SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

Composable Infrastructure and Computational Storage

SNIA BrightTalk Webcast

September 15, 2020

10:00 am PT

Today's Speakers

Moderator:
Alex McDonald
NetApp
Co-Chair, SNIA Compute, Memory, and Storage Initiative

Presenter:
Philip Kufeldt
Seagate Technology
Member, SNIA Computational Storage Technical Work Group

Presenter:
Eli Tiomkin
NGD Systems
Chair, CMSI Computational Storage Special Interest Group

SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

SNIA Legal Notice

- 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.

SNIA-at-a-Glance

185
industry leading organizations

2,000
active contributing members

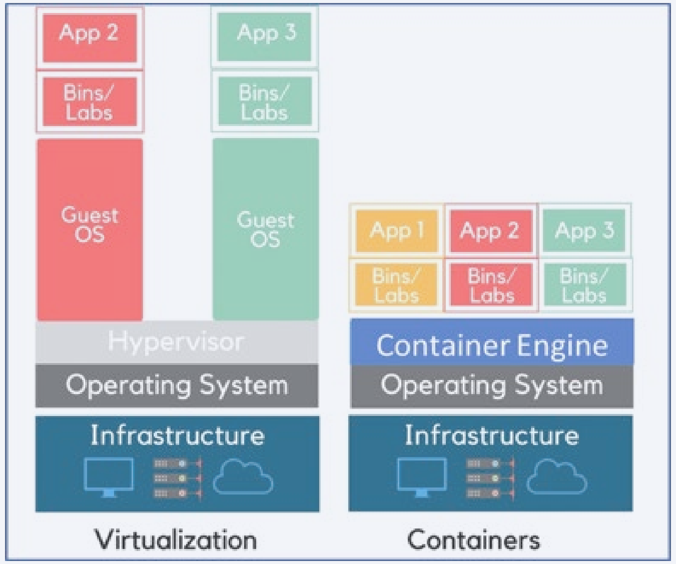
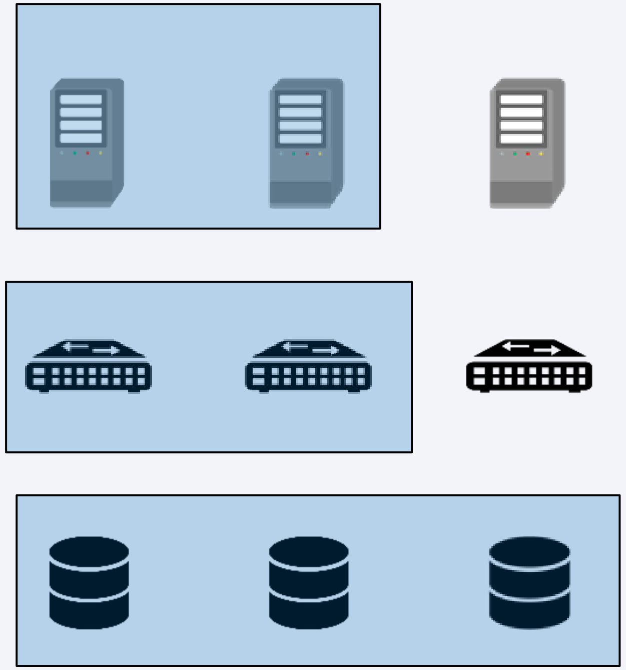
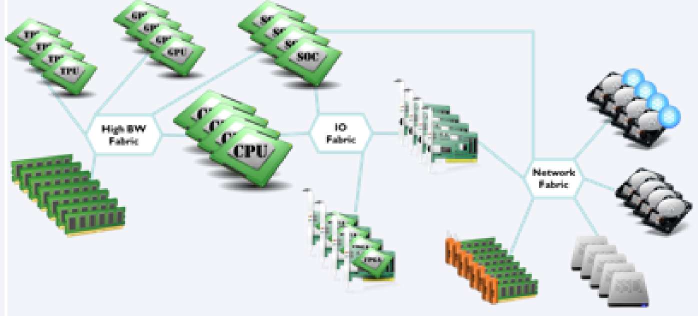
50,000
IT end users & storage pros worldwide

Agenda

- Need for Change
- What is Composable Infrastructure?
  - What are the use cases?
- What is Computational Storage?
  - What are the use cases?
- Are They Mutually Exclusive or Beneficial?

