Realizing the Next Generation of Exabyte-scale Persistent Memory-Centric Architectures and Memory Fabrics

Zvonimir Z. Bandic, Sr. Director, Next Generation Platform Technologies
Western Digital Corporation

Safe Harbor | Disclaimers

This presentation contains forward-looking statements that involve risks and uncertainties, including, but not limited to, statements regarding our product and technology positioning, the anticipated benefits of the integration of HGST and SanDisk into our company, market trends, business strategy and growth opportunities. Forward-looking statements should not be read as a guarantee of future performance or results, and will not necessarily be accurate indications of the times at, or by, which such performance or results will be achieved, if at all. Forward-looking statements are subject to risks and uncertainties that could cause actual performance or results to differ materially from those expressed in or suggested by the forward-looking statements.

Additional key risks and uncertainties include the impact of continued uncertainty and volatility in global economic conditions; actions by competitors; difficulties associated with the integration of SanDisk and HGST into our company; business conditions; growth in our markets; and pricing trends and fluctuations in average selling prices. More information about the other risks and uncertainties that could affect our business are listed in our filings with the Securities and Exchange Commission (the “SEC”) and available on the SEC’s website at www.sec.gov, including our most recently filed periodic report, to which your attention is directed. We do not undertake any obligation to publicly update or revise any forward-looking statement, whether as a result of new information, future developments or otherwise, except as otherwise required by law.

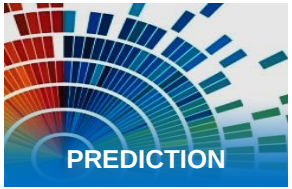
Agenda

- Rapid growth in the data ecosystem
  - Importance of purpose-built technologies and architectures
- Current distributed systems architecture
  - Where should we attach persistent memory?
  - RDMA networking: Lowest latency remote memory access
  - Compute state of the art: Compute acceleration
- Persistent memory scale out
  - Memory pool size requirements
  - CPU attached memories access via RDMA networking vs. memory fabrics
  - Memory fabrics latency requirements
- Conclusions

