SNIA | COMPUTATIONAL STORAGE

What happens when Compute meets Storage?

Scott Shadley, Co-Chair, NGD Systems
Nick Adams, Co-Chair, Intel
David Slik, NetApp

July 2019

Agenda

SNIA COMPUTATIONAL STORAGE

- **Introduction to Computational Storage**
  - How long has this idea been around, Why Now?
  - How the TWG was formed
- **Computational Storage Working Group Focus**
  - Taxonomy is Key, need the right TLAs
  - Scope is Critical, Roadmap to success
- **Architectural Discussions and Future**
  - A look at some current solutions
  - A view of people’s thoughts of future solutions

Many Factors driving a Need for Computational Storage

SNIA™ COMPUTATIONAL STORAGE

Keys To Harnessing The Data Tsunami

Jonathan Salem Baskin Contributor ⓘ
Jun 13, 2016, 10:00am • 1,486 views • #BigData

AI Weekly: Computing power is shaping the future of AI

KHARI JOHNSON @KHARIJOHNSON MAY 18, 2018 7:14 PM

The Big Data Tsunami

Author: Matt Ferrari
Chief Technology Officer
ClearDATA

NEAR-DATA PROCESSING: INSIGHTS

Near-Data Computation: Looking Beyond Bandwidth

Published in: IEEE Micro (Volume: 34, Issue: 4, July-Aug. 2014)

the Analytical Scientist

Defying the Data Tsunami

Three motivating factors for using Edge Computing

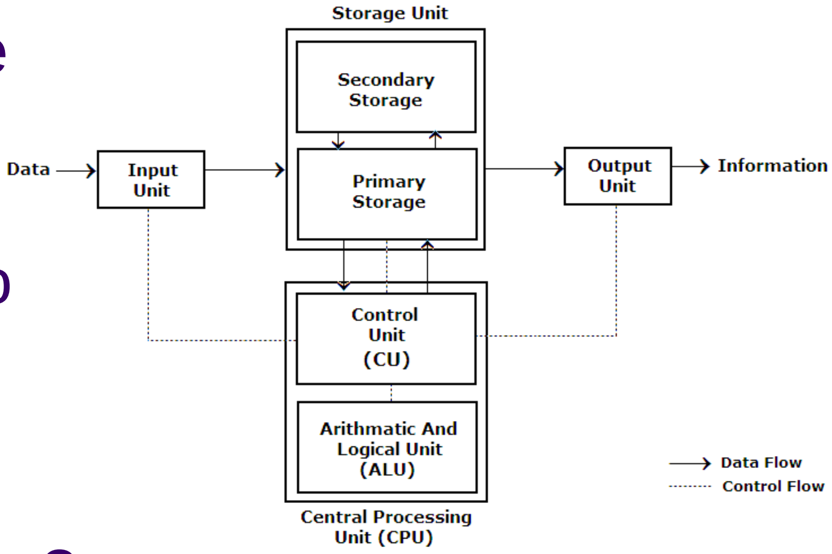
IBM

Internet of Things blog

1. Preserve privacy
2. Reduce latency
3. Be robust to connectivity issues

Compute, Meet Data

- Based on the premise that storage capacity is growing, but **storage architecture has remained mostly unchanged** dating back to pre-tape and floppy…
- How would you define changes to take advantage of Compute at Data?

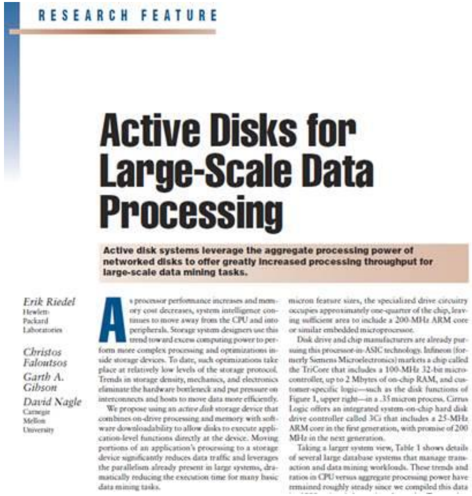


The Evolution of Computational Storage

- A delicate process to build an Ecosystem

- Great ideas! Time was needed to build it
  - Many technology papers exist around:
    - “Active Disks”, “CAFS”, “Near-Data”
    - “In-Storage”, “In-Situ”, “Near-Storage”

- So did some initial products!



RESEARCH FEATURE

Active Disks for Large-Scale Data Processing

Active disk systems leverage the aggregate processing power of networked disks to offer greatly increased processing throughput for large-scale data mining tasks.

Erik Riedel
Hewlett-Packard Laboratories

Christos Faloutsos
Garth A. Gibson
David Nagle
Carnegie Mellon University

As processor performance increases and memory cost decreases, system intelligence continues to move away from the CPU and into peripherals. Storage system designers use this trend toward excess computing power to perform more complex processing and optimizations inside storage devices. To date, such optimizations take place at relatively low levels of the storage protocol. Trends in storage density, mechanics, and electronics eliminate the hardware bottleneck and put pressure on interconnects and hosts to move data more efficiently.