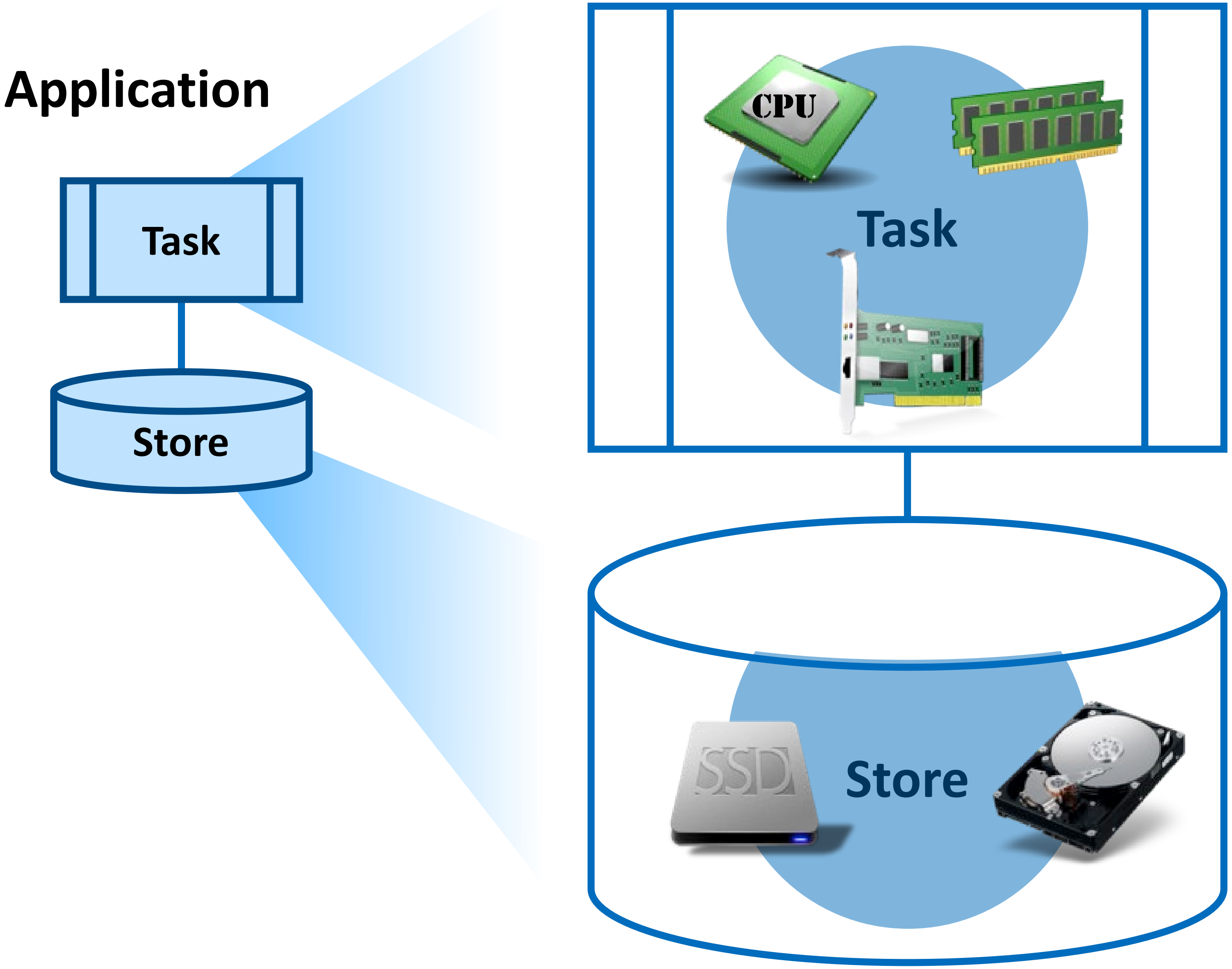
Need For Change

Where Are We Today?

Where We've Been	Where We Are	Where We Are Going
		

What Is An Application?

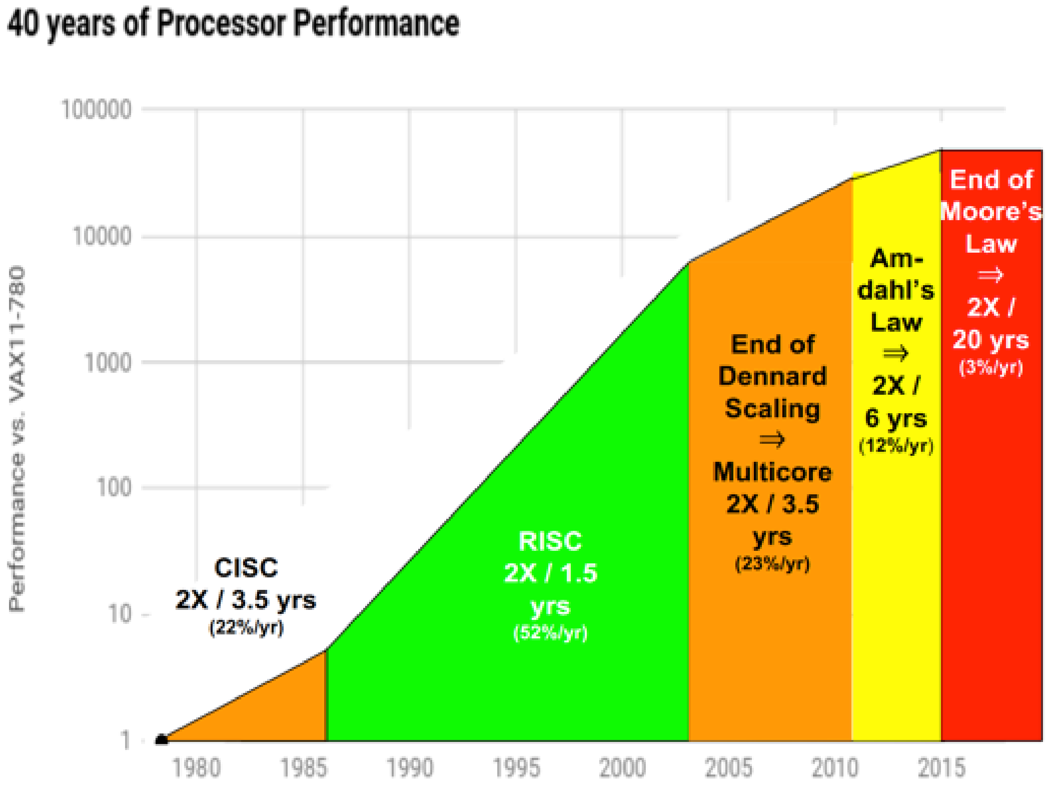
- Task
  - Apps need a system
    - They have requirements
      - CPU cores
      - Memory size
      - Network bandwidth
      - Network location
      - Availability
- Store
  - Most apps need a persistent store
    - They have requirements
      - Bandwidth (BW)
      - Latency
      - Capacity
      - Availability
- Examples
  - RDBMS
  - Web Servers
  - ML application



SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

While Apps Are GROWING

- Dennard's scaling ended
  - Power leakage and heat prevent cycle scaling
- Multicore hit Amdahl's law
  - Applications can only be parallelized so far
- Moore's law is ending
  - Physical size limitations
- What's left then?
  - Domain Specific Architectures
    - Graphics Processing Unit (GPU)
    - Offloading Network Interface Controllers (NIC)
    - Tensor Processing Unit (TPU)
    - FPGA Based Accelerators
  - This increases configuration complexity