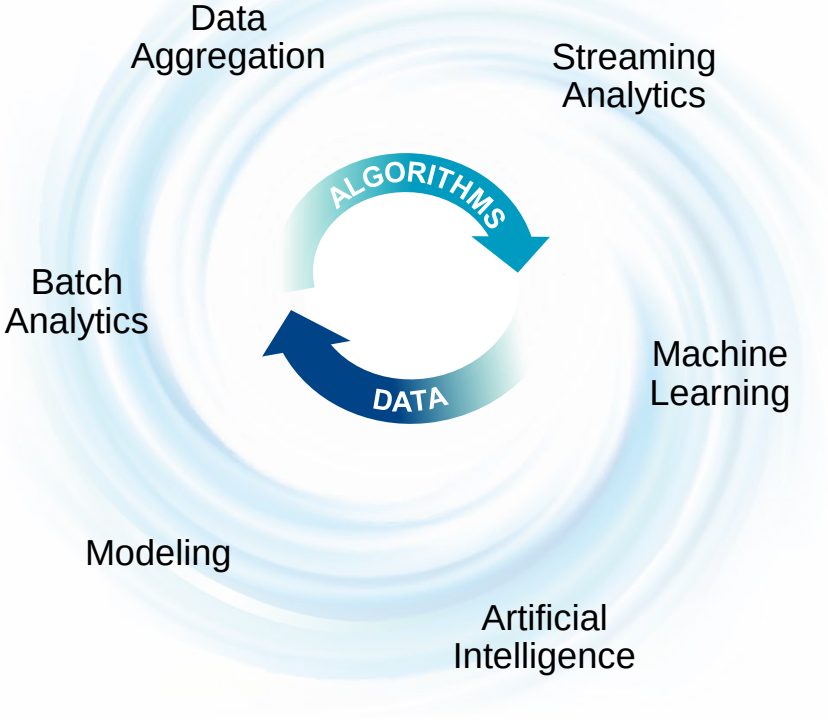
Diverse and connected data types

Tight coupling between Big Data and Fast Data

BIG DATA



Scale

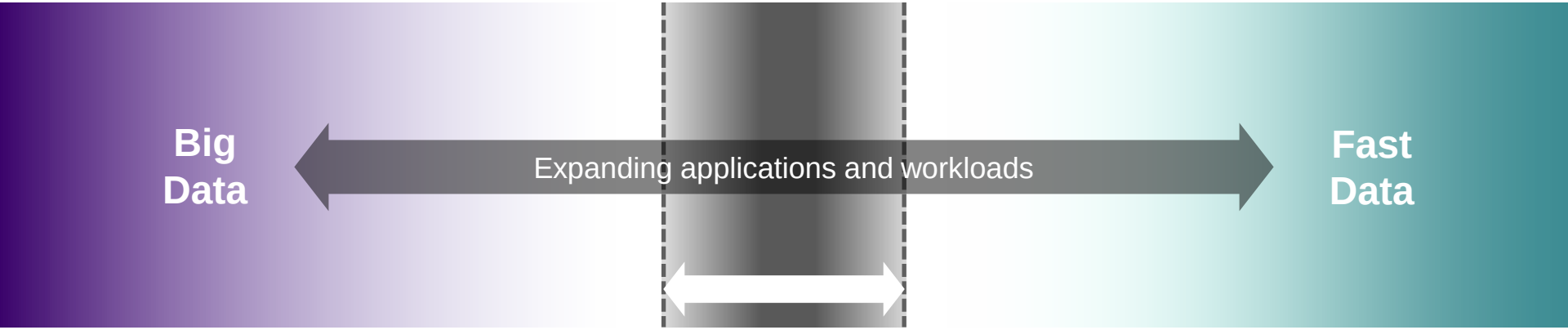


FAST DATA

Performance

From general purpose to purpose built

Architectures designed for Big Data, Fast Data applications

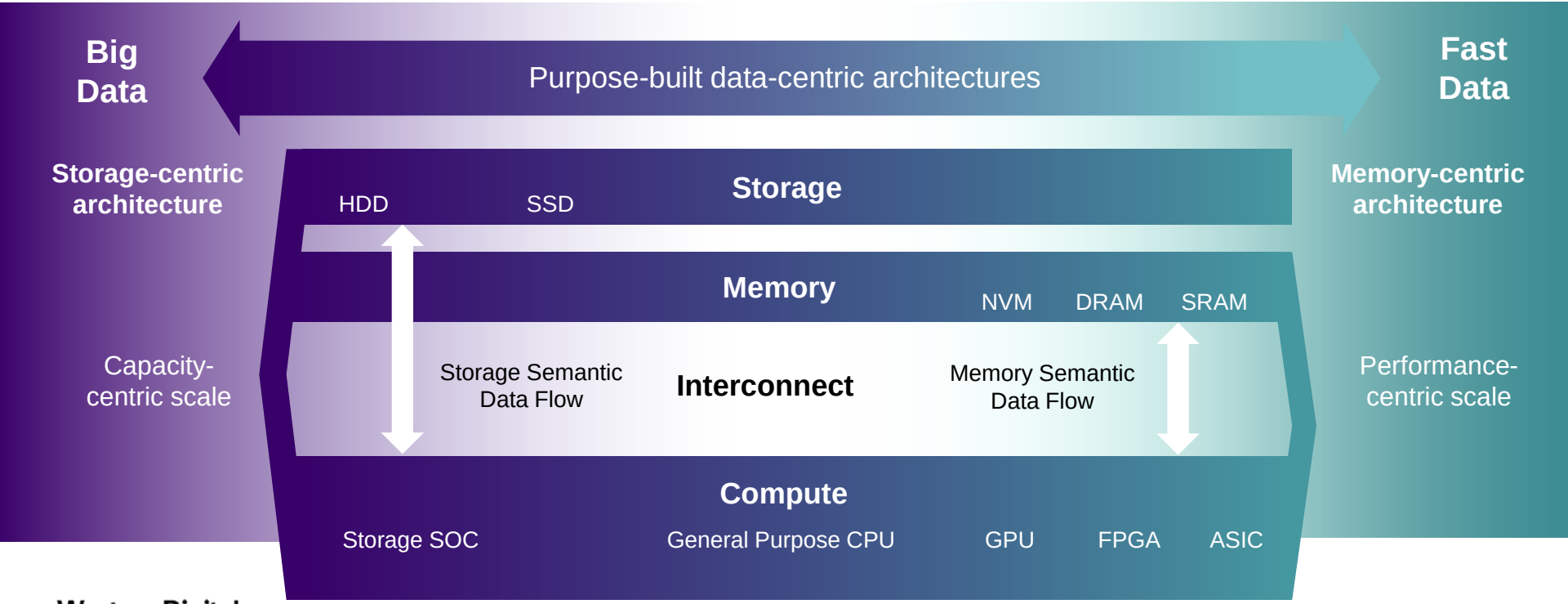


General purpose
compute-centric architecture

- Solutions
- Systems
- Platforms
- Devices

Workload diversity

Demanding diverse technologies and architectures

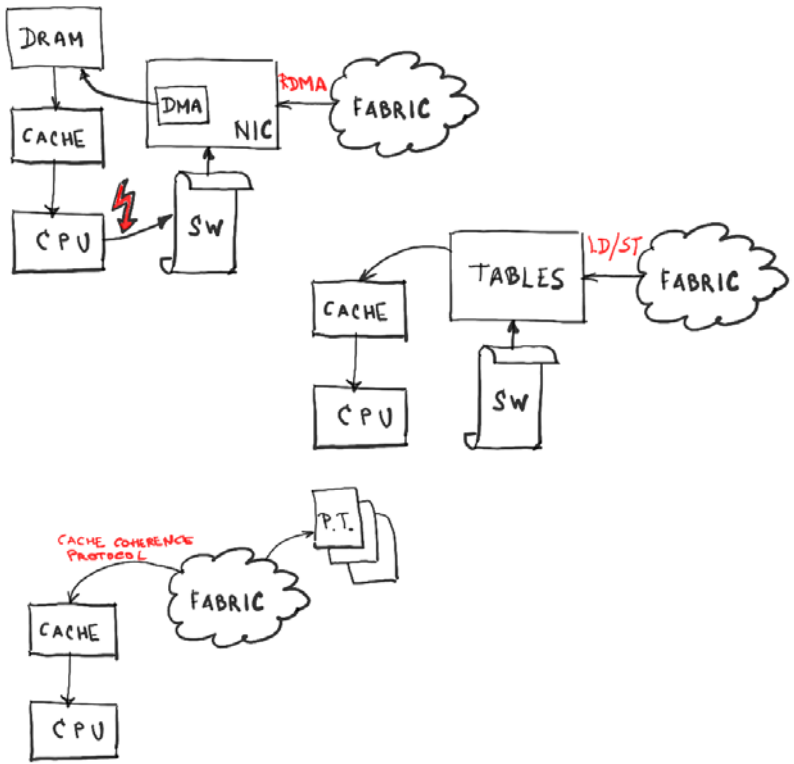


Where should we attach persistent memory?

	CPU BUS: PARALLEL	CPU BUS: SERIAL	SERIAL PERIPHERAL BUS
Physical interface	DIMM	DIMM/other	PCIe
Logical interface	Non-standard DDR4, NVDIMM-P	DMI for Power 8, CCIX, OpenCAPI 3.1, Rapid-IO, gen-Z	NVMe, DC express*
