What Software Defined Storage Means for Storage Networking

Live Webcast
October 22, 2019
10:00 am PT
SNIA-at-a-Glance

185 industry leading organizations

2,000 active contributing members

50,000 IT end users & storage pros worldwide

Learn more: snia.org/technical

@SNIA
Technologies We Cover

- Ethernet
- iSCSI
- NVMe-oF
- InfiniBand
- Fibre Channel, FCoE
- Hyperconverged (HCl)
- Storage protocols (block, file, object)
- Virtualized storage
- Software-defined storage
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

- SDS - Storage software on commodity hardware
- SDS versus Traditional
- Converged Infrastructure / Hyper-Converged Infrastructure
- Cloud models
- Comparisons
- Future trends
- Q&A
## Basic Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>N</td>
<td>A given server or storage processing element</td>
</tr>
<tr>
<td>Disks</td>
<td>M</td>
<td>A given SSD or storage capacity element</td>
</tr>
<tr>
<td>Shared Nothing</td>
<td>-</td>
<td>A given SSD is accessed by only 1 processing element</td>
</tr>
<tr>
<td>Shared Everything</td>
<td>-</td>
<td>A given SSD is accessed by more than 1 processing element</td>
</tr>
<tr>
<td>High Availability</td>
<td>HA</td>
<td>Redundancy to maintain data accesses in event of a failure</td>
</tr>
<tr>
<td>Failure Domain</td>
<td>FD</td>
<td>A domain of subsystems that can survive a subsystem failure</td>
</tr>
<tr>
<td>EN, FC, IB</td>
<td>EN, FC, IB</td>
<td>Ethernet, Fibre Channel, InfiniBand network links</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>BW</td>
<td>Transmission / reception data capacity of a network link</td>
</tr>
<tr>
<td>RAID Head</td>
<td>RBOD</td>
<td>Storage processing subsystem of traditional storage</td>
</tr>
<tr>
<td>Just a Bunch of Disks</td>
<td>JBOD</td>
<td>Storage capacity expansion behind a node or RBOD</td>
</tr>
<tr>
<td>Round Robin</td>
<td>RR</td>
<td>Approach to path selection where all paths have equal preference</td>
</tr>
<tr>
<td>Asymmetrical Logical Unit Assignment</td>
<td>ALUA</td>
<td>Approach to path selection where a specific path has preference</td>
</tr>
<tr>
<td>TeraByte</td>
<td>TB</td>
<td>2 to the power of 40 Bytes of storage capacity</td>
</tr>
</tbody>
</table>
What is SDS vs Traditional Storage

❖ **SDS = Software Defined Storage**
  - Comprised of several nodes of commodity server hardware
  - Comprised of storage focused software on each node
  - Clustered together to form a storage solution

❖ **Traditional Storage = Classic Storage Appliance**
  - Comprised of a purpose built storage hardware
  - Comprised of proprietary storage software
  - Together, forms a storage solution
SDS Introduction

- **SDS = Storage software on commodity hardware**
  - Storage software = Open or proprietary software
    - Linux based + specific storage services (Dedupe, snapshots, mirroring)
  - Commodity hardware = Standard server (usually X86 based)
    - Intel or AMD based servers
    - Modelled as hardware selection that “meets a few storage requirements”
  - Storage architecture = Shared nothing access model
    - Each server node or “controller” owns its respective drives
    - Non **HA**: Single node  **HA**: A node cluster with copies of data sprayed around
  - Solution = a scaled out cluster of nodes
    - \(N\) number of nodes each with \(M\) number of disks interconnected via Ethernet
  - Management: storage at solution level, Platform at node level
Traditional Storage Introduction

- Traditional = Storage software on purpose built appliance
  - Storage software = Proprietary software
    - Can be Linux based but many specific storage services added
  - Purpose-built hardware = Dual X86 controllers in single chassis
    - Intel / AMD CPUs but with power protection & inter controller communications
    - Modelled as “must buy software and hardware together” i.e. no “roll your own”
  - Storage architecture = Shared everything access model
    - Both controllers have access to common pool of drives
    - HA: controllers access common drives with failover from one controller to other
  - Solution = single or scaled out cluster of enclosures
    - Single enclosure with disks or multiple enclosures with associated disks
  - Management: storage and platform at appliance level
SDS vs Traditional Storage

Software Defined Storage

SDS Nodes
(Storage Software)
(HDDs and/or SSDs)