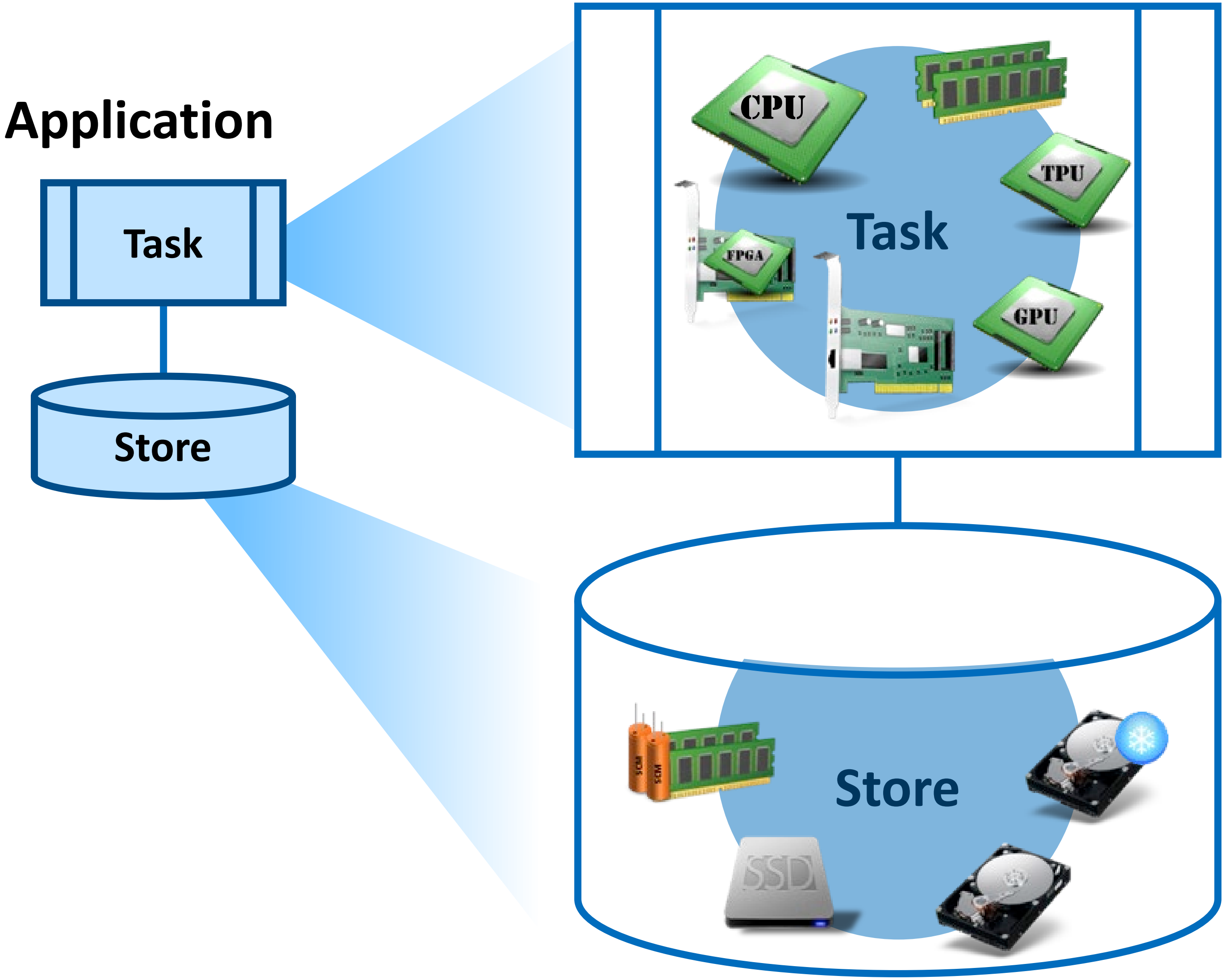


Based on SPECintCPU. Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e 2018

David Patterson's presentation at ISSCC2018
https://youtu.be/NZS2TtWcutc

Today's Applications

- Task
  - Many System options
    - CPUs/SoCs
    - Core counts
    - DDR Capacity
    - NICs
      - TOEs
      - RDMA
    - Accelerators
      - FPGAs
      - GPUs
      - TPUs
- Store
  - Many Store options
    - Cold stores
    - HDDs
    - SSDs
    - Persistent memory (SCM) devices
- All must go into a box
  - Dictated by App requirements
  - What and how much decided at purchase time
  - No going back, no evolution



What is the Problem

- Application requirements wide and varied
  - Complicated set of hardware requirements
- Application requirements quickly and constantly evolve
  - Agile development creates quickly evolving app requirements
  - Mapping occurs at purchase time and cannot evolve
  - Invalidates system design requirements
- Must map these requirements onto physical hardware
  - Due to high core counts, multiple apps must be mapped to single system
  - **Forces IT managers to be system designers**
  - **Forces overprovisioning inside the system**
  - Availability and Competition issues
- Growth rate of apps
  - Forces overprovisioning system counts for elasticity
- Ever growing classes of hardware systems
  - Lifecycle management (scaling, EOL, etc) becomes a multi-vectored problem

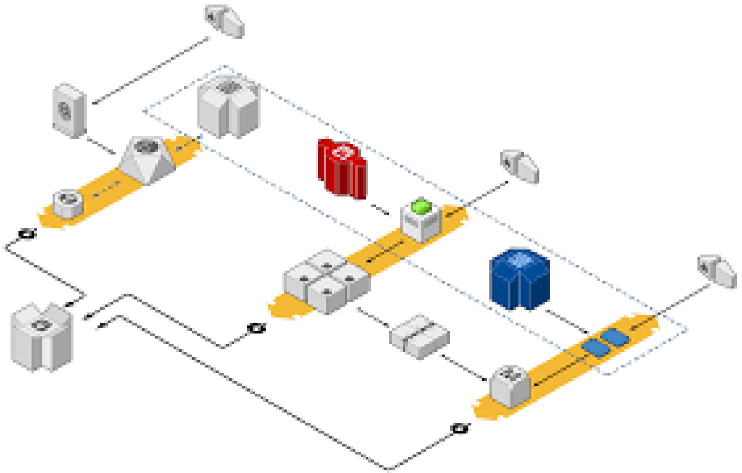
The multicore server as the unit of app allocation is now too big and complicated

What is Composable Infrastructure?

What is Composable Infrastructure?

