memory and Direct Attached Storage (DAS). While inexpensive to deploy, these solutions become harder to manage over time. Power and space requirements of Data Centers are difficult to meet with this type of solution. Object Drives further partition these object systems allowing storage to scale up and down by single drive increments.

This talk will discuss the current state and future prospects for object drives. Use cases and requirements will be examined and best practices will be described.

Learning Objectives

- Definition of object drives
- Learn the value that they provide
- Discover where they are they best deployed

What are Object Drives?

- Key/Value semantics (Object store) among others
- Hosted software in some cases
- Interface changed from SCSI based to IP based (TCP/IP, HTTP)
- Channel (FC/SAS/SATA) interconnect moves to Ethernet network

This work is ongoing in the SNIA Object Drive TWG. Please join us at:
https://members.snia.org/apps/org/workgroup/objecttwg/

Object Drive Usage Interface

- Key Value example is Kinetic
  - Metadata is not part of the interface
  - Metadata understood by higher levels would be stored as values
  - Not expected to be used directly by end user applications (similar to existing Block I/F in that regard)
- Higher level example is CDMI
  - Includes rich metadata (both user and system)
  - Expected to be used by end user applications (and is)
- Object Drives, because of their abstraction allow many more degrees of innovation freedom in the implementations behind this abstraction
  - Hosts needn't manage the mapping of objects to a block storage device
  - An object drive may manage how it places objects on the media without reference to host specified locations

What is driving the market?

- A number of scale out storage solutions expand by adding identical storage nodes incrementally
  - Typically use an Ethernet interface and may be connected directly to the Internet
- Open source examples include:
  - Scale out file systems
    - Hadoop's HDFS
    - Lustre
  - Ceph
  - Swift (OpenStack object storage)
- Commercial examples also exist

Who would buy Object Drives?

- System vendors and integrators
  - Enables simplification of the software stack
- Hyperscale Data Centers
  - Using commodity hardware and open source software
- Enterprise IT
  - Following the Hyperscale folks

Problem Statement (Current Solutions)

- For these solutions, typically a commodity server is used as a storage node with Direct Attached Storage, CPU, memory, networking
- These generalized solutions for specific use cases utilize multiple commodity servers and therefore consume more power and are more complex to manage
- Although less expensive to acquire than previous solutions, they still require higher long term ownership costs

