Hewlett Packard Enterprise

The Machine – Memory Driven Computing

Sharad Singhal

1st Workshop on Resource Disaggregation, April 13, 2019

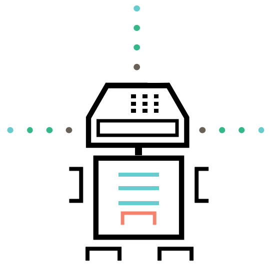
Outline

- Motivation for Memory-Driven Computing
- Initial experiences with Memory-Driven Computing
  - The Machine
  - How Memory-Driven Computing benefits applications
- Commercialization of Memory-Driven Computing
- Challenges for programming Memory-Driven Computing
- Summary

Hewlett Packard Enterprise

What’s driving the data explosion?

Record	Engage	Act



Electronic record of event	Interactive apps for humans	Machines making decisions
Ex: banking	Ex: social media	Ex: smart and self-driving cars
Mediated by people	Interactive	Real time, low latency
Structured data	Unstructured data	Structured and unstructured data

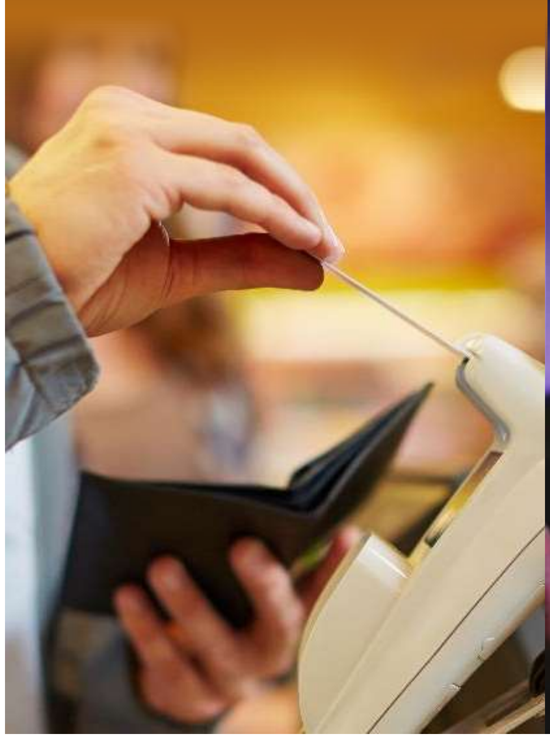
Hewlett Packard Enterprise

More data sources and more data

Record

40 petabytes

200B rows of recent transactions for Walmart's analytic database (2017)



Engage

4 petabytes a day

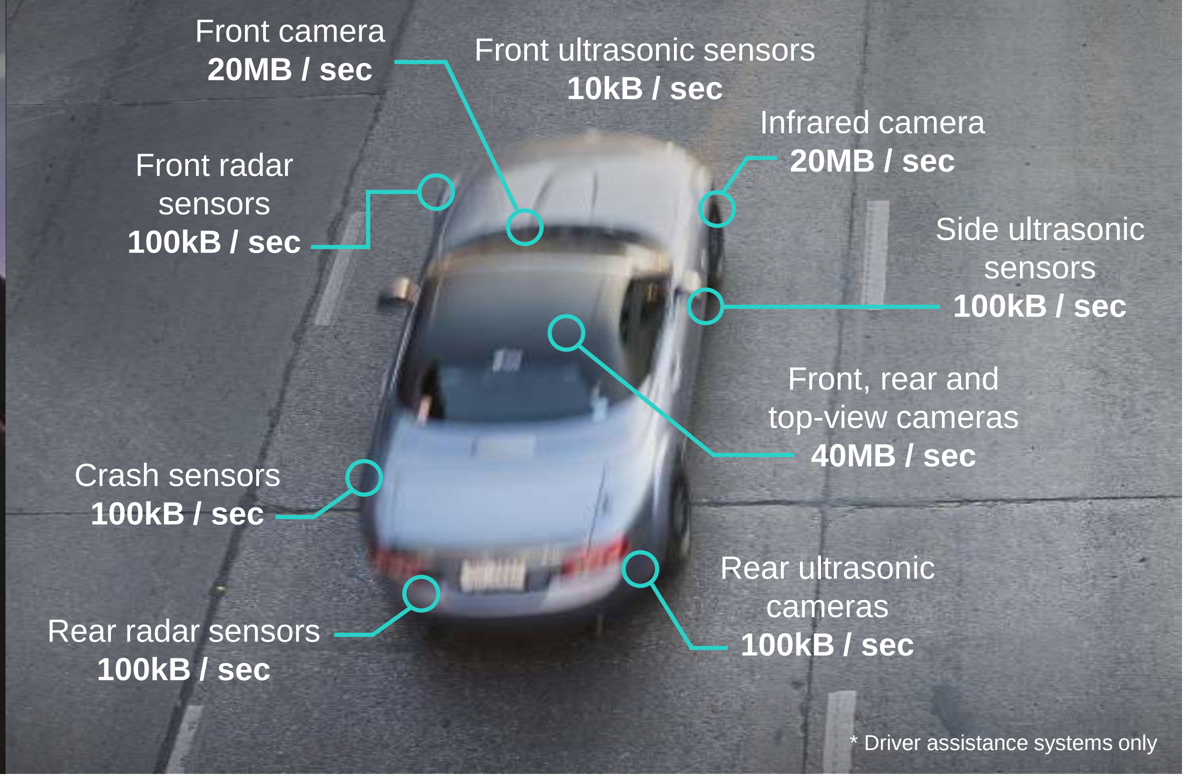
Posted daily by Facebook's 2 billion users (2017)

2MB per active user

Act

40,000 petabytes a day*

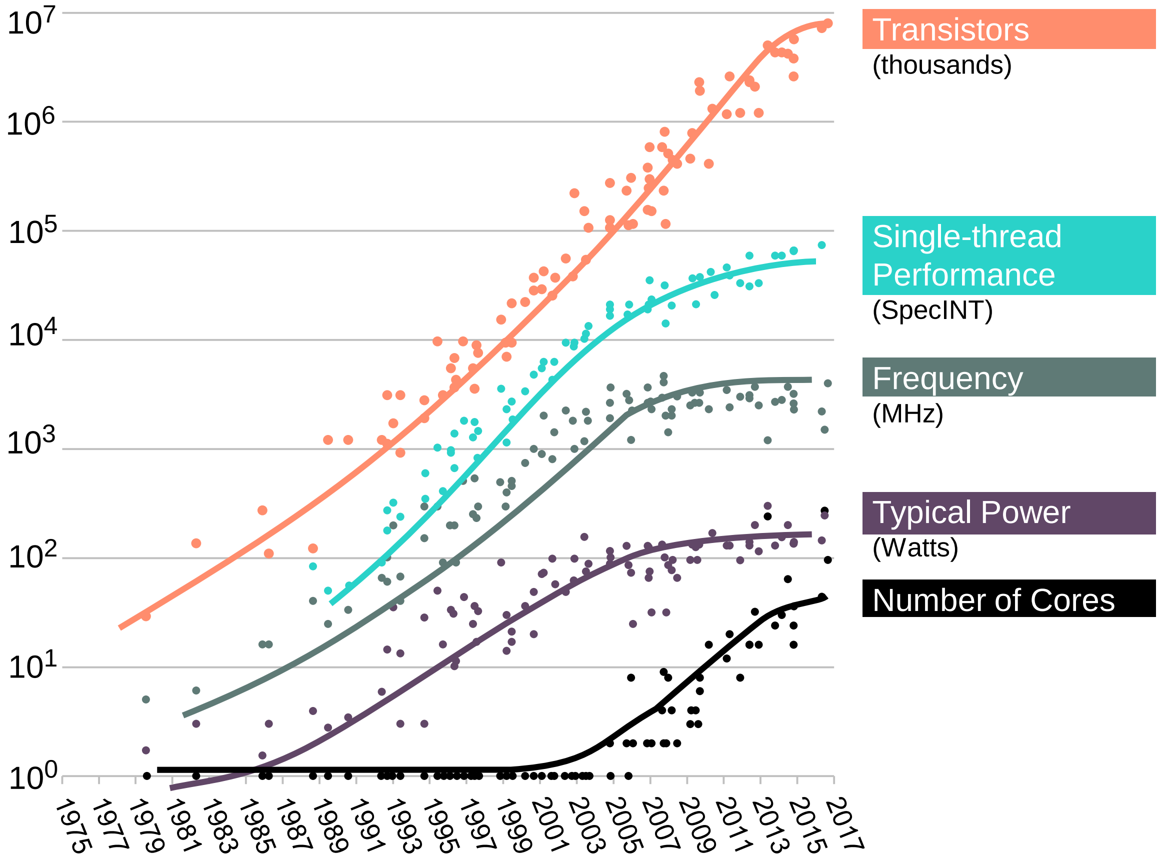
4TB daily per self-driving car
10M connected cars by 2020



