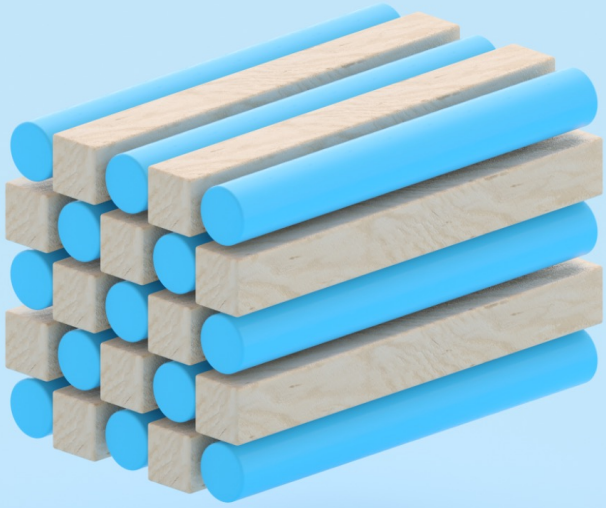
Enterprise Data Trends for Hybrid Multi-clouds

The semiconductor crisis, data growth, and new opportunities with cloud, AI, and models

Naresh Patel
Vice President & Chief Architect
NetApp

November 3, 2020

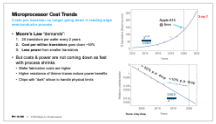
NetApp

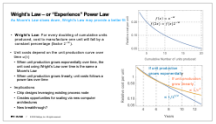


Overview

Layers × Lens

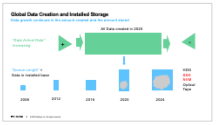
Tech Trends



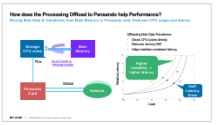


CRISIS

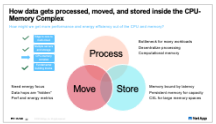
Global Data

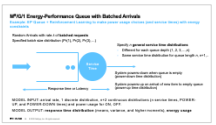


Servers/Storage



CPU complex

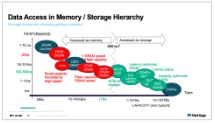




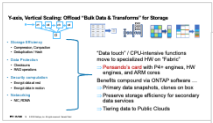
Process

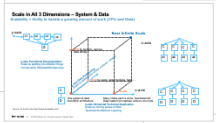
Move

Store



Scale Cube





Layers

TECHNOLOGY TRENDS

GLOBAL DATA / STORAGE SCALING AND PROPERTIES

SERVER AND STORAGE SCALING

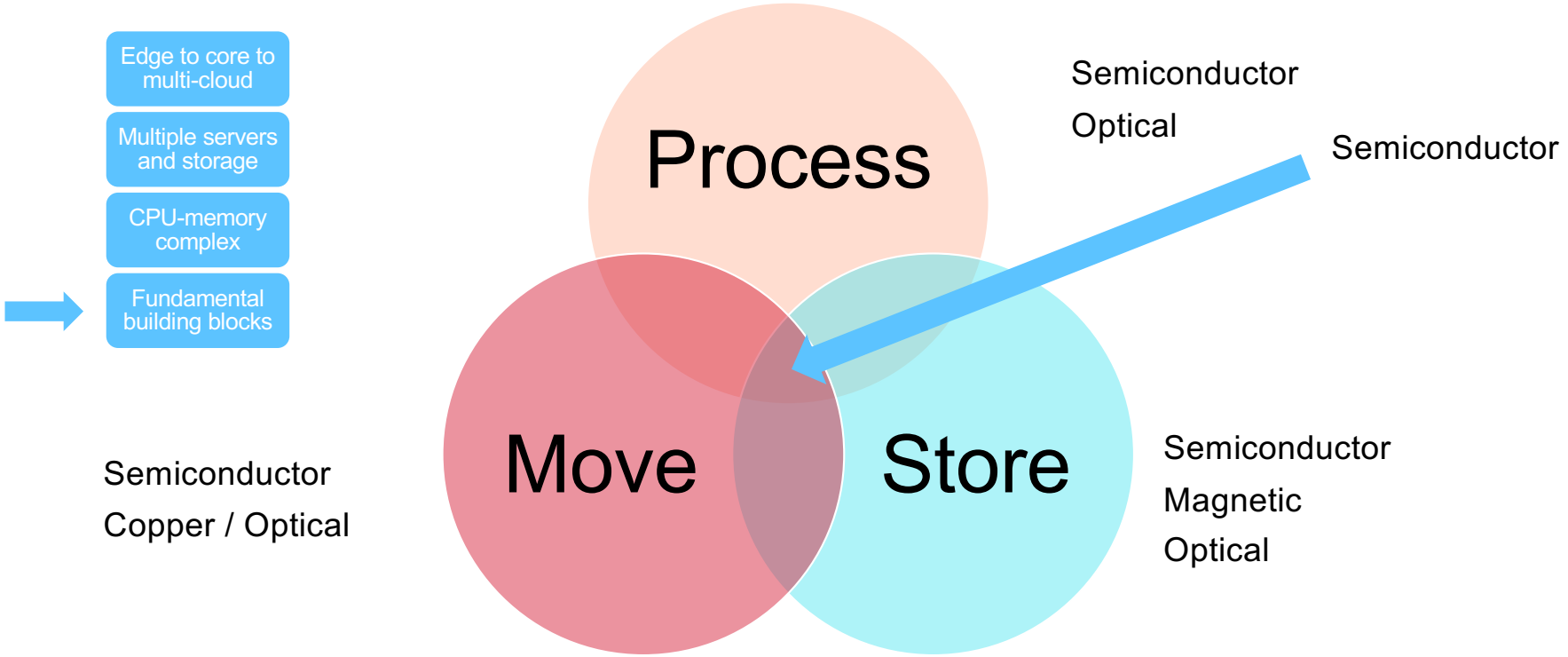
CPU / MEMORY COMPLEX

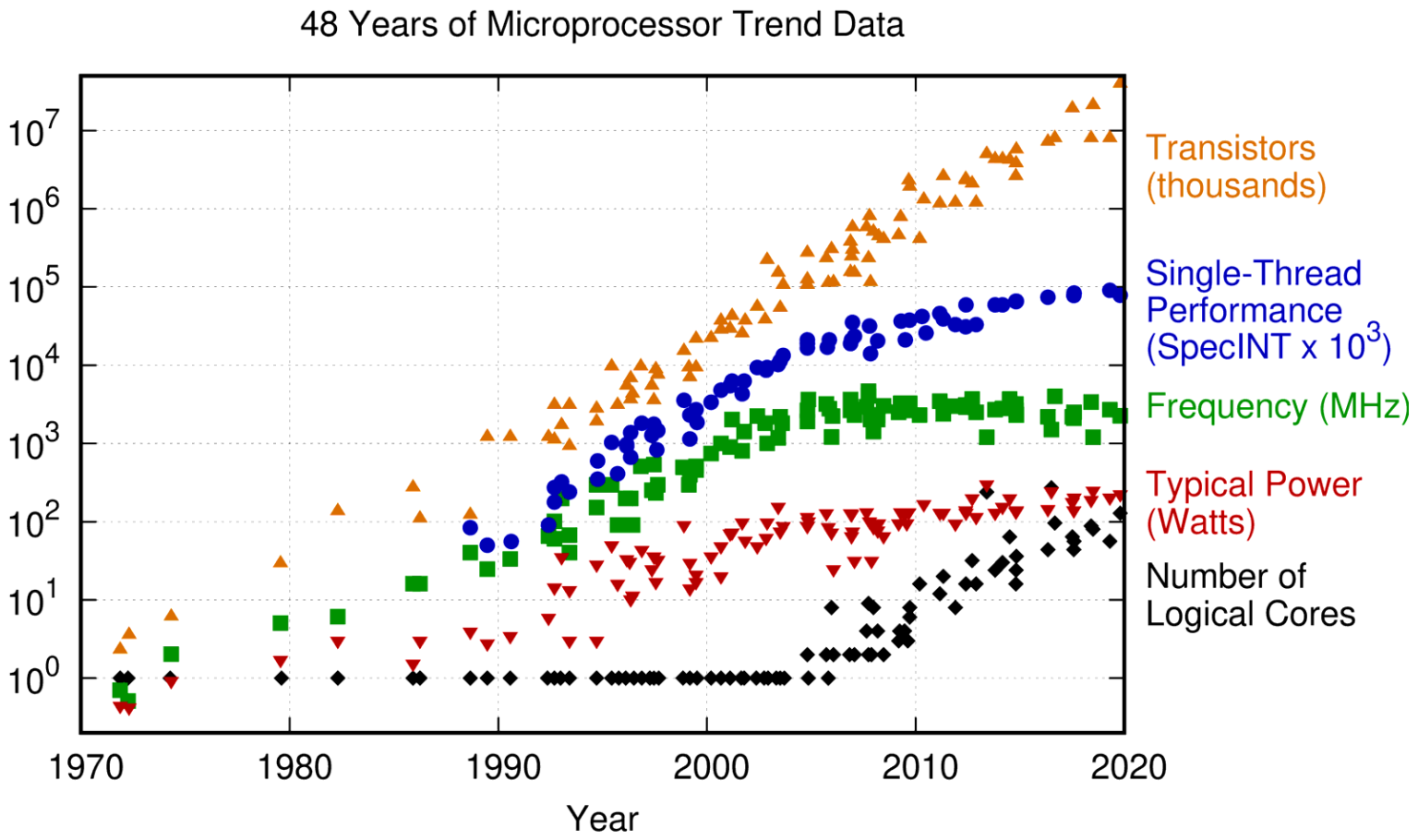
Lenses

- **Data-focused** interplay of the 3 things you can do with data:
  - Process it
  - Move it
