

# Compute and Storage Innovations Combine to Provide a Pathway to Composable Architecture

*Adam Roberts*

*Engineering Fellow, Western Digital Corporation*



# Forward-Looking Statements

## *Safe Harbor / Disclaimers*

This presentation contains certain forward-looking statements that involve risks and uncertainties, including, but not limited to, statements regarding: data center trends and market needs; the RISC-V Foundation and its initiatives; our contributions to and investments in the RISC-V ecosystem; composable infrastructure products; our business strategy, growth opportunities and technology development efforts; market trends and data growth and its drivers. Forward-looking statements should not be read as a guarantee of future performance or results, and will not necessarily be accurate indications of the times at, or by, which such performance or results will be achieved, if at all. Forward-looking statements are subject to risks and uncertainties that could cause actual performance or results to differ materially from those expressed in or suggested by the forward-looking statements.

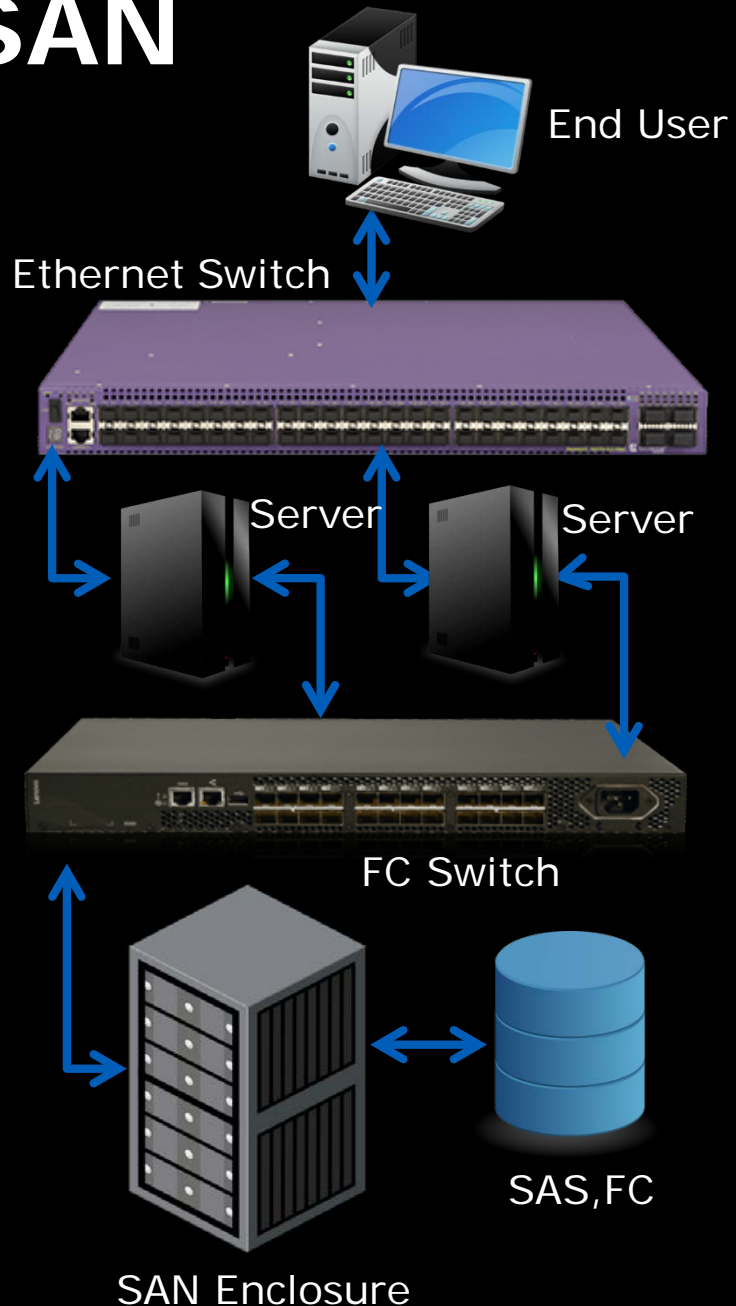
Additional key risks and uncertainties include the impact of continued uncertainty and volatility in global economic conditions; actions by competitors; business conditions; growth in our markets; and pricing trends and fluctuations in average selling prices. More information about the other risks and uncertainties that could affect our business are listed in our filings with the Securities and Exchange Commission (the "SEC") and available on the SEC's website at [www.sec.gov](http://www.sec.gov), including our most recently filed periodic report, to which your attention is directed. We do not undertake any obligation to publicly update or revise any forward-looking statement, whether as a result of new information, future developments or otherwise, except as otherwise required by law.



**We are on the precipice of a new data center  
paradigm**

**But first, some history to explain why it is  
happening into perspective...**

# SAN



Remote User

Ethernet Network

Application

File System

FC Network

iSCSI, FC, FCoE  
Network protocol

HA HW  
Protection

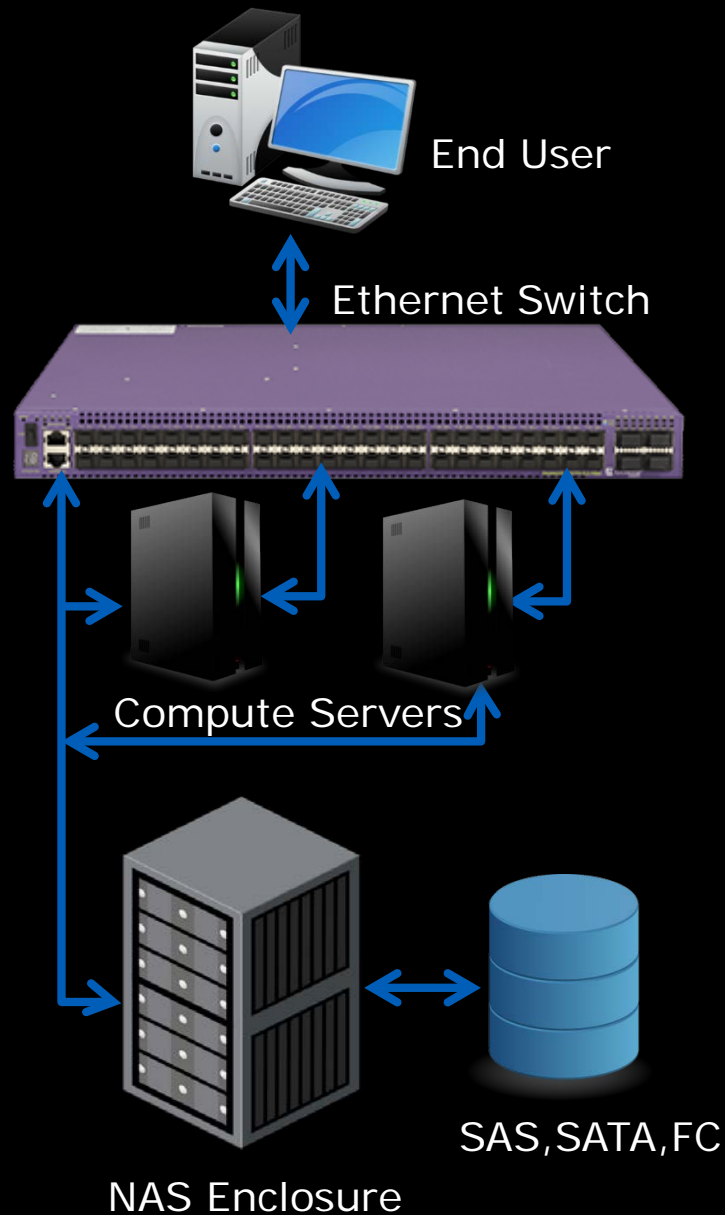
RAID

SAS, FC

SAN allows for individual servers to access large, scalable capacity points of storage with good performance, **but..**

- Very Expensive
  - Large single point of failure requires inefficient bullet proofing for HA
- Uses proprietary HW, SW and Management
  - Forces vendor lock-in
- Difficult to administer
  - Usually requires a specialty admin. team
- Scalability is complex
- Requires Architectural changes to work
  - Requires a separate storage network (Fibre Channel)
- Complex HW with no RDMA capability brings medium latency

# NAS



Remote User

Ethernet Network

Application

Application File System

NFS, CIFS, SMB  
(NAS File System)

HA HW  
Protection

RAID

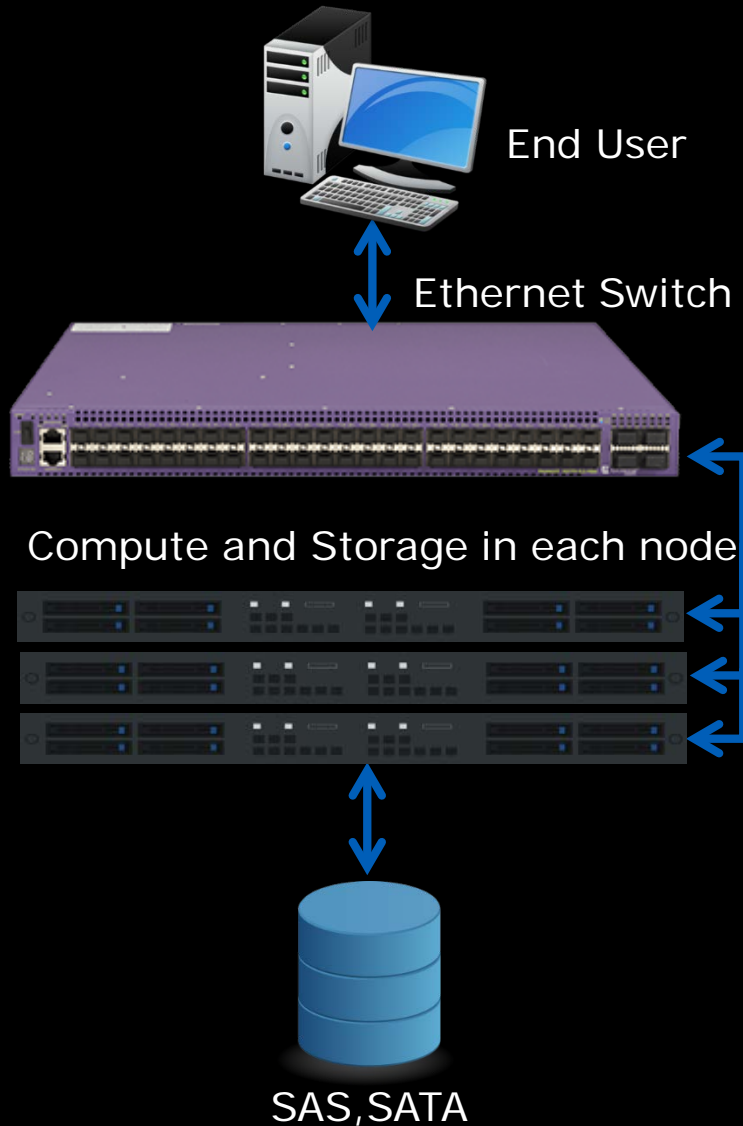
SAS, SATA, FC

NAS allows for individual servers to access large, scalable capacity points of storage with good performance, **but..**