* Driver assistance systems only

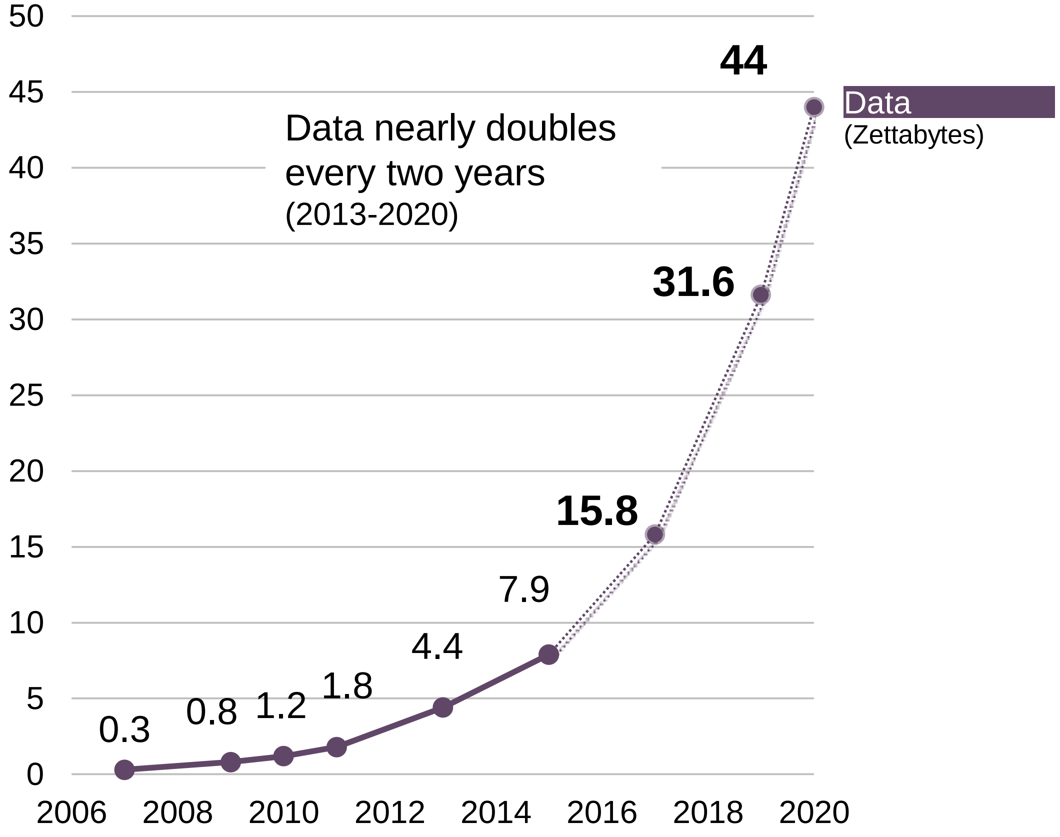
The New Normal: Compute is not keeping up

Microprocessors



Source: K. Rupp. 42 Years of Microprocessor Trend Data

Data growth



Source: Data Age 2025 study, sponsored by Seagate, April 2017

Hewlett Packard Enterprise

The New Normal: Interconnects are not keeping up

East-West traffic is exploding, driving a need for much higher bandwidth…

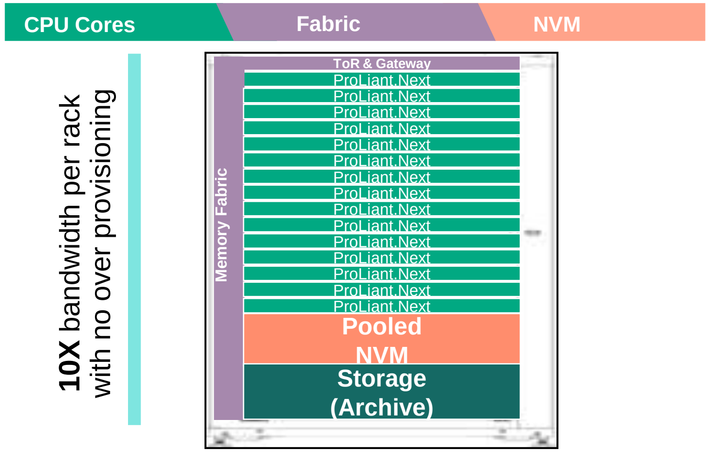
Hyperscale

Facebook Network Traffic[1]

Increasingly complex, AI driven workloads drive significant machine-to-machine traffic to deliver richer customer experiences and user value

… and enterprises are seeking rack-scale architectural composability

Enterprise

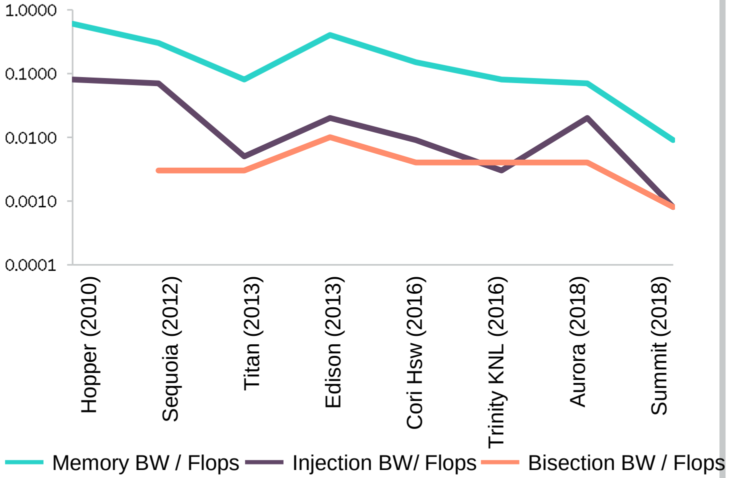


Requires significant bandwidth across the rack to dynamically compose resources

… and in HPC systems, bandwidth is not keeping up with compute

HPC

Bandwidth / Flops (Exascale Systems)[2]



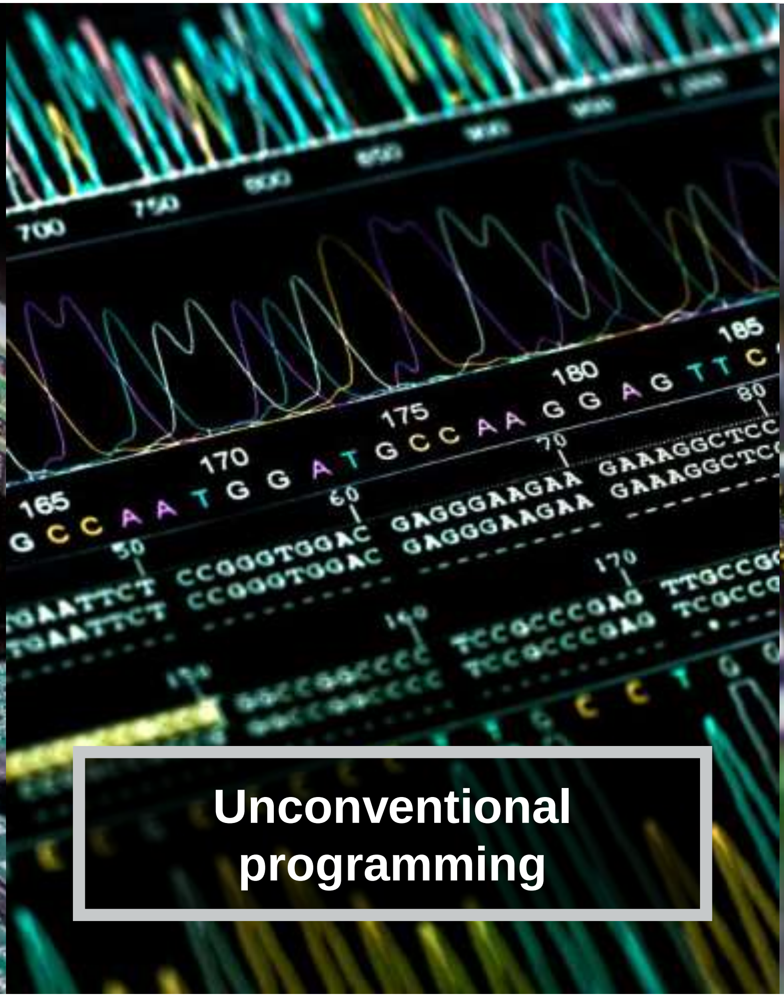
New solutions are required to meet emerging performance demands

