STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

Computational Storage Spark/Kubernetes

Containerized, Managed, Local, Accelerated

Scott Shadley, VP Marketing, NGD Systems
Director, Board of Directors of SNIA
Co-Chair, Computational Storage TWG

Bringing Real Value to the Data

Computational Storage Acceleration of Results on Stationary Data

The Market Evolution and Need for Local Compute

Our Friends at Gartner Say it best…

Structured Data is great for current infrastructure

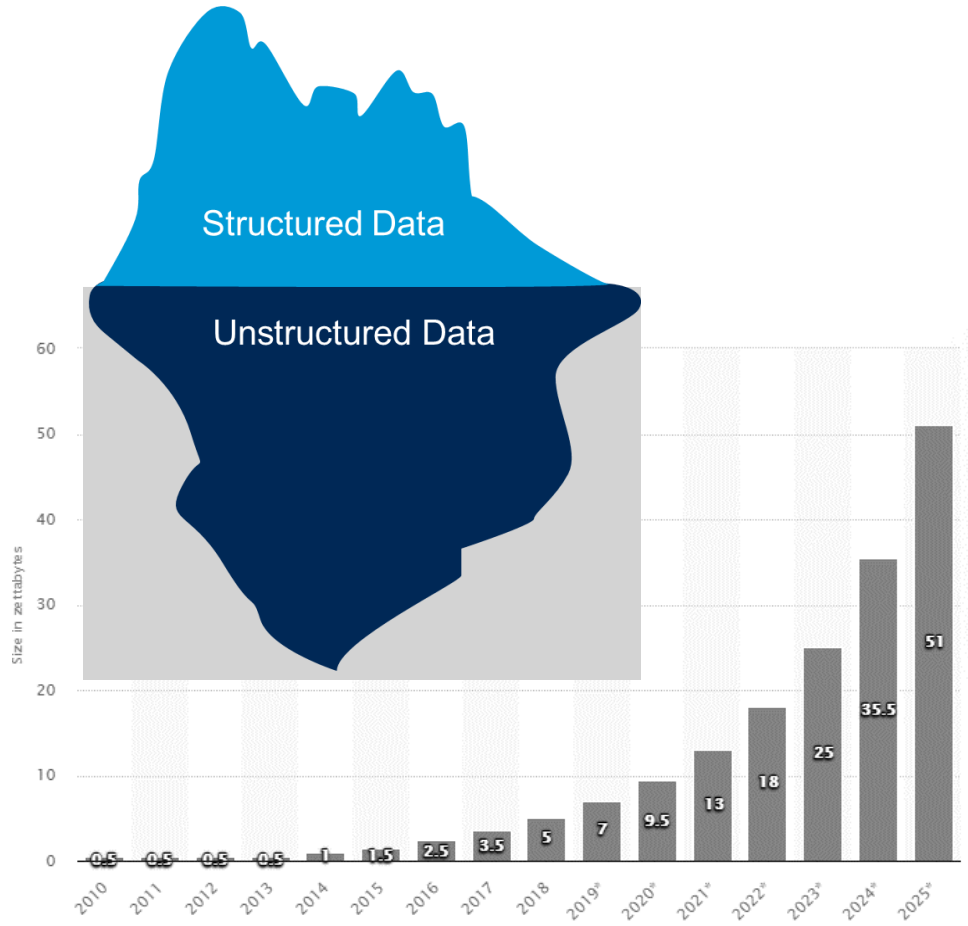
- Allows for ease of data movement, location, access, compute.
- Only a small subset of the real data Iceberg

Unstructured Data is the greatest threat to results

- As more and more data is generated, it is more random
- Needs to manage this data locally are key
- **Edge Computing is not able to scale** at data growth pace
- A new way to compute on random, local data is needed

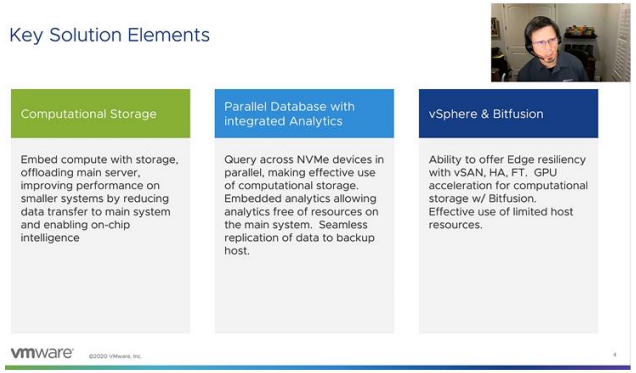
The Global DataSphere (Statista.com) shows how the data growth is overshadowing the compute growth

CHANGE IS NEEDED

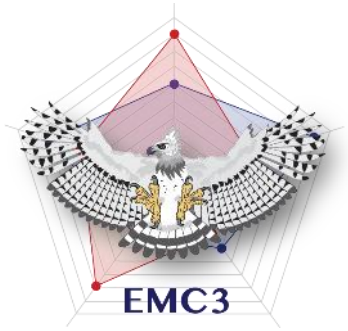


Partnerships Drive Innovation and Adoption

- VMware Virtualization and Database Acceleration
  - Customer Engaged, Joint Deployment focus
  - Edge Analytics, migrating Nodes from CPUs to CSDs



- Los Alamos National Labs
  - Partnership to evaluate the value of Distributed Processing
  - Spark via Kubernetes one of several projects



Los Alamos National Laboratory welcomes NGD Systems to the Efficient Mission Centric Computing Consortium

The collaborative effort will explore high capacity NVMe computational storage drive, and scalable computational offloads for HPC and scalable computing uses

The What and Why

What is Computational Storage and Why Spark?

EASY Steps Coming

- The NGD Systems Computational Storage Drive (CSD)
  is LINUX OS in a Drive… Let's just use an (ssh)