We propose using an *active disk* storage device that combines on-drive processing and memory with software downloadability to allow disks to execute application-level functions directly at the device. Moving portions of an application's processing to a storage device significantly reduces data traffic and leverages the parallelism already present in large systems, dramatically reducing the execution time for many basic data mining tasks.

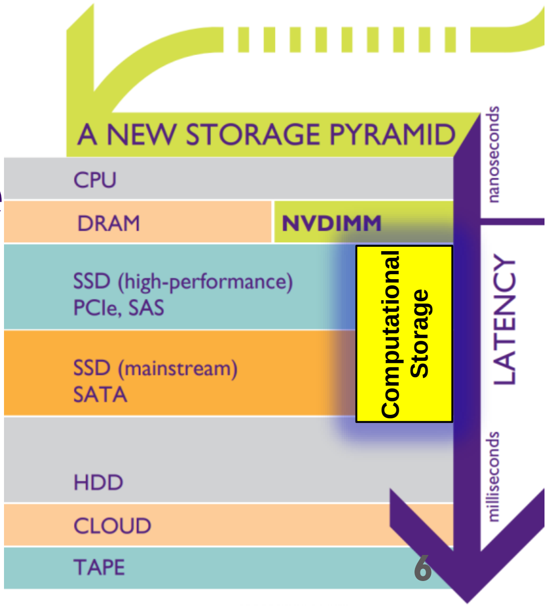
micron feature sizes, the specialized drive circuitry occupies approximately one-quarter of the chip, leaving sufficient area to include a 200-MHz ARM core or similar embedded microprocessor.

Disk drive and chip manufacturers are already pursuing this processor-in-ASIC technology. Infineon (formerly Siemens Microelectronics) markets a chip called the TriCore that includes a 100-MHz 32-bit microcontroller, up to 2 Mbytes of on-chip RAM, and customer-specific logic—such as the disk functions of Figure 1, upper right—in a .35 micron process. Cirrus Logic offers an integrated system-on-chip hard disk drive controller called 3Ci that includes a 25-MHz ARM core in the first generation, with promise of 200 MHz in the next generation.

Taking a larger system view, Table 1 shows details of several large database systems that manage transaction and data mining workloads. These trends and ratios in CPU versus aggregate processing power have remained roughly steady since we compiled this data

Playing Nice Together is Needed!

- Is this a solution replacing a solution?
- Complimentary work to the pyramid
- Another facet of advancement of compute
- In-Memory is needed, but some work can be offloaded all the way to storage!



So Now What?

The Progression of the TWG

40+ Participating Companies
148 Individual Members

SNIA COMPUTATIONAL STORAGE

Finding a Focus and Direction

- Initial focus on a definition list to ensure we covered questions on what it is and what products can be

- Drive to a Scope and path to universal usage model
  - Today we have custom… Tomorrow Standard… Sound Familiar?

Computational Storage

TWG Focus Areas

TWG Charter Overview

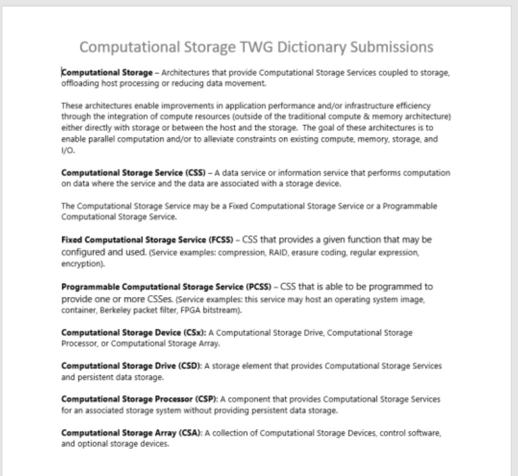
- Prioritize Industry Level Requirements
  - Collect and prioritize feature requests for Computational Storage Interfaces
- Develop Standard Interfaces & Protocols
  - Enable device vendors to supply Computational Storage features using extensions to existing standard interfaces and enable development of SW against those interfaces
- Align the Industry
  - Coordinate the submission of new standard proposals to accommodate the new features or create a new standard as a SNIA Architecture
- Facilitate and Drive SW Development
  - Work with relevant industry organizations to implement the feature's interface using the new version of the underlying standard that adds the feature.
- Educate
  - Promote Computational Storage paradigms through the industry at large

Starting the Standards Work

SNIA COMPUTATIONAL STORAGE

- Multiple F2F sessions have been focused on what we can accomplish and what we will leave for later

- Management
- Security
- Operation



Computational Storage TWG Dictionary Submissions

Computational Storage – Architectures that provide Computational Storage Services coupled to storage, offloading host processing or reducing data movement.

These architectures enable improvements in application performance and/or infrastructure efficiency through the integration of compute resources (outside of the traditional compute & memory architecture) either directly with storage or between the host and the storage. The goal of these architectures is to enable parallel computation and/or to alleviate constraints on existing compute, memory, storage, and I/O.

Computational Storage Service (CSS) – A data service or information service that performs computation on data where the service and the data are associated with a storage device.