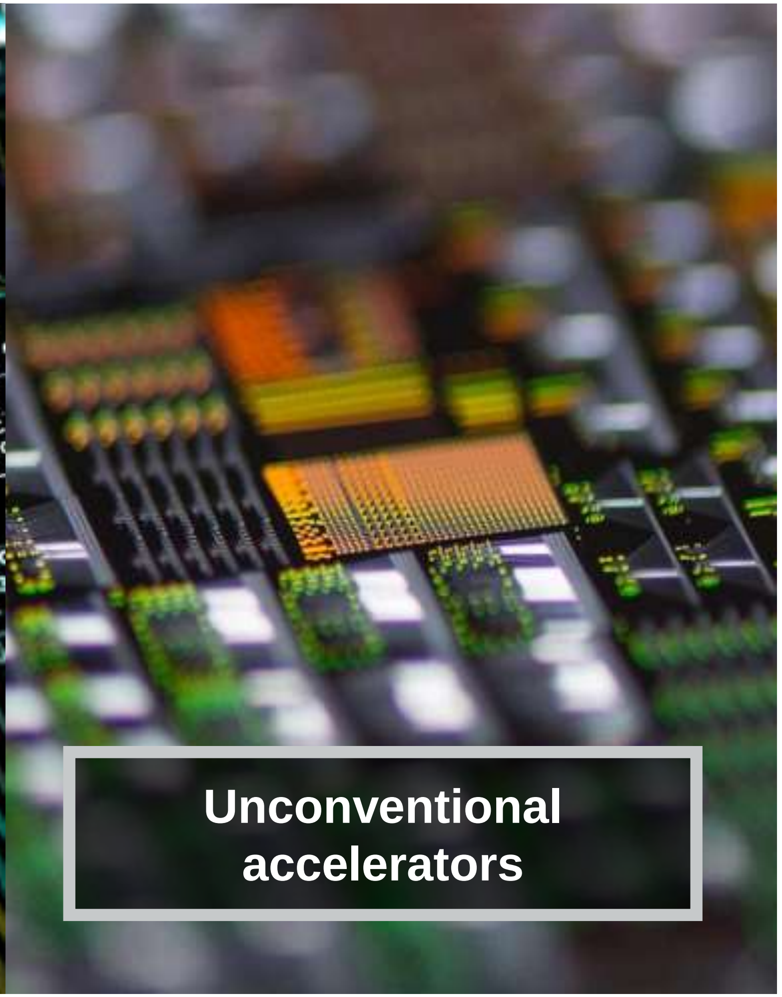
Hewlett Packard Enterprise

Source: 1) Katherine Schmitdke, Facebook; 2) HPC Team

We are radically rethinking our approach to computing

Advancing computing without relying on Moore’s Law

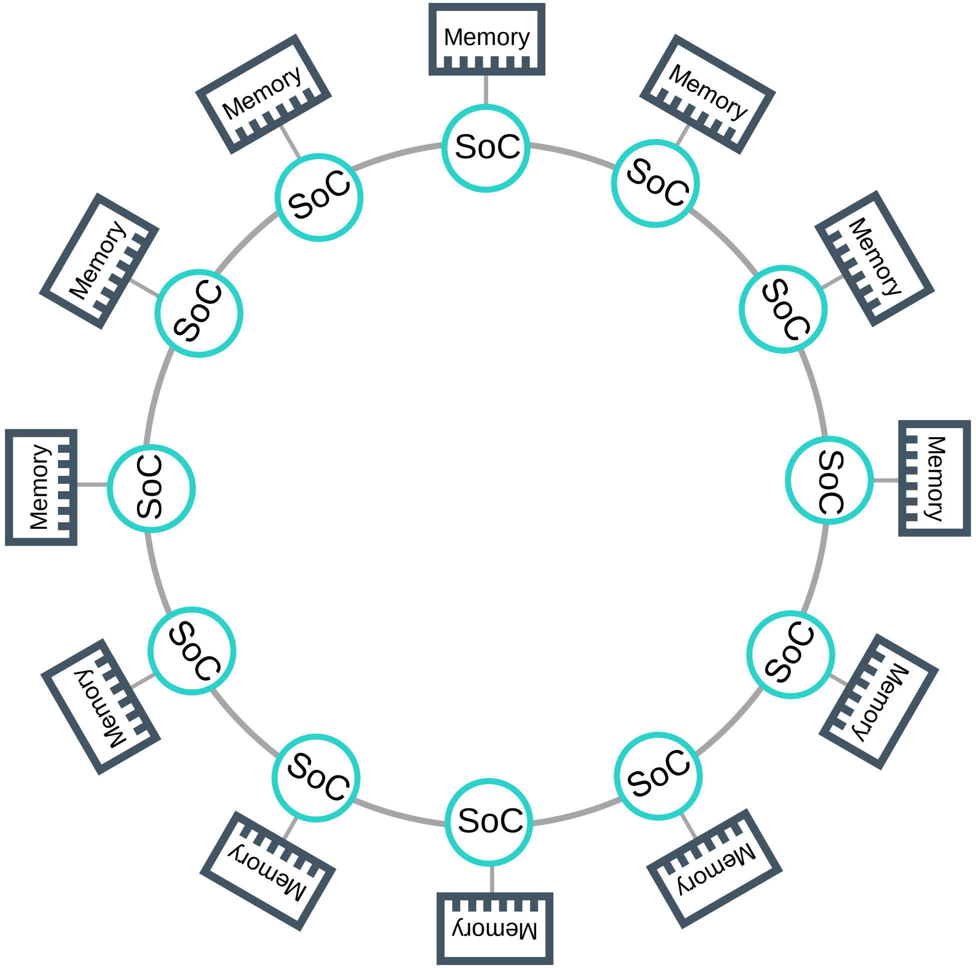




Hewlett Packard Enterprise

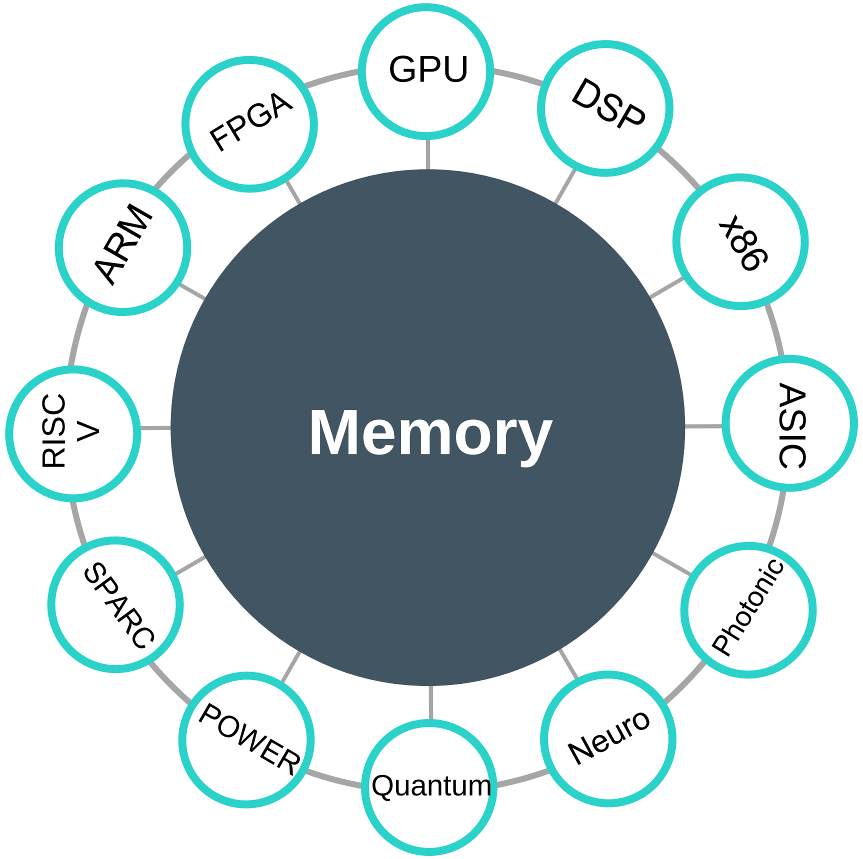
Today's architecture

From processor-centric computing

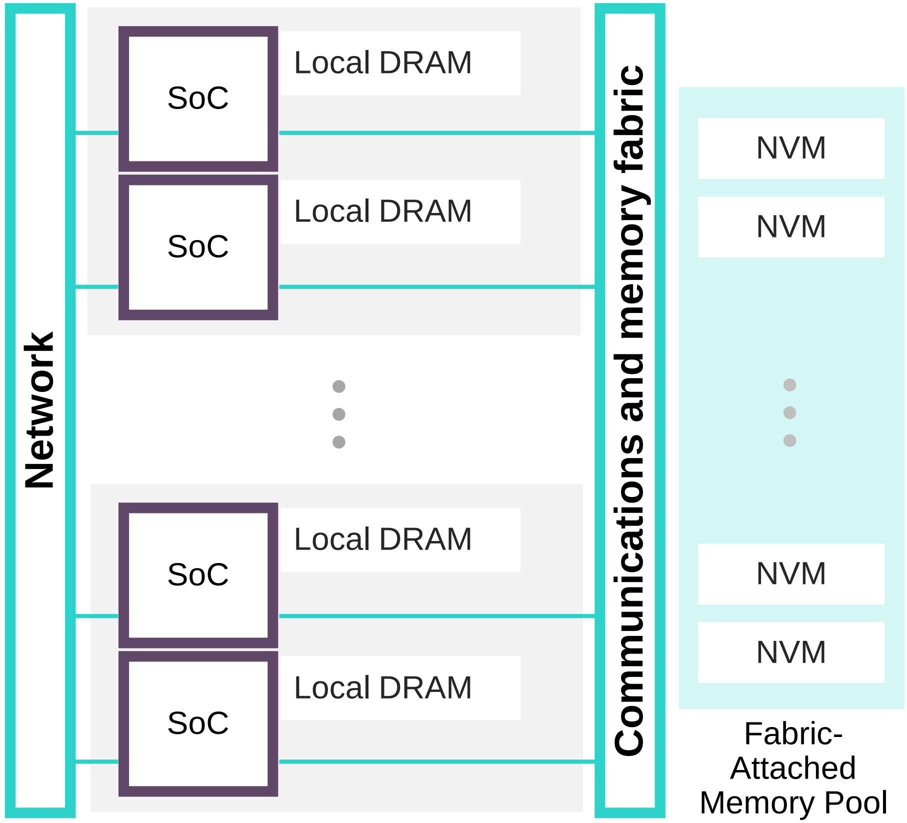


Future architecture

Memory-Driven Computing



Fabric-attached memory (FAM) architecture



- **Convergence of memory and storage**
  - Byte-addressable non-volatile memory accessible via memory operations
  - Local volatile memory provides lower latency, high performance tier
- **High capacity disaggregated memory pool**
  - Fabric-attached memory pool is accessible by all compute resources
  - Low diameter networks provide near-uniform low latency
- **Distributed heterogeneous compute resources**
  - Enables mix of processors to work together
- **Software**
  - Memory-speed persistence
  - Direct, unmediated access to all fabric-attached memory across the memory fabric
  - Non-coherent concurrent accesses and data sharing by compute nodes