- Compose – to form by putting together

- Infrastructure – the underlying foundation or basic framework (as of a system or organization)

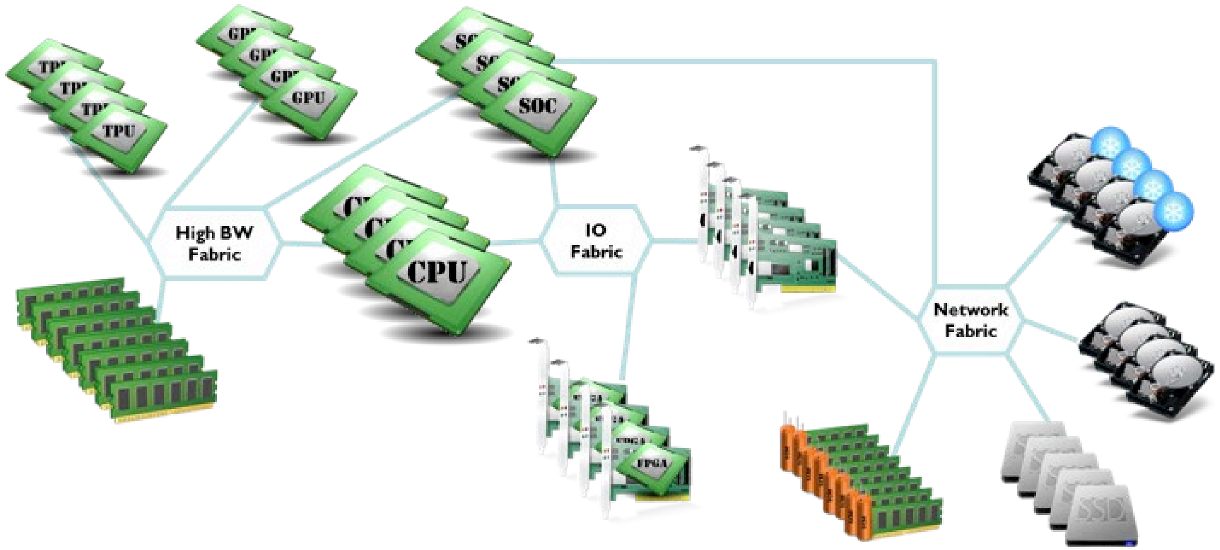


Source: Merriam Webster dictionary

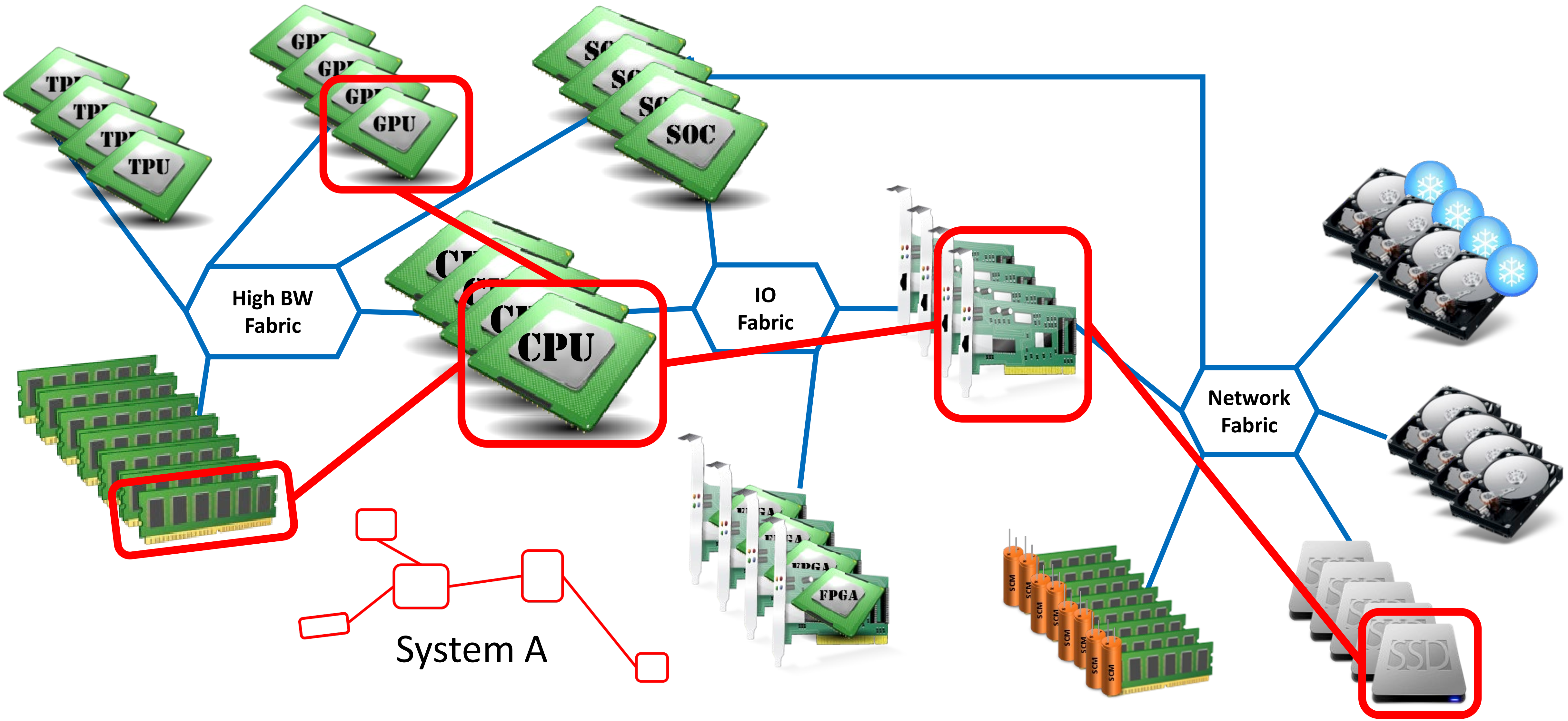
SNIA. CMSI | COMPUTE, MEMORY, AND STORAGE

What is Composable Infrastructure?