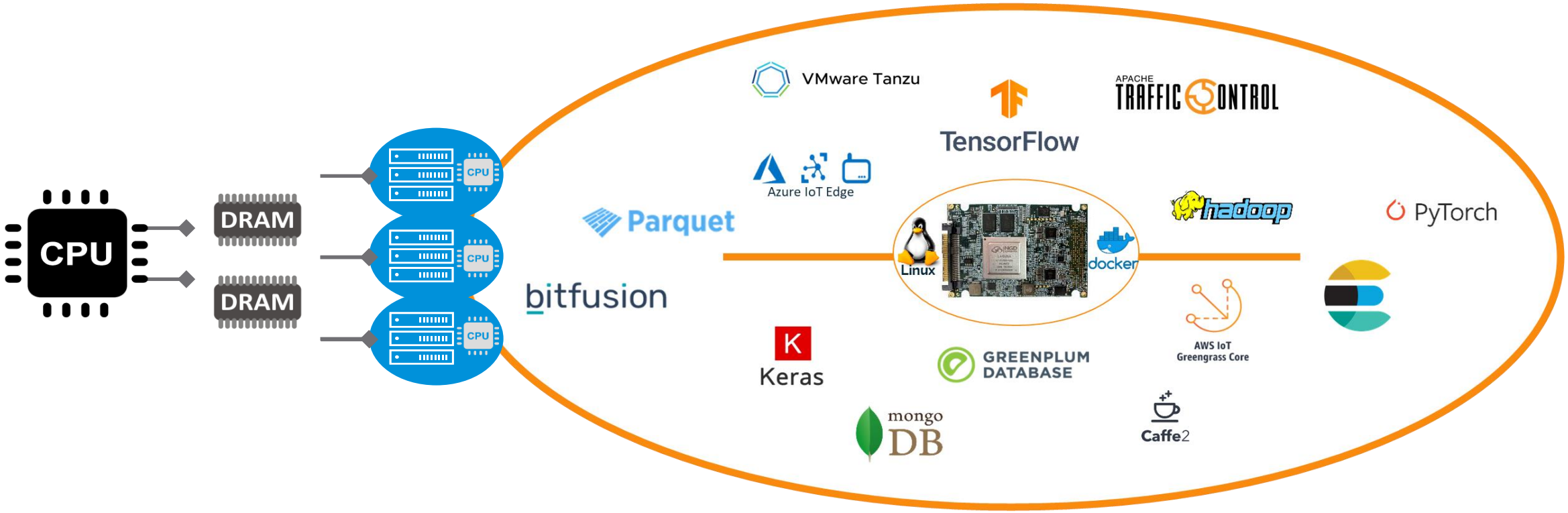
- All code used in the following slides is "COTS"
  - Consumer Off the Shelf – NOT CUSTOM ☺

- Standard NVMe Storage, Linux OS instructions

- Intern or Experienced, It is 'that easy'

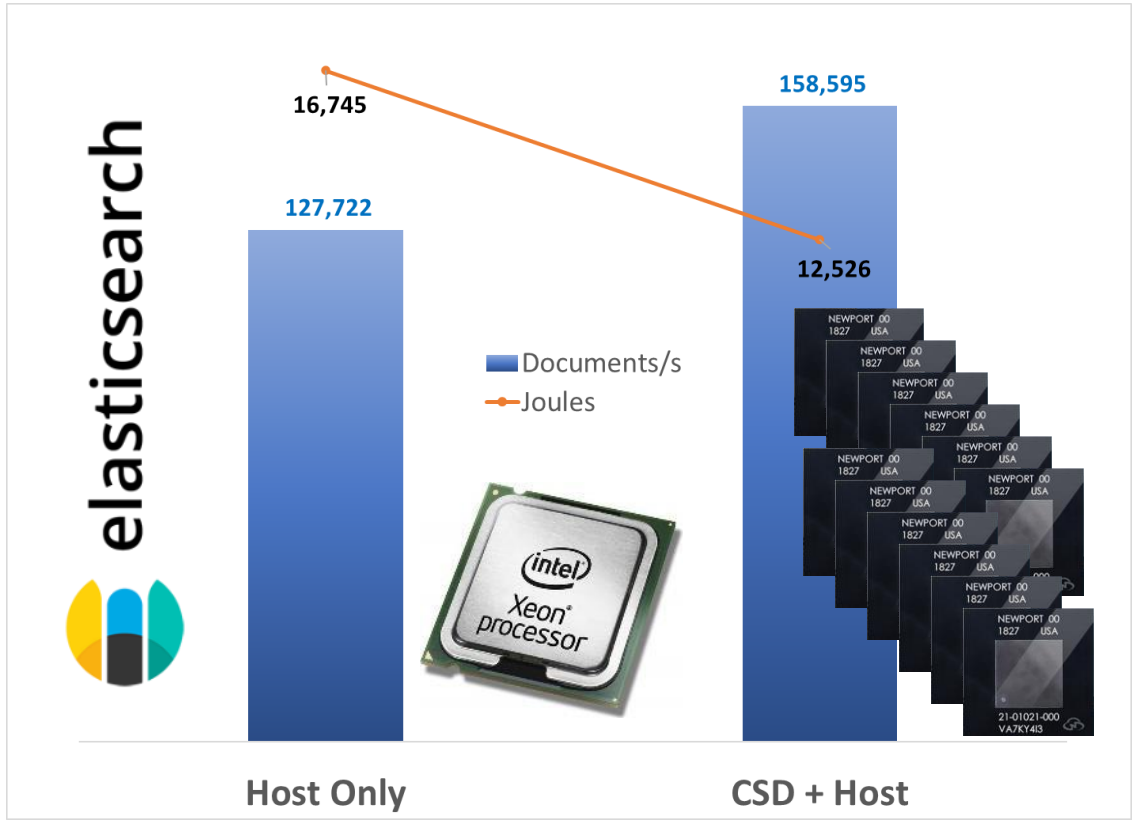
Before we get to Spark, Let's See History…

- **Keep It Simple & Seamless** – K.I.S.S.
  - The best way to move forward is to leverage architectures already in use