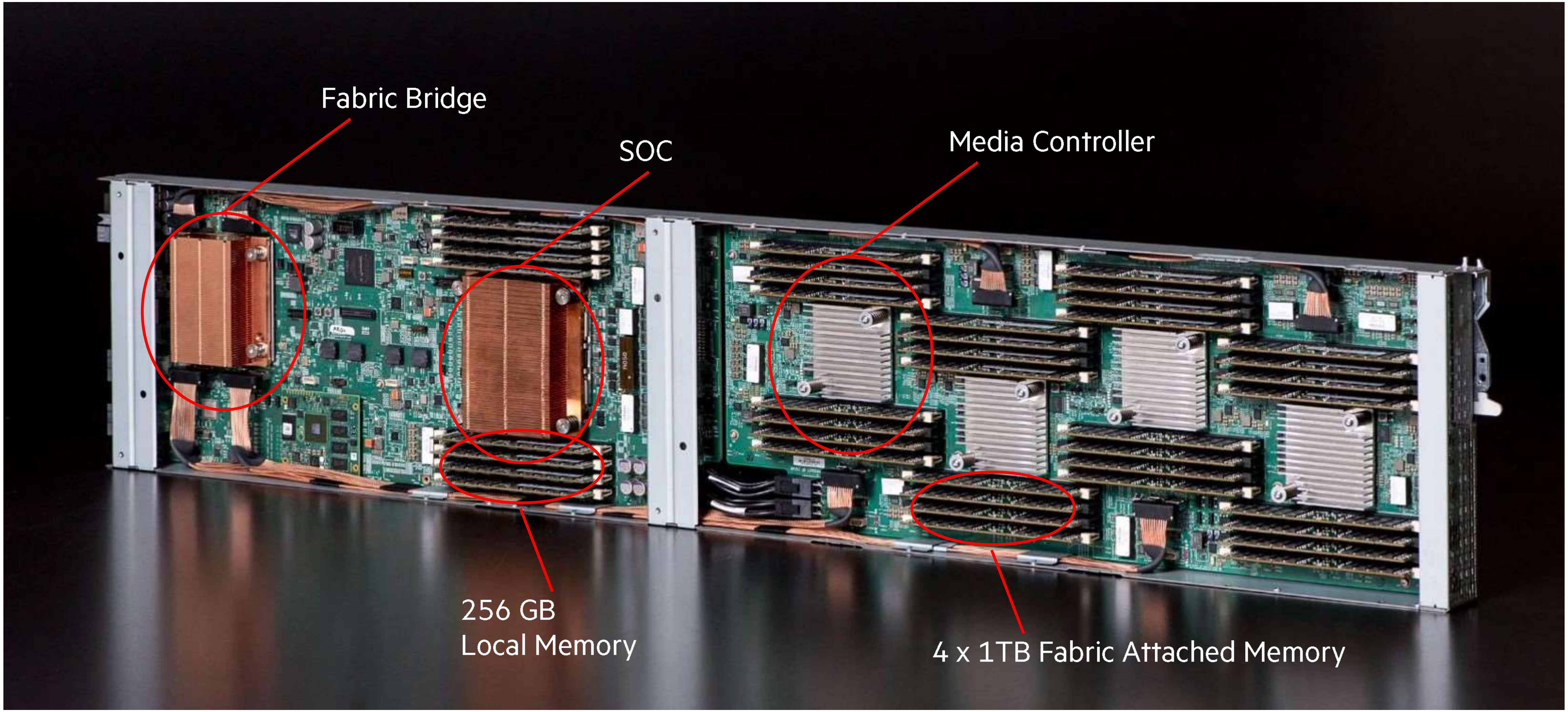
Hewlett Packard
Enterprise

The Memory Fabric Testbed – The Machine

- **The Machine** prototype (May 2017)
- 160 TB of fabric-attached, shared memory
- 40 compute nodes
  - ARM-based Cavium ThunderX2 SoC
  - 256 GB node-local memory
  - Optimized Linux-based operating system
- High-performance fabric
  - Photonics/optical communication links with electrical-to-optical transceiver modules
  - Protocols are early version of Gen-Z
- Software stack designed to take advantage of abundant fabric-attached memory

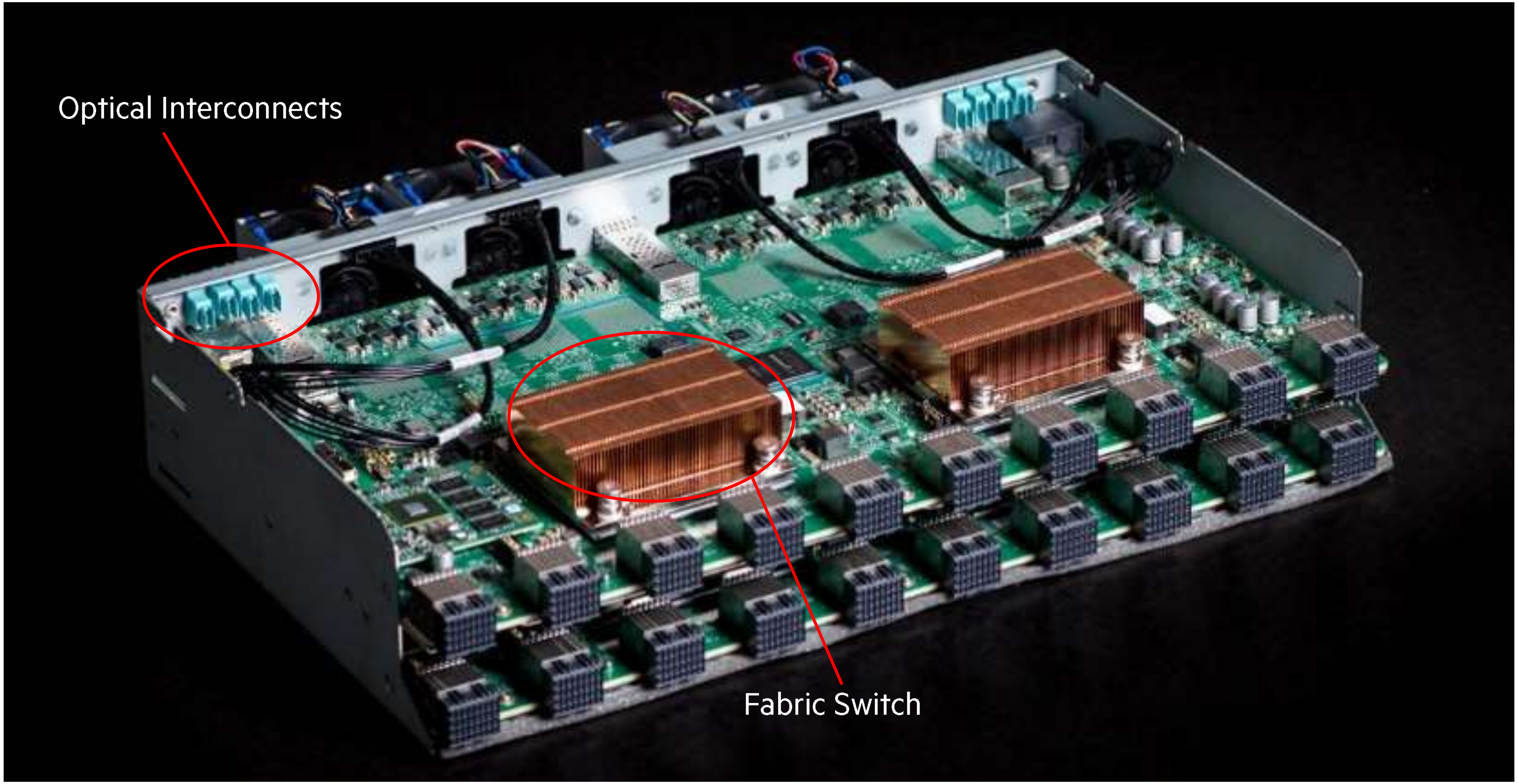
https://www.nextplatform.com/2017/01/09/hpe-powers-machine-architecture/

Hardware design: Memory fabric testbed

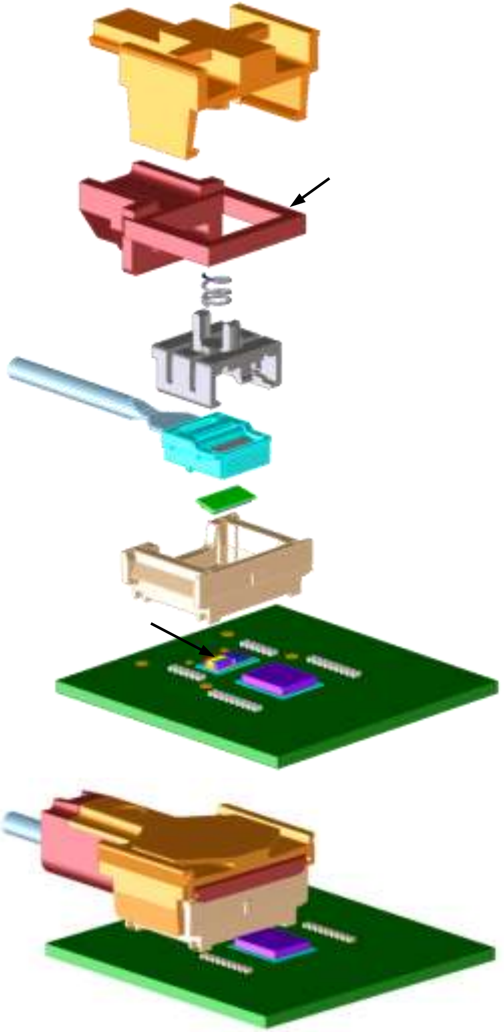


Hewlett Packard Enterprise

Hardware design: Memory fabric testbed



The Machine Program: Memory fabric testbed



Transform performance with Memory-Driven programming

Modify existing frameworks

In-memory analytics

15x

faster

Similarity search

40x

faster

New algorithms

Large-scale graph inference

100x

faster

Completely rethink

Financial models

10,000x

faster

Hewlett Packard Enterprise

Large in-memory processing for Spark

Spark with Superdome X

Spark

– Our approach:

– In-memory data shuffle

– Off-heap memory management

  – Reduce garbage collection overhead

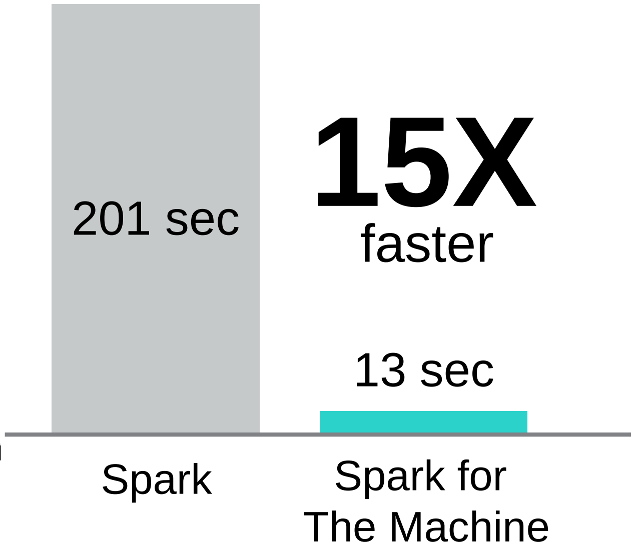
  – Exploit large NVM pool for data caching of per-iteration data sets

– Use case: predictive analytics using GraphX

– Superdome X: 240 cores, 12 TB DRAM

M. Kim, J. Li, H. Volos, M. Marwah, A. Ulanov, K. Keeton, J. Tucek, L. Cherkasova, L. Xu, P. Fermando, “Sparkle: optimizing Spark for large memory machines and analytics,” *Proc. SOCC*, 2017.
https://github.com/HewlettPackard/sparkle
https://github:com/HewlettPackard/sandpiper

Dataset 1: web graph
101 million nodes
1.7 billion edges



Dataset 2: synthetic
1.7 billion nodes
11.4 billion edges

Spark for The Machine: 300 sec
Spark: *does not complete*

Hewlett Packard
Enterprise

Memory-Driven Monte Carlo (MC) simulations

Traditional

Step 1: Create a parametric model $y = f(x_1,\dots,x_k)$
Step 2: Generate a set of random inputs
Step 3: Evaluate the model and store the results
Step 4: Repeat steps 2 and 3 many times
Step 5: Analyze the results