The Computational Storage Service may be a Fixed Computational Storage Service or a Programmable Computational Storage Service.

Fixed Computational Storage Service (FCSS) – CSS that provides a given function that may be configured and used. (Service examples: compression, RAID, erasure coding, regular expression, encryption).

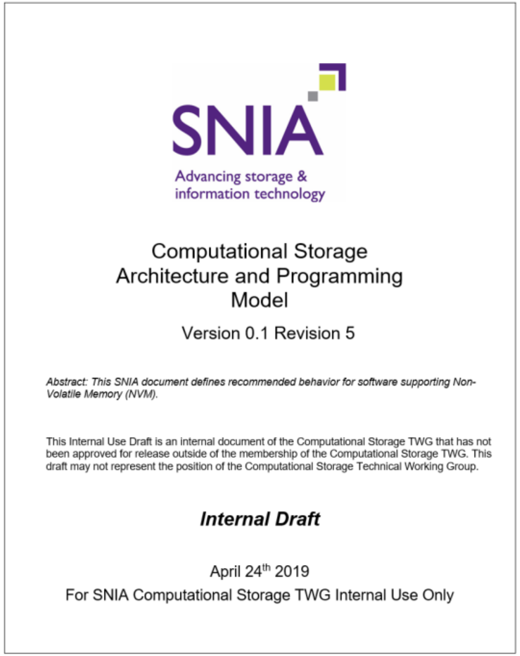
Programmable Computational Storage Service (PCSS) – CSS that is able to be programmed to provide one or more CSSes. (Service examples: this service may host an operating system image, container, Berkeley packet filter, FPGA bitstream).

Computational Storage Device (CSx): A Computational Storage Drive, Computational Storage Processor, or Computational Storage Array.

Computational Storage Drive (CSD): A storage element that provides Computational Storage Services and persistent data storage.

Computational Storage Processor (CSP): A component that provides Computational Storage Services for an associated storage system without providing persistent data storage.

Computational Storage Array (CSA): A collection of Computational Storage Devices, control software, and optional storage devices.



SNIA
Advancing storage & information technology

Computational Storage Architecture and Programming Model

Version 0.1 Revision 5

Abstract: This SNIA document defines recommended behavior for software supporting Non-Volatile Memory (NVM).

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

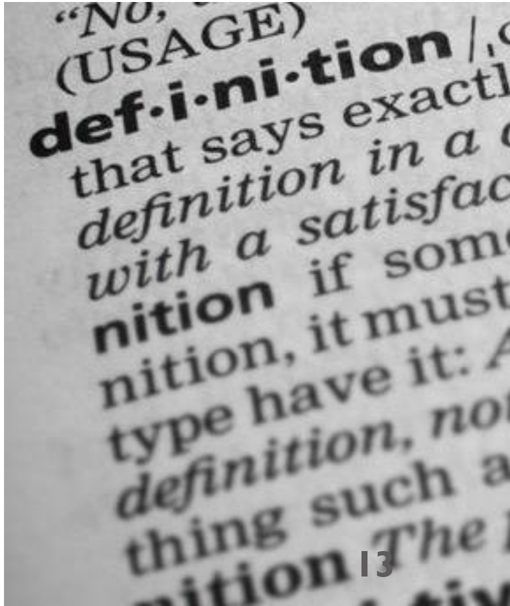
April 24th 2019
For SNIA Computational Storage TWG Internal Use Only