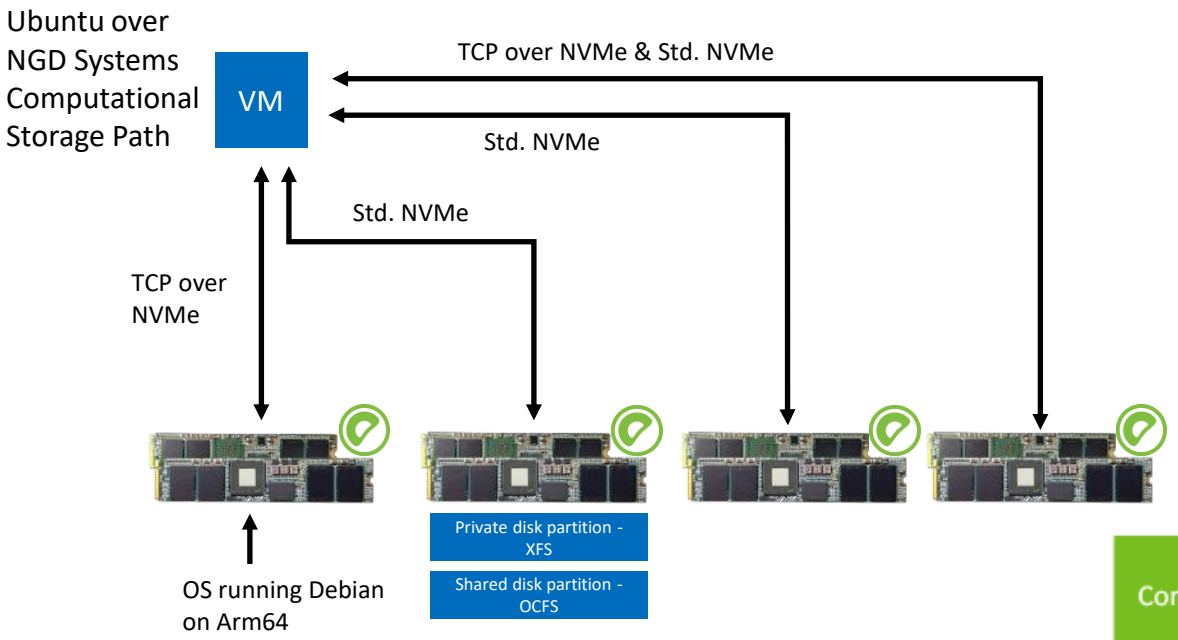


Elasticsearch – Memory Focus

- **Total Performance Improves**
  - **20% Better** Results
- **Reduced Power Consumption**
  - **30% LESS** Power
- **DRAM Usage Reduced by >50%**
  - Host Only used **25GB**
  - Hybrid used **12GB**
- **CPU Usage Utilization Reduced by >50%**
  - Host Only used **24%**
  - Hybrid used **10%**



Integration of NGD Systems Devices to vSphere



1. VM Directpath IO for NVMe devices
    1. up to 15 into a VM
2. TCP connected jump box allows addressing of devices from network.
3. Two partitions
    1. one shared w/OCFS
    2. one dedicated to Greenplum

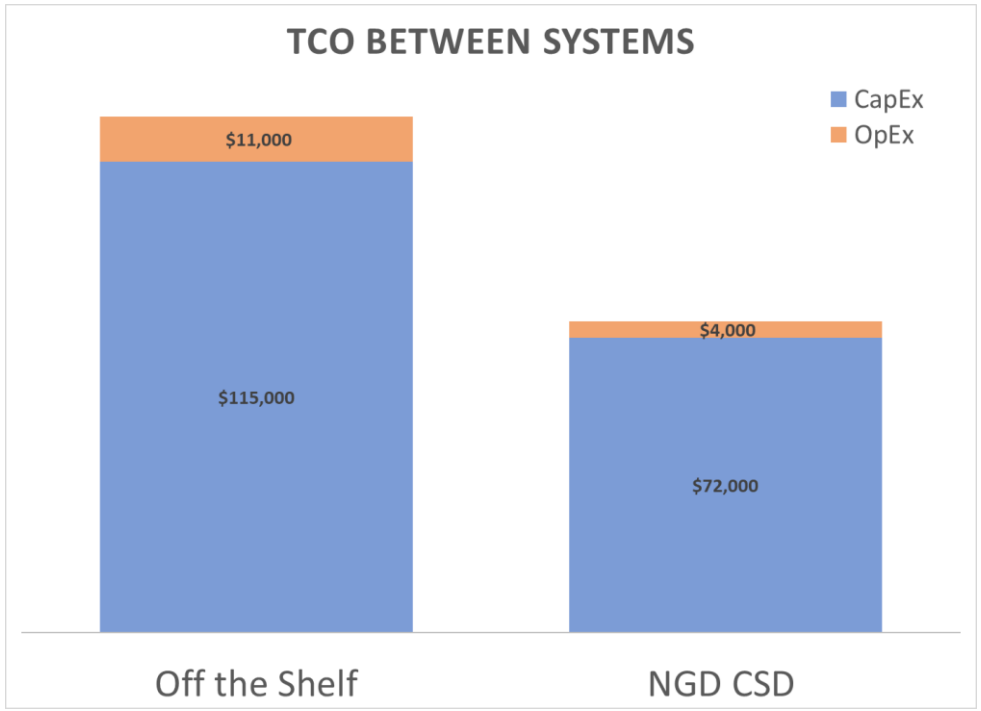
[LINK TO ONLINE DEMO – VMWorld 2020]

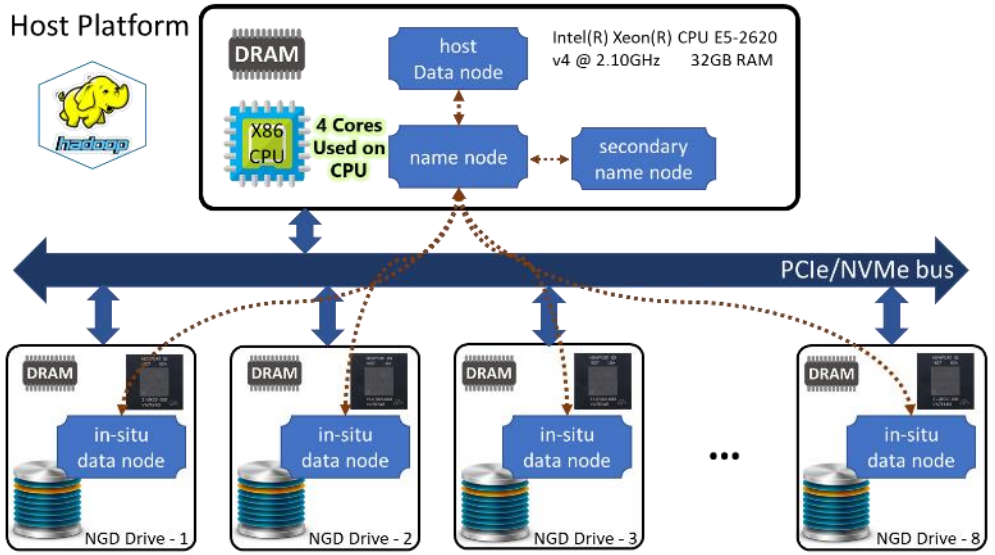
Key Points

- 1 GB / sec transfer per NVMe device
- 16TB capacity per device
- Simultaneous addressing as storage device & as remote compute node
- PCI passthrough allows native use by VMs
- Greenplum running on each node