  - Store it
- **AI-infused** decision making using real-time data vs. HW/SW designer making choices
- Opportunities for **performance/energy models & metrics**

How data gets processed, moved, and stored at the component level?

New data can be processed, moved and stored. Semiconductors play a key role each.



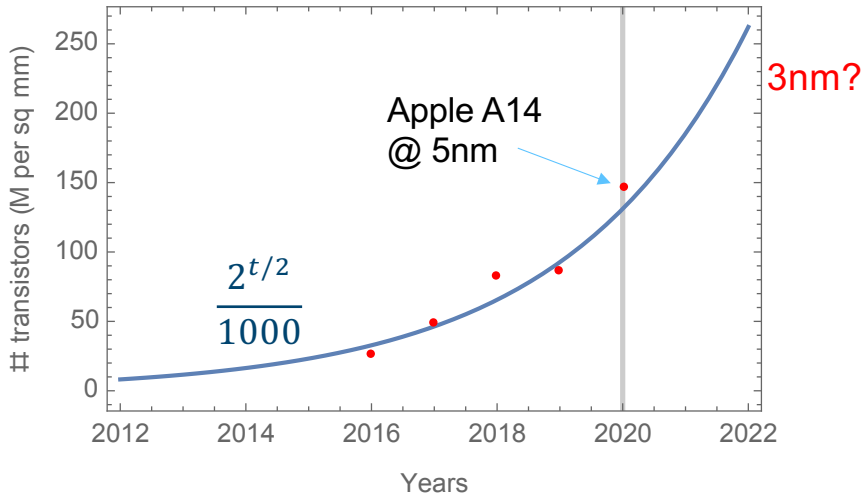


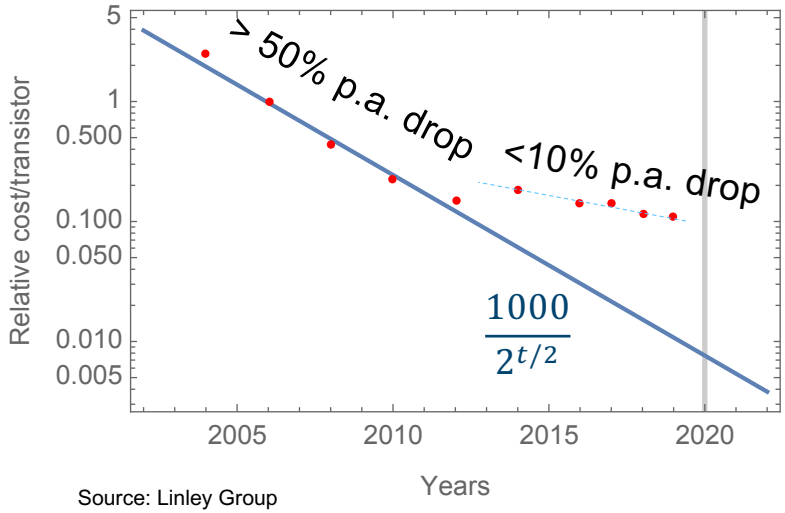
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

Microprocessor Cost Trends

Costs per transistor no longer going down in leading edge semiconductor process

- **Moore's Law** "demands":
  1. 2X transistors per wafer every 2 years
  2. **Cost per million transistors** goes down ~50%
  3. **Less power** from smaller transistors
- But costs & power are not coming down as fast with process shrinks
  - Wafer fabrication costs are higher
  - Higher resistance of thinner traces reduce power benefits
  - Chips with "dark" silicon to handle physical limits





Source: Linley Group

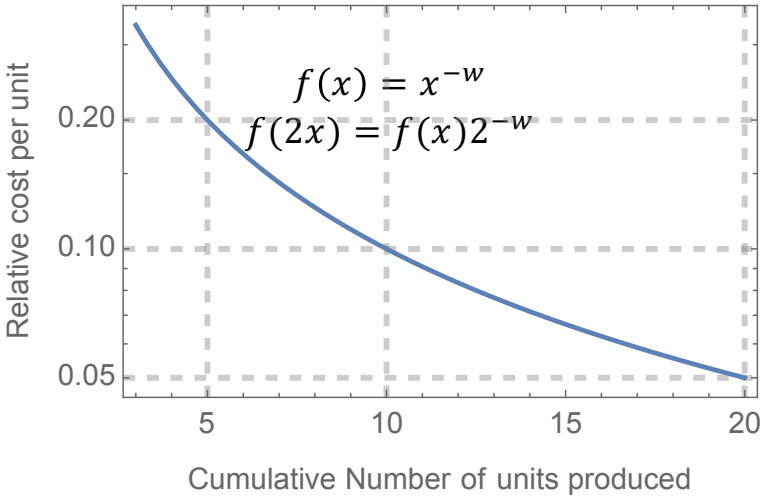
Wright's Law – or "Experience" Power Law