Speaking the Same Language

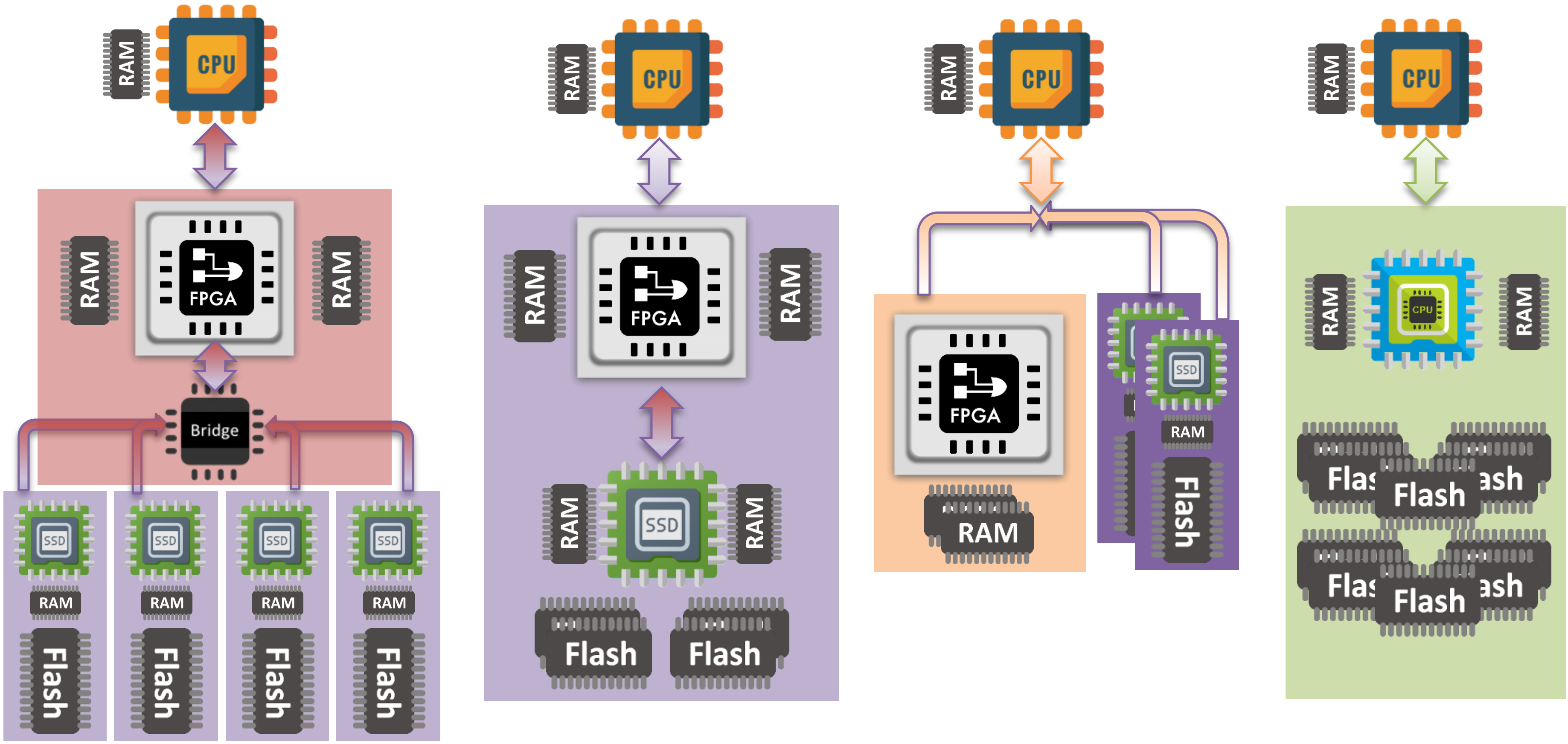
SNIA COMPUTATIONAL STORAGE

- Computational Storage:
    - Architectures that provide Computational Storage Services coupled to storage offloading host processing and/or reducing data movement.

- Two Foundational Constructs
    - Computational Storage Devices (CSx)
    - Computational Storage Services (CSS)

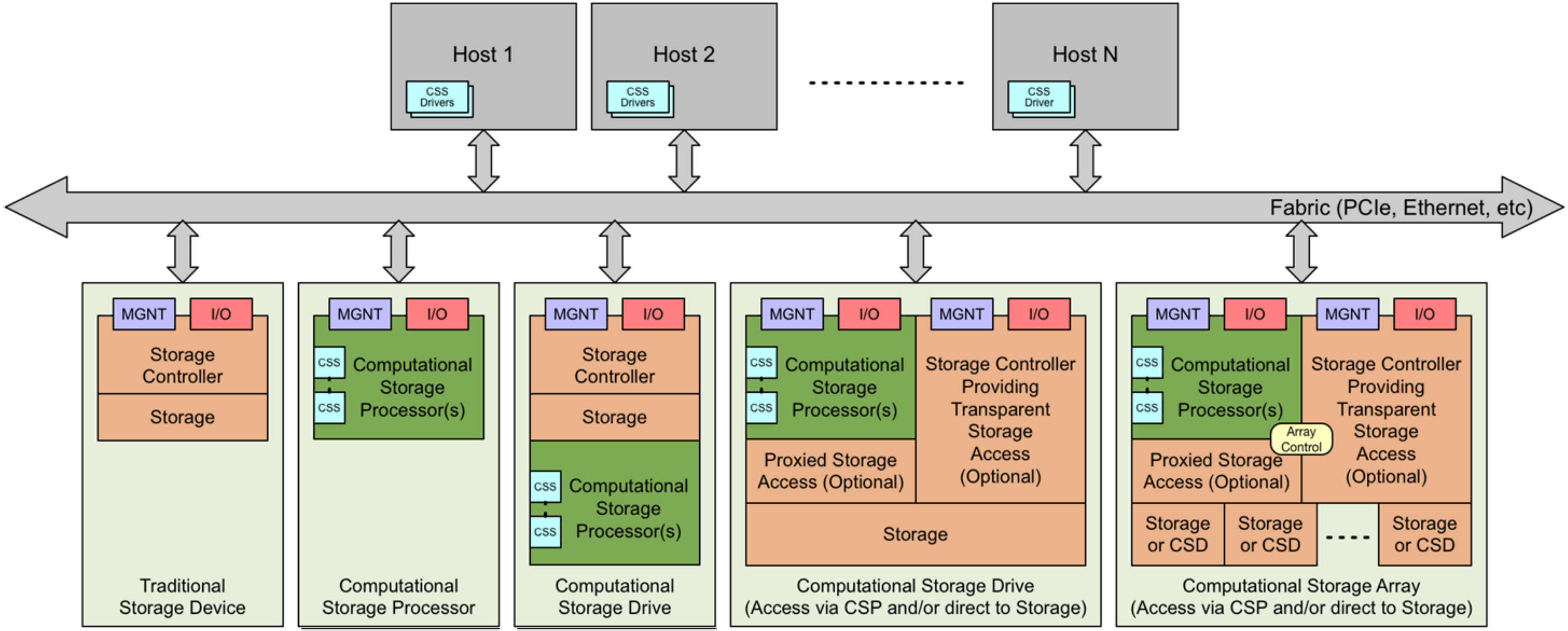


Current Instances of Computational Storage



Computational Storage Devices (CSx)