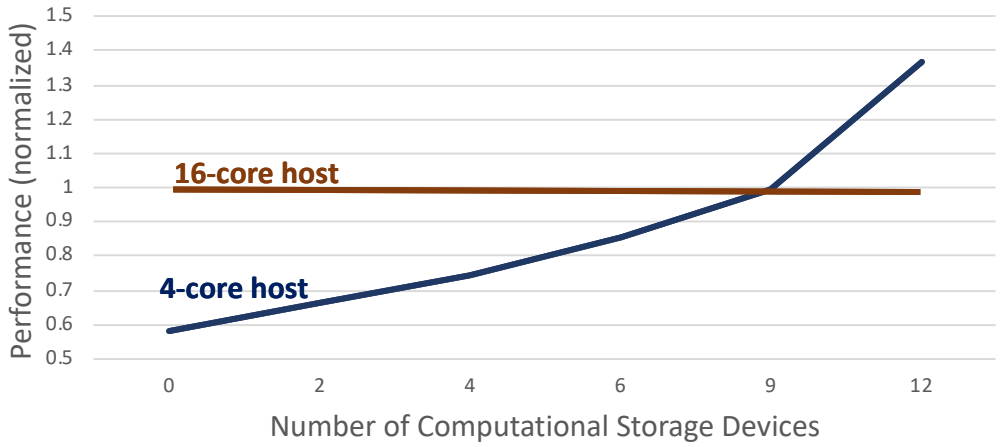
Computational Storage	Parallel Database with integrated Analytics	vSphere & Bitfusion
Embed compute with storage, offloading main server, improving performance on smaller systems by reducing data transfer to main system and enabling on-chip intelligence	Query across NVMe devices in parallel, making effective use of computational storage. Embedded analytics allowing analytics free of resources on the main system. Seamless replication of data to backup host.	Ability to offer Edge resiliency with vSAN, HA, FT. GPU acceleration for computational storage w/ Bitfusion. Effective use of limited host resources.

What about Mongo? Hadoop?





Terasort performance



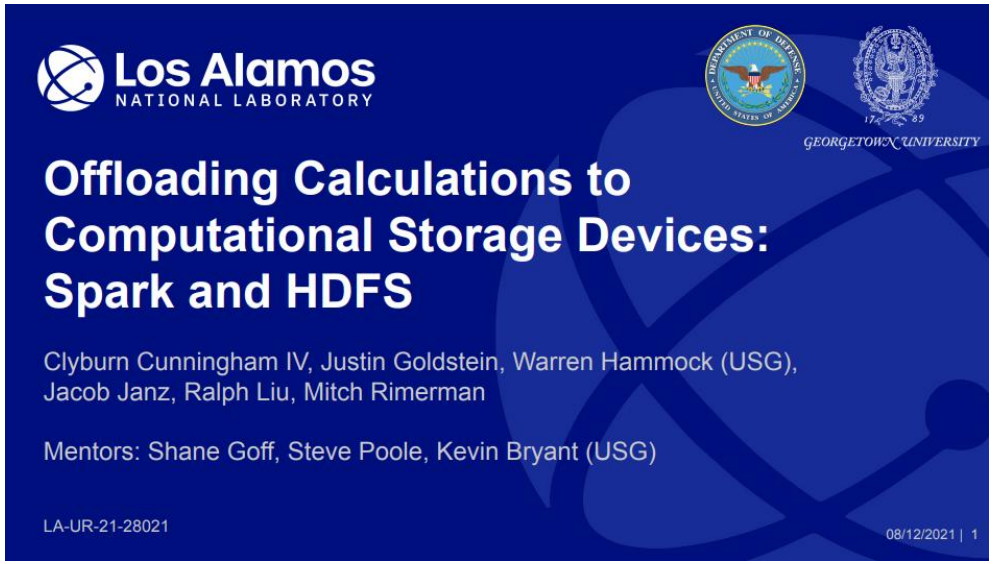
The Focus of this Discussion

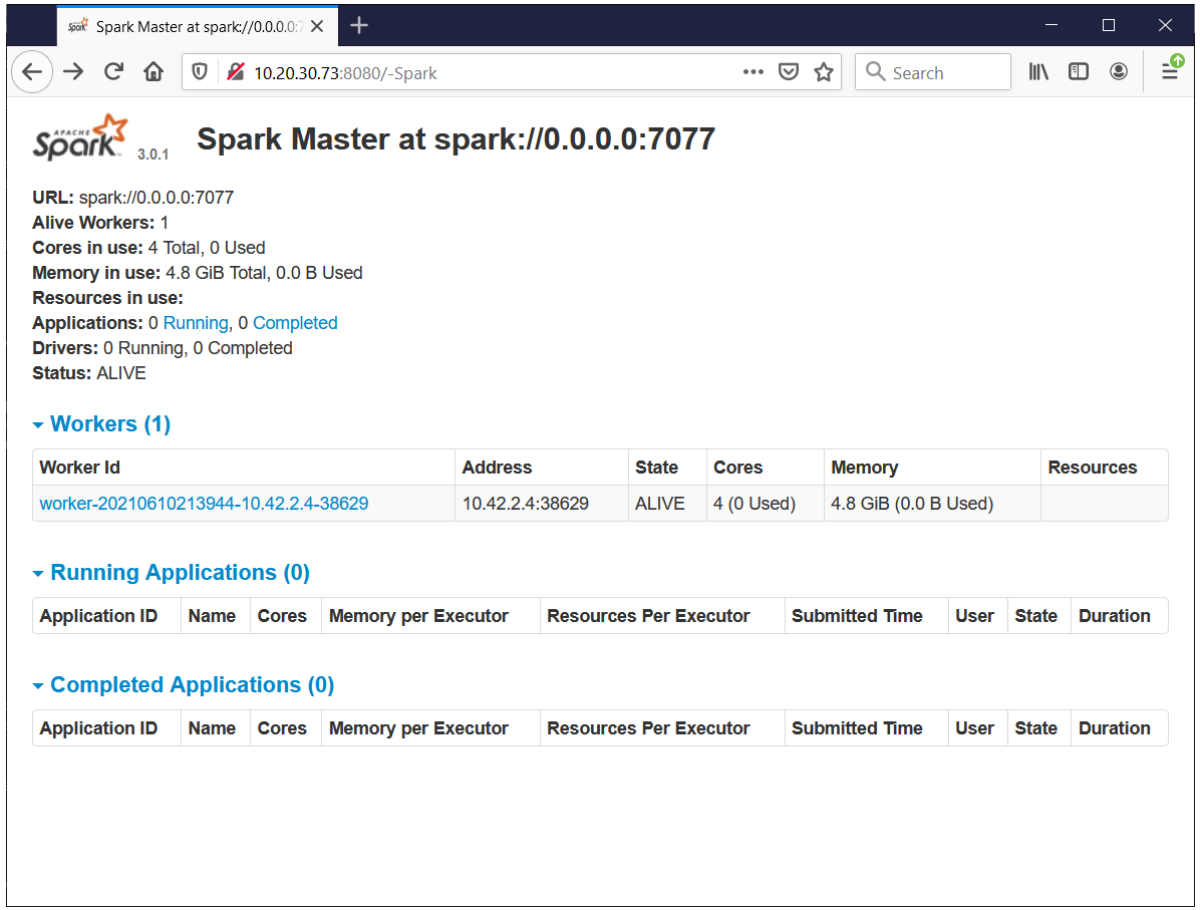
Let’s Get Back to Spark

Developed, Deployed, Demonstrated

NGD Systems

Spark Cluster – Newport deployment guide





What is Required To Get There?