Memory-Driven

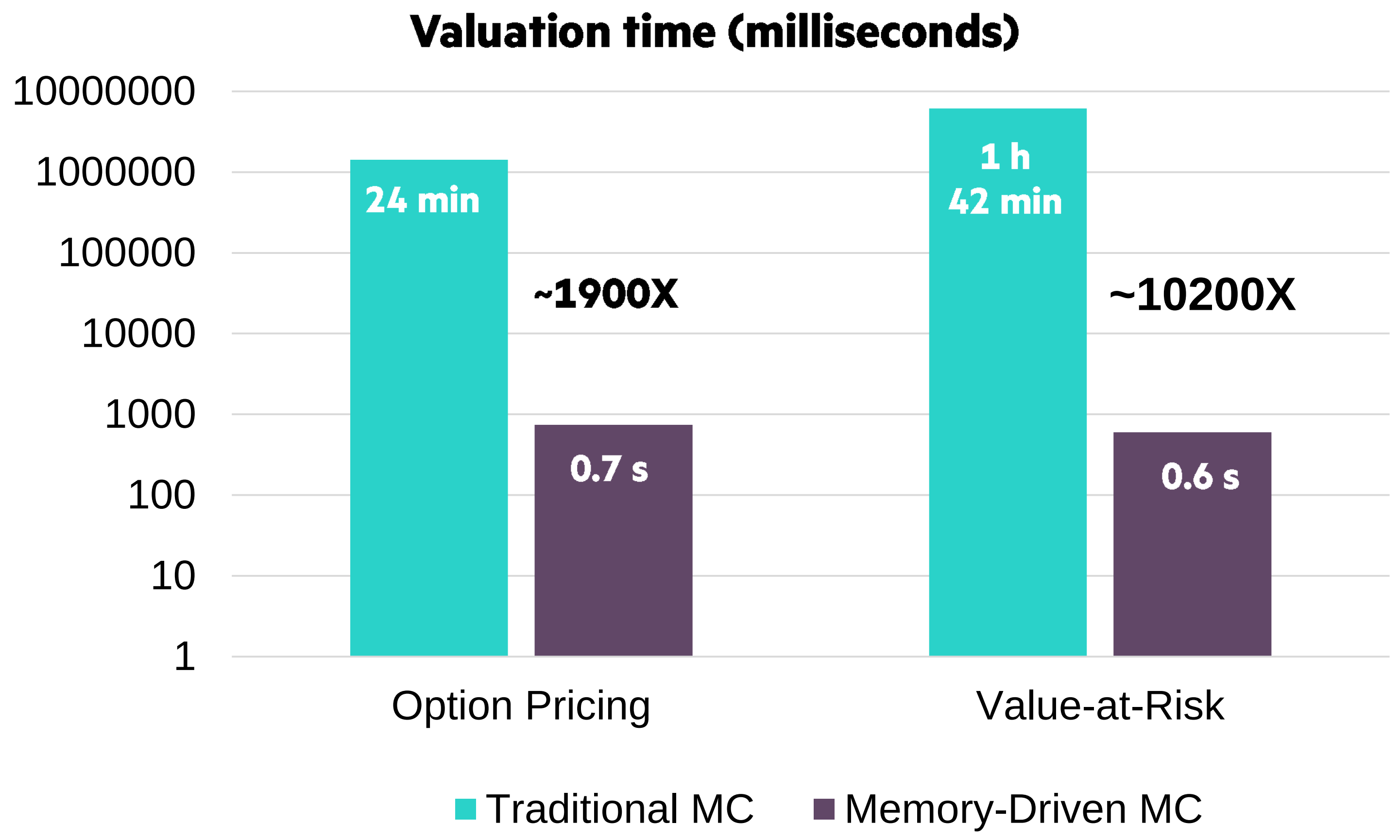
Replace steps 2 and 3 with look-ups, transformations

- Pre-compute representative simulations and store in memory
- Use transformations of stored simulations instead of computing new simulations from scratch

Hewlett Packard Enterprise

Experimental comparison: Memory-driven MC vs. traditional MC

Speed of option pricing and portfolio risk management



Option pricing

Double-no-Touch Option with 200 correlated underlying assets

Time horizon (10 days)

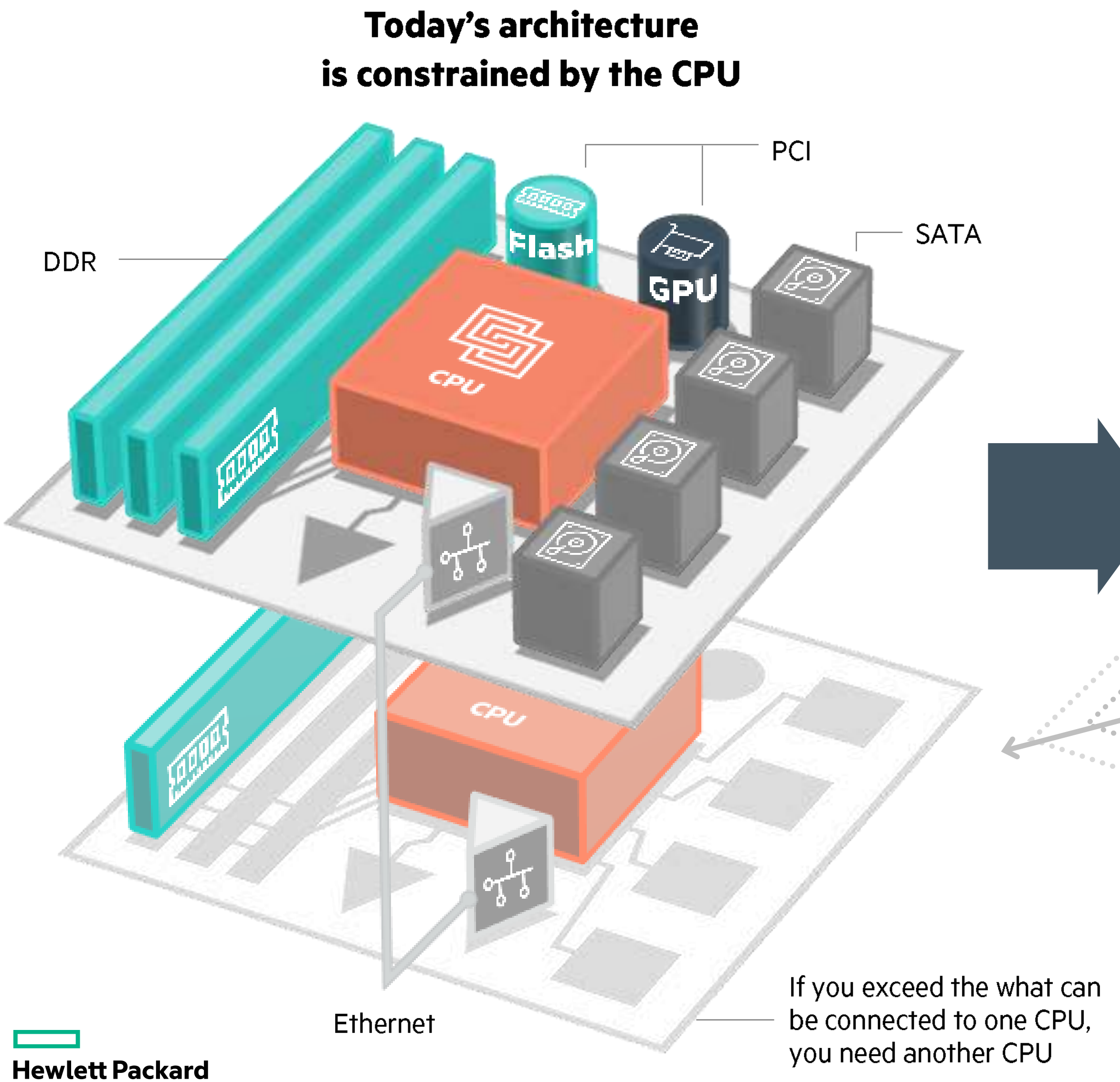
Value-at-Risk

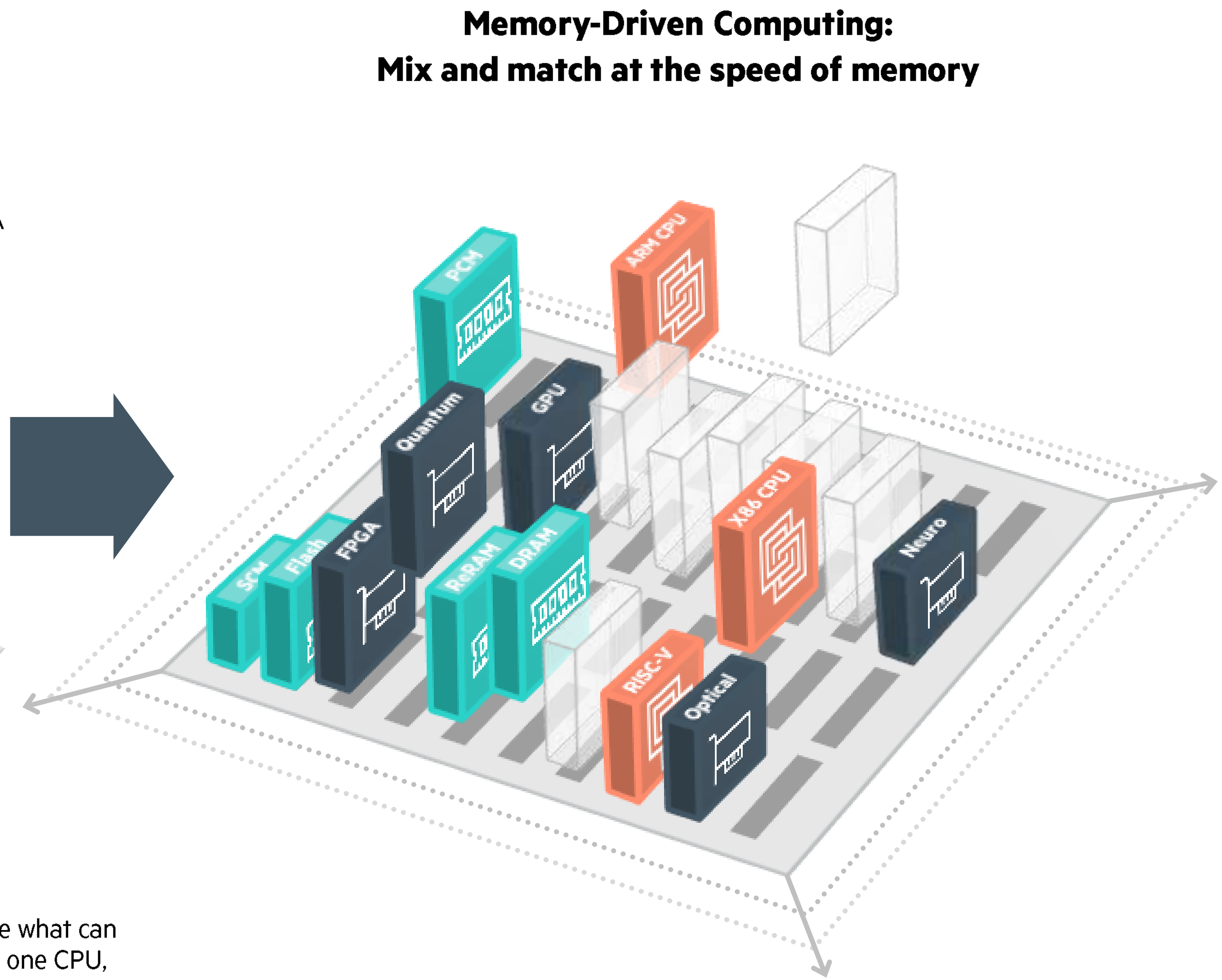
Portfolio of 10000 products with 500 correlated underlying assets