- More expensive than DAS
- Scalability issues
  - Growing existing workloads shouldn't be strapped across multiple disparate HW instances
- File System solution only (tied to Linux only)
  - NAS failure leads to difficulty recovering the data
- Proprietary management
- Tool sets less robust than SAN



# DAS



Remote User

Ethernet Network

Application

File System

HA HW  
Protection

RAID

SAS, SATA

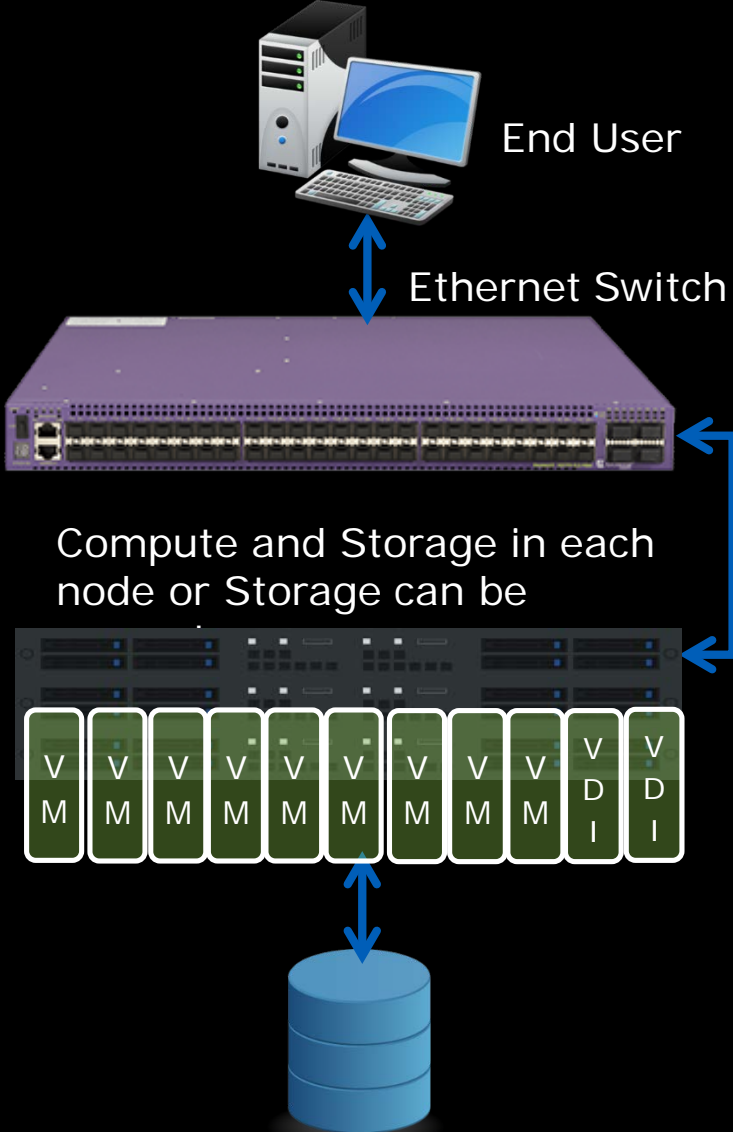
This is the simplest approach architecturally, and also the least expensive on the surface, **but..**

- In reality most customers overprovision DAS solutions to prevent having to open the chassis later
  - 60-70% OP in traditional datacenters for mission critical applications\*
- Scalability issues
- Weak HA component (Compute + Storage must all be duplicated per server)
- Performance tied to limited storage device count
- Compute and storage resource ratios are locked to HW configuration
- No file sharing possible
- Upgrading HW is difficult and requires downtime or complex workload migration

\* Ericsson's "An Economic Study of the Hyperscale Data Center"

# Virtualized DAS

(with possibility for storage element from SAN)



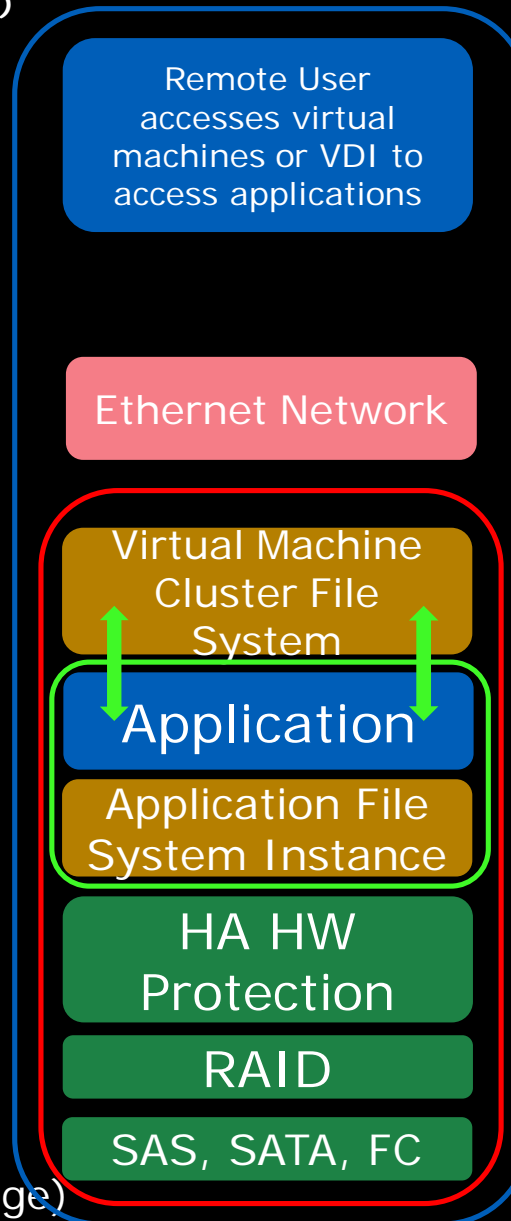
SAS, SATA, (FC if SAN attach for storage)

Western Digital

©2017 Western Digital Corporation or its affiliates. All rights reserved.

Best HW efficiency of the legacy methods **but..**

- Most Expensive HW solution
  - Large, expensive HW platform with multiple virtual machines residing inside this single compute environment
- Most expensive SW solution
  - Proprietary options that lock the user to a single solution
- Complex architecture and management
- Migration of workloads is necessary when cloudburst occurs and the VM no longer fits on existing HW
  - Network is inefficient



# How do we fix these things?

## Let's duplicate what the Hyperscalers did!!

Yes.. Attempts were made to “duplicate” the cost effective DAS storage revolutionized by the Hyperscalers. What did they do and how did it work out?



### SW Defined Storage (SDS) 1.0

- Hyperscalers wrote code around a common HW platform. In reality everyone else has endless combinations, exposing endless issues.
- The dream of software that can turn any hardware into a storage solution turned into a nightmare for generic users



# Let's try again!



## SDS 1.1 – Converged Infrastructure

- An attempt was made to resolve the issue of incompatible hardware by testing configurations; e.g., Vrail, etc.
- While tested configs sound good, it still requires validation at the customer.
  - This is still a case by case approach. Essentially a custom solution
  - Risk is inherent here
- Most customers don't have the capabilities or resources to carry out such tasks.
  - Why can't it just work?



# Back to where we started!

## SDS 1.5 – Hyperconverged Infrastructure

- **History repeats itself...**

To get around the endless combinations and endless test cycles, an SDS box includes compute, memory and storage, in a proprietary solution that's margin stacked. The very problems that SDS set out to resolve.

- Overprovisioning in the server to cover future growth
- Disparity between life cycles of different components locked in the box
- Locked in ratios of resources in the box
- Proprietary and expensive
- Poor Scaling





# Identify the problems, and actually fix them

## What pain points sent us down this SDS path?

# SDS 2.0! Let's learn from past lessons and get it right!

What's really needed to create a useful SDS solution?

- Pooled SDS that meshes well with pooled compute
- Pooled SDS with scalable access, and dynamic add or removal
- A standard protocol for accessing the pooled storage
  - a. No proprietary solutions for accessing data.
  - b. Multiple sources for the HW
- A very robust, reliable and low latency connection
- A single network for storage and compute
- The best of both worlds! Scalability of a SAN, with DAS like performance and cost from multiple sources!

# Well, what are the problems we face?

- **Datacenters are facing bottlenecks and unbalanced HW driven in part by the emergence of SSDs as general purpose storage and in part by rapid growth**
  - The capacity density disparity between SSDs and HDDs cause a need to change how we connect it all. More connect bandwidth per device and/or enclosure is needed to realize the full potential of the new Data Center.
  - The latency disparity between SSDs and HDDs require a need to change how we connect it all.
  - Rapid growth leaves little time to study the inefficiencies
- **DAS connections have been seen as the only way to take advantage of the SSD BW and latency, but that causes a whole new set of problems**
  - DAS solutions create a fundamental problem with EOL servicing of compute and storage. My CPUs can't keep up but my storage is still good, do I replace it all or tear it apart and salvage what I can? etc..
  - DAS solutions rarely provide efficient ratios of server versus compute and memory
  - DAS solutions have growth issues in mission critical environments as it is difficult to take the node down and add CPU, memory or storage



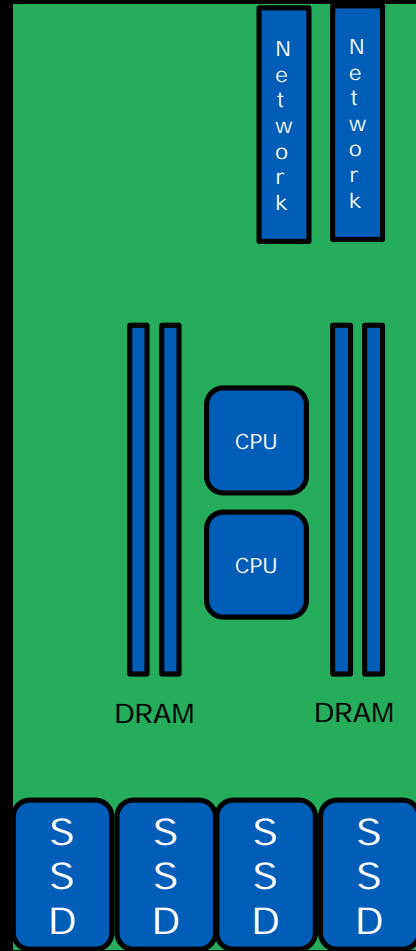
**In the DAS configurations, we  
see a need to dis-aggregate both  
Storage and Compute.**