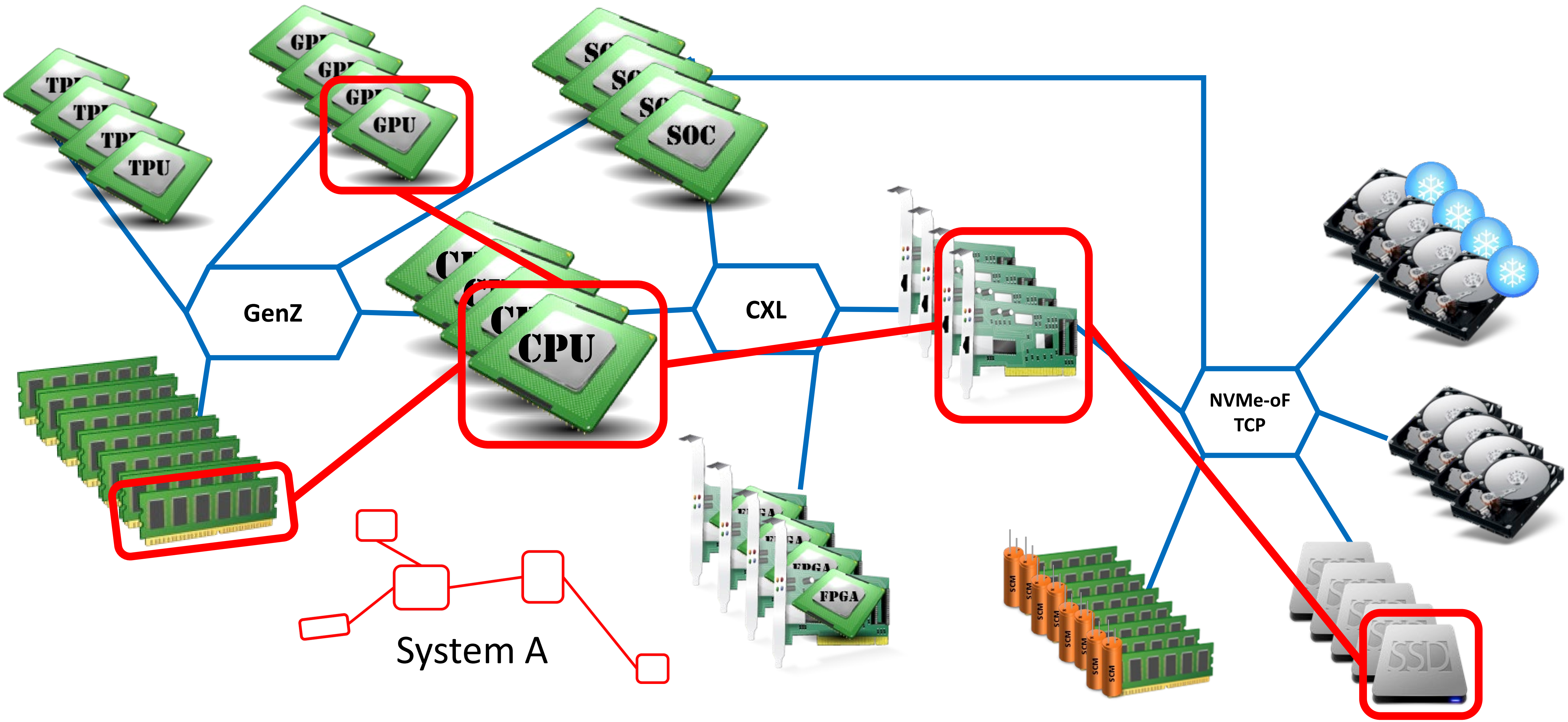
- ✓ Disaggregate the server
  - ✓ Separate compute, memory storage, networking components, accelerators
  - ✓ Use high speed low latency fabrics to interconnect
- ✓ Create pools of resources
  - ✓ Don’t need to be physically proximate
  - ✓ Horizontal scaling is in expanding pools
- ✓ Compose and decompose as needed via orchestration
  - ✓ API driven (autonomous operation)
  - ✓ Vertical scaling by combing more resources
- ✓ Orchestration driven by dynamic application needs



