SNIA™ | COMPUTATIONAL STORAGE

What happens when Compute meets Storage?

Leah Schoeb, AMD
Eli Tiomkin, NGD Systems

Feb 2020

Agenda

- **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

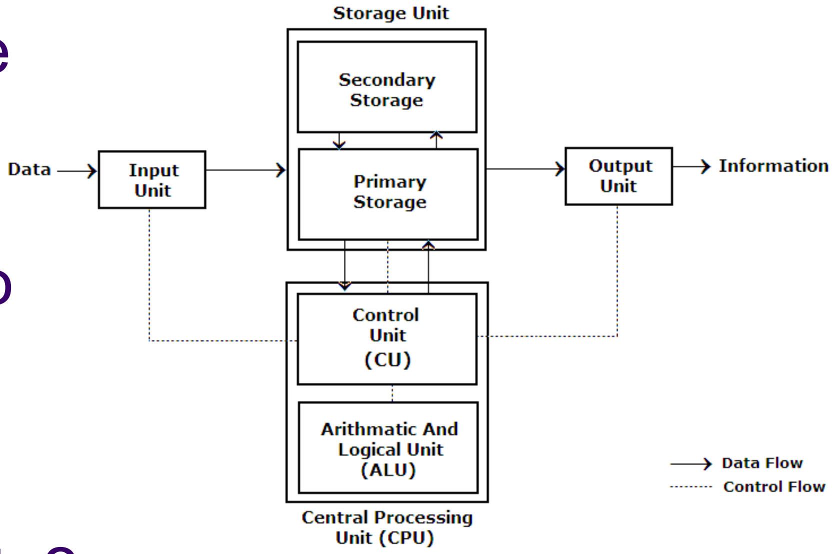
Internet of Things blog

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

Compute, Meet Data

SNIA COMPUTATIONAL STORAGE

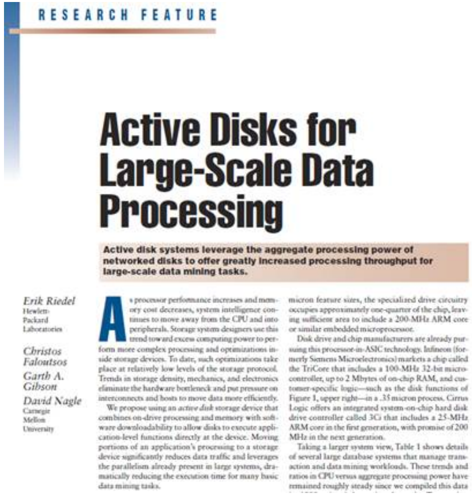
- Based on the premise that storage capacity is growing, but **<u>storage architecture has remained mostly unchanged</u>** dating back to pre-tape and floppy…
- How would you define changes to take advantage of Compute at Data?



The Evolution of Computational Storage

SNIA. 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.

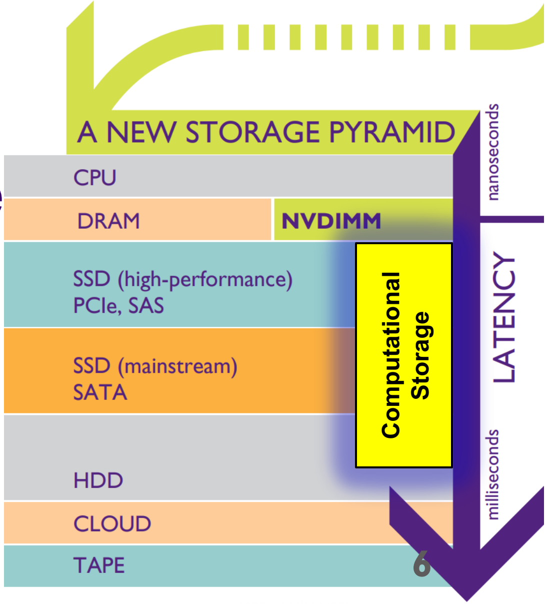
micron feature sizes, the specialized drive circuitry occupies approximately one-quarter of the chip, leaving sufficient area to include a 200-MHz ARM core or similar embedded microprocessor.