**Here is why.....**



# Growing Pains in the Server

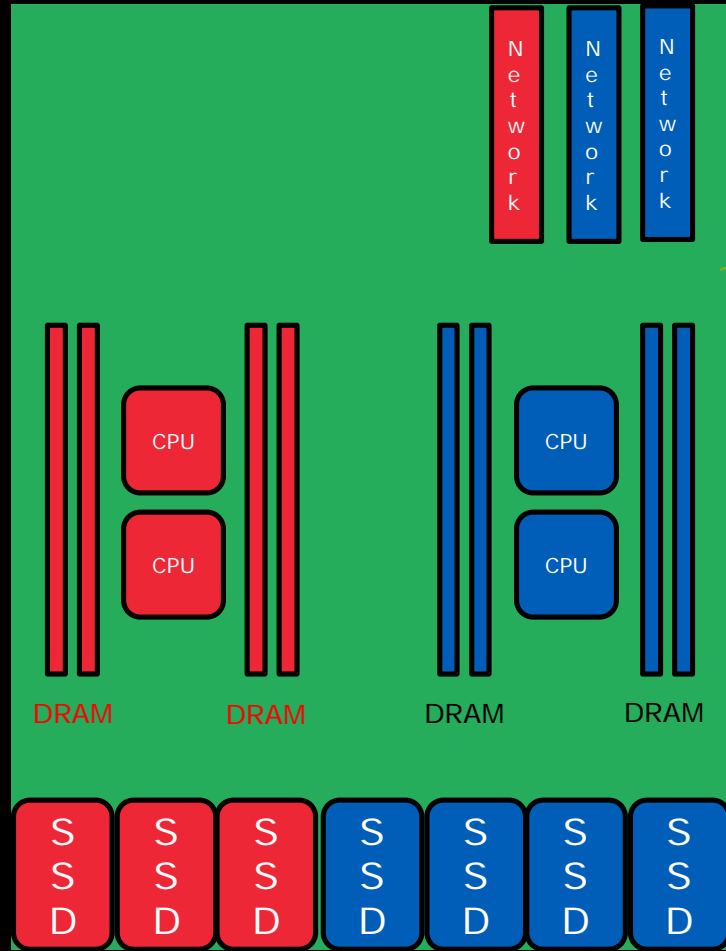
Mission Critical environments aren't meant to be opened



- CPU, Memory and other PCIe peripherals such as network never get hot-added and adding storage without compute memory to utilize it is futile
- The server must last 3 to 5 years and has a need to grow
- The user will overprovision the box to allow room for future growth.
  - Some studies have shown the OP to be as high as 60-70%

We need this

# Growing Pains in the Server



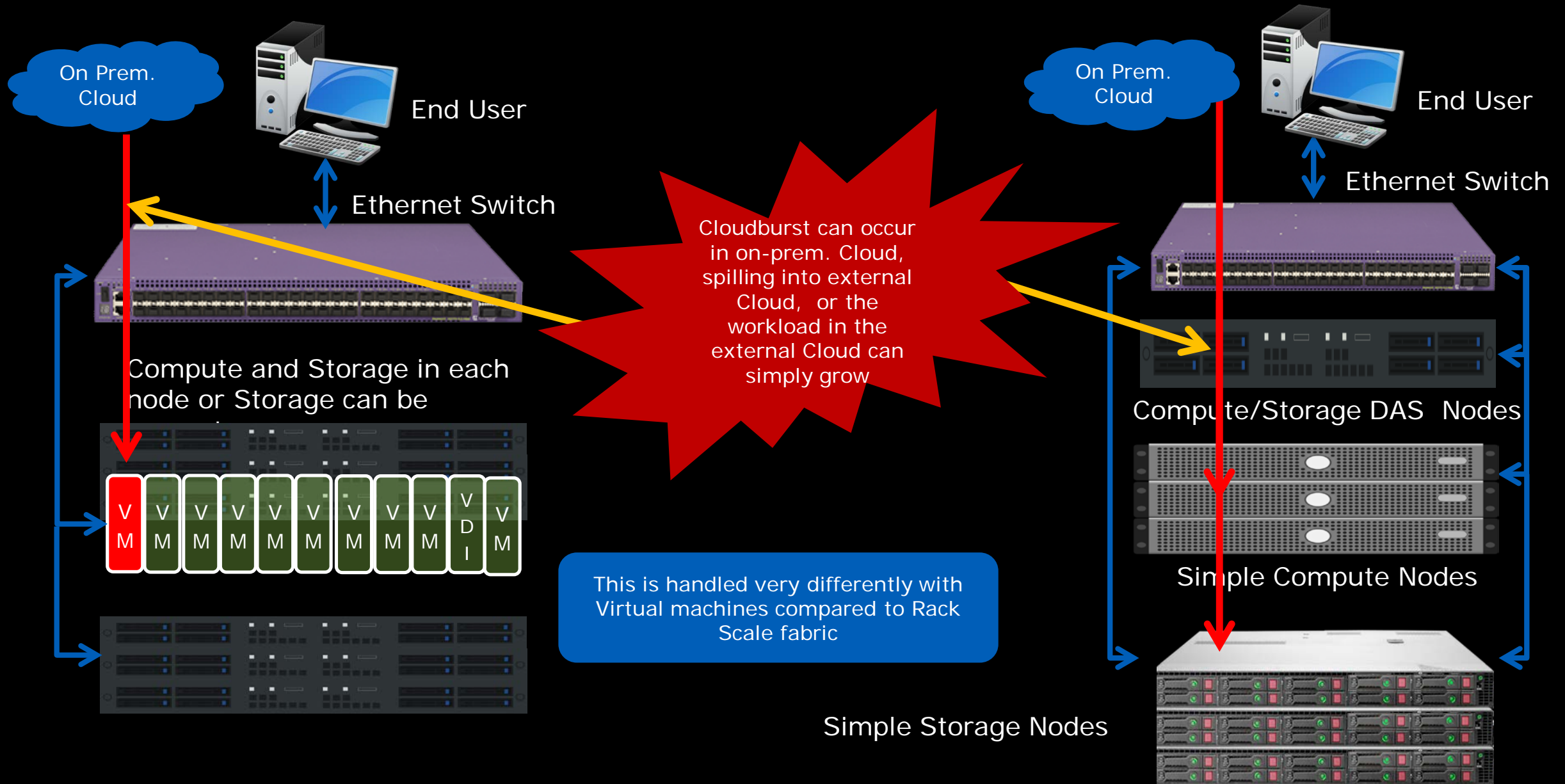
But end up building this to allow for growth

- We buy additional resources at today's prices for function we won't need till later
  - Even in idle mode they burn power and can fail with zero hours of use, or you can run everything at an inefficient partial load
- What we need is a way to add resources without opening individual chassis, but still keep our DAS performance.