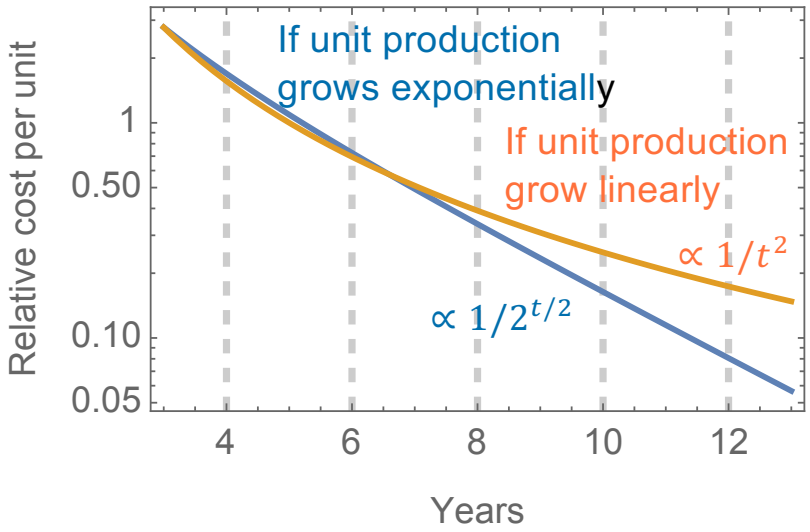
As Moore's Law slows down, Wright's Law may provide a better fit

- **Wright's Law**: For every doubling of cumulative units produced, cost to manufacture one unit will fall by a constant percentage (factor 2^{-w}).

- Unit costs depend on the unit production curve over time $u(t)$
  - When unit production grows exponentially over time, the unit cost using Wright's Law over time is the same a Moore's Law
  - When unit production grows linearly, unit costs follows a power law over time

- Implications
  - Chip designs leveraging existing process node
  - Creates opportunities for scaling via new computer architectures
  - New breakthrough?





Emerging CPU and Memory Trends

General purpose CPUs being augmented for speed and efficiency

Moore's Law scaling slows

- Manufacturing improvements continue
- Better packaging: Chiplet approach to overcome single monolithic die challenges for 3D scaling beyond 5nm
- Novel domain-specific architectures, esp. AI.

General purpose CPU becoming system bottleneck

- Composable architectures (including disaggregated resources) via SW APIs
- Scale-out architectures for public and private clouds
- Scale-up with HW accelerators and domain-specific architectures: GPUs, AI/ML,…

Persistent memory

- Memory-based computing provide synchronous access for simpler / efficient programming
  - Multi-TB scale

Processing inside memory

Emerging storage media trends

Ongoing Flash $/GB reduction by adding layers, but new breakthrough needed to reduce cost per Flash cell

New approaches for differentiated workloads

- Bring compute closer to media to take advantage of high internal BW
- Wider adoption of AI/ML techniques for large immutable data sets

Cost/GB reduction

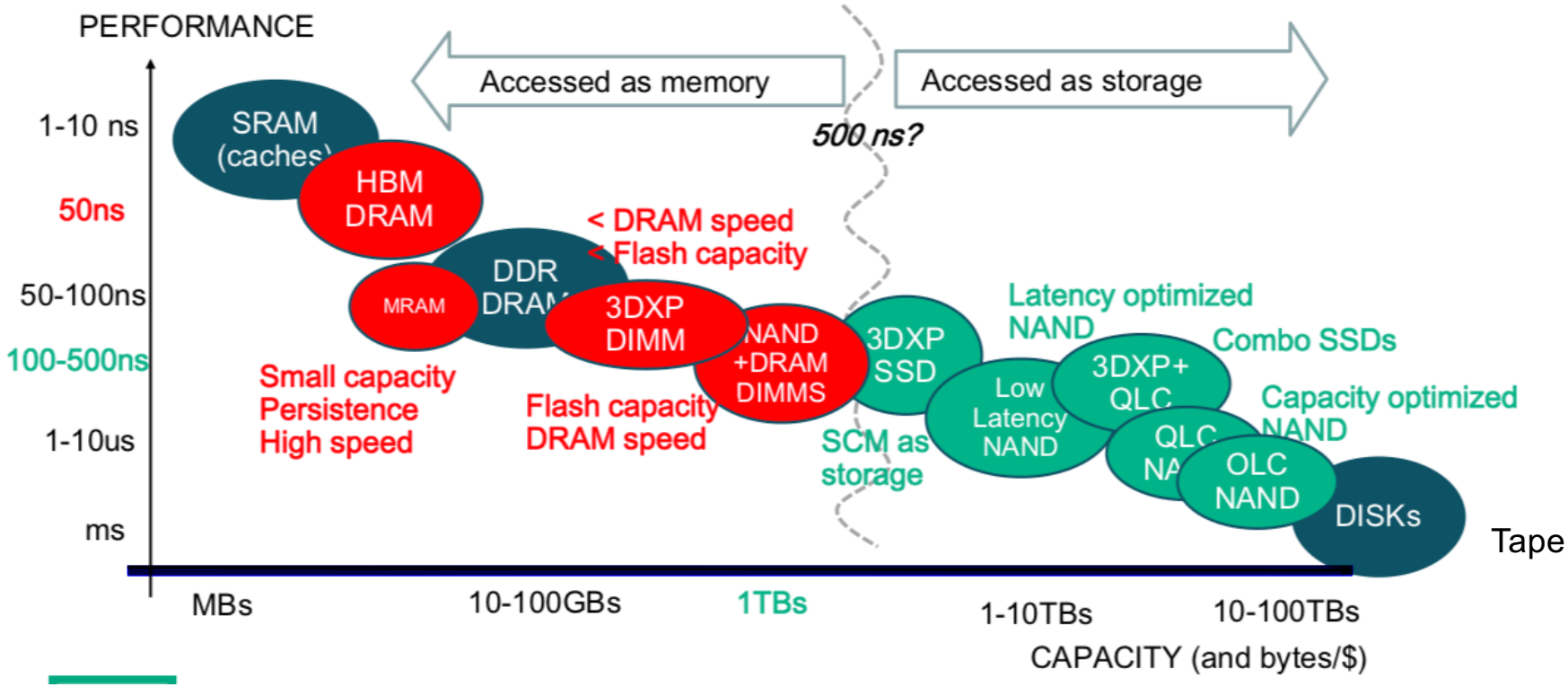
- Flash cost reduction via stacking layers continues
- Cost reduction from QLC with endurance trade-off
- Need breakthrough media to challenge NL-SAS HDDs in $/GB