Time horizon (14 days)

Hewlett Packard Enterprise

Traditional vs. Memory-Driven Computing architecture

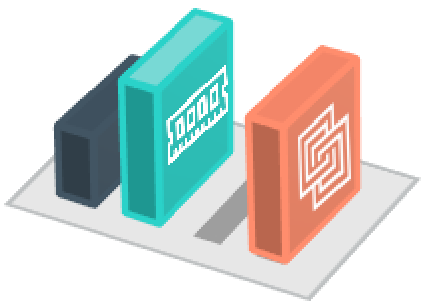




Hewlett Packard
Enterprise

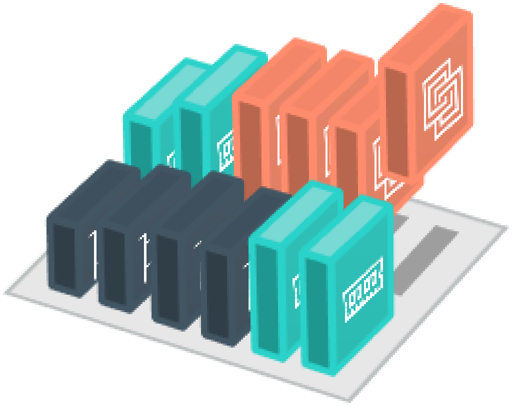
Memory-Driven Computing is the future for every kind of computing

Edge device



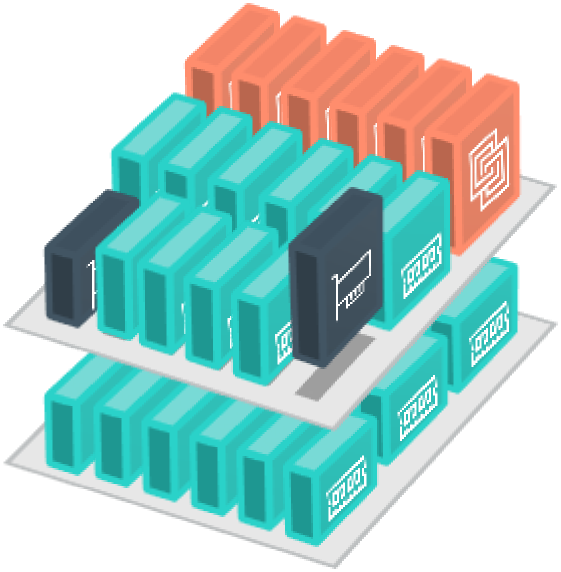
- Near-zero power
- Persistent memory
- AI task-specific accelerator

Cloud infrastructure



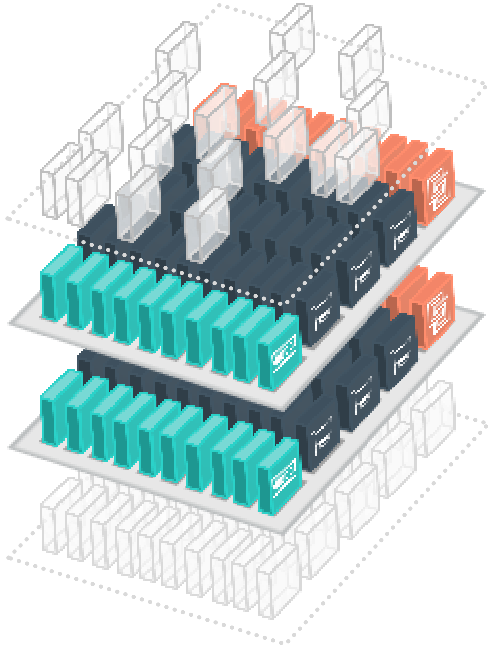
- Composable infrastructure from every edge to any cloud
- Microservices in microseconds at massive scale

Big memory machine



- High-performance data analytics
- Large shared memory
- Monte Carlo, graph analytics applications, etc.

Exascale computer



- 100,000+ components
- Ultra-fast message passing and checkpointing
- 20x more energy-efficient than state-of-the-art

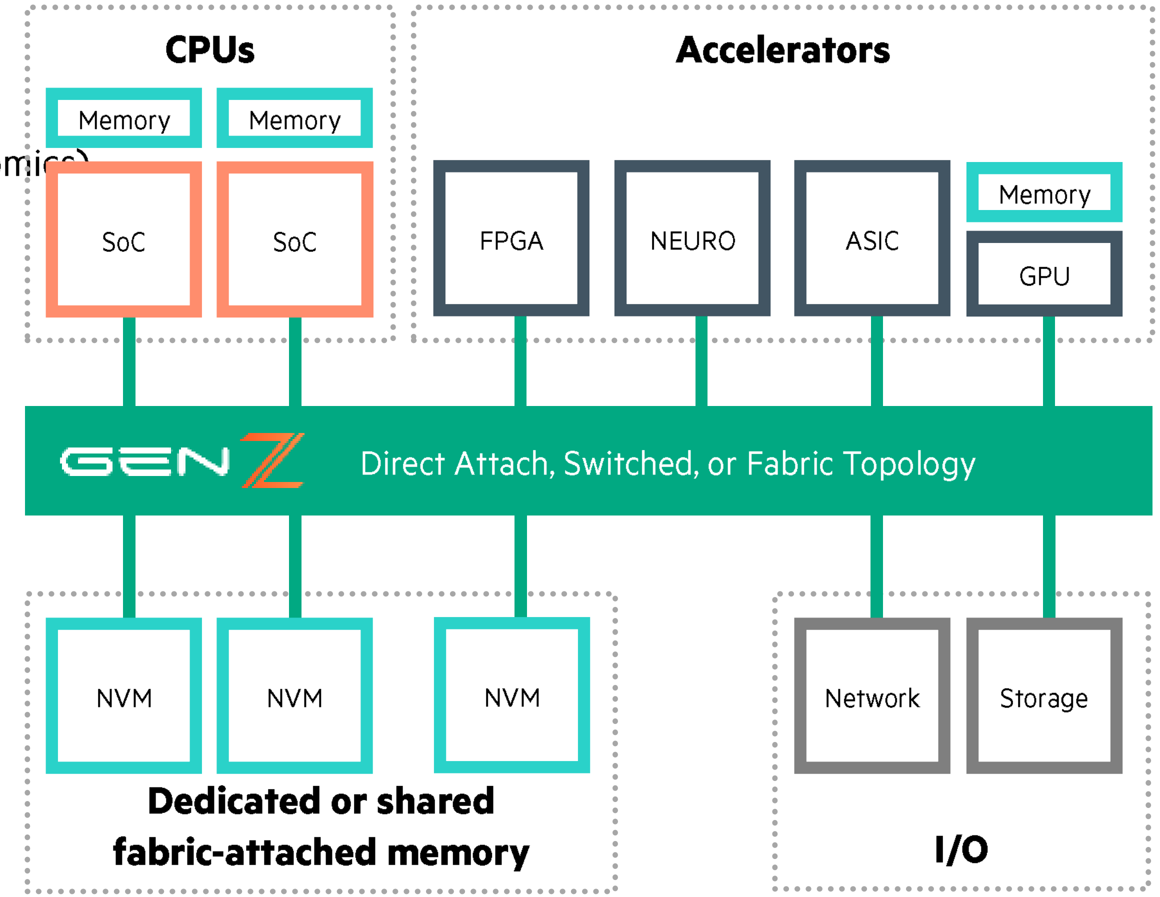
Hewlett Packard Enterprise

Gen-Z: open systems interconnect standard

http://www.genzconsortium.org

Open Standard

- Open standard for memory-semantic interconnect
- Memory semantics
  - All communication as memory operations (load/store, put/get, atomics)
- High performance
  - Tens to hundreds GB/s bandwidth
  - Sub-microsecond load-to-use memory latency
- Scalable from IoT to exascale
- Spec available for public download



Hewlett Packard Enterprise

Consortium with broad industry support

GEN Z Consortium Members (65)

System OEM
- Cisco
- Cray
- Dell EMC
- H3C
- Hitachi
- HP
- HPE
- Huawei
- Lenovo
- NetApp
- Nokia
- Yadro

CPU/Accel
- AMD
- Arm
- IBM
- Qualcomm
- Xilinx

Mem/Storage
- Everspin
- Micron
- Samsung
- Seagate
- SK Hynix
- Smart Modular
- Spintransfer
- Toshiba
- WD