**While we are at it, don't forget the growing  
pains in the virtualized world triggers  
costly migrations**

# Virtual Machines need to Migrate for growth





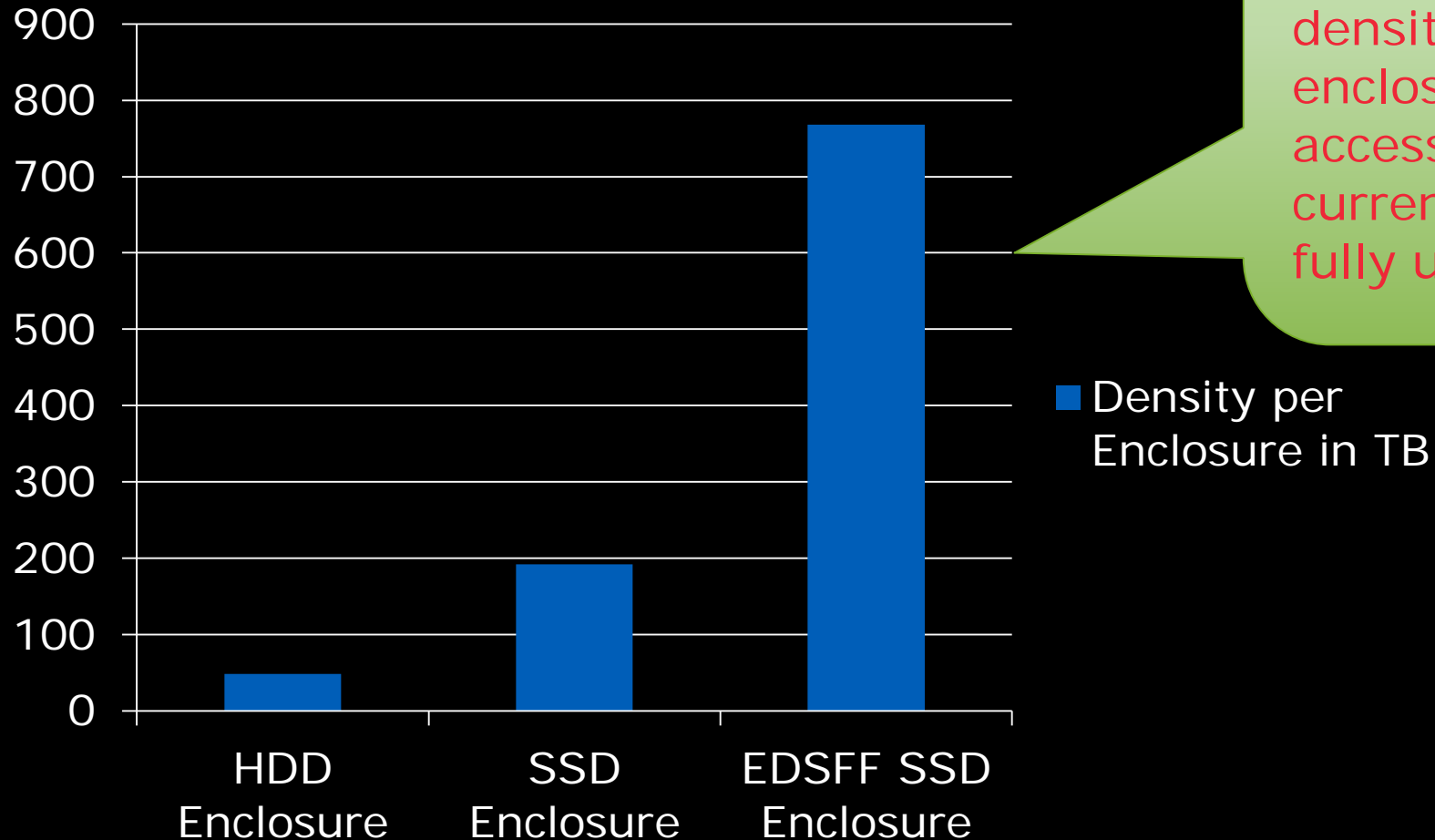


**DAS won't work, so we look to Fabric connect, but pooled resources on a legacy network has it's own challenges.**



# Denser storage enclosures mean device/enclosure BW must increase

Density per Enclosure in TB \*

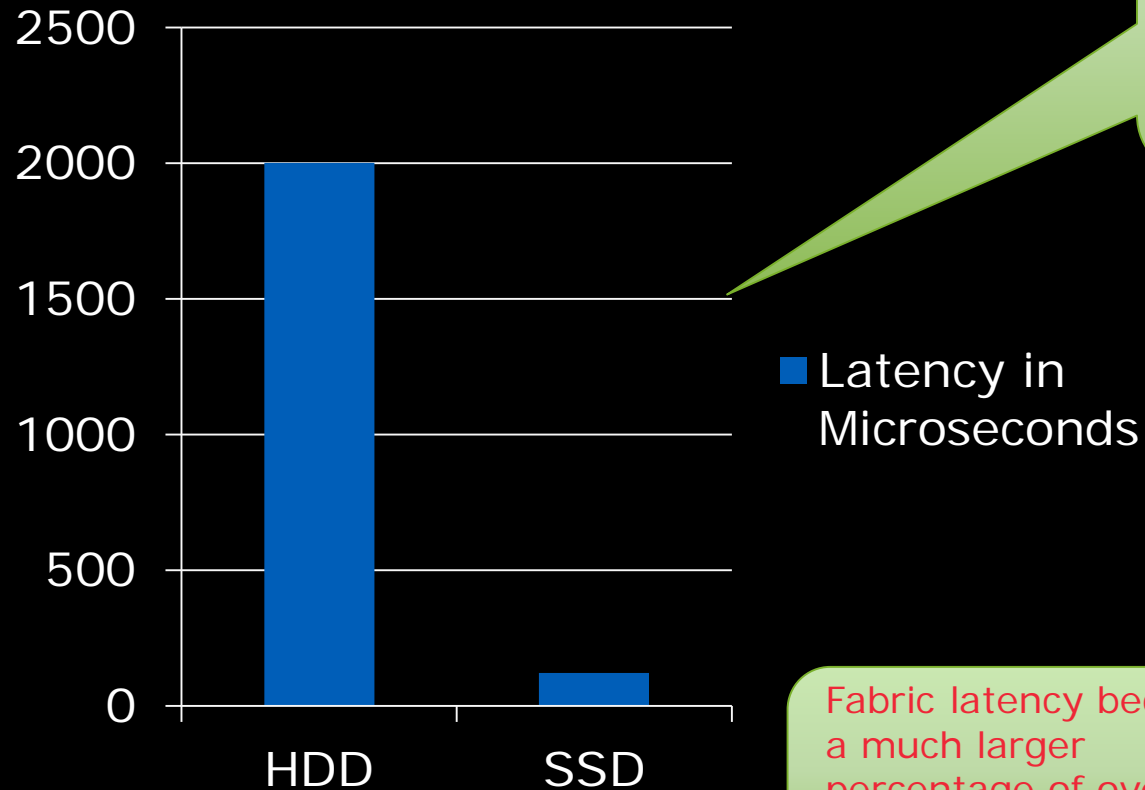


Denser SSDs mean denser capacity per enclosure. To service this density, we need more BW per enclosure just to keep performance access density at parity with current solutions, and even more to fully utilize the solution.

\*Assumes 2TB\*24 performance HDD, 8TB\*24 SSD, and 32TB EDSFF\*24

# Lower latency SSDs require lower latency network connections

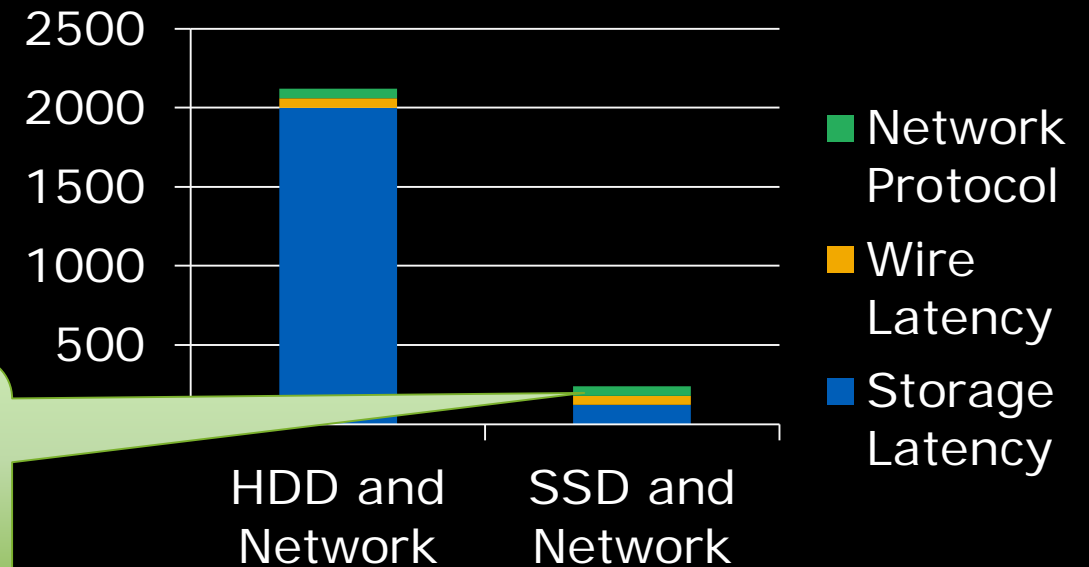
## \*Latency in Microseconds



To take advantage of SSD latency we need:

- More efficient data path
  - Why were we running SSDs on SAS?
- Lower latency data path

Fabric latency becomes a much larger percentage of overall latency as SSDs get faster



\*Assumes 2MS average latency for HDD and 120 microseconds average latency for SSDs. Assuming 60 microseconds for the fabric wire and protocol

OK, so SAN is too expensive and proprietary, DAS is too rigid in design and ratio and legacy networks aren't fast enough...

Here is what we can do!

- We create a network with DAS like performance and latency.
- We add in RDMA to allow "Big Data" to be moved with no un-needed CPU pass thru to get to memory
- We provide fast, inexpensive 100Gb lanes with 200Gb and 400Gb lanes coming.
- Open SW to manage and dynamically configure it all
- Let's throw in some open source compute while we are at it

It's easy to say all of that but what actually needed to happen to realize this better way of running? Here is where we learned to do better!

# A Perfect Storm Enables an Electrifying Opportunity

Compute Centric Fabric Connect      Universal Management Methods

Universal connectivity

RDMA

Rack Level Thinking

Enter NVME

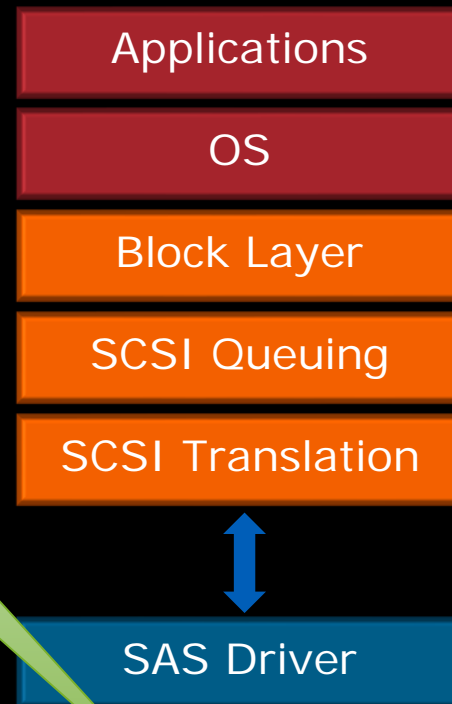
Multiple source solution that relieves  
price concerns

## New Scalable Datacenter

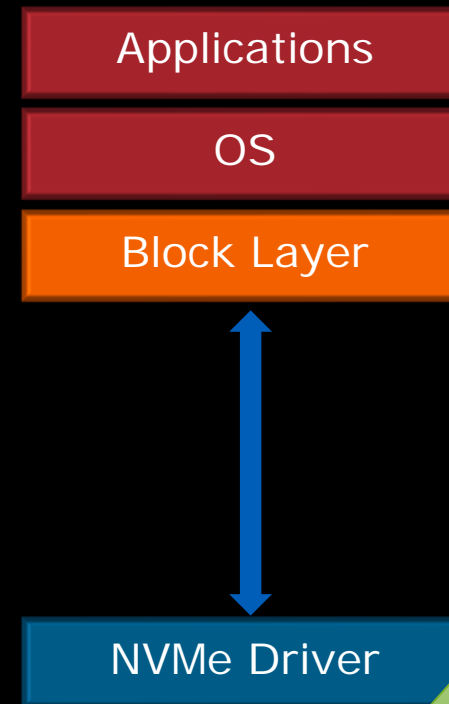
# First, fix the device protocol. We don't need SCSI layers in the way!