Low latency media (better latency than TLC NAND Flash)

- As a read cache
- As a "write absorber”
- As network-attached replacement for server-side SSDs

Data Access in Memory / Storage Hierarchy

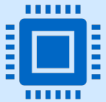
Storage hierarchy choices getting crowded



Source: P. Faraboschi, "The Data Access Continuum", SC'19 MCHPC

Emerging Networking Trends

Network scaling continues upward trend



Link speeds continue upward trend beyond 100 GbE

Switch ports going to 400 GbE, using 4 lanes of 100G SerDes
Public clouds creating faster transitions to next generation
Next step is 800 GbE

Smart NICs with processing capability

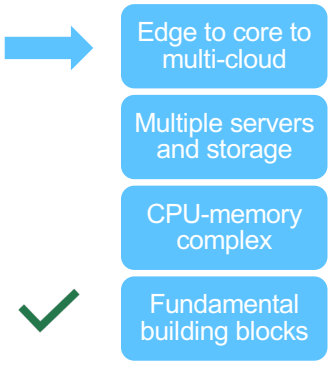
Offer HW offload capability and relatively easy programmability
Direct placement benefits of RDMA on client and storage side
Security features

NVMe-oF for storage connectivity scaling well to 1000's of nodes

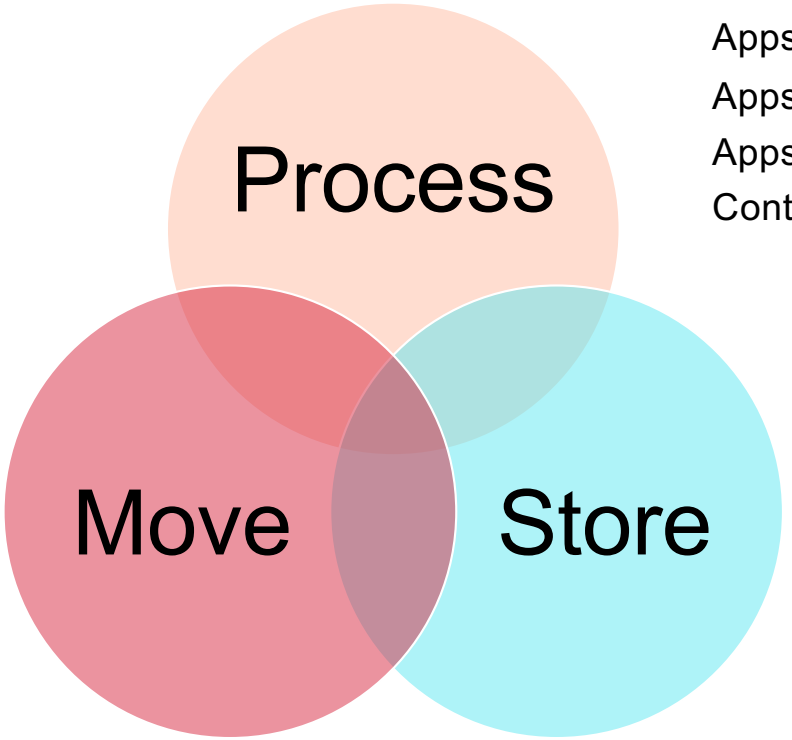
RoCEv2, TCP-IP for most robust transport layer

How data gets processed, moved, and stored across the Edge, Core, Multi-Cloud

How might we find the “best home” for a workload across Edge, Core, and Multi-cloud?



Data fabric to allow movement across edge, core, multi-cloud
Move the right data, to the right place, at the right time!

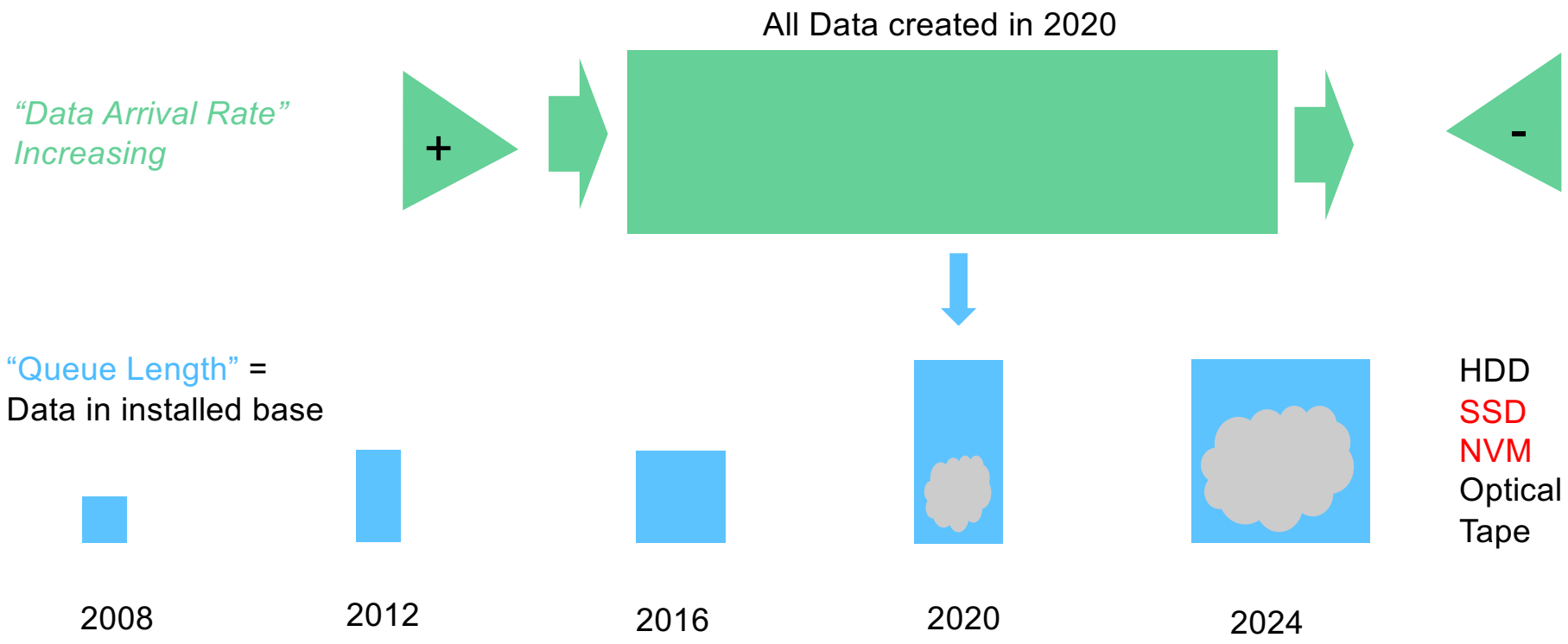


Apps at the edge for real-time
Apps in core DC for security/compliance
Apps in cloud for scale, cloud native
Containers and serverless FaaS

Most data created at the edge
Endpoints, Cloudlets
Core for sovereignty
Increased cloud storage for consume