- Combination of Tools
  - Containers
  - Kubernetes
  - Hadoop
  - Spark

```
root@In-Situ-machine2:~# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda           8:0    0 931.5G  0 disk
├─sda1        8:1    0   9.3G  0 part [SWAP]
├─sda2        8:2    0     1K  0 part
├─sda5        8:5    0 186.3G  0 part /
└─sda6        8:6    0   736G  0 part /media/data
sdb           8:16   0 931.5G  0 disk
├─sdb1        8:17   0   300G  0 part /media/diskB
└─sdb2        8:18   0 631.5G  0 part /media/2_diskB
nvme0n1     259:0    0   5.8T  0 disk
├─nvme0n1p1 259:1    0   100G  0 part
└─nvme0n1p2 259:2    0     5T  0 part
root@In-Situ-machine2:~#
```

```
root@In-Situ-machine2:~# apt install nvme-cli
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

Code & Scripts

The source code for the experiments in this report is located at the public repository ngd_docker_images.

What Does It Take to Get an ssh?

- The ONLY required file to Activate…
  - `In-Situ_Control_Newport.sh`
- Effectively TCP over NVMe
- That's it… One simple tool
- Then you have an OS on the drive
  - Optimized OS for Drive Level Use

```
root@In-Situ-machine2:~# nvme io-passthru /dev/nvme0n1 -o 0xe0 datalen=4096 -r
NVMe command result:00000000
root@In-Situ-machine2:~# ./In-Situ_Control_Newport.sh start
 Starting Tunel nvme0n1 tap1 10.1.1.1 255.255.255.0
nohup: redirecting stderr to stdout
 starting Device Agent daemons
Warning: Permanently added '10.1.1.2' (ECDSA) to the list of known hosts.
Creating iptables rules for accepting packets from/to tunel interface tap1
Warning: Permanently added '10.1.1.2' (ECDSA) to the list of known hosts.
 Changed the device name back to default :
localhost
 Device name has changed:
node1
 Creati
 Adding
root@In
```

```
root@In-Situ-machine2:~# ssh ngd@node1
The authenticity of host 'node1 (10.1.1.2)' can't be established.
ECDSA key fingerprint is SHA256:sb1P6g/5yXByuQR4kYKUZoEwj34hvOSGU1cLj64G6nM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'node1' (ECDSA) to the list of known hosts.
ngd@node1's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 4.14.1_newport_4.1_5.2+ aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.
```


Time to Get Kubernetes Running

- To make it easier to manage a large number of CSD nodes from the host with minimum effort, we will install Spark as part of a Kubernetes Cluster.
- For that we need to install docker runtime and all the necessary dependencies to get Kubernetes cluster (K3S) up and running.
- “Off the Shelf” commands…

```
root@In-Situ-machine2:~# k3s kubectl get node
NAME              STATUS     ROLES                  AGE     VERSION
in-situ-machine2  Ready      control-plane,master   3m    v1.21.1+k3s1
root@In-Situ-machine2:~#
```

```
root@In-Situ-machine2:~# k3s kubernetes get node
NAME              STATUS     ROLES                  AGE     VERSION
node1             Ready      <none>                   1m    v1.21.1+k3s1
in-situ-machine2  Ready      control-plane,master    45m    v1.21.1+k3s1
root@In-Situ-machine2:~#
```

- To make it easier to deploy the Spark service as a cluster, there is a Git repository with the necessary scripts and a few examples.

Let’s Fire up Spark

```
root@In-Situ-machine2:~#~/ngd_docker_images/kubernetes/bigdata2$ ./deploy.sh csd
Deploying in csd mode
Clearing data in host ...
Clearing data in node1 ...
host cleared.
Done!
pod/hadoop-primary created
daemonset.apps/spark-worker-csd created
daemonset.apps/hadoop-worker-csd created
pod/spark-primary created
service/spark-primary created
service/hadoop-primary created
Summary: ContainerCreating=8, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=0
Summary: ContainerCreating=8, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=0
Summary: ContainerCreating=7, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=1
Summary: ContainerCreating=7, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=1
Summary: ContainerCreating=7, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=1
Summary: ContainerCreating=7, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=1
Summary: ContainerCreating=6, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=2
Summary: ContainerCreating=6, Pending=0, ErrImagePull=0, ImagePullBackOff=
Running=2
Summary: ContainerCreating=6, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=2
Summary: ContainerCreating=6, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=2
Summary: ContainerCreating=5, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=3
Summary: ContainerCreating=4, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=4
Summary: ContainerCreating=4, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=4
Summary: ContainerCreating=2, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=6
Summary: ContainerCreating=2, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=6
Summary: ContainerCreating=2, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=6
Summary: ContainerCreating=0, Pending=0, ErrImagePull=0, ImagePullBackOff=0, CrashLoopBackOff=0, Error=0,
Running=8
Done!
root@In-Situ-machine2:~#
```

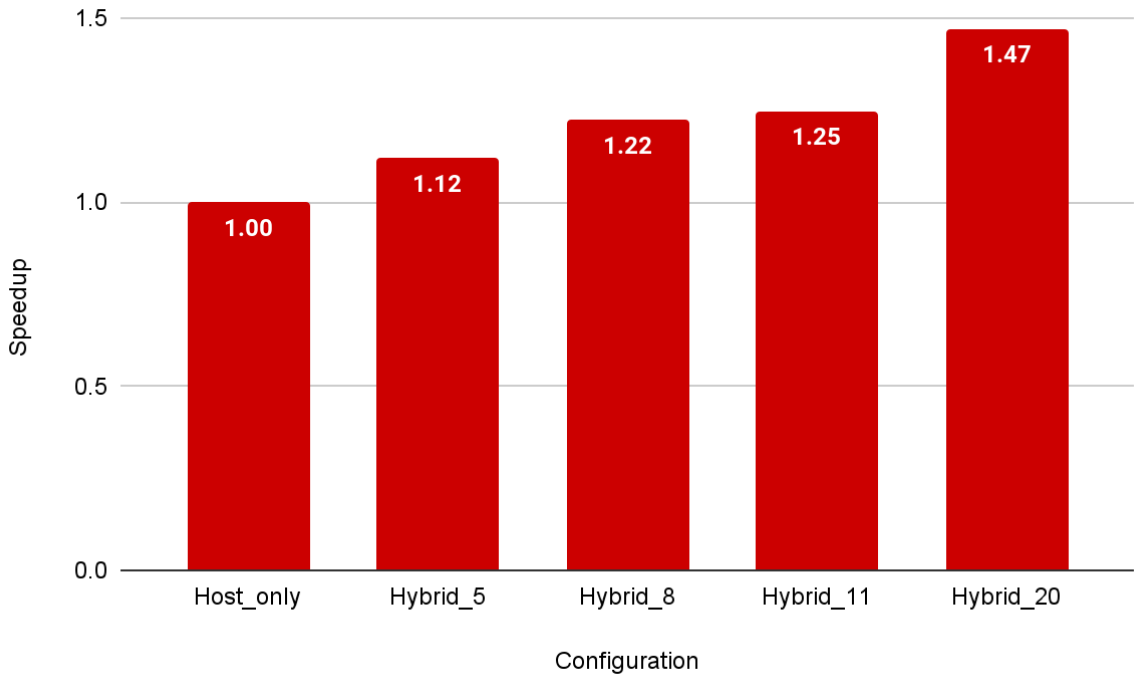
- One Line Of Code…