SDS Mgmt
(Solution & Platform)

Interconnect Fabric
(Usually EN)

Host / App Server

Covers

Reads

writes

Traditional Defined Storage

Traditional Storage
(Dual controllers)

(Storage mgmt)

(Shared drives)

Host / App Server

Storage

Interconnect Fabric
(FC / EN Usually)

Reads

writes

Copies
(Temporary)
(Cache Sync)

© 2019 Storage Networking Industry Association. All Rights Reserved.
Converged Infrastructure = SDS or traditional storage + compute & network
- Same storage architectures as SDS / traditional storage
- Solution management now includes the hosts, fabrics, and apps
- Independent scaling of app/compute, storage, & fabrics

Hyper converged infrastructure = Fixed Building Block
- Compute + network + storage in a fixed ratio within a single platform
- Solution management covers all subsystems in platform
- Fixed ratio(s) scaling, since app, storage, fabrics scale with N platforms
  - Ratios may vary (heavy vs light compute), but they are still in groups of functions
**CI vs HCI**

**Converged Infrastructure**

- **SDS Nodes** *(Storage Software)* *(HDDs and/or SSDs)*
- **SDS Mgmt** *(Solution)* *(Platform)*
- **Host / App Server**
- **Interconnect Fabric** *(Usually EN)*

**Hyper – Converged Infrastructure**

- **SDS Nodes** *(Storage Software)* *(HDDs and/or SSDs)*
- **SDS Mgmt** *(Solution)* *(Platform)*
- **Host / App Server**
- **Interconnect Fabric** *(Usually EN)*
- **Platform** *(Server + Storage + Fabric)*

**Collection of Discrete Servers, Network, Storage**

**Servers, Network, Storage in One Platform**
Networking Challenges

- Transition from traditional to SDS/CI/HCI introduces challenges
  - Physical network typically shared between multiple elements
  - Logical segregation leads to fragmentation
  - Variable latencies undesirable for storage
- “Bump-in-the-wire” services commonly implemented in VMs
  - Requires tuning/optimization to avoid bottlenecking
Public vs. Hybrid Cloud

- Public Cloud (everything on Cloud Service Provider (CSP) hardware)
  - Large (virtually infinite) capacity / (typically) modest performance / simplicity
  - All storage in the cloud, solution management interfaces to cloud
  - CSP implements storage services in opaque fashion to customer
    - Tenant specifies “what” not “how”; Service-Level Agreement (SLA) drives contract

- Hybrid Cloud (workload and storage span from CSP to on-premises hardware)
  - On-Premises: typically provides modest capacity / higher performance
  - Off premises: large capacity / modest performance / simplicity (limited by cost)
  - Data movement costly between public cloud and on premises
  - On premises: Same storage architectures as SDS, CI, HCI
  - New services required to provide right data service to solution architecture
    - Data caching, backup svcs, workload migration, edge-to-cloud data mgmt, and much more
Cloud with On & Off Premises

Hybrid Cloud

- SDS Mgmt (Solution & Platform)
- SDS Nodes (On Premise)
  - (Storage Software)
  - (HDDs and/or SSDs)
- Host / App Server
- Interconnect Fabric (Usually EN)
- Cloud Storage (Off Premise)
  - (Cloud Storage Abstractions)

On Premises can be Hyper-Converged as well
Distilling Approaches

- **SDS (and CI, HCI)**
  - Simpler to implement (no traditional storage “corner” cases)
  - Despite the perception, SDS can be expensive just from the sheer node count
  - So some degree, performance customizable based on underlying HW
    - Storage software involved contribute far more to the performance characteristics

- **Cloud (without on premises)**
  - Further simplification since storage is abstracted
  - Pure off premises = No storage infrastructure but lower performance

- **Traditional storage**
  - Complex but can be higher cost and higher performance and scales capacity and performance better
## Storage Architecture Comparison