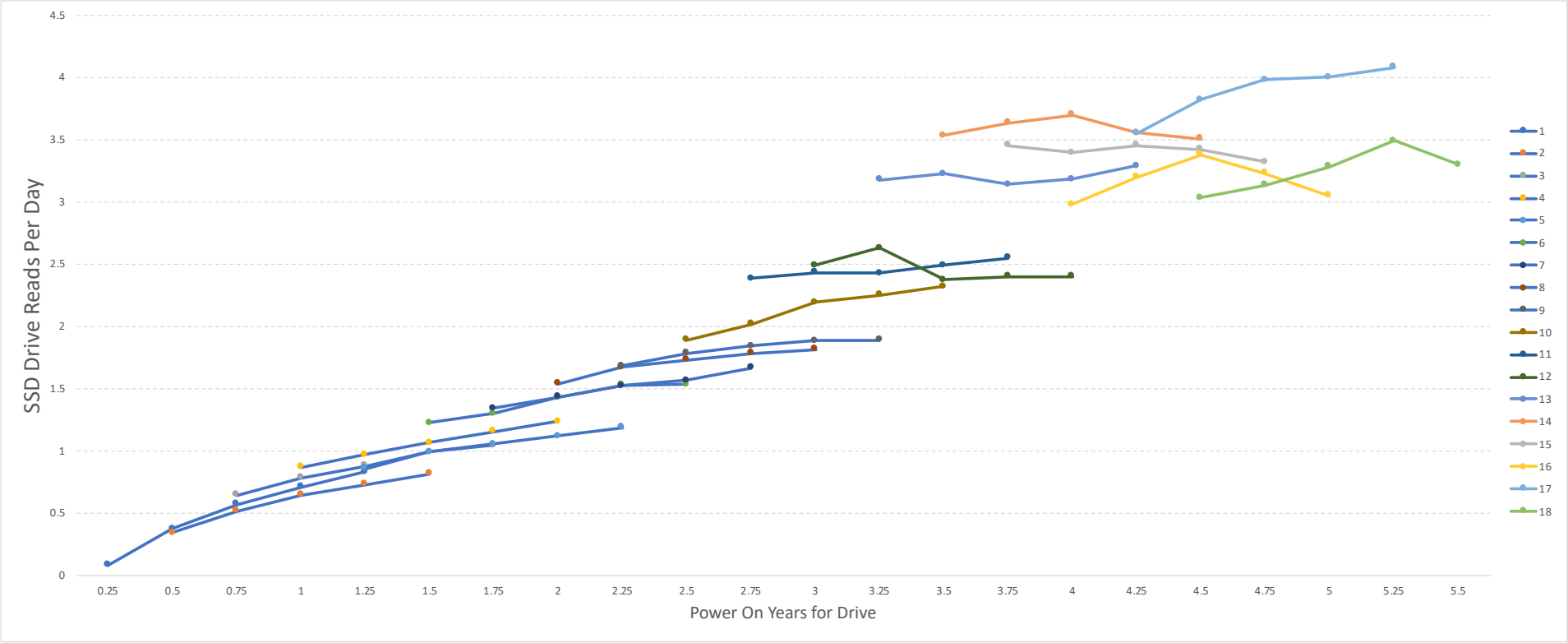
Global Data Creation and Installed Storage

Data growth continues in the amount created and the amount stored



Example: Evolution of “data reads per day” on a set of drives

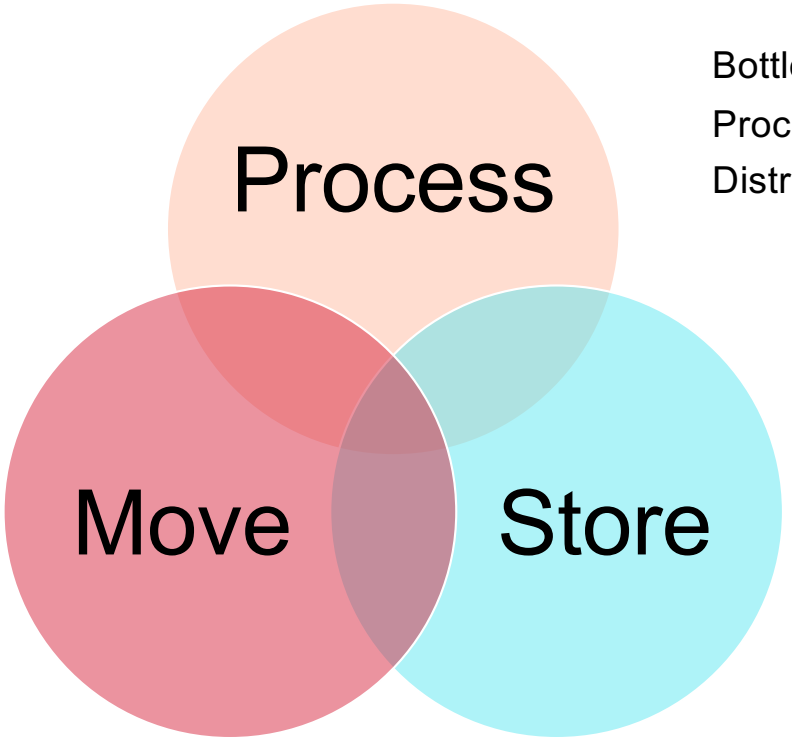
Post-processing the evolution of drives provides useful insights across the population. Real-time analysis of collective drives’ lifetime of experience enables better prediction and decision making.



How data gets processed, moved, and stored across server nodes?

How might we find the right balance of processing, movement and storage for my workload?

- Edge to core to multi-cloud
- Multiple servers and storage
- CPU-memory complex
- Fundamental building blocks



Bottleneck for many workloads

Process in memory / storage / network

Distribute processing across servers

Data movement getting faster

PCIeG4/5 or Ethernet

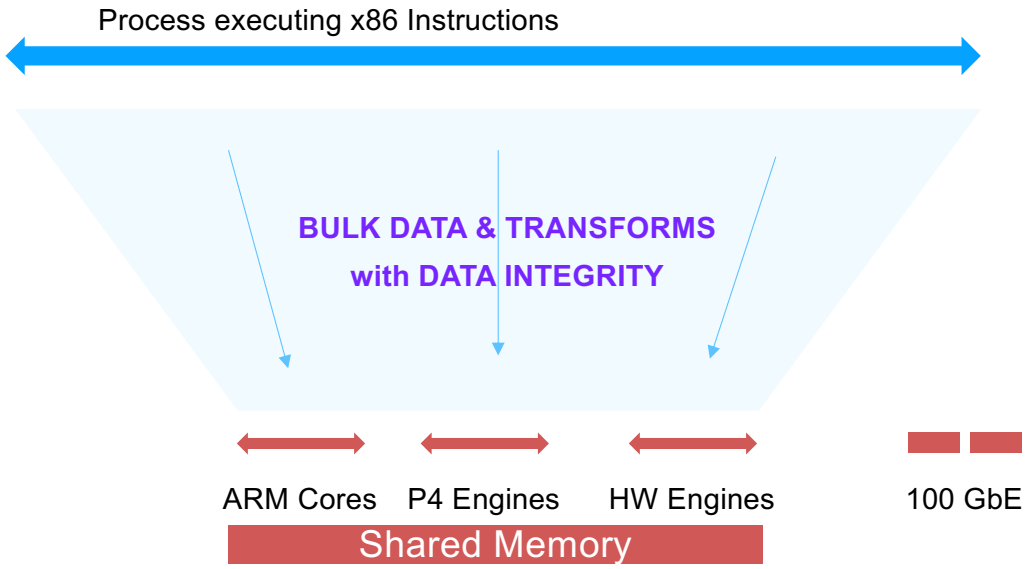
Less movement saves energy

Rapid shift to SSDs for high IOPS

Unlock internal memory/SSD BW

What Type of Processing is being Offloaded from x86 CPU?