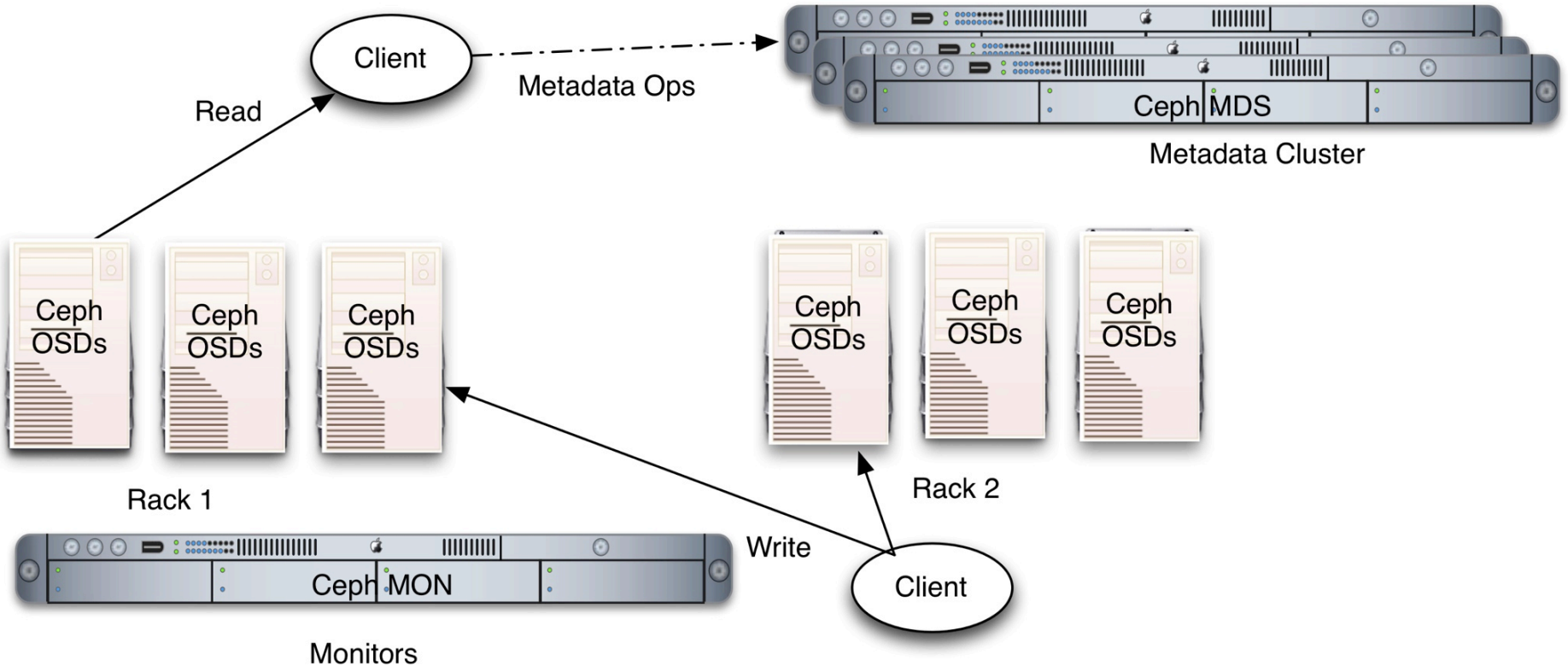
How do Object Drives Solve this?

- CPU power is moved to the drive where it can be optimized to the task (I/O) – does not need to be general purpose
  - Similar to what happened with (dumb) cell phone processors
  - For Smart phones, as desired applications increase their needs, cell phone resources have grown