Increased Fabric BW comes with newer less expensive 100 Gb links..  
but what about fixing our inefficient data path? **NVMe™ saves the day here!!**

## Storage Stack



**SAS**



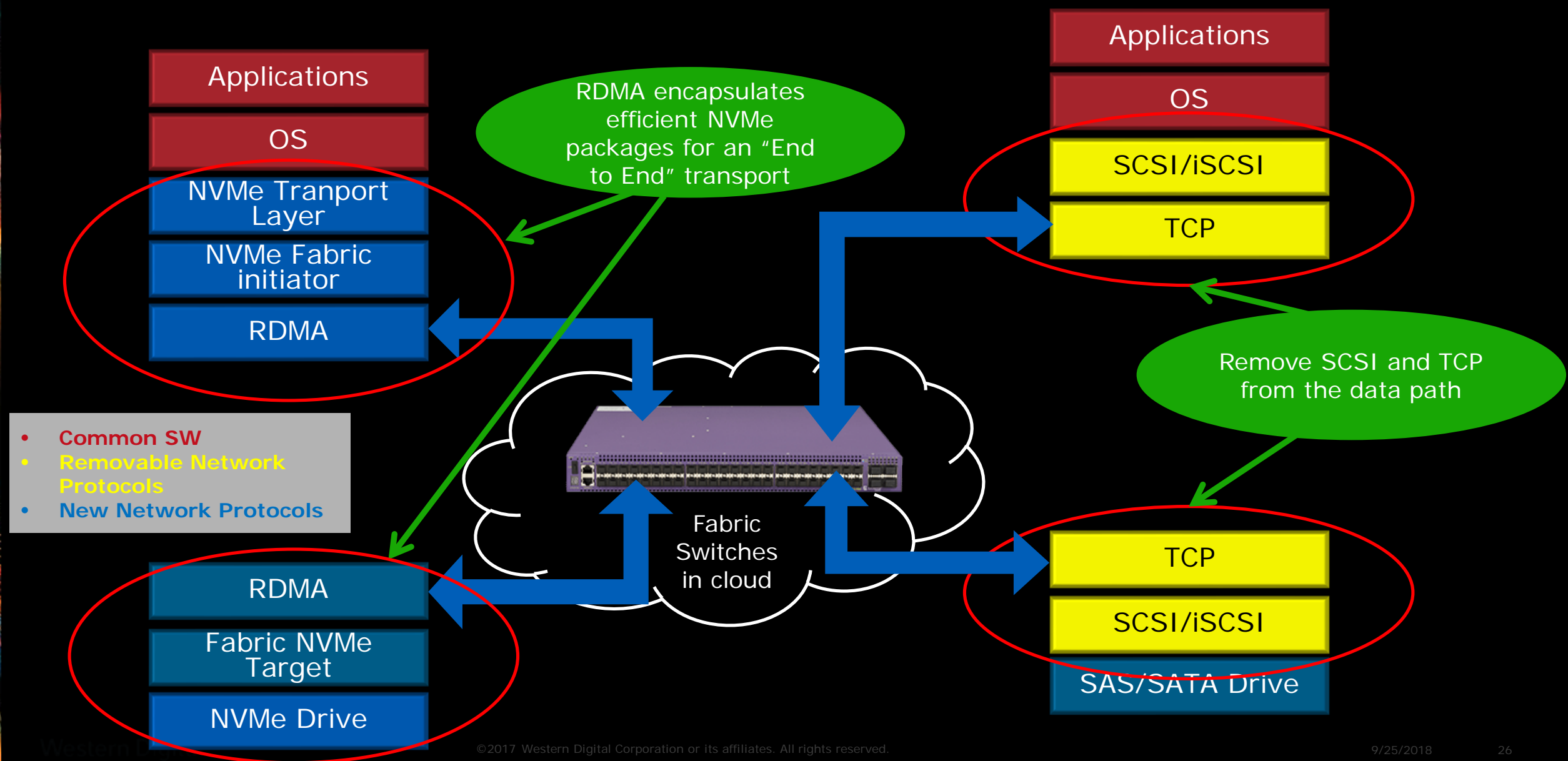
**NVMe™**

6 microsecond  
data path

Less than 3 microsecond  
data path

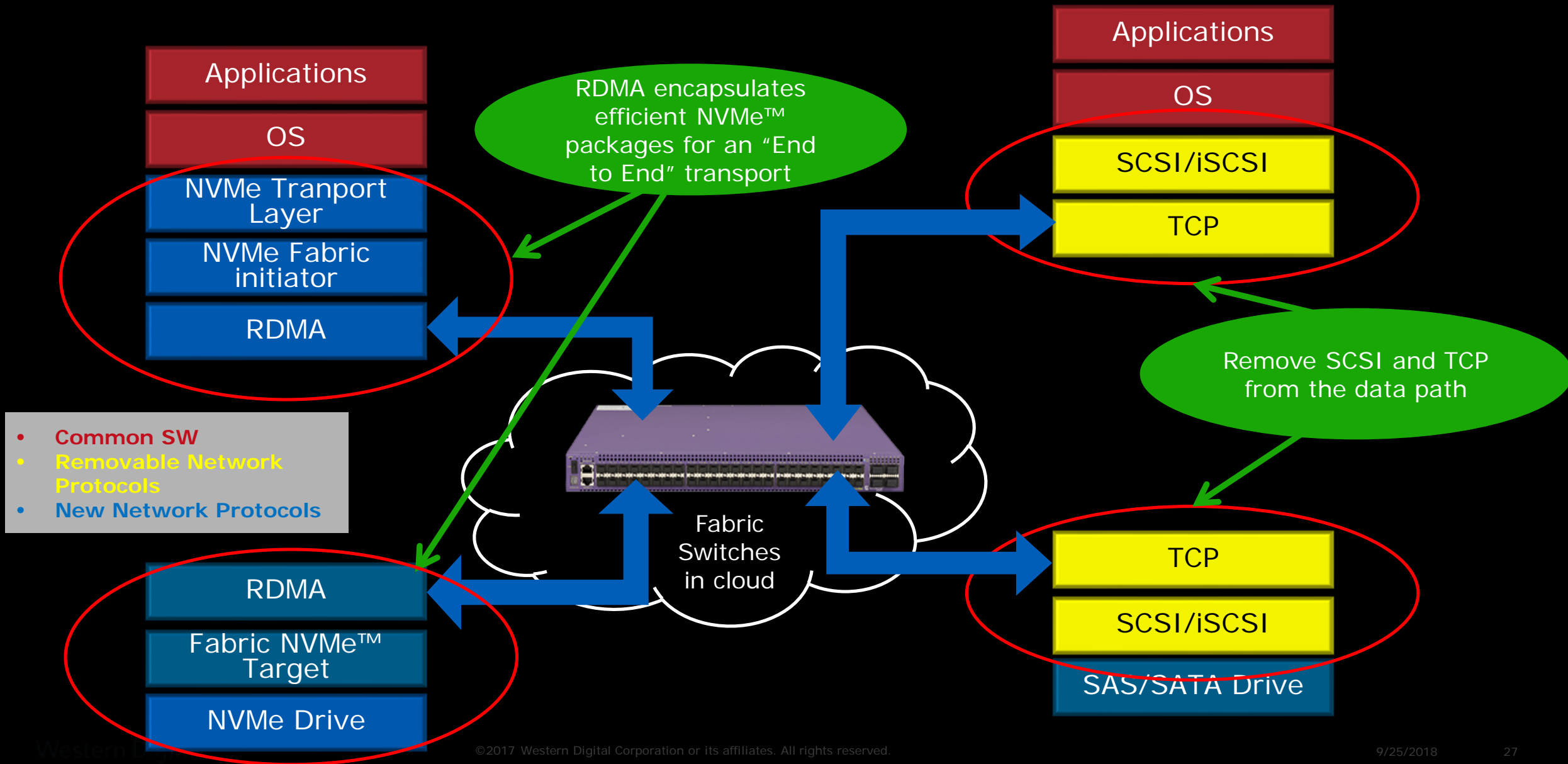
- Customer visible SW
- Transport modules
- Drivers

# Next, we simplified network path. Who needs SCSI or TCP in the way?





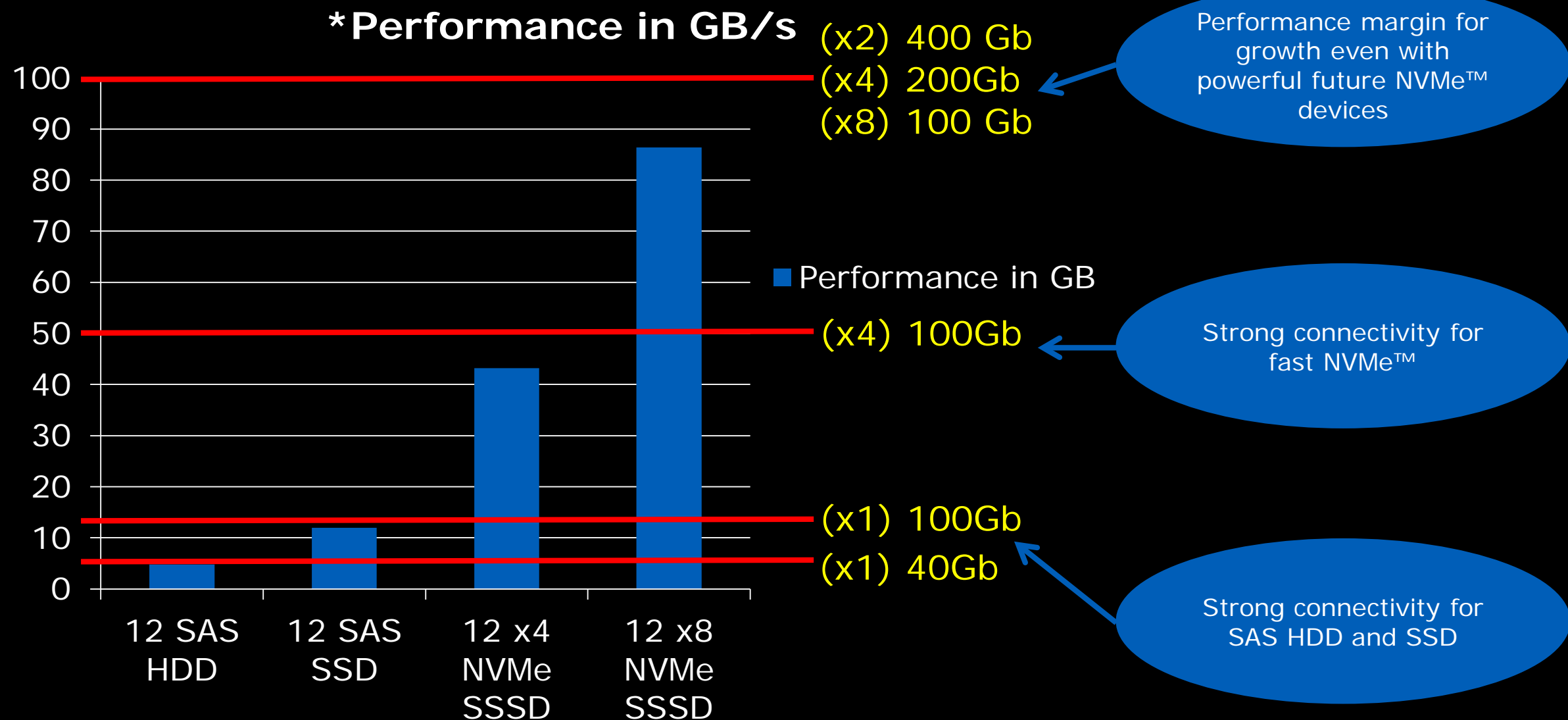
# While we are at it.. Who needs the CPU in the way if we are just moving data? RDMA makes it happen!