<table>
<thead>
<tr>
<th>Parameter</th>
<th>CI, HCI using SDS Storage</th>
<th>CI using Traditional Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block :: File :: Object</td>
<td>Yes :: Yes :: Yes</td>
<td>Yes :: Yes :: Yes but not usually</td>
</tr>
<tr>
<td>Storage Architecture</td>
<td>Several nodes</td>
<td>Single platform</td>
</tr>
<tr>
<td></td>
<td>Each shared nothing architecture</td>
<td>Shared everything architecture</td>
</tr>
<tr>
<td>Storage Host Interface</td>
<td>EN, FC &amp; IB Rare</td>
<td>EN, FC, IB</td>
</tr>
<tr>
<td>Storage High Availability</td>
<td>Node errors: Node failover</td>
<td>Controller errors: Failover to peer controller</td>
</tr>
<tr>
<td></td>
<td>Disk errors within node: RAID, etc</td>
<td>Disk errors: RAID, etc</td>
</tr>
<tr>
<td>Storage Failure Domain</td>
<td>CI or SDS: Groups of nodes</td>
<td>Within enclosure</td>
</tr>
<tr>
<td></td>
<td>HCI: Within building block</td>
<td></td>
</tr>
<tr>
<td>Storage Scaling</td>
<td>CI or SDS: More nodes and/or more disks</td>
<td>More disks and/or JBODs and disks</td>
</tr>
<tr>
<td></td>
<td>HCI: More building blocks (compute network storage)</td>
<td></td>
</tr>
<tr>
<td>Storage Costs</td>
<td>Higher due to number of nodes</td>
<td>Lower due single storage platform</td>
</tr>
<tr>
<td>Storage Efficiency</td>
<td>Lower due to number of copies</td>
<td>Higher due to full use of drives</td>
</tr>
</tbody>
</table>
Futures Trends to Consider

- **NVMe™ et al**
  - Increased performance
    - Enhancements to the use of namespaces (shared, streams, etc)
    - Speculation around NVMe paradigm shifting to HDDs (simplify storage stacks)

- **Cloud – How best to use it?**
  - Data = Security concerns whether app in cloud or data in cloud
  - Processing = Performance concerns
    - App processing on premises / data in cloud: Cloud latency for data access
    - App processing in cloud / data on premises: “batch” jobs / limited interactive jobs?
    - App processing and data in clouds: “batch jobs” & latency
Futures Trends to Consider

- SDS vs Traditional storage
  - Mindshare around SDS despite complexity and cost
    - Simpler to implement ("throw HW at the problem", limited storage skill needed)
      - Limited shared everything corner cases
  - Limited programming skills for Traditional Storage
    - Throttles traditional storage innovation
    - Creates barrier to market entry (need deep pockets for special skills)
  - High solution cost
    - Speculation regarding opportunities for ARM to drive down costs
    - Disaggregate storage processing from storage capacity
      - Cost optimized storage processing nodes ⇒ cost optimized capacity nodes?
Summary

- **SDS = Storage Stack on Commodity Hardware**
  - Shared nothing architecture (HA via copies et al)

- **Traditional Storage = Storage Stack on targeted appliances**
  - Shared everything architecture (HA inherent)

- **CI = Leverage either SDS or traditional storage**
  - Traditional storage + separate servers and networking
  - SDS storage + separate servers and networking

- **HCI = SDS storage, servers, & networking in one platform**
  - Tradition storage may have a role in 2nd or 3rd tier of storage

- **On premises Cloud = Collection of CI or HCI**

- **Off premises Cloud = CI or HCI interacting with public cloud**
After This Webcast

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Networking Storage Forum (NSF) website and available on-demand at [www.snia.org/forums/nsf/knowledge/webcasts](http://www.snia.org/forums/nsf/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-NSF blog: [sniansfblog.org](http://sniansfblog.org)
- Follow us on Twitter @SNIANSF
More Webcasts

❖ Introduction to Incast, Head of Line Blocking, and Congestion Management:
  On-Demand Webcast: https://www.brighttalk.com/webcast/663/356343
  SNIAVideo YouTube: https://youtu.be/WKAtGUph-hw

❖ New Landscape of Network Speeds:
  On-Demand Webcast: https://www.brighttalk.com/webcast/663/356464
  SNIAVideo YouTube: https://youtu.be/KMkFSwWsEWE

❖ File vs. Block vs. Object Storage:
  On-Demand Webcast: https://www.brighttalk.com/webcast/663/308609
  SNIAVideo YouTube: https://youtu.be/Akpu35MKzrA

❖ SNIA resources on SDS
  Article - Software Defined Storage: https://www.snia.org/sites/default/files/SDS_Article.pdf
  Educational page - https://www.snia.org/sds
  SNIA Dictionary - https://www.snia.org/education/online-dictionary/term/software-defined-storage
Thank You