Composing a System

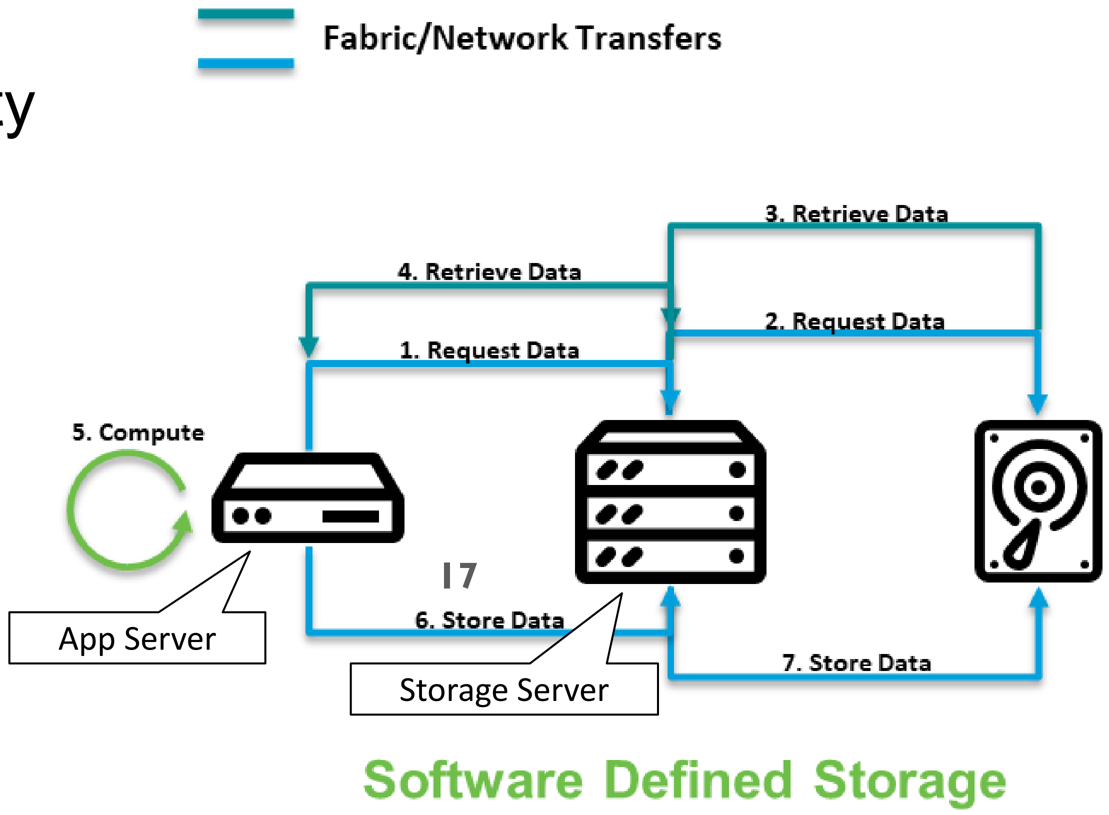


Example



Current Storage View

- SDS already creates disaggregated scalable composable storage
  - Storage is disaggregated
  - Allocation/Deallocation can be automated
- Demonstrates key issues with composability
  - App servers waits for storage
    - Adds latency
  - Consumes DDR BW
    - SSDs can easily consume DDR BW
    - Does this on two systems
  - Consumes Fabric/Network
    - Increased power needs
    - Increased transfers
    - Increased provisioning costs



SNIA. CMSI | COMPUTE, MEMORY, AND STORAGE

What Is Computational Storage?

Need A New Way to Look at Storage

Pain Points

Physical Space

Available Power

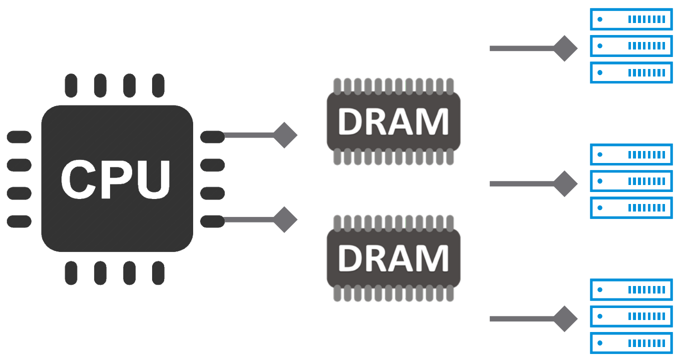
Scaling Mismatch

Bottleneck Shuffle

Scaling requirements are not met with existing solutions

One CPU to many storage devices creates bottlenecks

These bottlenecks exist, we currently just shift where they reside

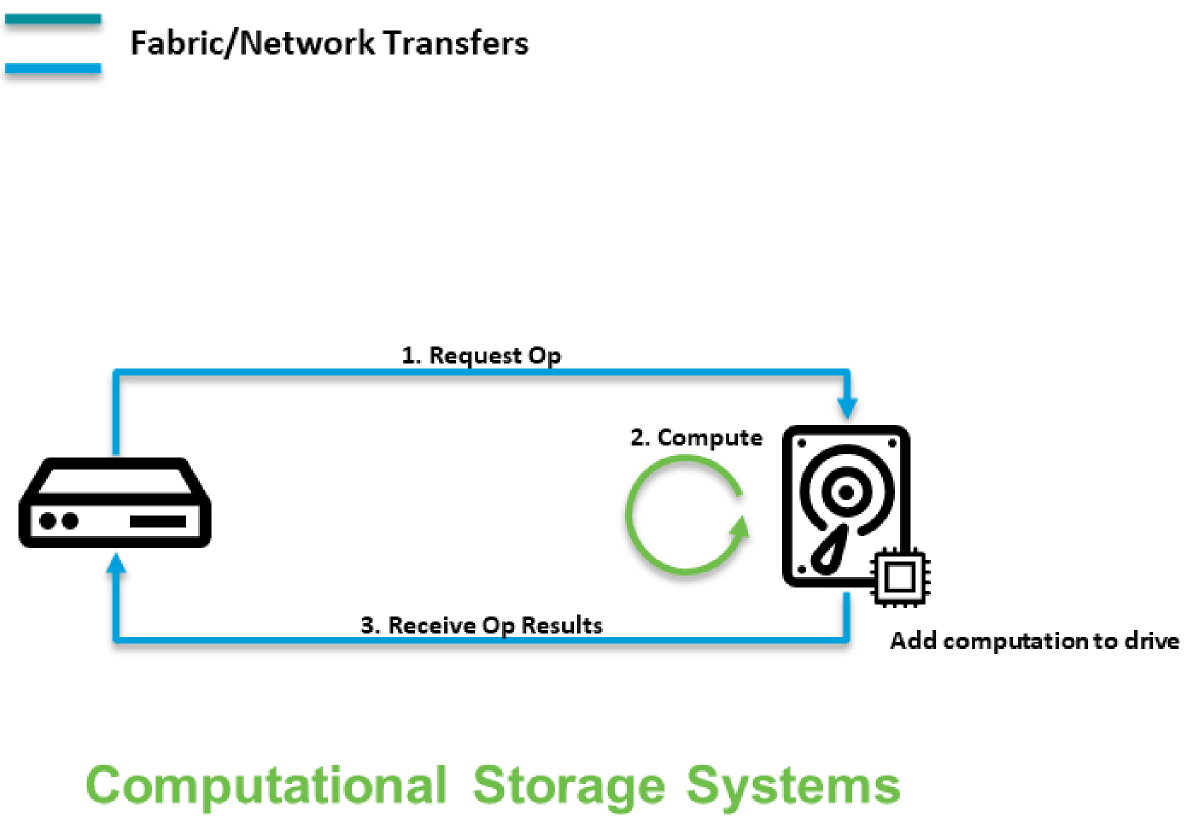


Technologies that ‘compose’ these elements just exacerbate the bottleneck

A way to augment and support without wholesale change is needed

Computational Storage View

- Computational Storage Function (CSF)
  - Send compute request to the drive
  - Allow drive to reduce data
  - Only return the results
  - Can be local or fabric attached
  - Reduces fabric and DDR BW consumption
  - Costs Saving
    - Reduced transfers
    - Reduced power
    - Free up host cycles
    - Potential for server removal
  - Potential for massively parallel compute

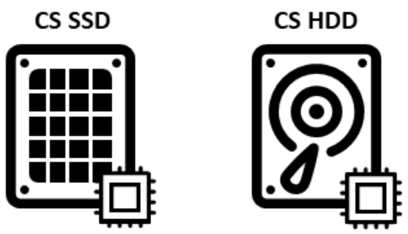


SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

Computational Storage Devices

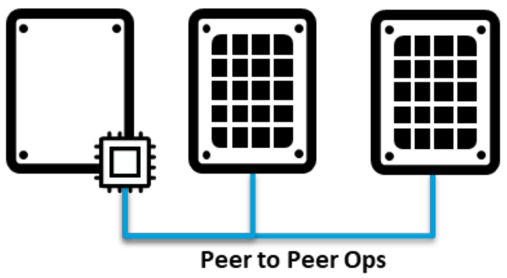
- **Computational Storage Drive (CSD)**:

  A storage element that provides Computational Storage Function and persistent data storage.