- CPU/Memory/Network/Storage is one component, managed as such
  - Volume shipments drives the cost of this down
- Enable the resources to be matched/tuned with each other and sized appropriately

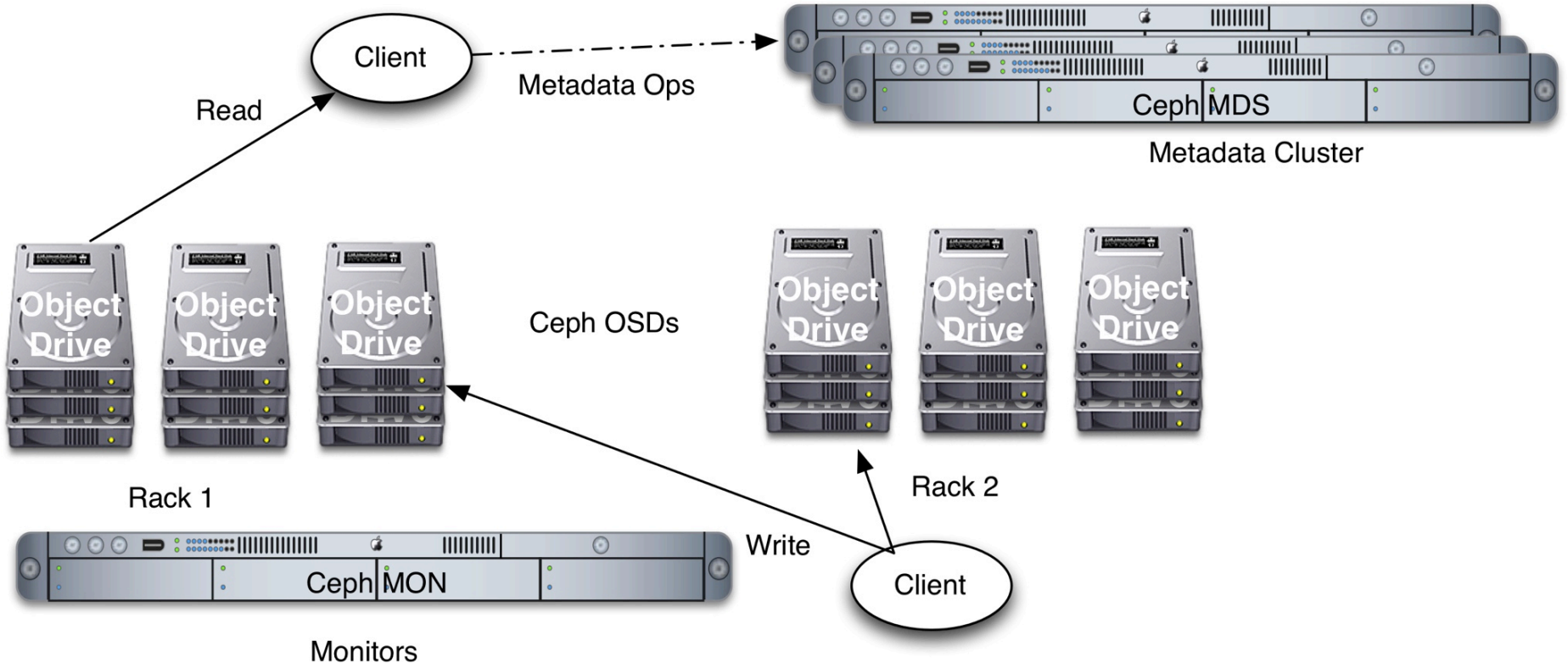
What do Object Drives NOT Solve? (today)