```
root@In-Situ-machine2:~# sudo k3s kubectl get pod -o wide
NAME                          READY   STATUS    RESTARTS   AGE    IP           NODE              NOMINATED NODE   READINESS
GATES
svclb-spark-primary-gnxfm     4/4     Running   0          5m6s   10.42.1.4    node1             <none>           <none>
svclb-spark-primary-w6d4n     4/4     Running   0          5m6s   10.42.0.10   in-situ-machine2  <none>           <none>
svclb-hadoop-primary-h2kjw    45/45   Running   0          5m5s   10.42.1.6    node1             <none>           <none>
svclb-hadoop-primary-jqmdl    45/45   Running   0          5m5s   10.42.0.12   in-situ-machine2  <none>           <none>
hadoop-primary                1/1     Running   0          5m6s   10.42.0.9    in-situ-machine2  <none>           <none>
spark-primary
spark-worker-csd-w691m
hadoop-worker-csd-7sfvc
root@In-Situ-machine2:~#
```

```
root@In-Situ-machine2:~# k3s kubectl exec -it spark-primary -- spark-shell
21/06/10 23:01:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
classes where applicable
Spark context Web UI available at http://spark-primary:4040
Spark context available as 'sc' (master = local[*], app id = local-1623366081445).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.0.1
      /_/

Using Scala version 2.12.10 (OpenJDK 64-Bit Server VM, Java 1.8.0_292)
Type in expressions to have them evaluated.
Type :help for more information.