- **Computational Storage Processor (CSP)**:

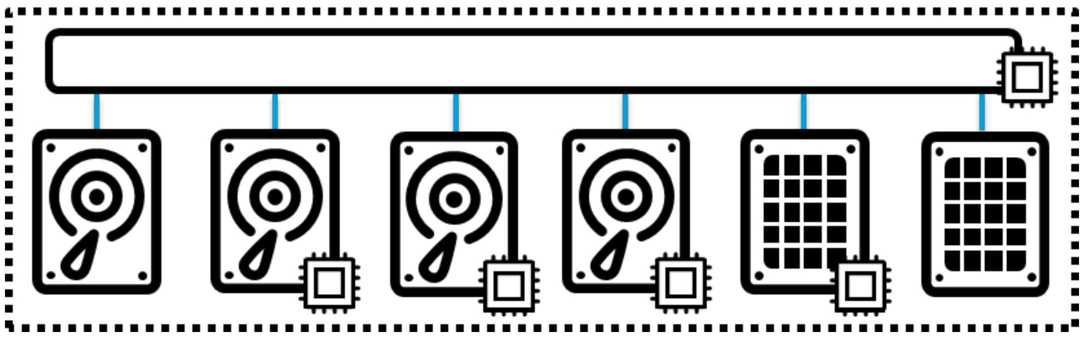
  A component that provides Computational Storage Functions for an associated storage system without providing persistent data storage.



- **Computational Storage Array (CSA)**:

  A collection of Computational Storage Devices, control software, and optional storage devices.

  (Many options here)



Using Computational Storage

Benefits

- ✓ Distributed Processing
- ✓ Faster Results
- ✓ Lower Power
- ✓ Smaller Footprint

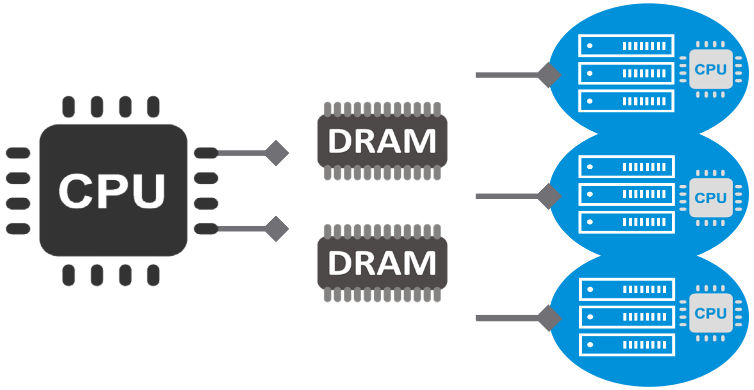
Reduced data transfers

Reduced fabric provisioning

Scaling compute resources with storage provides access to results faster

Computational Storage resources ‘offload’ work from the overtasked CPU

Seamless architectures create new ‘servers’ with each storage device added



Additional CPU resources without added rack space

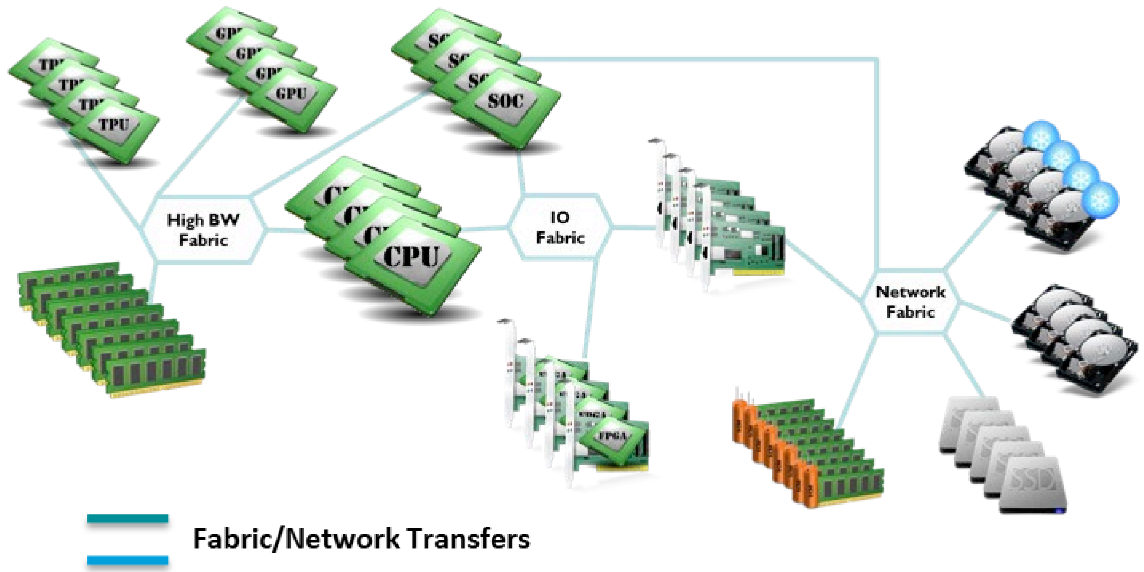
SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

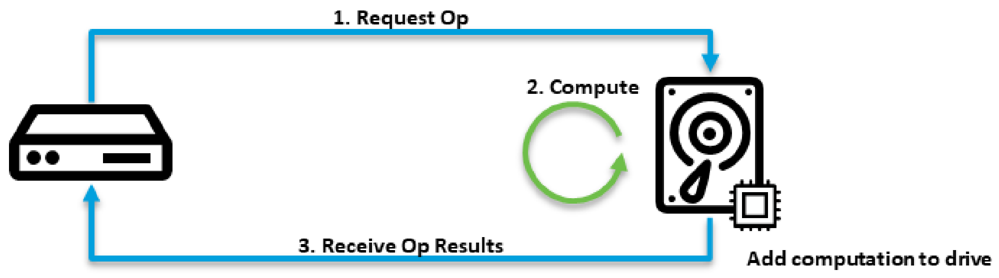
Are Composable Infrastructure and Computational Storage Mutually Exclusive . . .

or Mutually Beneficial?