- Management Complexity
  - 10x or more of things to manage
  - Still no management software included with the drives
  - Perhaps the best place to do scale out management is above the object abstraction
- End to end security
  - Data is secured on the drive
  - End user applications don't do this securing of the data
    - No secure multi-tenancy in reality
  - Higher level software will typically be trusted for authentication and access control
    - Drive has basic message security support (private key)
- End to end integrity
  - Ensure that data is not altered during I/O
  - e.g. T10 DIF

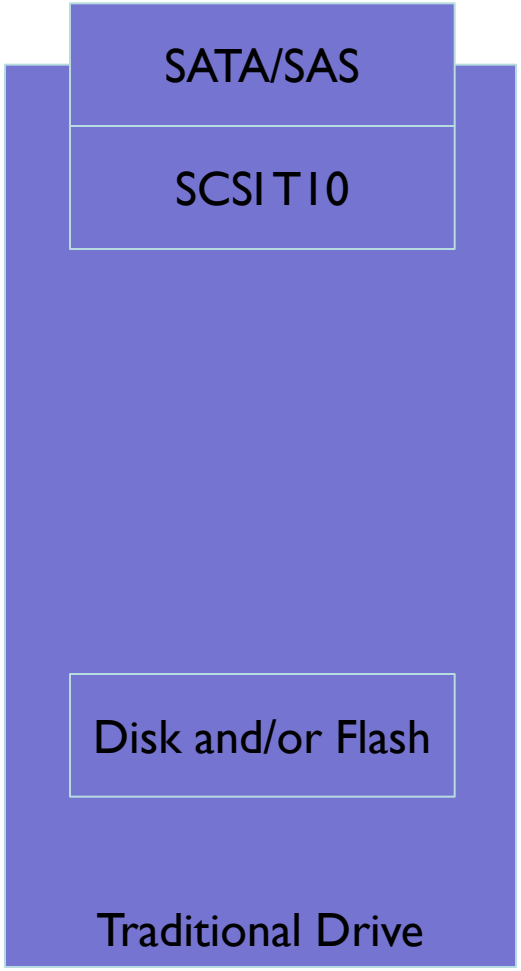
Example: Ceph Architecture



Ceph with Object Drives

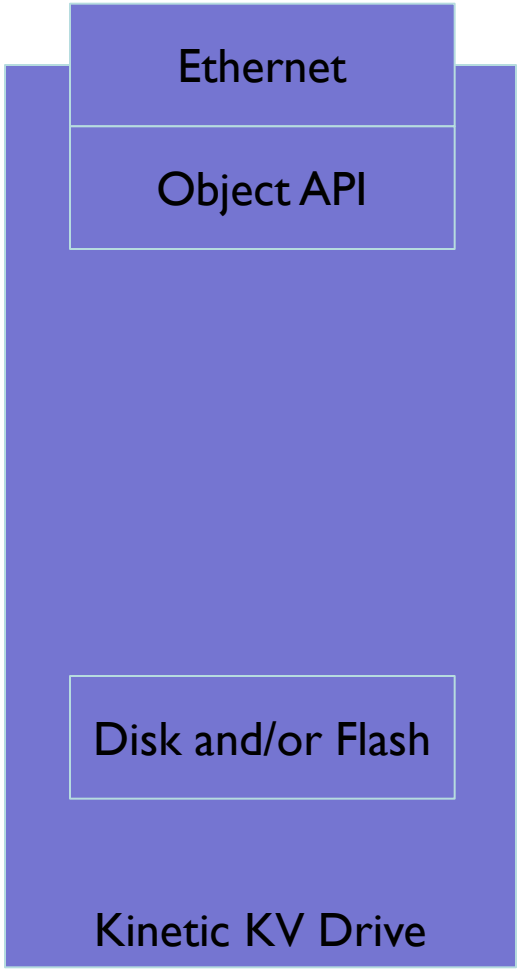


Traditional Hard Drive



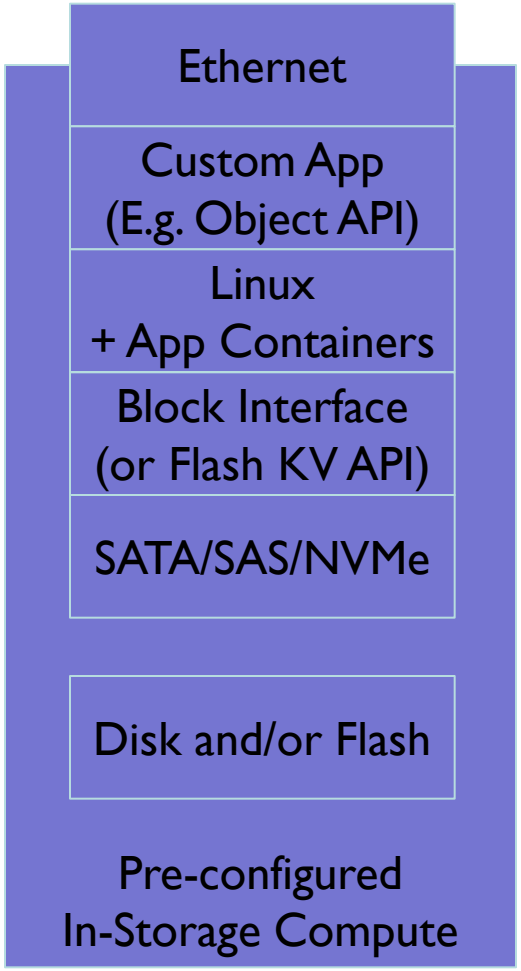
- Interconnect via SATA/SAS
  - Limited routability
  - High development costs
  - Typically single host
- T10 SCSI Protocol
  - Low-level storage interface
  - Not designed for lossy network connectivity
  - Not typically used in multi-client concurrent access use cases