Pensando Distributed Services Card: 100 GbE Networking and easy programmable offloads, but with ASIC-like speeds



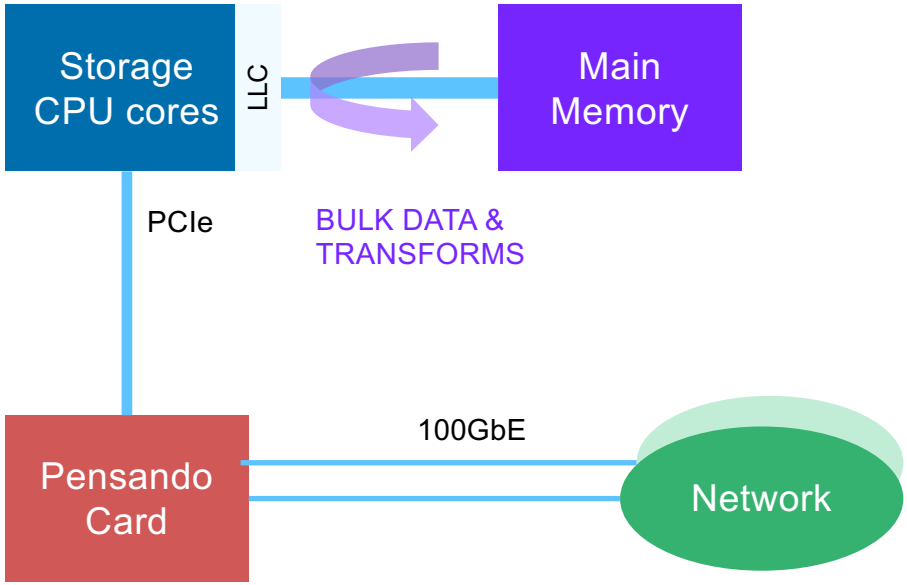
x86 CPU

Storage Controller

Pensando
PCIe Card

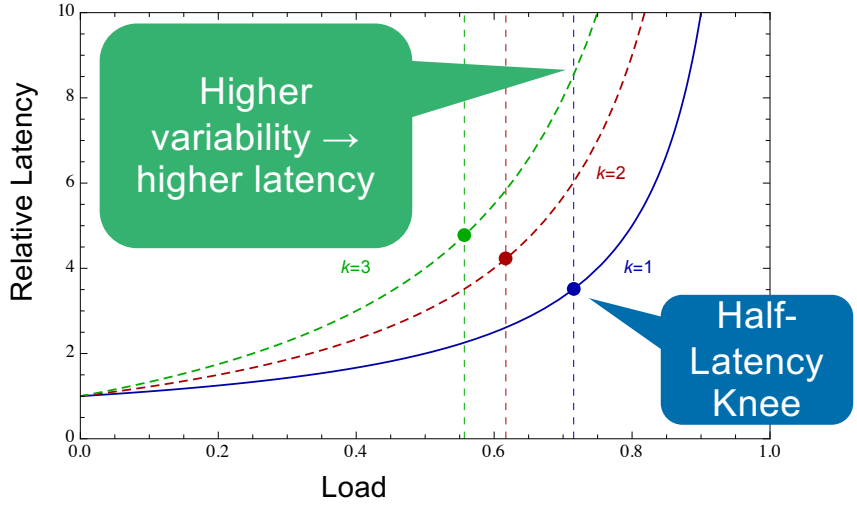
How does the Processing Offload to Pensando help Performance?

Moving Bulk Data & Transforms from Main Memory to Pensando card: Reduces CPU usage and latency

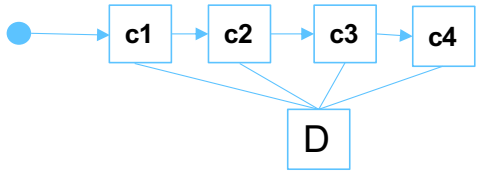


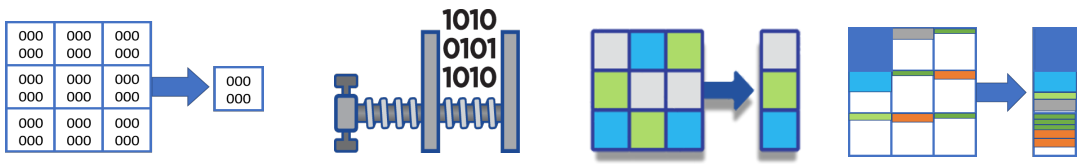
Offloading Bulk Data Transforms:

- Saves CPU cycles directly
- Reduces memory BW
- Helps maintain consistent latency



Y-axis, Vertical Scaling: Offload “Bulk Data & Transforms” for Storage





- Storage Efficiency
  - Compression, Compaction
  - Deduplication / Hash
- Data Protection
  - Checksums
  - RAID operations
- Security computation
  - Encrypt data at rest
  - Encrypt data in motion
- Networking
  - NIC, RDMA

“Data touch” / CPU-intensive functions move to specialized HW on “Fabric”

⇒ Pensando’s card with P4+ engines, HW engines, and ARM cores

Benefits compound via ONTAP software …

⇒ Primary data snapshots, clones on box

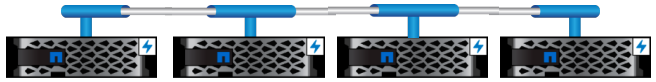
⇒ Preserve storage efficiency for secondary data services

⇒ Tiering data to Public Clouds

Z-Axis, Scale-out for High-Tech Industry Applications

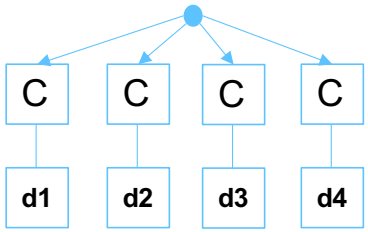
FlexGroup on A400 with Pensando for clustering: a scalable, high-performance data container

Apps for electronic design automation,
high-tech, oil and gas,
media and entertainment