SNIA COMPUTATIONAL STORAGE



Computational Storage Services

SNIA COMPUTATIONAL STORAGE

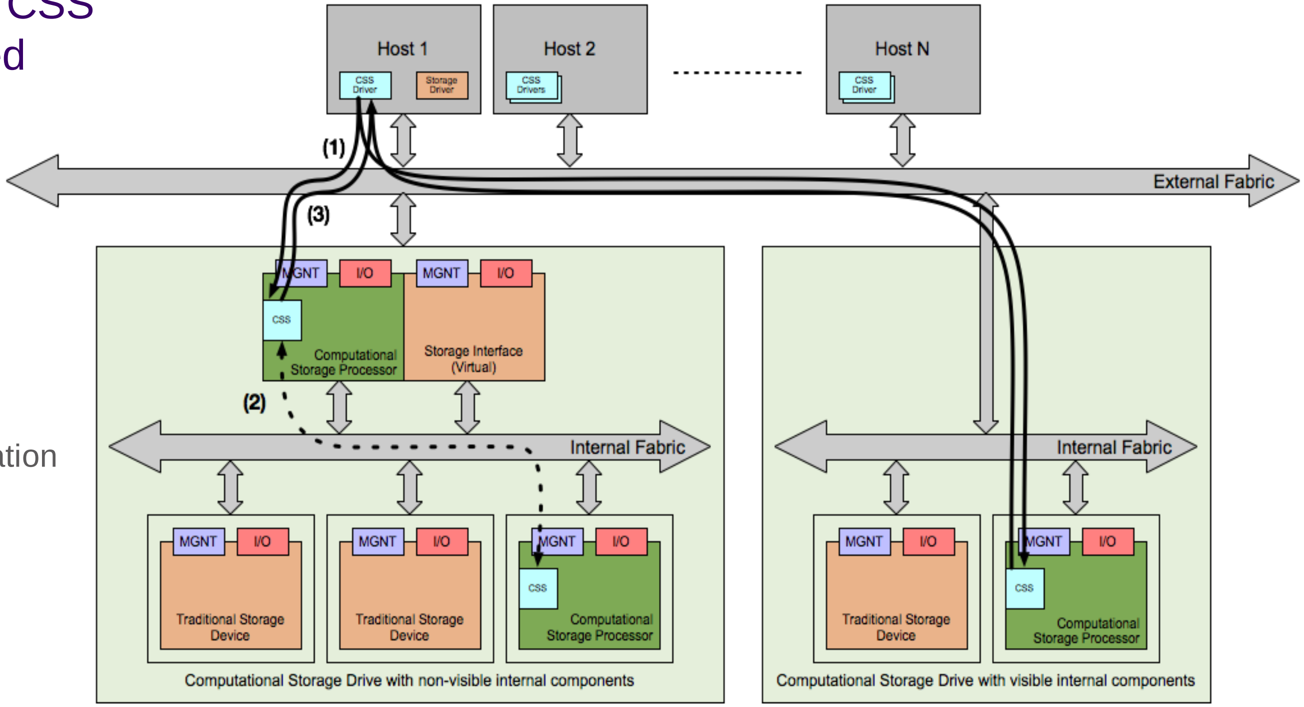
- Fixed Computational Storage Service (FCSS)
  - CSS that is well-defined
  - Consumable by the Host Agent for a well-defined purpose
  - Examples: Compression, RAID, Erasure Coding, or Encryption
- Programmable Computational Storage Service (PCSS)
  - Configured by the Host Agent to provide one or more CSSes
  - Examples: May host an Operating System image, Container, Berkeley Packet Filter, or FPGA Bitstream

Define the Scope & Prioritize

- **Management**
  - **Discovery.** Identify and determine the capabilities and functions.
  - **Configuration**. Parameters for initialization, operation, and/or resource allocation
  - **Monitoring**. Reporting mechanisms for events and status
- **Security**
  - **Authentication**. Host Agent to CSx and CSx to Host Agent.
  - **Authorization**. Mechanism for secure data access and permissions control.
  - **Encryption**. Mechanisms to perform computation on encrypted data.
  - **Auditing**. Mechanisms to generate and retrieve a secure log.
- **Operation**
  - Mechanisms for the CSx to store and retrieve data.
  - Host Agent interaction may be explicit or transparent.