Kinetic Key Value Drive



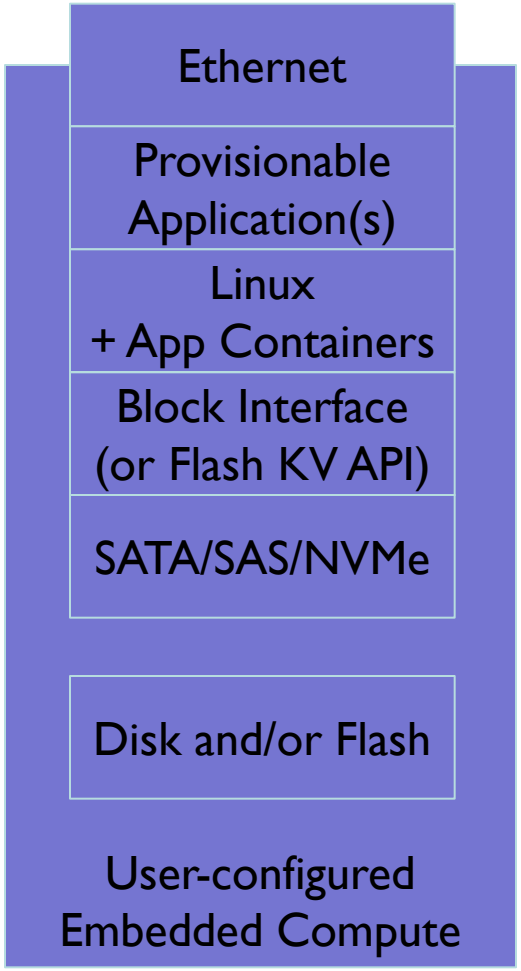
- Interconnect via Dual Ethernet
  - Full routability
  - Lower development costs
  - Intended for multi-client access
  - Path failover for availability
- Kinetic Protocol
  - Higher-level Key Value interface
  - Designed for lossy network connectivity (TCP/IP)
  - Typically used in multi-client concurrent access use cases

Pre-Configured In-Storage Compute Drive



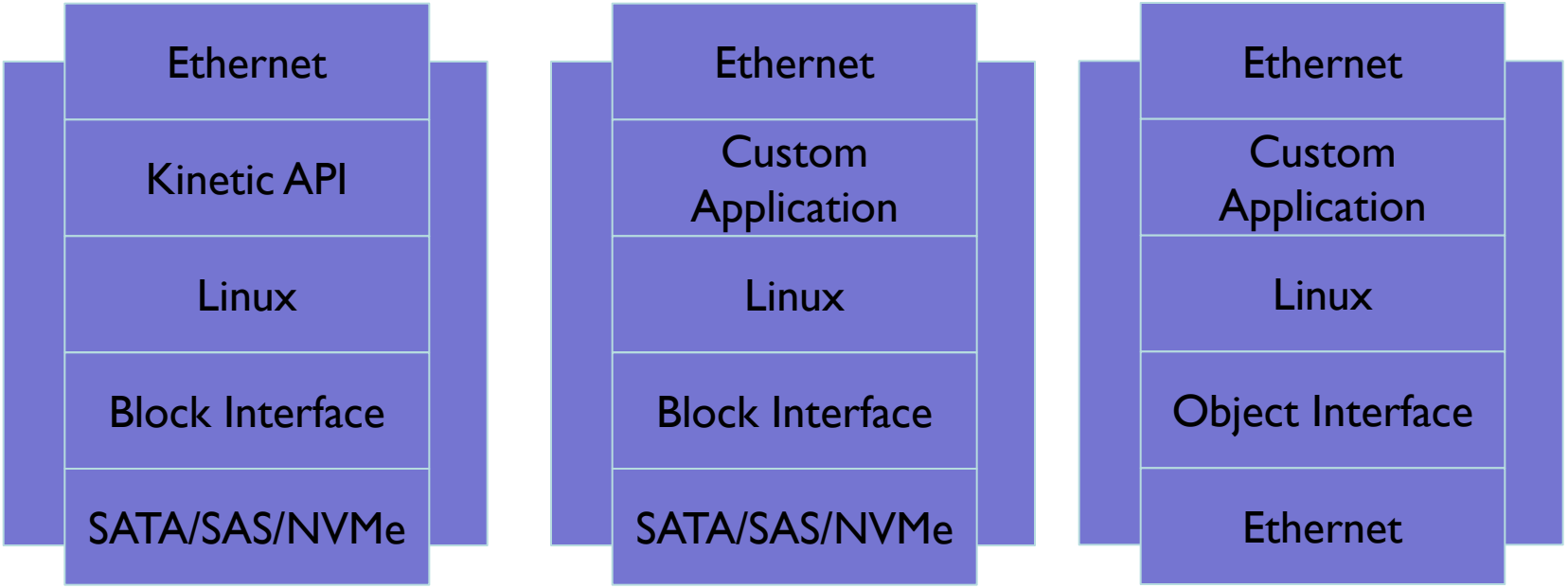
- Interconnect via Ethernet
  - Full routability
- Custom Applications installed at factory or provisioning time
  - No pre-determined client-facing interface
  - Example is Ceph, or Kinetic API
  - Custom application can run in a container in an embedded Linux environment
  - Custom application accesses storage via standard Linux interfaces