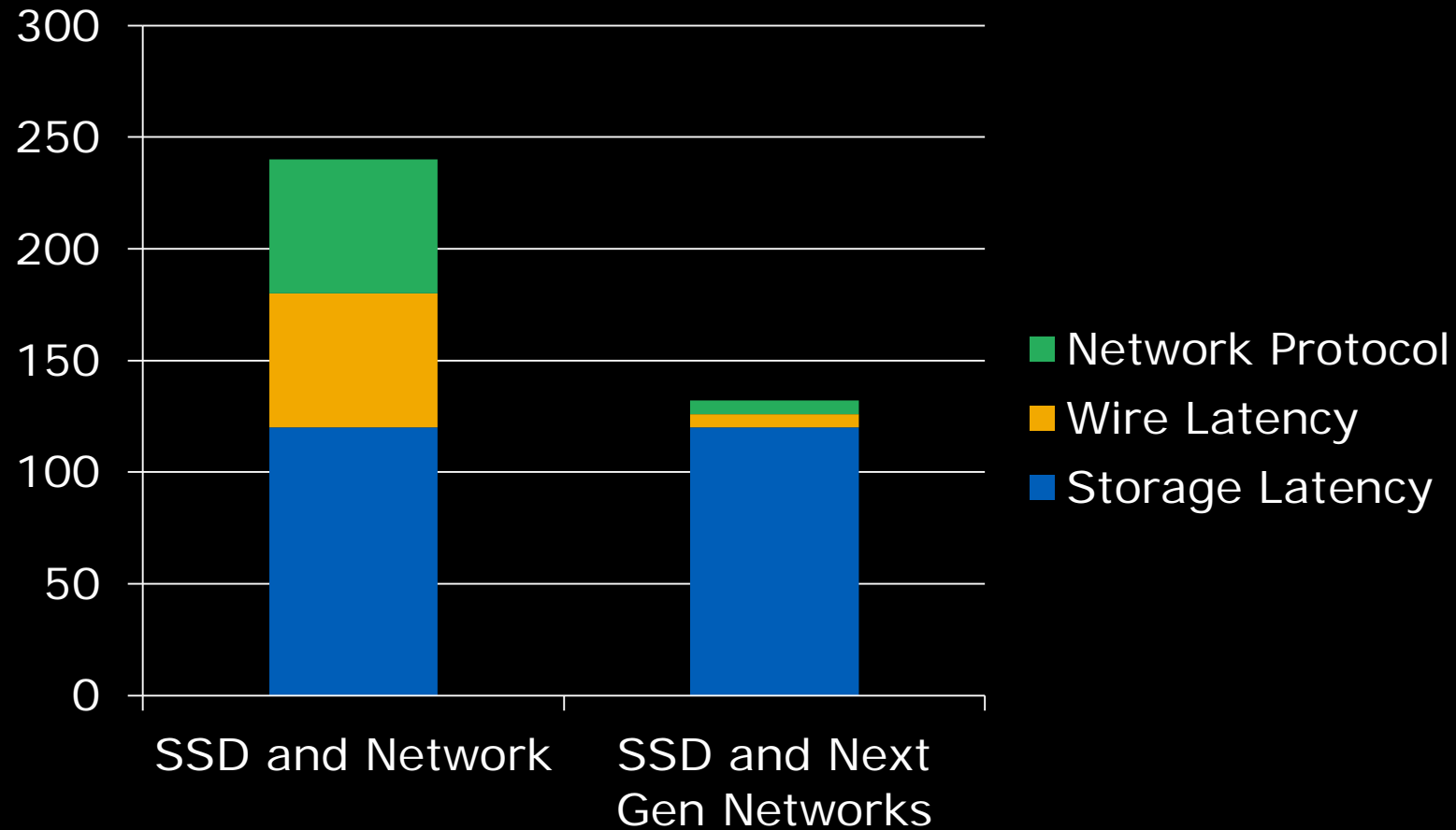
**Of course we need BW to handle current and future NVMe™ devices....**

# Fabric can connect both HDDs and SSDs for a common attach point



\*Assumes 400 MB/sec per HDD, 1 GB/s per 12 Gb SAS SSD, 3.6 GB/sec per x4 NVME SSD, 7.2 GB/sec per x8 NVME SSD

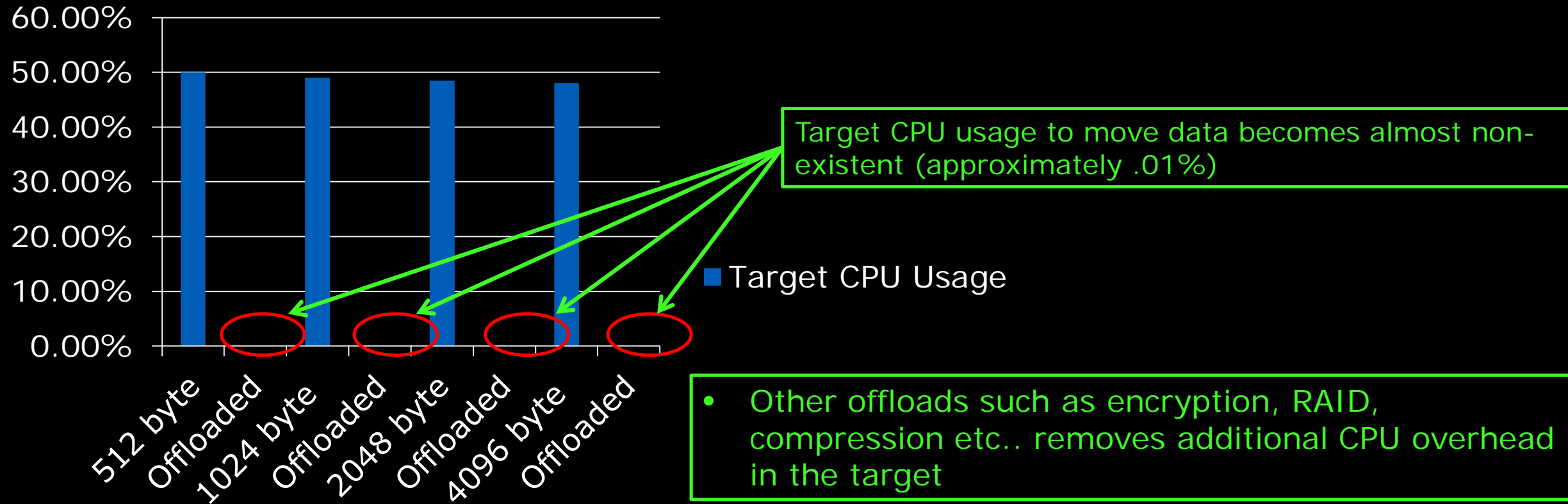
**\*Network and Storage latency are being improved as well....**



\*Assumes 120 microseconds latency storage, 1 microseconds latency protocol thru 3 hops to host and 3 hops back, 1 microsecond wire with 3 hops to host and 3 hops back

**\*Target CPU efficiency can be vastly improved as well....**

## CPU usage just to move data



\*512byte thru 4kbyte transfers consume approximately 50% Host CPU resources. Offloads with RDMA take this number down to approximately .01% From internal testing