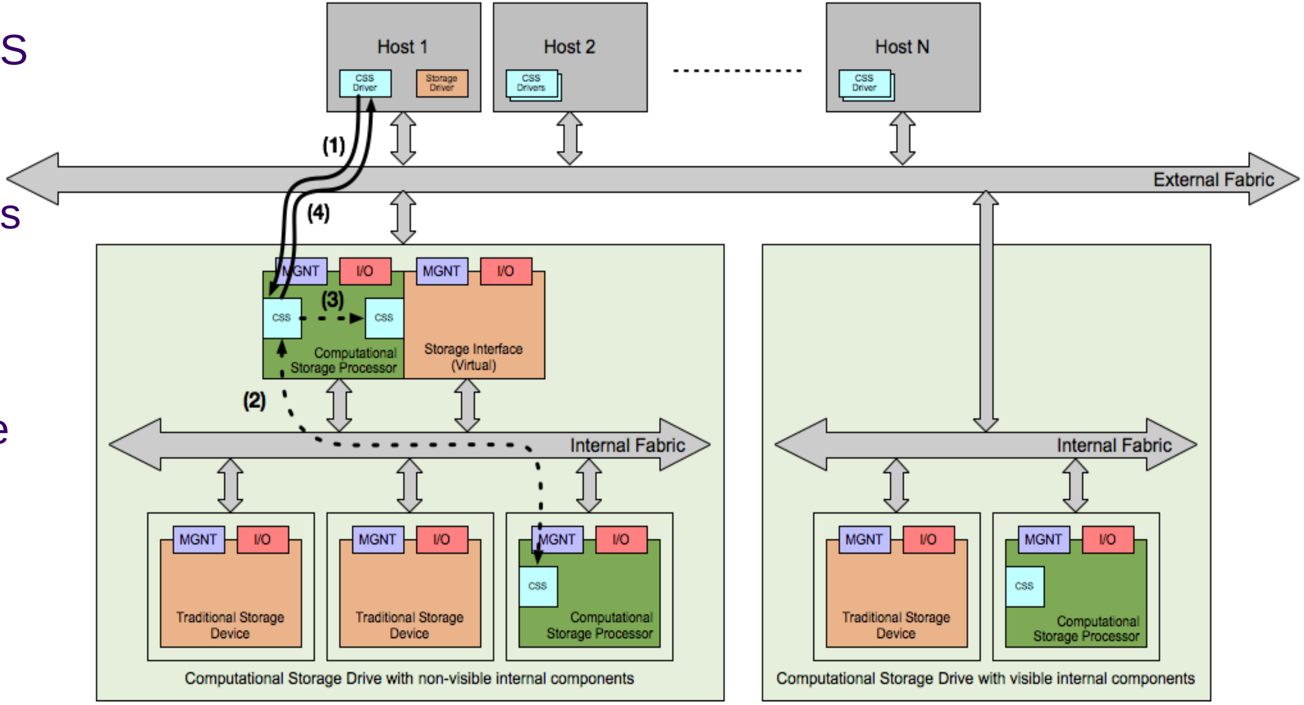
Discovery

- Host Agent discovers the CSS capabilities of the attached CSX devices
- Each CSx returns info on available CSSes
  - CSx ID
  - CSS ID(s)
  - Vendor
  - Type & Subtype
  - State & Reservation Information
  - Configuration Schema
  - Active Configuration
  - Error Information