Pros	• Low latency • High bandwidth • Power proportional • Coherent through memory controller	• High bandwidth • Significant pin reduction • Higher memory bandwidth to CPU • Coherent through memory controller, or in some cases can even present lowest point of coherence	• Standardized • CPU/platform independent • Latency low enough for storage • Easy RDMA integration • Hot pluggable
Cons	• CPU memory controller has to implement specific logical interface • Not suited for stochastic latency behavior • Not hot pluggable • BIOS needs change	• CPU memory controller has to support • May have higher power consumption	Higher latency (~1us)

The emergence of memory fabric

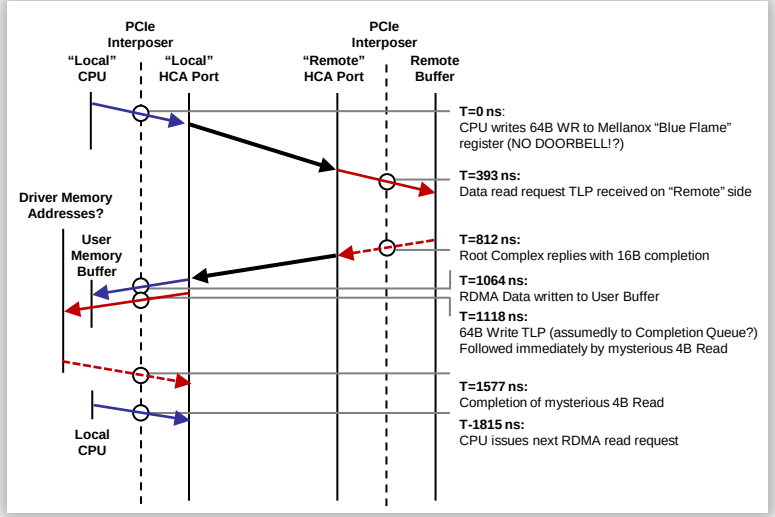
- Memory fabric may mean different things to different people:
  - Page fault trap leading to RDMA request (incurs context switch and SW overhead)
  - Global address translation management in SW, leading to LD/ST across global memory fabric
  - Coherence protocol scaled out, global page management and no context switching