Mutually Beneficial

- ✓Enable computational storage
  - ✓Reduce the data movement between the storage and the host
- ✓Compose and decompose as needed via orchestration
  - ✓Run some applications in storage
- ✓Orchestration driven by dynamic application needs
  - ✓Utilize computational storage to reach a higher level of system efficiency





Computational Storage Systems

Finding the Needles in Haystacks with AI and CSDs

Problem Statement

- Databases growing at exponential rates

10 M	1 Billion	1 Trillion
2007	2017	2021

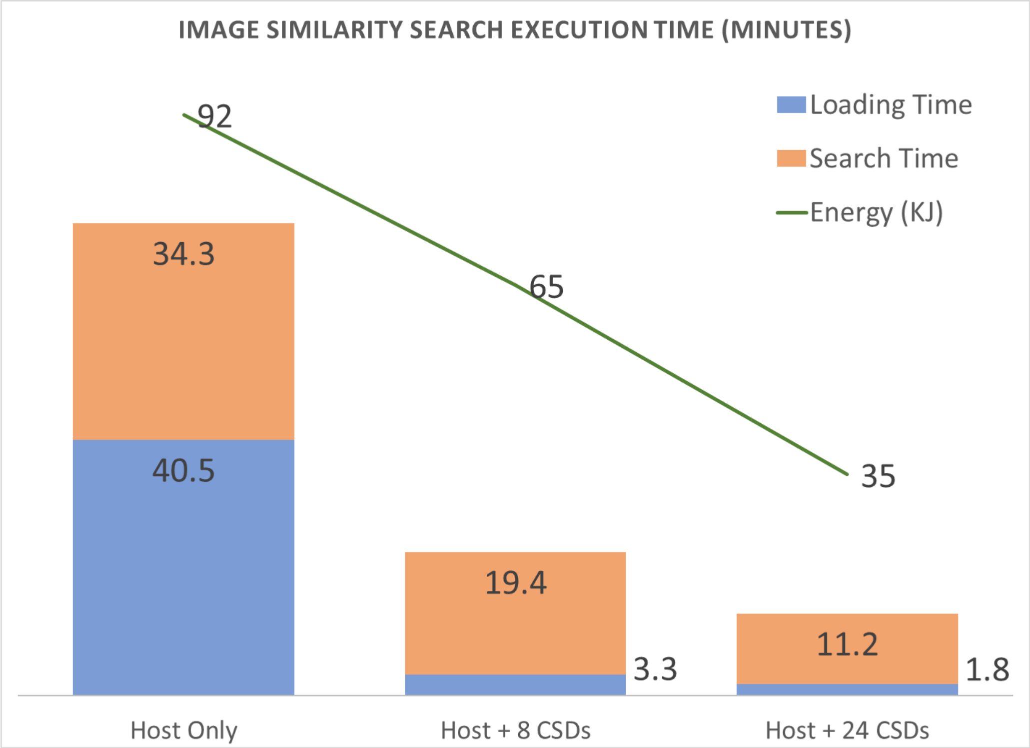
- Load and Search time key blocks in getting results

Computational Storage Solution

- Determine best way to increase performance
- Load Time Reductions due to CSD Offload of AI code

Results are Proven:

- **Load Time Reduced > 95%**
- **Search Time Reduced > 60%**
- **Power Savings of > 60%**



Association for Computing Machinery

Technical paper to be published in the ACM journal on Computational Storage

Thanks for Watching Our Webcast

- Please rate this webcast and provide us with feedback
- A link to this webcast and the PDF of the slides are posted to the SNIA Compute Memory and Storage Initiative website at https://www.snia.org/forums/cmsi/knowledge/articles-presentations
- You can also find this [webcast](#) and many other videos and presentations on today's topics in the [SNIA Educational Library](#)
- A Q&A from this webcast will be posted to the SNIA [Compute, Memory, and Storage Blog](#)
- Learn more about computational storage at www.snia.org/computational

SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

Where To Find Out More About Composable Infrastructure and Computational Storage

- Website resources
  - www.snia.org/CMSI
- Twitter
  - @sniacomputational
- Blog
  - SNIAComputeMemory&Storage
- Videos
  - https://www.youtube.com/user/SNIAVideo/playlists
- Educational materials
  - https://www.snia.org/educational-library
- Joining SNIA and the Compute, Memory, and Storage Initiative
  - https://www.snia.org/member_com/join-SNIA

SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

The ndustry leading companies of the SNIA Compute, Memory, and Storage Initiative (CMSI) support the industry drive to combine processing with memory and storage, and to create new compute architectures and software to analyze and exploit the explosion of data creation over the next decade.

CMSI Engages and Educates

- Computational Storage
- Persistent Memory
- PM and SSD Performance
- Solid State Drives
- Solid State Systems
- SSD Form Factors

CMSI Accelerates Standards

- Computational Storage Architecture Model
- Persistent Memory Programming Model
- PM Hardware Threat Model
- Solid State Storage Performance Test Specifications
- SSD Form Factor Specifications

CMSI Propels Technology Adoption

- Persistent Memory Programming Bootcamps
- PM Remote Access for High Availability White Paper
- SSD Form Factors Explained
- Compute, Memory, and Storage Demos at live and online technology events
- Interactive Webcasts with Industry Experts
- Technology Videos on the SNIA Video YouTube Channel

Learn more: **snia.org/csmi** @SNIA

SNIA CMSI | COMPUTE, MEMORY, AND STORAGE