Single namespace

z-axis



Linear scale for performance and capacity

Operational simplicity

- Single mount point with automated load and space distribution

Consistent high performance

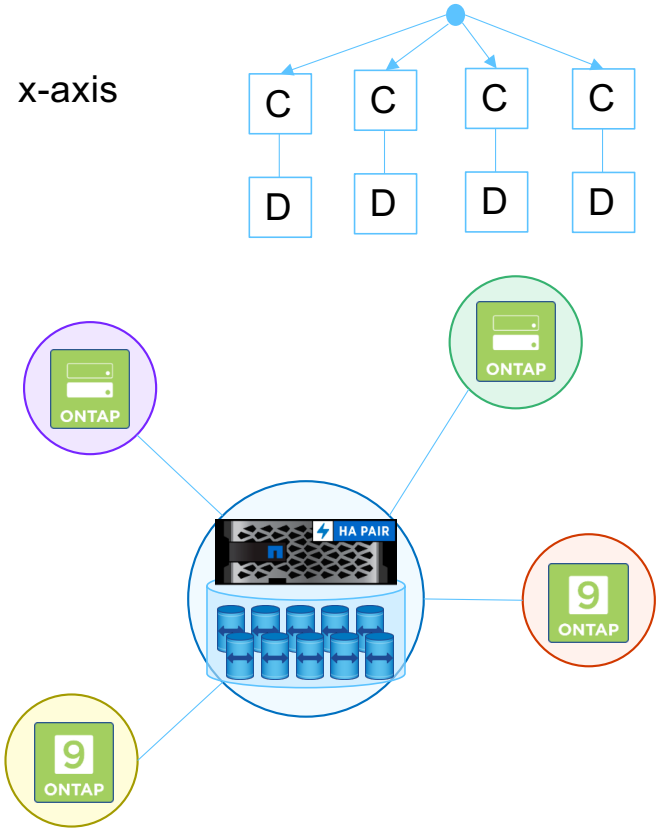
- Predictable, consistent low latency
- All-flash containers

High resiliency

- NetApp® ONTAP® nondisruptive operations

X-axis, Horizontal Scaling - Accelerate Your Global Namespace

FlexCache volumes distributes hot data over Pensando cluster ports



NetApp® FlexCache® volumes

- Sparsely populated volumes that can be cached on the same cluster or a different cluster as the origin volumes to accelerate data access

Performance acceleration for hot volumes

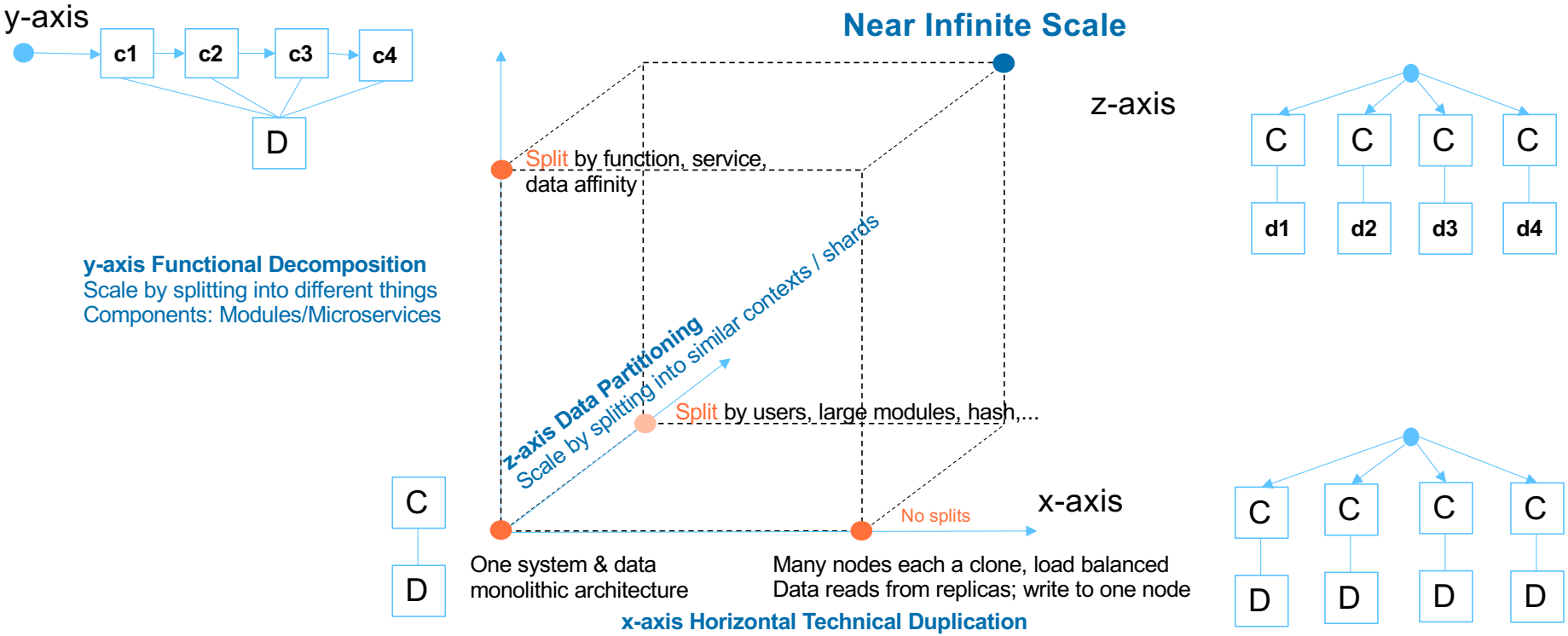
- Cache read and metadata for CPU-intensive workloads
- Provides different mount points to avoid hot volumes
- Cache data within the cluster (intra-cluster)

Cross data center data distribution

- Cache across multiple data centers to reduce WAN latencies
- Bring data closer to compute and users
- Between Netapp AFF, FAS, or ONTAP Select systems

Scale in All 3 Dimensions – System & Data

Scalability = Ability to handle a growing amount of work (CPU and Data)

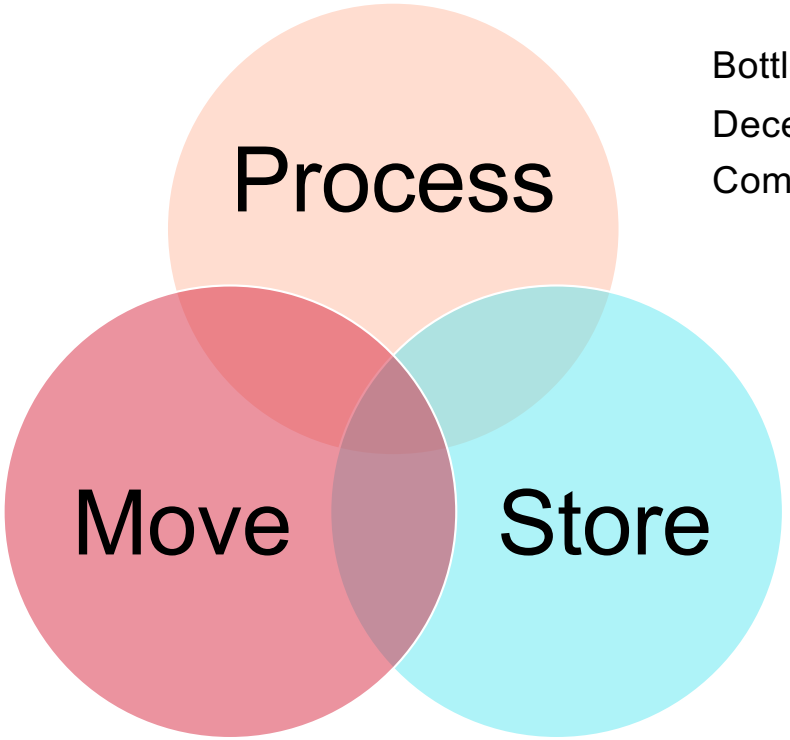


Source: the Scale Cube http://theartofscalability.com/

How data gets processed, moved, and stored inside the CPU-Memory Complex

How might we get more performance and energy efficiency out of the CPU and memory?