scala>

scala>
```


So… Running, Now OUR Results…

- We selected the SparkPI example, present in the Spark examples folder.
  - This example has the characteristic that it is a CPU-intensive operation with very little network communication



- As observed, the performance gain keeps growing as we add more nodes into the system.

So… Running, Now THEIR Results…

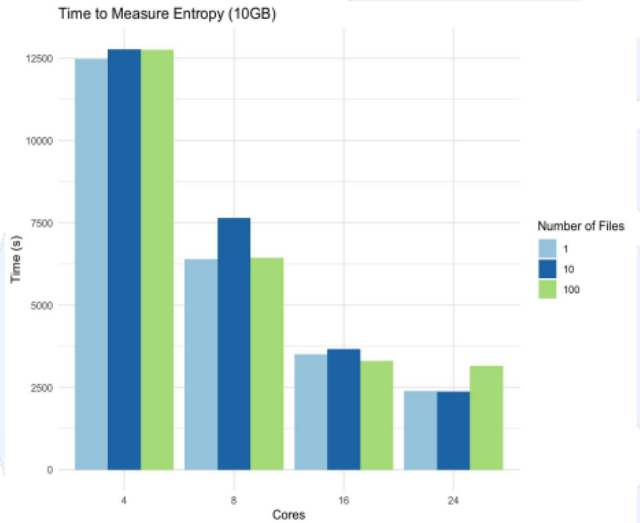
Experimental Objective and Design

- Objective → Evaluate the capabilities of multiple CSDs (provided by NDG Systems) using Hadoop Filesystem and Apache Spark
  - Use native Spark libraries, such as SparkSQL and DataFrames, to perform matrix operations on datasets
- Independent Variables
  - # of CSDs → 0, 1, 2, 4, or 6
  - Size of dataset → 1 GB, 5 GB, 10 GB
  - Type of dataset → One large file with all of the data, 10 files, 100 files
- Dependent Variables
  - Job time
  - Execution time
- Constants
  - Operations on dataset

Los Alamos NATIONAL LABORATORY

Number of Files

- Increased performance with more CSD's
- Similar observations for different file amounts
- Lesser improvement for more nodes with large amount of files

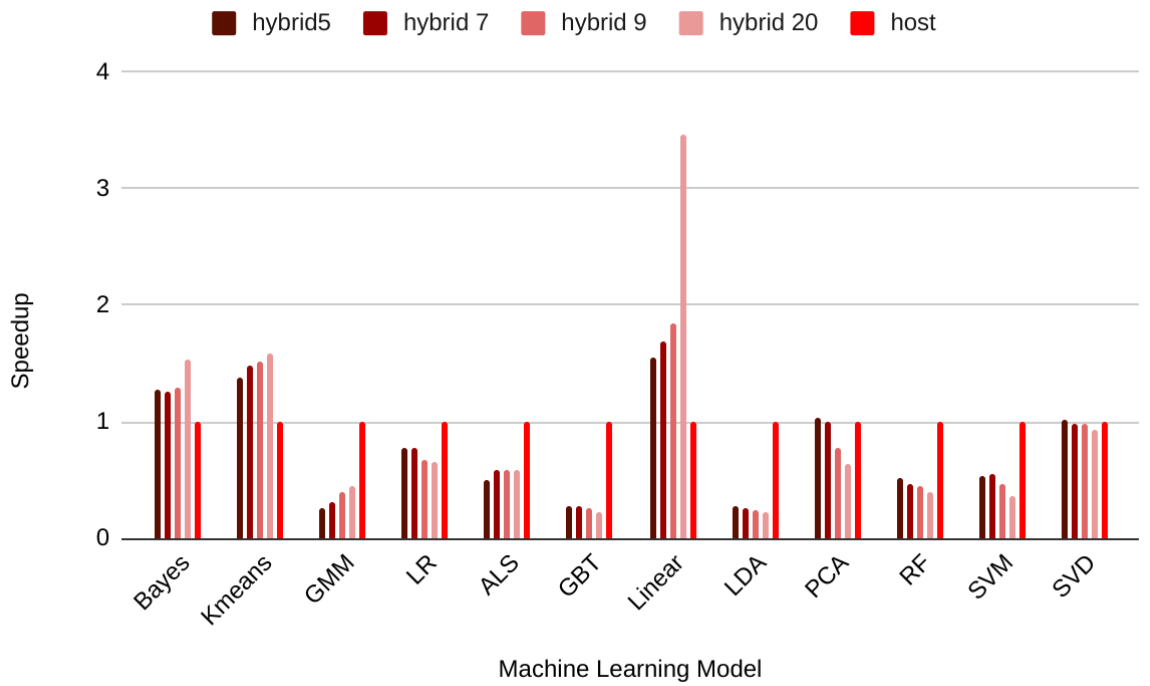


Los Alamos NATIONAL LABORATORY

08/12/2021 | 10

So, What's the Catch?

- THIS IS NOT A ONE SIZE FITS ALL
- There are many ways to 'Solve' problems…
  - Memory-Centric Tools need Memory
  - Storage-Centric Tools Get Faster
  - Less Data, Less Acceleration
- NOT A CPU replacement!!
  - This is support/augment
- If Data MOVES, we can Help!!



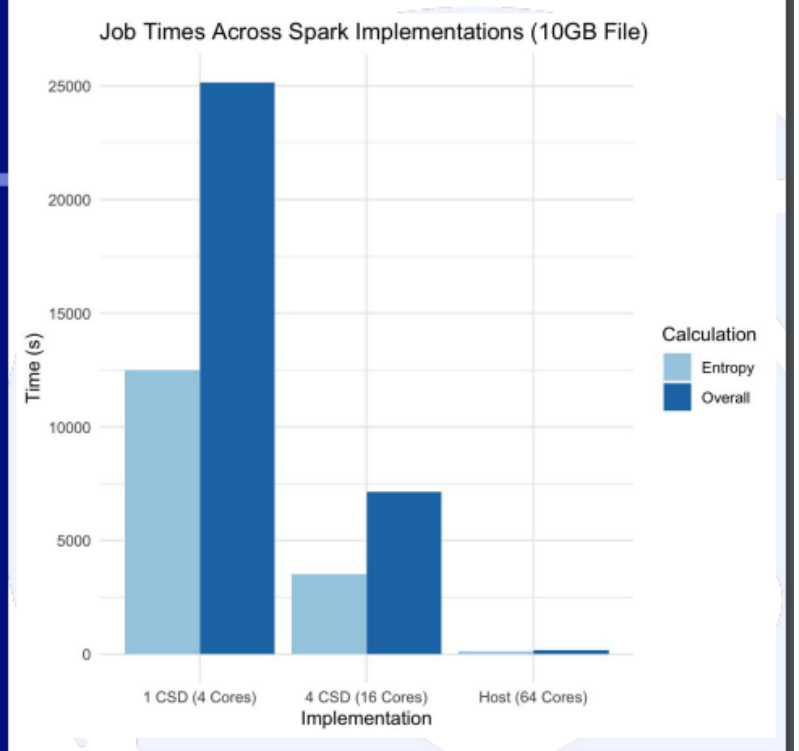
Results on SMALL Files

- Core to Core performance – Host Faster – Augmentation Still There!

Comparison to Host

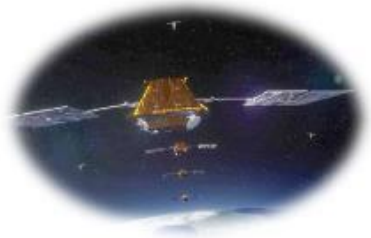
- Much faster on Host
- Assuming uniform scaling, achieving host performance would not be possible with any amount of CSDs

Los Alamos NATIONAL LABORATORY



08/12/2021 | 13

Ongoing Partnerships



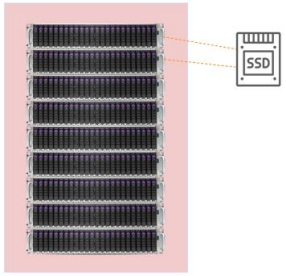
UNITED STATES SPACE FORCE

Problem/Opportunity

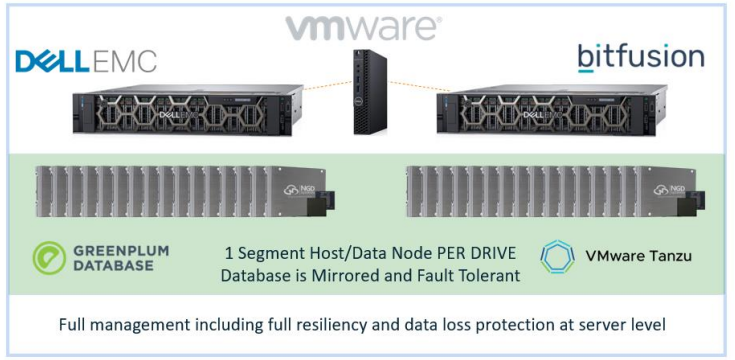
The Big Data paradigm shift, where processing is pushed as close to the data as possible, enhances the potential for Machine Learning applications generating cognitive representations of the battlespace to inform future autonomous decisions for small satellites on-orbit.

- Continued work with the Labs
- Customer PoC count increasing
  - Remote Access Lab Environments
- Scale (down) while Scale Up
  - Ongoing Edge Deployment work

Computational Storage allows it to be drive level.
Reducing footprint, server cost, while still offering full fault tolerance