Tech Svc Provider
- Google
- Microsoft
- Node Haven

Silicon
- Broadcom
- IDT
- Marvell
- Mellanox
- Microsemi
- Sony Semi

IP
- Avery
- Cadence
- Intelliprop
- Mentor
- Mobiveil
- PLDA
- Synopsys

Eco/Test
- Allion Labs
- Keysight
- Teledyne LeCroy

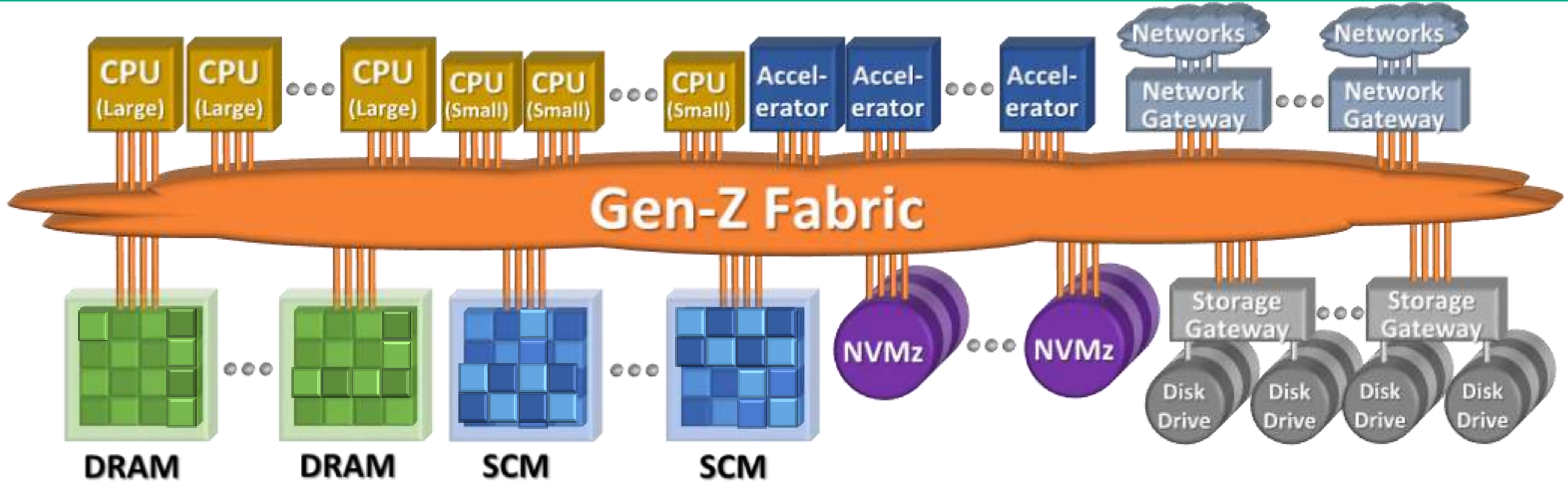
Connect
- Aces
- AMP
- FIT
- Genesis
- Jess Link
- Lotes
- Luxshare
- Molex
- Samtec
- Senko
- TE
- 3M

Software
- Redhat
- VMware

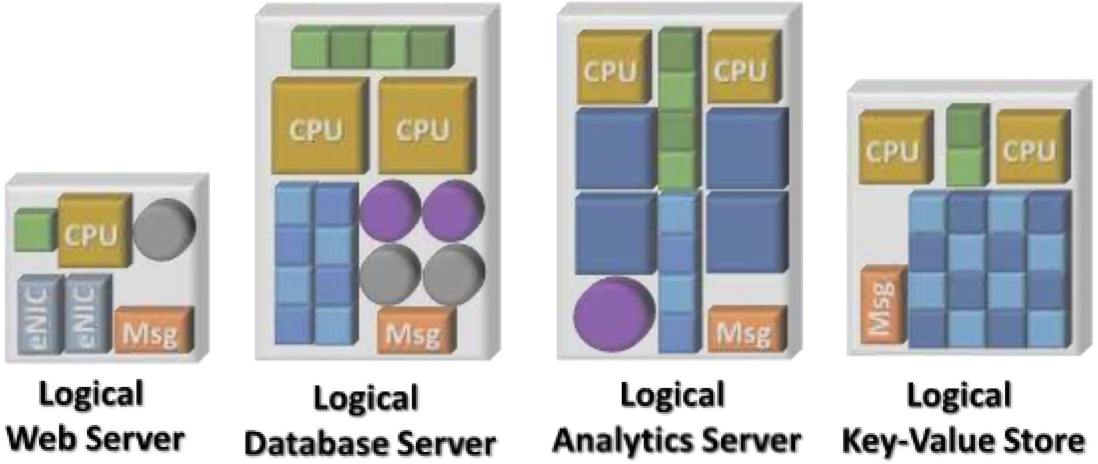
Govt/Univ
- ETRI
- Oak Ridge
- Simula
- UNH
- Yonsei U
- ITT Madras

Hewlett Packard Enterprise

Enabling Right-Sized Solutions



- Logical systems composed of physical components
  - Or subparts or subregions of components (e.g. memory/storage)
- Logical systems match exact workload requirements
  - No stranded resources overprovisioned to workloads
- Facilitates data-centric computing via shared memory
  - Eliminates data movement: Do more with less, reduces cost

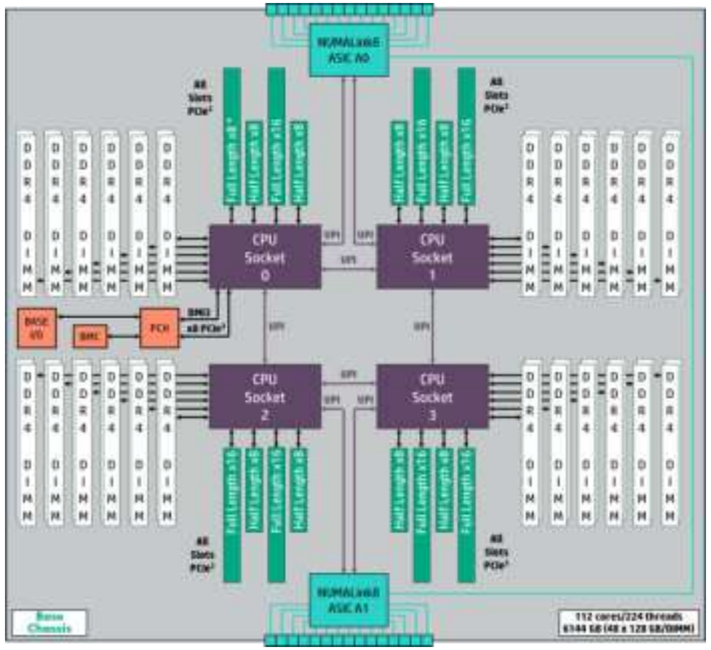


Hewlett Packard Enterprise

HPE Superdome Flex

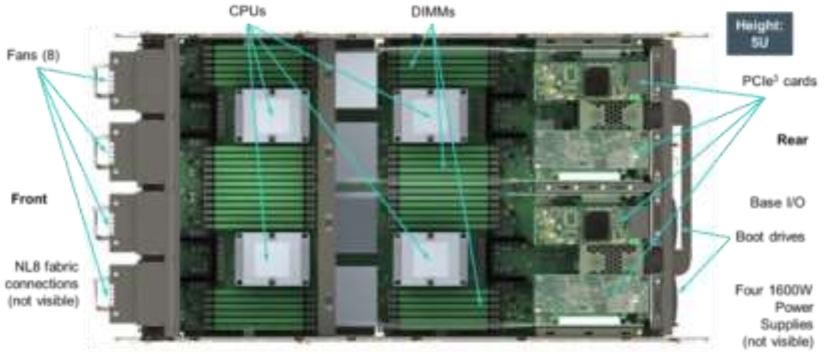
Modular design for maximum flexibility and performance

- Technology supports 5U 4-socket modular chassis, scaling from 4 to 32+ sockets in 4-socket increments. General release of >8s scale (to 32s) will roll-out over time.
- Full and flexible connectivity with simple interconnect cabling architecture – **higher bandwidth and lower latency than Superdome X and MC990X** for ‘extreme scale performance’ - Unique in the industry!
- High availability (HA) features enable fabric fault tolerance
- Flexible stand-up PCIe Gen3 card format supported

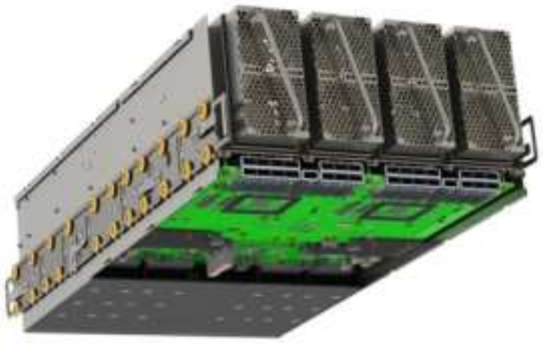


Memory	Compute	I/O
48 DIMMs 6 TB capacity	4 sockets	9 (x8) slots 7 (x16) slots boot storage

Top view



Bottom view

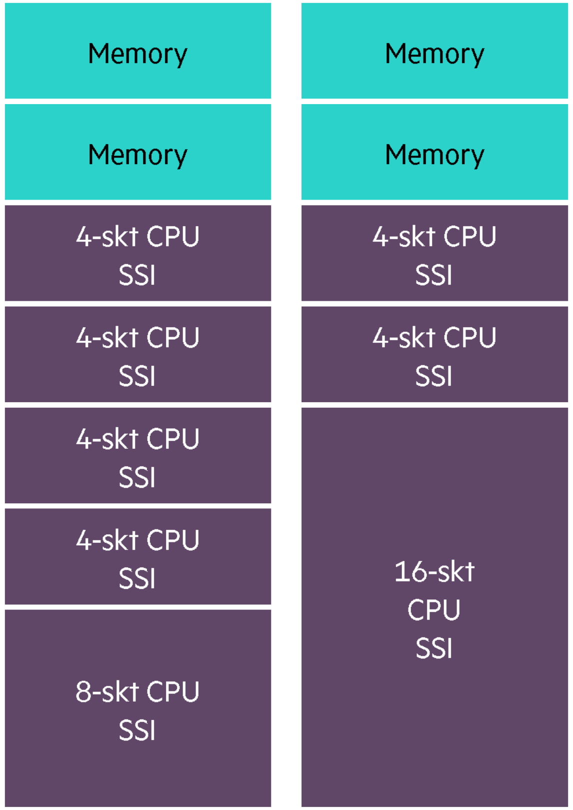


For even larger workloads

12S	• up to 336 cores / 672 threads • up to 18TB • up to 48 PCIe 3.0 card slots
16S	• up to 448 cores / 896 threads • up to 24TB • up to 64 PCIe 3.0 card slots
20S	• up to 560 cores / 1120 threads • up to 30TB • up to 80 PCIe 3.0 card slots
24S	• up to 672 cores / 1344 threads • up to 36TB • up to 96 PCIe 3.0 card slots
28S	• up to 784 cores / 1568 threads • up to 42TB • up to 112 PCIe 3.0 card slots
32S	• up to 896 cores / 1792 threads • up to 48TB • up to 128 PCIe 3.0 card slots