- ✓ Edge to core to multi-cloud
- ✓ Multiple servers and storage
- → CPU-memory complex
- ✓ Fundamental building blocks



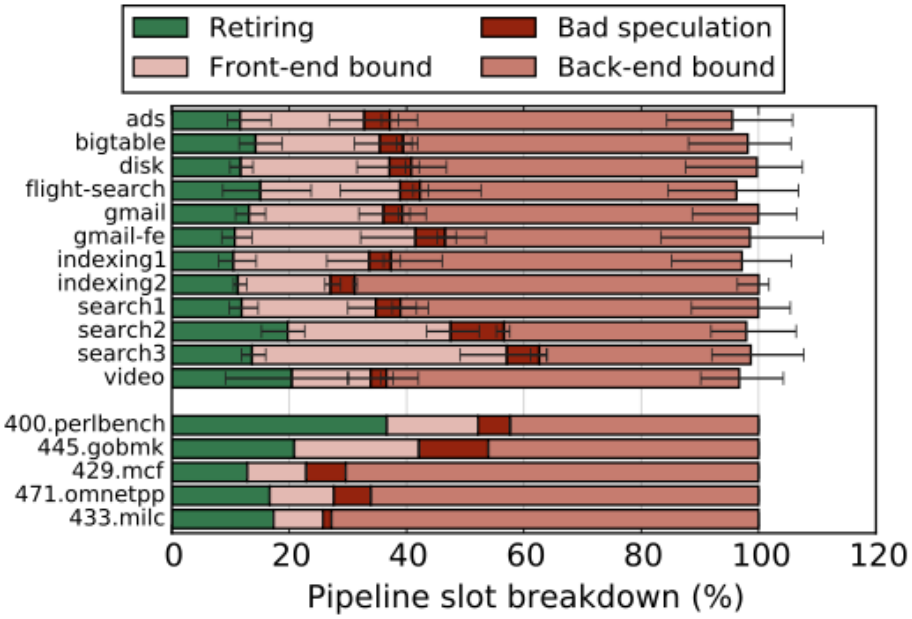
Bottleneck for many workloads
Decentralize processing
Computational memory

Need energy focus
Data hops are “hidden”
Perf and energy metrics

Memory bound by latency
Persistent memory for capacity
CXL for large memory spaces

CPU Pipeline is stalled for >80% of the time for many workloads!

"100% CPU Utilization" hides the underlying bottlenecks – cache hierarchy and memory latency.



Source: S. Kanev et al. Profiling a warehouse-scale computer, ICSA'15

- Many data center workloads have low instructions per cycle
  - Back-end bound → waiting to fill caches / memory latency
  - Dependent data accesses → serial memory accesses
- Characteristics of data intensive workloads
  - Request and response over the network
  - Simultaneously access many contexts (fan in)
  - Areas of "tax": Protocol buffers, RPCs, memory moves, compression, memory allocation, hashing
- Memory latencies going higher with bigger sockets
  - Moving these "data intensive" functions away from CPU to ethernet fabric-attached offload HW
  - Move processing into memory for better latency and energy efficiency
  - Symmetric Multi-Threaded cores

$M^B/G/1$ Energy-Performance Queue with Batched Arrivals

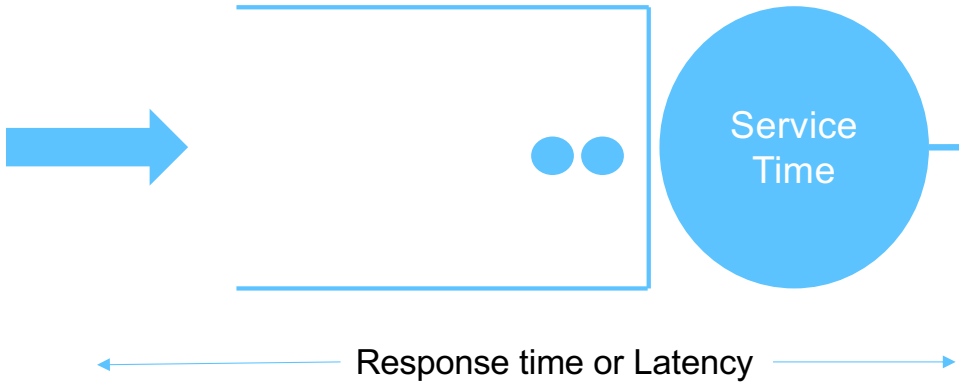
Example: EP Queue + Reinforcement Learning to make power usage choices (and service times) with energy constraints

Random Arrivals with rate λ of **batched requests**

Specified batch size distribution (Pr(1), Pr(2), Pr(3),…)

Specify *n* **general service time distributions**

- Different for each queue depth (1, 2, 3,…,*n*)
- Same service time distribution for queue length *n*, *n*+1,..



System powers-down when queue is empty (power-down time distribution)

System powers-up on arrival of new item to empty queue (power-up time distribution)

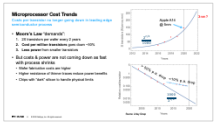
MODEL INPUT: arrival rate, 1 discrete distribution, n+2 continuous distributions (n service times, POWER-UP, and POWER-DOWN times) and power usage for ON, OFF.

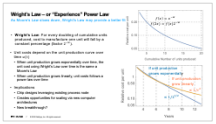
MODEL OUTPUT: **response time distribution** (means, variance, and higher moments), **energy usage**

Summary

Layers × Lens

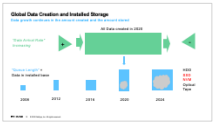
Tech Trends





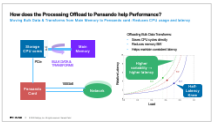
CRISIS

Global Data



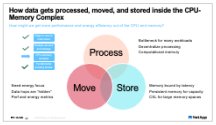
AI-infusion

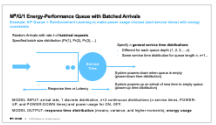
Servers/Storage



Renaissance

CPU complex



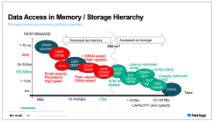


Performance & Energy metrics

Process

Move

Store



Scale Cube