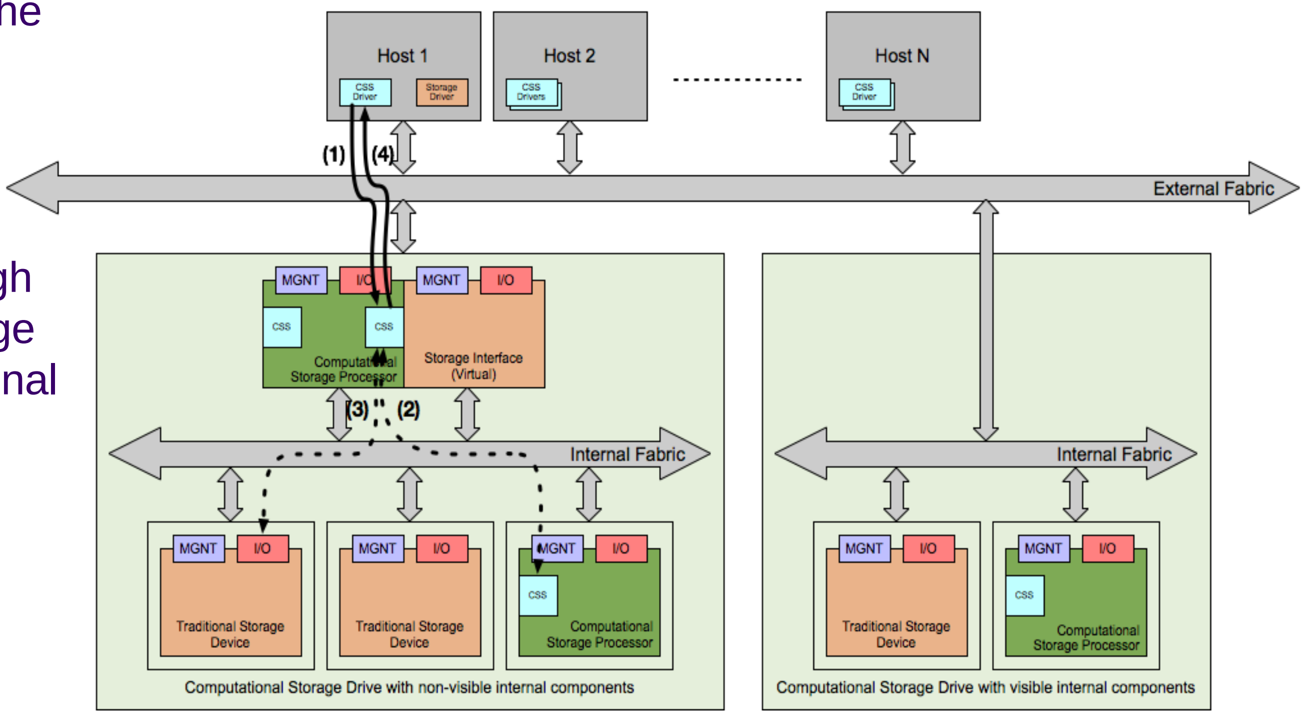
Configuration

- Host Agent sends the CSS Configuration request to each CSx
- For Programmable CSSes can result in creation of additional CSSes
- Readies the CSS for use by the Host Agent via one of two usage models – Direct & Transparent



Direct Operation Usage Model

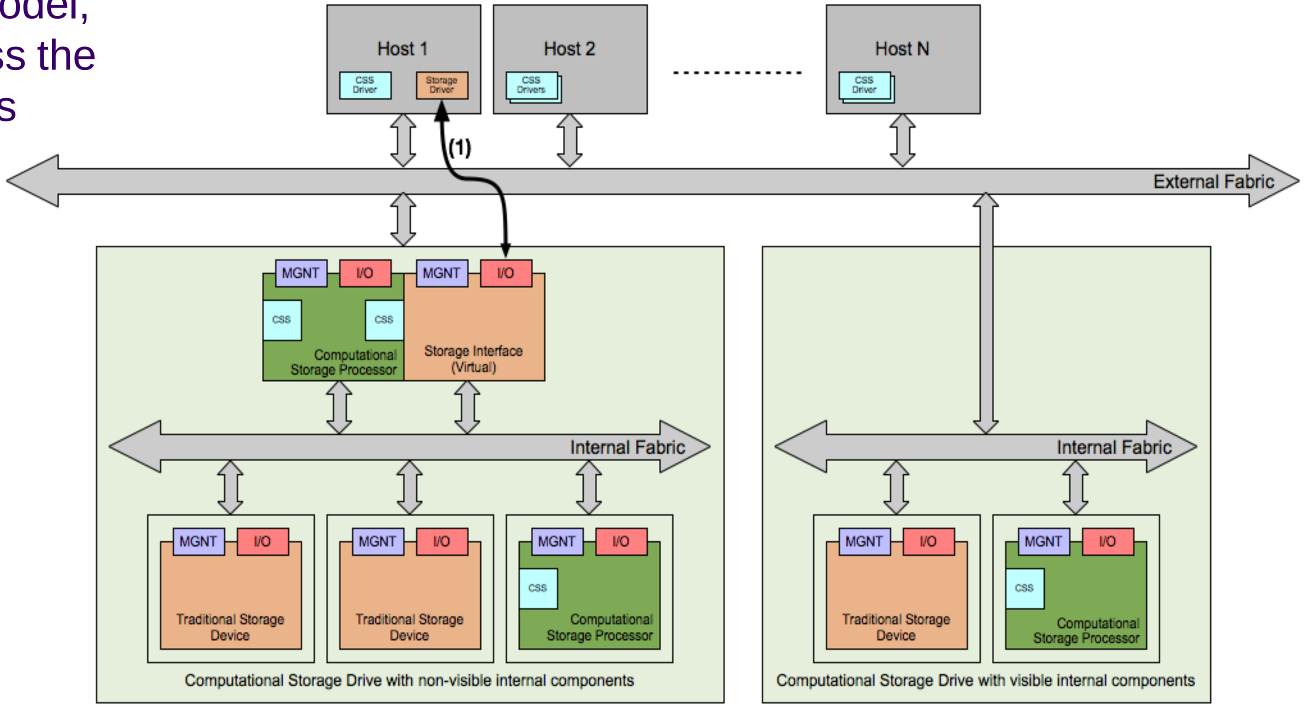
- Using the Direct model, the Host Agent will have a specific Computational Storage API required for interaction
- Commands will go through the Computational Storage Processor or Computational Storage Drive interface



Transparent Operation Usage Model

SNIA COMPUTATIONAL STORAGE

- Using the Transparent model, the Host Agent will access the computational capabilities through a standard storage API
- I/O commands will go through the traditional Storage interface



Computational Storage SW Infrastructure

SNIA COMPUTATIONAL STORAGE

Management

nvme-cli
nvme-of

Applications

GitHub
libcsnvme

intel
SPDK

Userspace

OS

Windows

vmware

CSDs, CSPs and CSAs

Hardware

Computational Storage
Example Workload Improvements Provided by Member Companies

RAID or Compression Offload

The case for Peer-2-Peer (P2P) processing…

- PCIe End-Points (EPs) are getting faster and faster e.g. NVMe SSDs, RDMA NICs & GPGPUs
- Bounce buffering all IO data through system memory is a waste of system resources and reduces QoS for CPU memory