Hewlett Packard Enterprise

Composing Superdome Flex Systems with Software Defined Scalable Memory



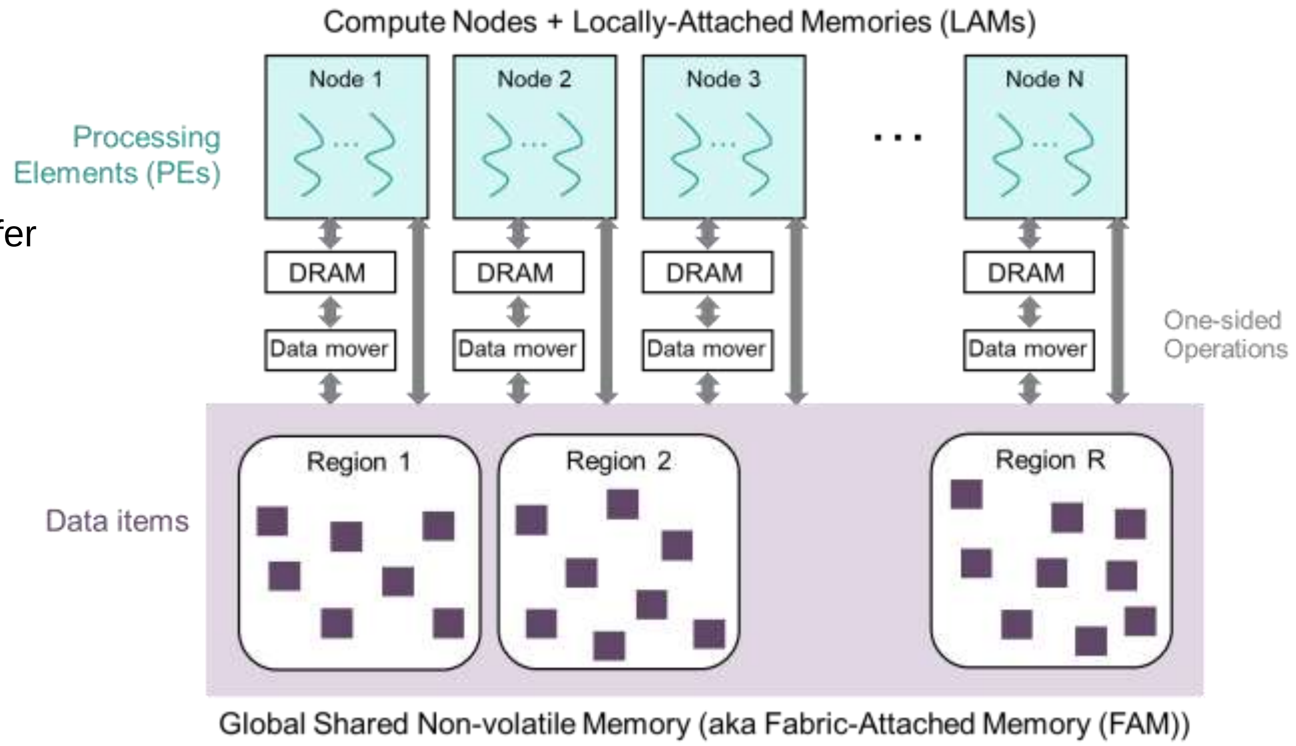
Intel release of CascadeLake, with 3DXPoint DIMMs, will enable potential for

- 12TB persistent memory per chassis, 96TB per rack
- Dynamically assigned to compute partitions
- Memory chassis becomes the most powerful intelligent storage element in HPE portfolio

Forward looking direction – no production commitment

OpenFAM: programming model for fabric-attached memory

- FAM memory management
  - Regions (coarse-grained) and data items within a region
- Data path operations
  - Blocking and non-blocking get / put, scatter / gather: transfer memory between node local memory and FAM
  - Direct access: enables load / store directly to FAM
- Atomics
  - Fetching and non-fetching all-or-nothing operations on locations in memory
  - Arithmetic and logical operations for various data types
- Memory ordering
  - Fence (non-blocking) and quiet (blocking) operations to impose ordering on FAM requests



K. Keeton, S. Singhal, M. Raymond, “The OpenFAM “The OpenFAM API: a programming model for disaggregated persistent memory,” *Proc. OpenSHMEM 2018*.

Draft of OpenFAM API spec available for review: https://github.com/OpenFAM/API
Email us at openfam@groups.ext.hpe.com

Hewlett Packard Enterprise

MDC Programming Opportunities

Data sharing in one large globally-addressable memory

- Pass by reference, rather than copy
- Multi-process – share large pool of data in shared memory
- Use global shared memory for messaging

Focus on in-memory data formats

- Filesystem vs database vs direct use of byte-addressable persistent memory
- Opportunity to move away from having multiple data formats in memory and storage; single data format used as both in-memory representation and data storage
- Reduce number of software layers – simpler to develop and maintain software

MDC Programming Challenges

Practicalities of using new technologies

- Accessing memory: persistent memory and fabric-attached memory
- Allocating and managing memory

Data consistency in face of failures

- Vulnerability of data in persistent memory – failures that result in corruption or loss
- Memory can be persistent but not consistent – can’t turn it off and on again
- Need ability to update data in persistent memory from one consistent state to another, even in presence of failures

Hewlett Packard Enterprise

Designing for disaggregation

- Challenge: how to design data structures and algorithms for disaggregated architectures?
  - Shared disaggregated memory provides ample capacity, but is less performant than node-local memory
  - Concurrent accesses from multiple nodes may mean data cached in node's local memory is stale

- Potential solution: "distance-avoiding" data structures
  - Data structures that exploit local memory caching and minimize "far" accesses
  - Borrow ideas from communication-avoiding and write-avoiding data structures and algorithms

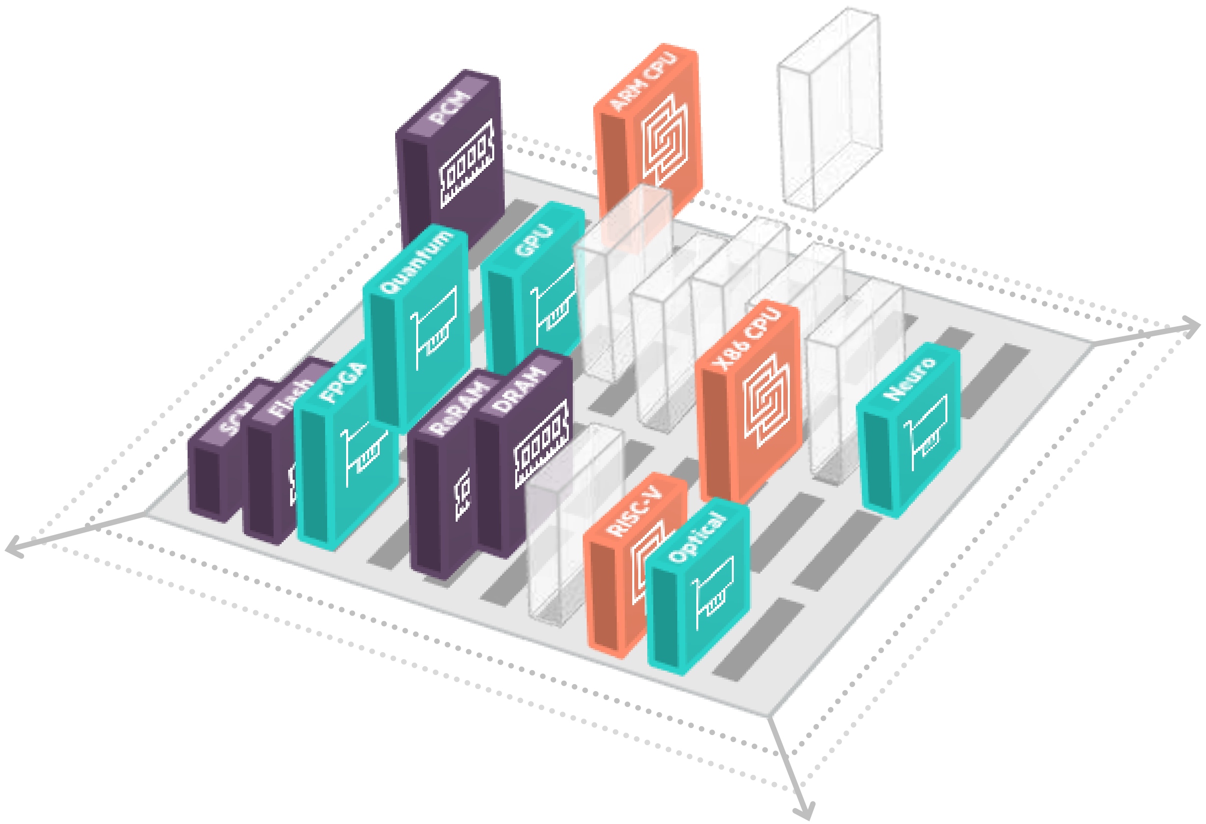
- Potential solution: hardware support
  - Ex: indirect addressing to avoid "far" accesses, notification primitives to support sharing
  - What additional hardware primitives would be helpful?

Marcos K. Aguilera, Kimberly Keeton, Stanko Novakovic, and Sharad Singhal. 2019. Designing Far Memory Data Structures: Think Outside the Box. In Workshop on Hot Topics in Operating Systems (HotOS '19), May 13–15, 2019, Bertinoro, Italy. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3317550.3321433 (To appear)

Wrapping up

- New technologies pave the way to Memory-Driven Computing
  - Fast direct access to large shared pool of fabric-attached (non-volatile) memory
- Memory-Driven Computing
  - Mix-and-match composability with independent resource evolution and scaling
- Combination of technologies enables us to rethink the programming model
  - Simplify software stack
  - Operate directly on memory-format persistent data
  - Exploit disaggregation to improve load balancing, fault tolerance, and coordination
- Many opportunities for software innovation

- How would you use Memory-Driven Computing?



Questions

Hewlett Packard Enterprise