**Now that we have efficient device and network protocols that allow DAS  
like performance lets talk Architecture!**

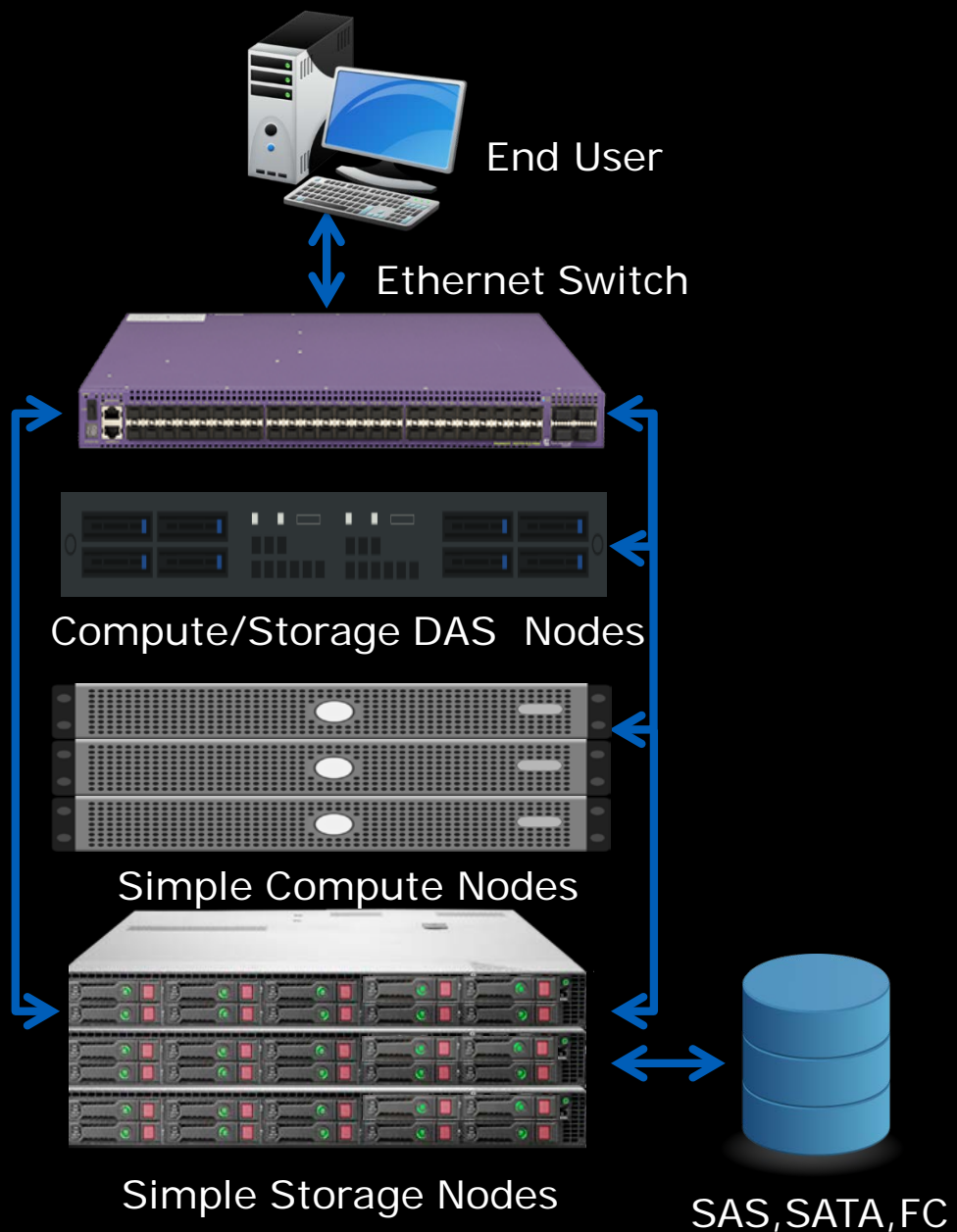


# Independent Scaling Demands Openness

*Rapid adoption of new open source technologies and standards*



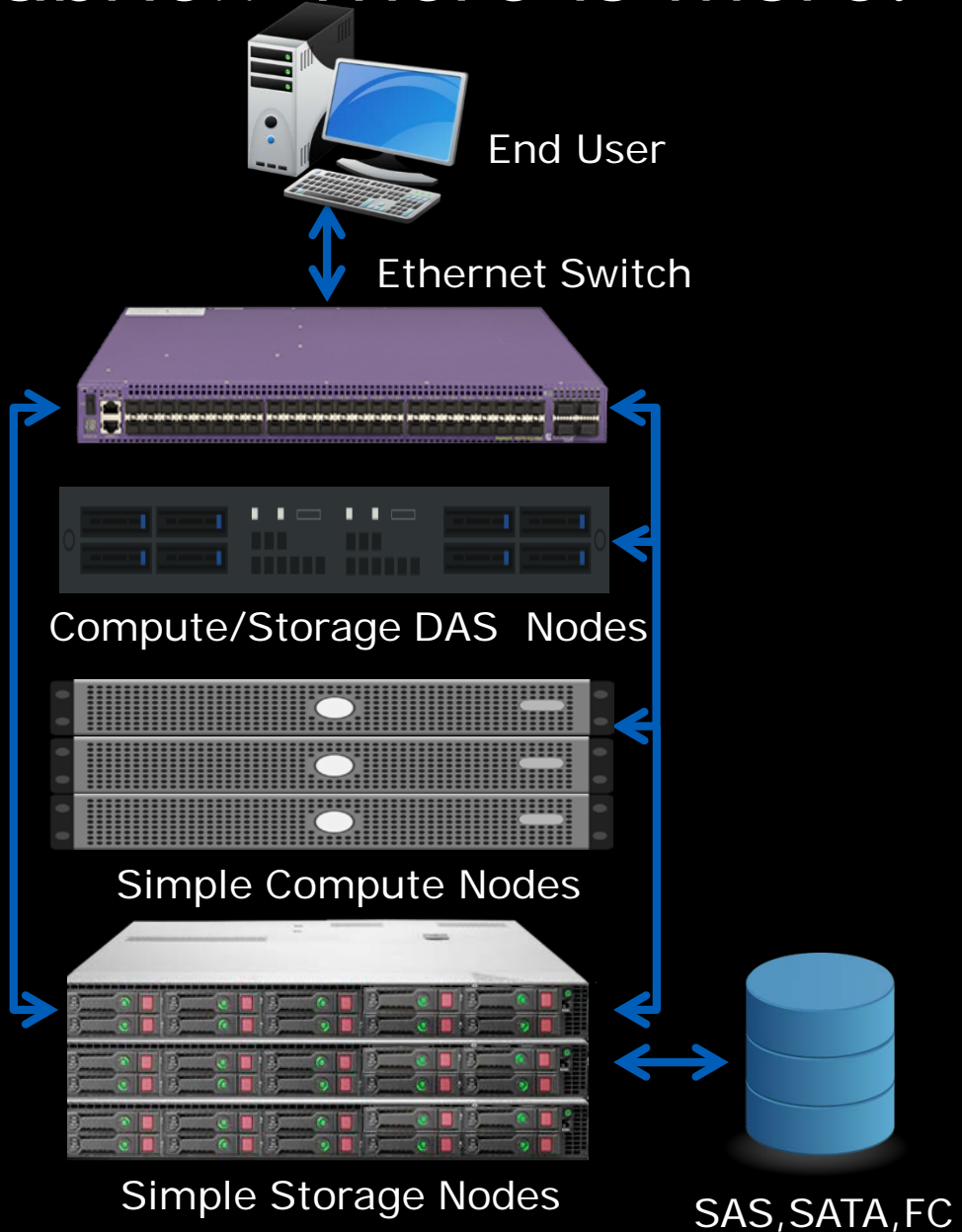
# Fabric..



## A New way to build your solution:

- Logically looks like DAS solution with none of the DAS resource ratio issues or overprovisioning burdens
- Efficient for small, medium and large configurations
  - Allows for seamless growth with no switchover points
- Inexpensive appliance style infrastructure
  - Redundancy is handled without the single point of HW failure without the need for heavy bulletproofing
  - Reliability, availability and serviceability handled in a clustered type of method
- Universal nature allows for no vendor lock in
  - Universal Commodity Hardware
  - Universal Ethernet connectivity (no dual networks)
- Scalability is simple with modular design approach
  - No need to test huge configurations with POC as each fabric attached module is a standalone entity in a bigger solution
- Low latency and highest bandwidth at lowest cost
- Provisioning is simpler than SAN
  - Dynamic provisioning designed for cloud use cases

# Fabric.. There is more!



## A New way to build your solution:

- Administration handled by the existing network admin team and no need for specialty Admins
- In 2018 most new x86 servers will have RoCE support built in so ability to build fabric solutions will be default in every server
- For those that want to continue using existing virtualized SW programs they can
  - It all still works, just on less expensive, universal HW
- New configuration solutions designed for fabric are available for those that want more functionality than open source solutions provide, and are still less expensive than existing legacy solutions
- All nodes can talk to all other nodes (both compute and storage)
  - Provides ability for compute in storage nodes
  - Provides ability for peer to peer storage device communication

# We can get the best of a DAS configuration along with the capacity and accessibility of SAN without sacrificing

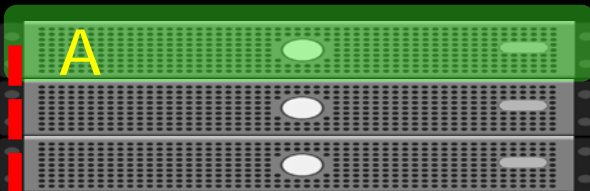
Fabric



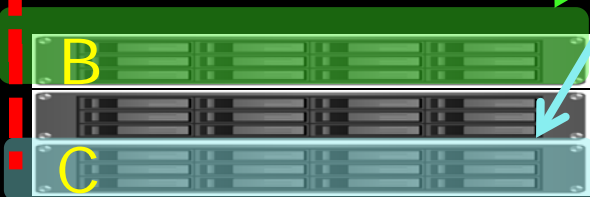
Ethernet Switch



Compute/Storage DAS Nodes



Simple Compute Nodes

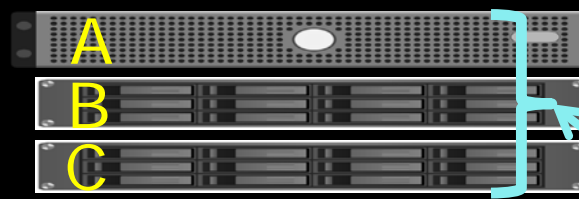


Simple Storage Nodes

SAS, SATA, FC



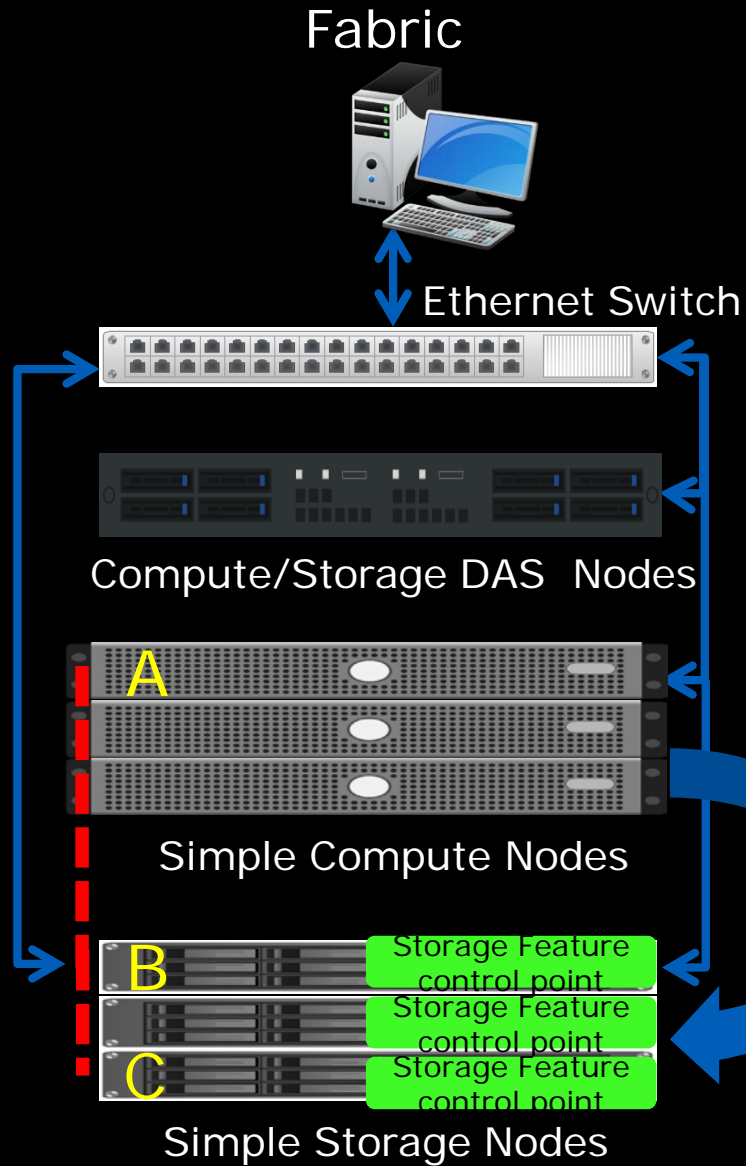
- Compute and storage nodes can be configured together as a server.
- If more capacity is needed we simply add another storage element via SW
- DAS type of performance and configuration without the resource ratio issues and overprovisioning costs
- SAN type capacity points without the cost and complexity
- Nodes become less tightly coupled and become universal building blocks that couple together with nearly the ease of attaching a laptop to WiFi



Logical server configured from units A and B

Additional capacity is needed so unit C is added into the configuration with no need to physically open any HW configuration

# Storage nodes can provide global storage features while still remaining as a universal multisource attach point



- Storage enclosures allow global storage features
  - Multi drive data traffic
    - Avoid read/write collisions
  - Endurance pools
  - Enclosure RAID
- Database Queries and other data-centric “data scrubbing” actions such as Genomics analyzing can be handled locally, removing fabric BW burden