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

43 Participating Companies
175 Member Representatives

SNIA COMPUTATIONAL STORAGE

AMD
arm
BROADCOM
CALYPSO Systems
DELL EMC
EIDETICOM
EXTEN TECHNOLOGIES
FADU
FUJITSU
GIGAIO
Hewlett Packard Enterprise
HITACHI
IBM
INNOGRIT 英韧科技
inspur
intel
JETIO
KALRAY
KIOXIA
Lenovo
MARVELL
Mellanox TECHNOLOGIES
Micron
Microsemi a MICROCHIP company
NEC
NetApp
NETINT
NGD systems
NYRIAD
ORACLE
PLIOPS
PURESTORAGE
SAMSUNG
ScaleFlux
SEAGATE
SiliconMotion
SK hynix
SUPERMICRO
TOSHIBA
vmware
WD Western Digital
XILINX

Finding a Focus and Direction

SNIA COMPUTATIONAL STORAGE

- 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 through PMCS Initiative**
  - Promote Computational Storage paradigms through the industry at large

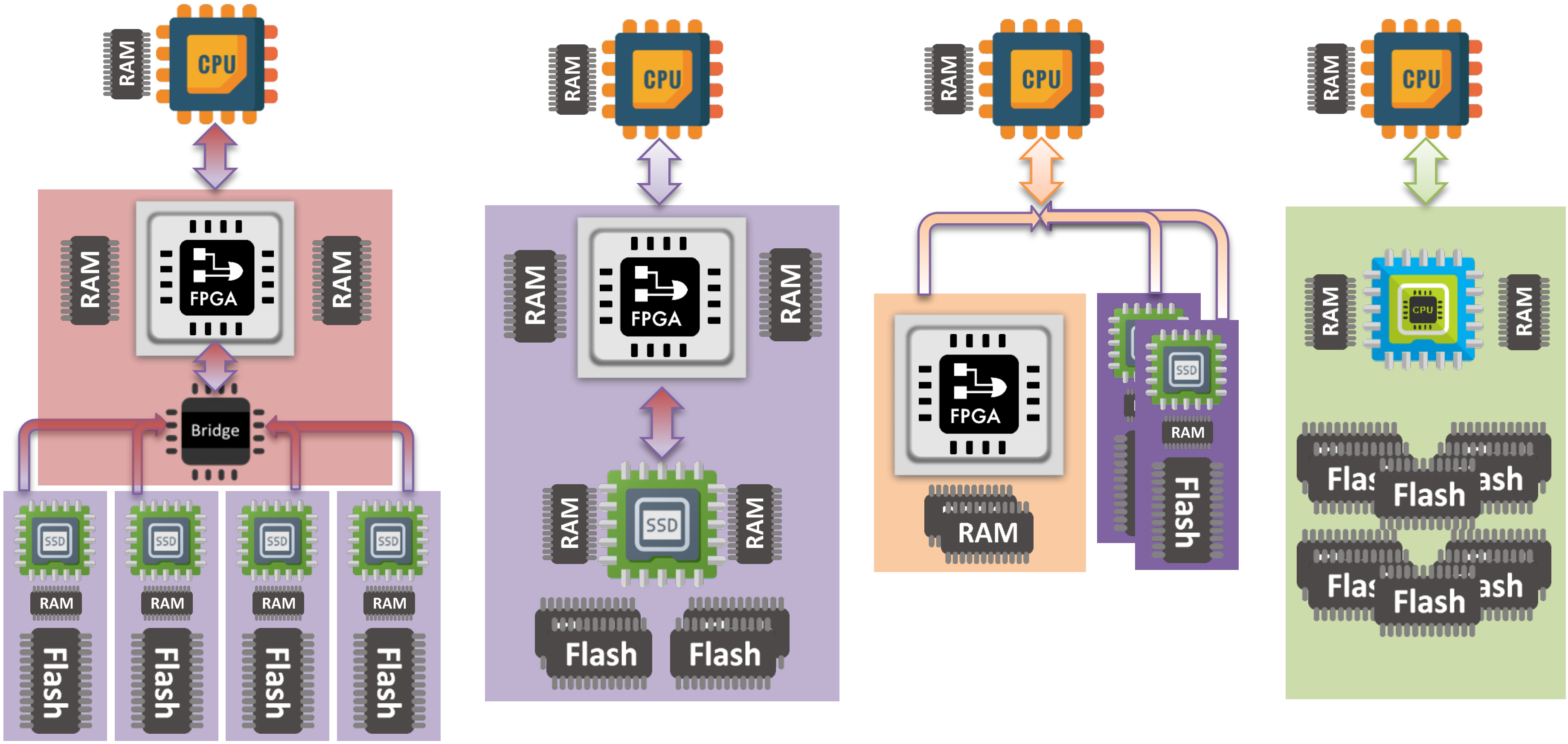
Speaking the Same Language

SNIA COMPUTATIONAL STORAGE

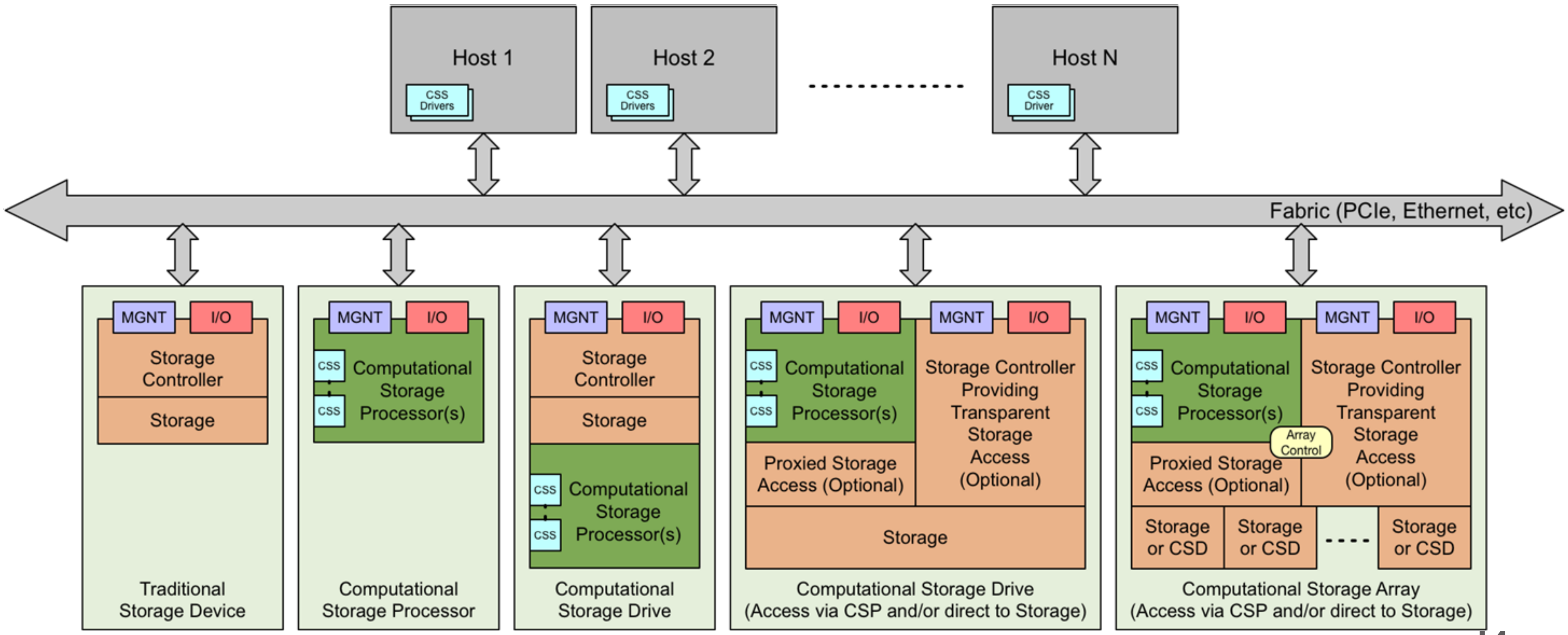
- **Computational Storage:**
  - Architectures that provide Computational Storage Services coupled to storage offloading host processing and/or reducing data movement.

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

Current Instances of Computational Storage



Computational Storage Devices (CSx)



Computational Storage Services

- 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

SNIA COMPUTATIONAL STORAGE

- **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.

Computational Storage

Example Services Provided from the Ongoing Architecture Document V0.3

Example Fixed Services

Documented Fixed Computational Storage Services

1. **Compression FCSS**

A compression CSS reads data from a source location, compresses or decompresses the data, and writes the result to a destination location.

CSS configuration specifies the compression algorithm and associated parameters.

CSS command specifies the source address and length and the destination address and maximum lengths.

2. **Database Filter FCSS**

A database filter CSS reads data from source location(s), performs a database projection (column selection) and filter (row selection) on the data according to projection and filter conditions, and writes the result(s) to destination location(s).

CSS configuration specifies the database format, table schema, selection and filter conditions, and associated parameters.

CSS command specifies the source address and length, and the destination addresses and lengths.

Example Programmable Services

Documented Programmable Computational Storage Services

1. **Operating System Image Loader PCSS**

An Operating System Image CSS allows an operating system image to be loaded and executed. The operating system may implement one or more additional CSSes.

CSS configuration specifies the location where the image is able to be obtained, and integrity/security verification information.

CSS command specifies pause/resume/start/stop/unload operations.

2. **Container Image Loader PCSS**

A Container Image CSS allows a container image to be loaded and executed. The container may implement one or more additional CSSes.

CSS configuration specifies the location where the container is able to be obtained, and integrity/security verification information.

CSS command specifies pause/resume/start/stop/unload operations.

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**

Thank You!!

www.SNIA.org/Computational