The solution:

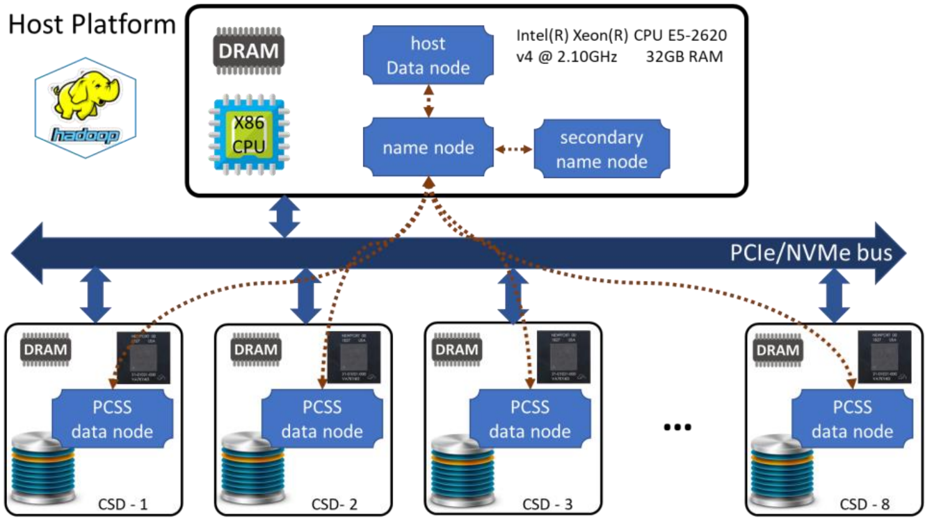
- A CSP + p2pmem Linux kernel framework for allowing PCIe EPs to DMA to each other while under host CPU control
- CPU/OS still responsible for security, error handling etc.
- 99.99% of DMA traffic now goes direct between EPs
- **Application:** P2P Compression offload

Get your FPGA's "out of the box" and shared across the datacenter

- Emerging ecosystem allows CSPs to be accessed/shared across network fabrics such as Ethernet
- FPGA acceleration being shared across the network fabric enables FPGA disaggregation

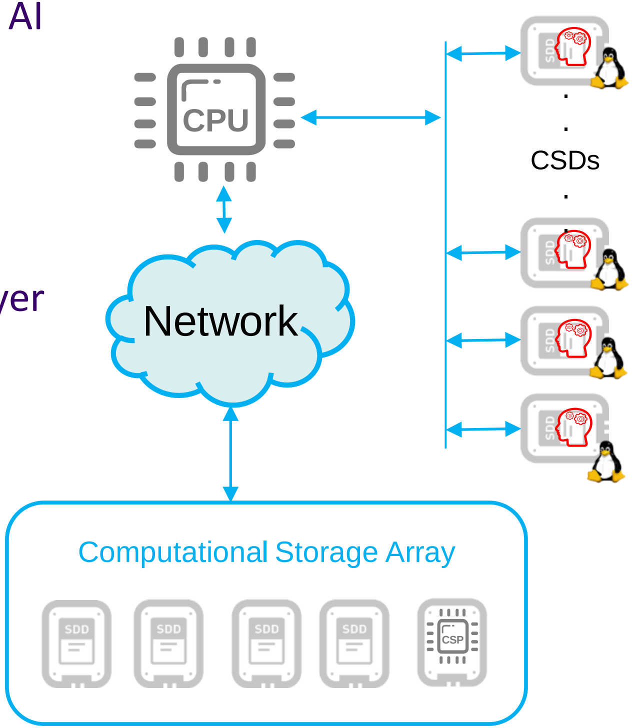
Hadoop Cluster Improvement via CSD

- Ability to Migrate Data Nodes into CSDs
- Allow for user to reduce Host CPU Core count via CSD usage
- Migrating the data node via a PCSS into an array of CSDs attached to the host systems
- Scalable across nodes, HW and datacenter space



AI Inference at the Storage

- Generate Metadata database (e.g tags) over a large set of unstructured data locally with an integrated AI inference engine
- Operation may be:
  - Triggered by a host processor
  - Done offline as a background task (batches)
- Metadata database may be then used by upper layer Big Data Analytics software for further processing
- Can work both on direct attached storage or on remote, over the network storage
- EXAMPLES:
  - Video search, Ad insertion, Voice call analysis
  - Images, Text scan, etc.



In Summary – Call to Action

- Computational Storage is a Real Market
  - Customers are deploying today
- Solutions exist and will continue to grow
  - Making the interface ‘uniform’ helps adoption
- Standardizing the host interaction is vital
  - We NEED more Support from Users/SW Solutions
- Working across the industry will be crucial

SNIA COMPUTATIONAL STORAGE

Thank You!!

www.SNIA.org/Computational