**I mention a new control point for compute closer to the media. Let's expand on that a bit.**



# Diverse and Connected Data Types

*Tight coupling between Big Data and Fast Data*

## Big Data

Insight



Prediction



Prescription



Scale

## Fast Data

Mobility



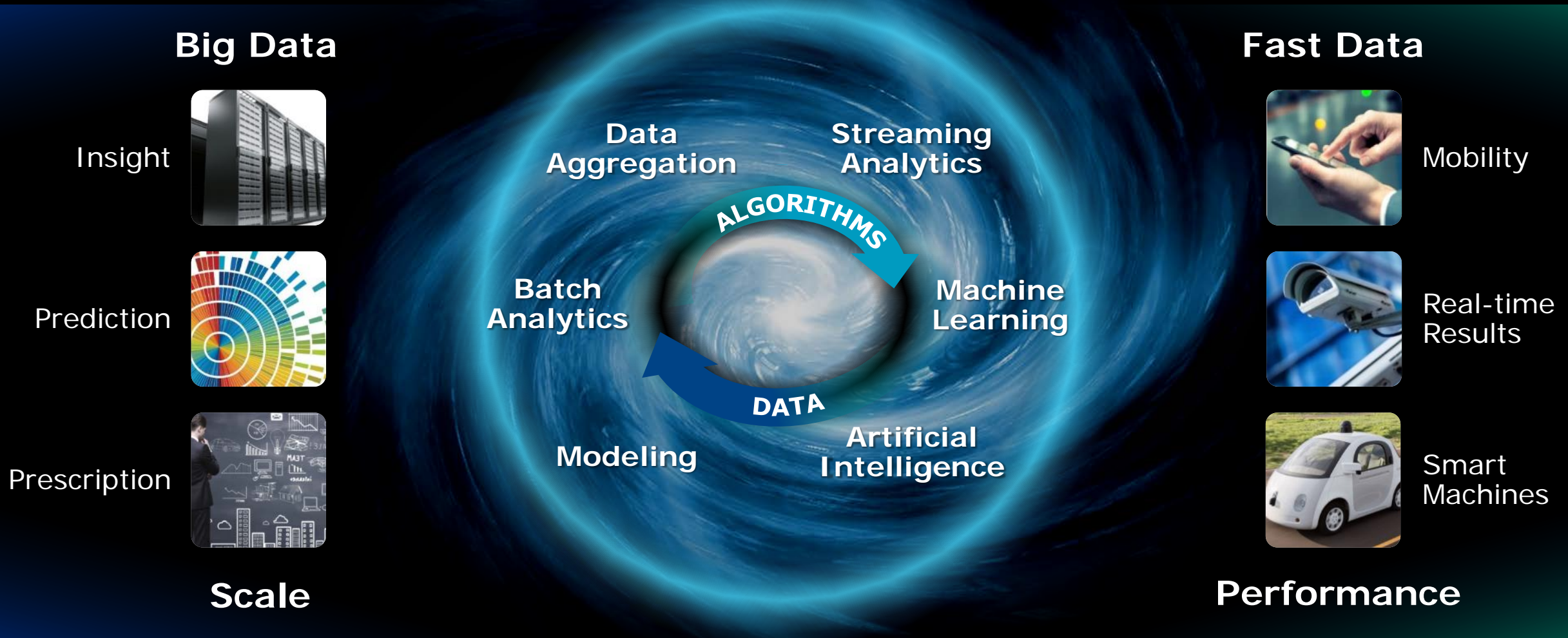
Real-time Results



Smart Machines



Performance



# From General Purpose to Purpose Built

*Architectures designed for Big Data, Fast Data applications*

**Big  
Data**

Expanding applications and workloads

**Fast  
Data**

General purpose  
compute-centric architecture

Solutions

Systems

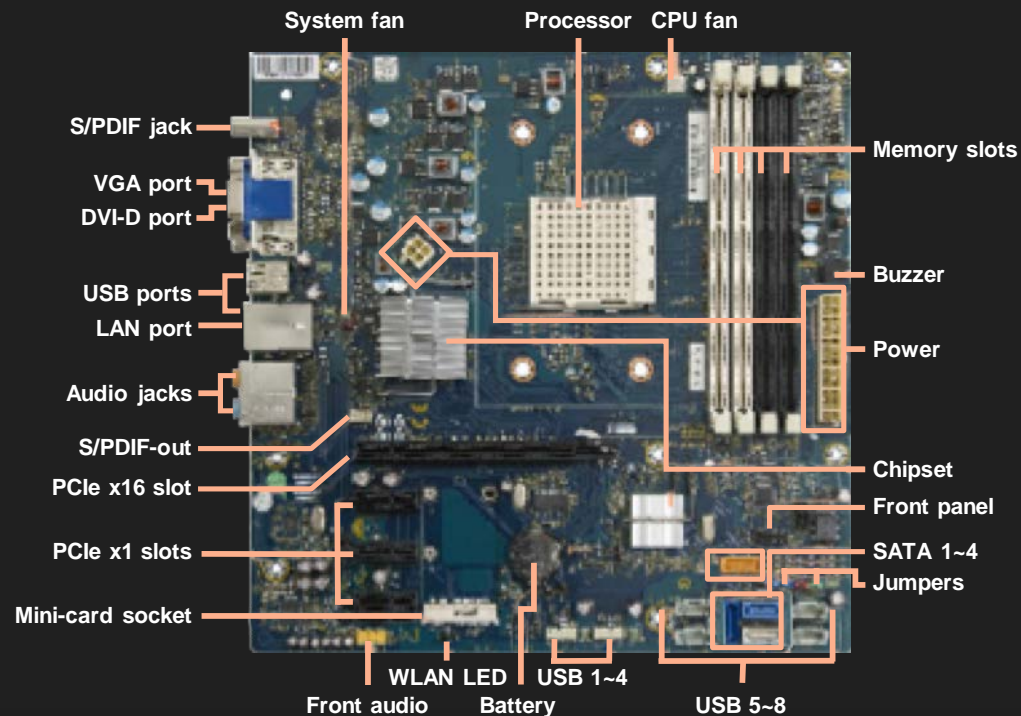
Platforms

Devices

# General Purpose Architectures No Longer Sufficient

*Big Data and Fast Data workloads exceed capability of uniform resource ratios*

## General Purpose Compute Architecture



- Predetermined ratios of:
  - OS/App Processor
  - Specialty Processor
  - Memory
  - Storage
  - Interconnect
- Overhead of “PC” logic
- CPU-centric



# Data-Centric Environments

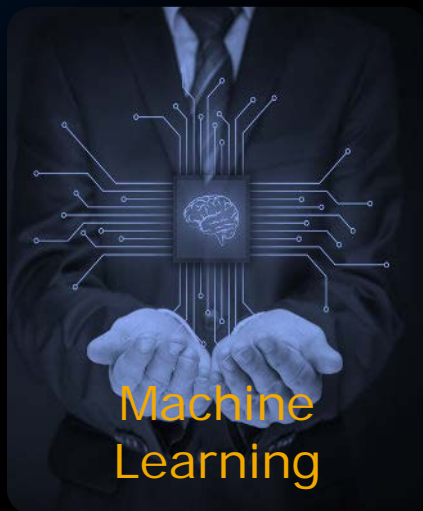
*Big Data and Fast Data workloads need independent scaling of resources*

## Big Data



Analytics

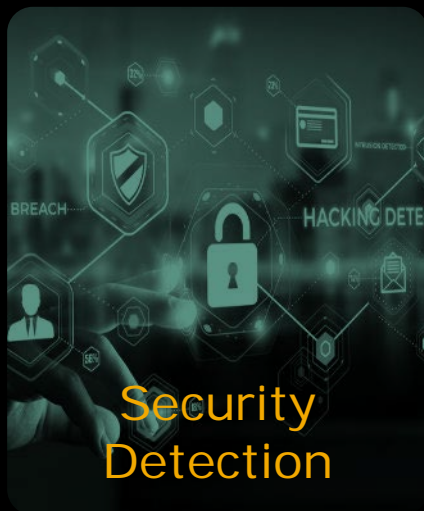
- Massive Storage
- Moderate Processing



Machine Learning

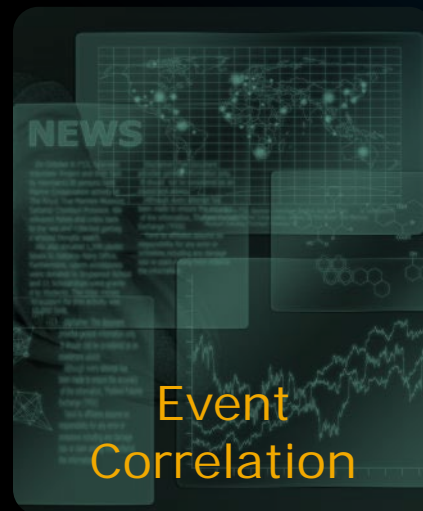
- Massive Storage
- Massive Specialty Processing

## Fast Data



Security Detection

- Large Memory
- Specialty Processing



Event Correlation

- High-bandwidth interconnect
- Large Memory and Specialty Processing



Blockchain

- High-bandwidth interconnect
- Large Specialty Processing

# RISC-V Meets the Needs of Big Data and Fast Data

*Provides a foundation for purpose-built, data-centric compute environments*

## Big Data

### Move Compute to Data

- CPU for device, platform, system
- Minimize data movement
- Offload workload to “smart” storage
- Localized machine learning

## Fast Data

### Memory Centric Compute

- Highly scalable main memory
- Minimize data movement
- Heterogeneous processor support
- Scalable accelerators/offload engines



- Open and free
- Enables modular chip designs
- From 16 to 128-bit

- Scales from embedded to enterprise
- Direct integration with specialty accelerators
- Extensible ISA (for special purpose functions)

# RISC-V Meets the Needs of Big Data and Fast Data



# Summary

- Fabric friendly protocols and methodologies are ready and going into next generation solutions as we speak
  - Allows for inexpensive multi-source compute and storage elements on a universally connected fabric with open source configuration/management
- TCA considerations on design need to take into account the SW savings with open source code now currently possible
- Customers needing to use existing Virtualized SW can still do so, just with less expensive HW
- Large storage capacity points have created a need for performing data centric compute near the actual storage, which we can now service as an offload
- Composable resource bricks of compute and storage can allow for granularity and dynamic growth for the most efficient solution.



An abstract graphic on the left side of the slide, consisting of numerous thin, flowing lines in shades of red, orange, yellow, and blue. These lines are layered and curved, creating a sense of movement and depth against the black background.

# Thank You!

Western Digital and the Western Digital logo are registered trademarks or trademarks of Western Digital Corporation or its affiliates in the US and/or other countries. Hadoop®, and the yellow elephant logo are either registered trademarks or trademarks of the Apache Software Foundation in the United States and/or other countries. The Ceph logo is a registered trademarks of Red Hat, Inc. in the U.S. and other countries. RISC-V is a trademark of the RISC-V Foundation. TensorFlow, the TensorFlow logo and any related marks are trademarks of Google Inc. All other marks are the property of their respective owners.