Provisionable In-Storage Compute Drive



- Interconnect via Ethernet
  - Full routability
- Custom Applications installable at any time

Interposers



- Allows existing drives to be used as Ethernet-connected Object Drives
- Allows collections of drives to be virtualized as Ethernet-connected Object Drives

Types of Object Drives

- Key Value Protocol (Object Drive)
  - Minimal incremental CPU/Memory requirements
  - Simple mapping to underlying storage
- In-Storage Compute (Object Drive)*
  - Enough CPU/Memory for Object Node Software to be embedded on the drive
  - General purpose download or factory installed
  - May have additional requirements such as solid state media and more/higher bandwidth networking connections
- In both cases, the interface abstracts the recording technology

*Jim Gray memorial

Key Value Protocol Object Drives

- Eliminates existing parts of the usual storage stack
  - Block drivers, logical volume manager, file system
  - And the *associated* bugs and maintenance costs
- Existing applications need to be re-written, or adapted
  - Mainly used by green field developed applications
  - Firmware is upgraded as an entire image
  - Hyperscale customers are already doing this
    - Using open source software and creating their own “apps” – Facebook, Google, etc.
  - Key Value organization of data is growing in popularity
    - Examples: Cassandra, NoSQL

In-Storage Compute Object Drives

- Same advantages as Key Value protocol plus
  - No need for a separate server to run Object Node service (other services still need a server but scale separately)
    - scaling is smoother – only adding drives
  - Additional features of the Object Node software can be deployed independently
  - Fewer hardware types that need to be maintained (for selected use cases)
  - Failure domains are more fine grained, thus overall data availability is enhanced

In-Storage Compute Future

- As data on the drive becomes colder, CPU/Memory becomes less utilized
  - Possible to host software that then uses this spare resource and works against the cold data
    - Extracting metadata
    - Performing preservation tasks
    - Other data services: advanced data protection, archiving, retention, deduplication, etc.
    - Data Analysis off-load

Ethernet Connectivity

- Ethernet speeds standardized at 1Gbps, 10 Gbps
- Object Drives are high volume, low margin devices
  - 10 Gbps too expensive for the drive form factor for the foreseeable future
- Desire to have standardized speeds of 2.5 Gbps, 5 Gbps
  - With auto-negotiation among them
- SFF 8601 effort to specify auto-negotiation using existing silicon implementations (switch firmware upgrade – done in a few months)
- 802.3 proposed effort to standardize single lane 2.5 and 5 Gbps speeds in silicon (multi-year effort)

Other FMS Tutorials

- After lunch, attend these tutorials:

Check out SNIA Tutorials:

Flash Storage for Backup, Recovery and DR

Utilizing VDBench to Perform IDC AFA Testing

NVDIMM Cookbook – A Guide to NVDIMM Integration

Attribution & Feedback

The SNIA Education Committee thanks the following Individuals for their contributions to this Tutorial.

Authorship History

Mark Carlson and the Object Drive TWG
September 2015

Additional Contributors

David Slik
Robert Qiuxin
Paul Suhler

Please send any questions or comments regarding this SNIA Tutorial to ***tracktutorials@snia.org***