Traditionally Segment Host and Data Nodes were at the **server level**.



- Redefining Workloads for Many Segments
  - AI/ML, Databases, M&E, CDN, Video Surveillance, “Pac-Man” trucking…

NVMe On-Drive Linux SSDs

- Large breadth of **STANDARD SSDs**
- Leading **TB/W**
- Industry's **Largest capacity**
- Quad-Core **Computational Storage CPUs**

Form Factor	Availability	Raw Capacity TLC (TB)	MAX Power (W)
M.2 2280	CQ4'21	up to 4	8
M.2 22110	NOW	up to 8	8
U.2 15mm	NOW	up to 32	12
EDSFF E1.S	NOW	up to 12	12
EDSFF E3	CQ1'22	up to 64	15

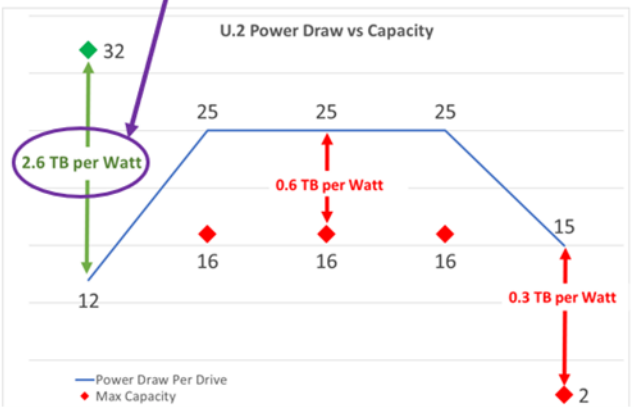
M.2

E1.S

U.2

NGD is the ONLY Provider of Capacity over Power

Another Paradigm Shift in the Market

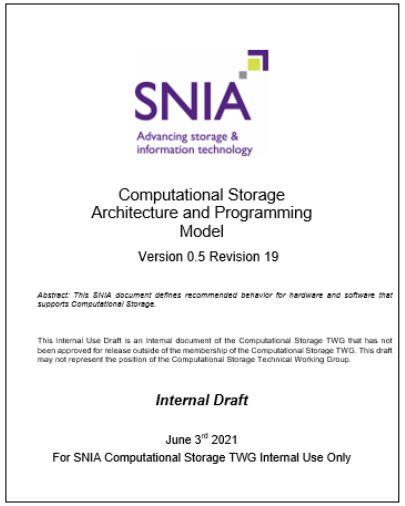


NGD Competitors

Computational Storage Solves Data Movement

SNIA is driving for an Architectural Solutions

NVM Express is working on Protocol


SNIA
Advancing storage & information technology

Computational Storage Architecture and Programming Model

Version 0.5 Revision 19

Abstract: This SNIA document defines recommended behavior for hardware and software that supports Computational Storage.

This Internal Use Draft is an internal document of the Computational Storage TWG that has not been approved for release outside of the membership of the Computational Storage TWG. This draft may not represent the position of the Computational Storage Technical Working Group.

Internal Draft

June 3rd 2021

For SNIA Computational Storage TWG Internal Use Only

nvm EXPRESS®

NVMe Computational Storage Task Group

The charter of Computational Storage Task Group is to develop features associated with the concept of **Computational Storage on NVM Express devices**.

The target audience consists of the vendors and customers of **NVMe Storage Devices** that support computational features.

Industry Prototyping and Deploying Now

vmware®

DELL EMC

arm

Hewlett Packard Enterprise

Microsoft

Lenovo

WD

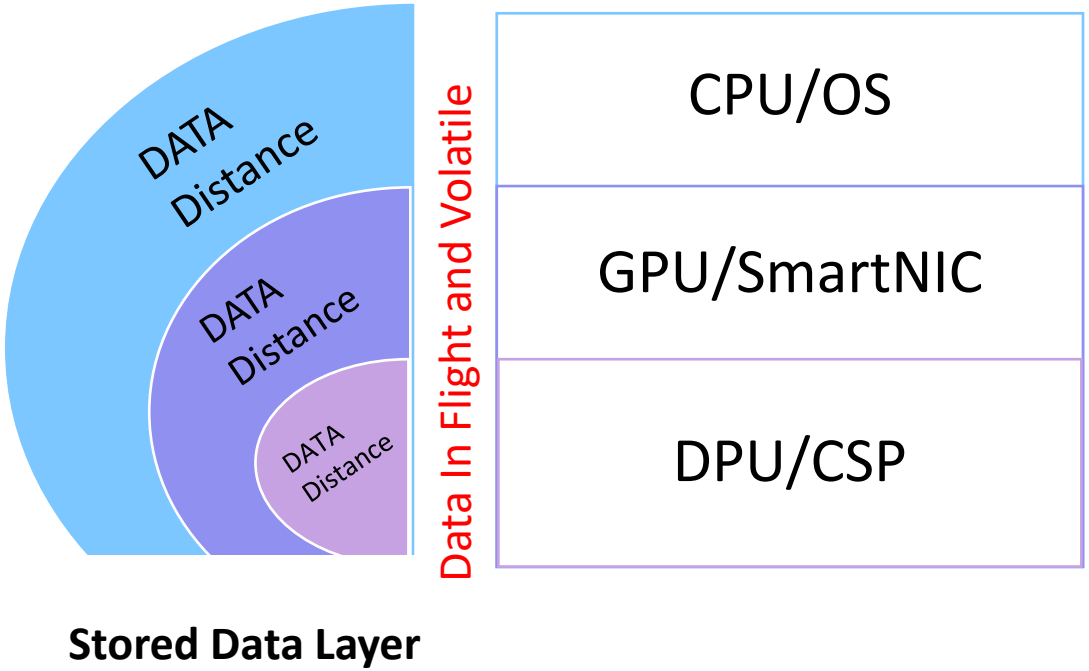
Booking.com

IBM

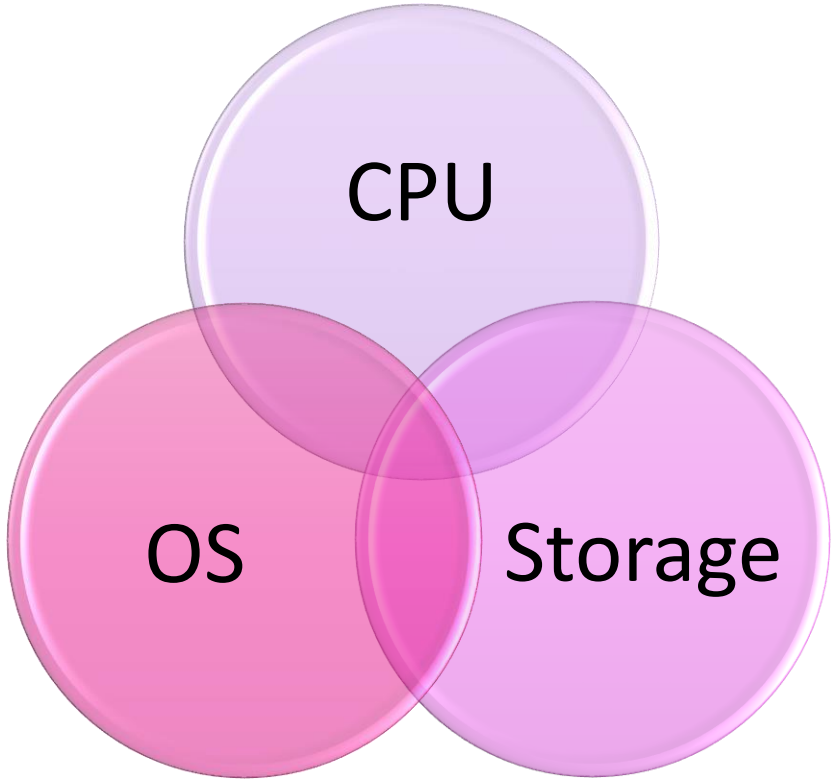
SEAGATE

A Comparison & NGD Systems CSDs Add Value

Today's Standard Infrastructure – **Data Distant**



Linux-Based Computational Storage Drive – **Data Local**



NGD Systems

Scott Shadley

https://www.ngdsystems.com/

https://www.ngdsystems.com/page/Los-Alamos-National-Laboratory-Welcomes-NGD-Systems-to-the-Efficient-Mission-Centric-Computing-Consortium

https://www.lanl.gov/org/ddste/aldsc/hpc/recruiting/intern-showcase.php

Please take a moment to rate this session.

Your feedback is important to us.