RDMA networking: Latency

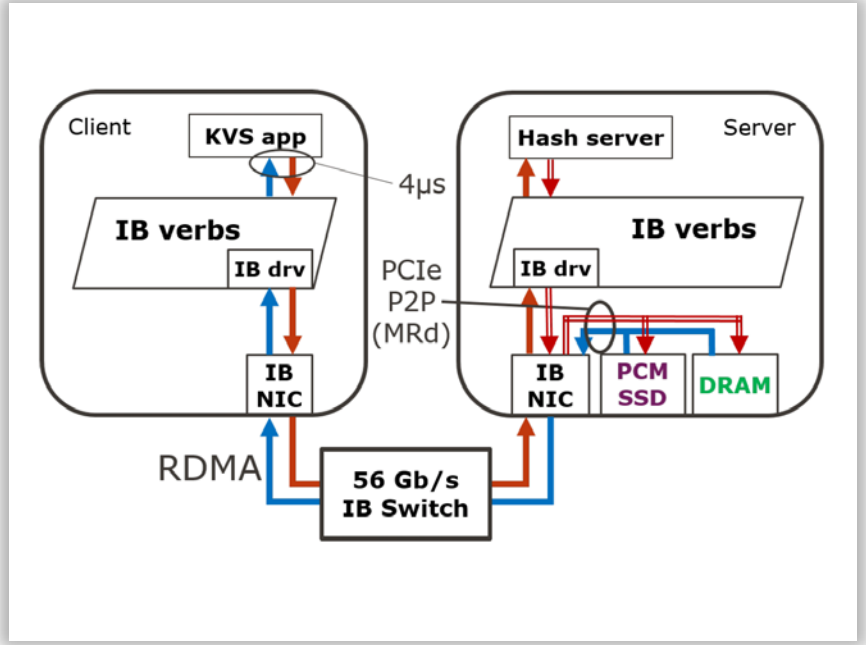
- As a direct memory transfer protocol RDMA is already suited to accessing remote byte-addressable persistent memory devices
- Typical latency for RDMA reads: 1.6-1.8us
  - For PCIe attached RNICs on host and target nodes
- Simultaneous PCIe tracing reveals as little as 600ns spent actually transferring data over the network
  - 70% of latency is spent in memory or PCIe-protocol overhead
- How much can be saved by attaching memory directly to network fabric (i.e., no PCIe overhead)
  - Maximal benefit (~1.0 us) would require integrated RNIC interfaces on both Initiator and Target

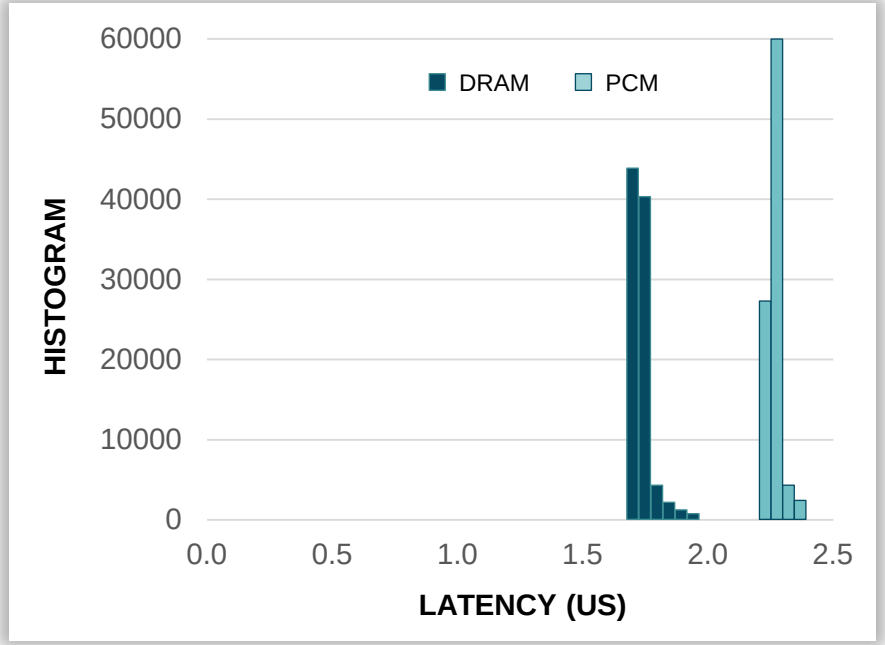
BREAKDOWN OF A 16-BYTE RDMA_READ REQUEST



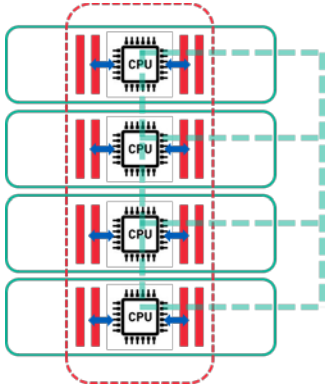
RDMA to persistent memory latency

Raw RDMA access to remote PCM via PCIe peer2peer is 26% slower than to remote DRAM

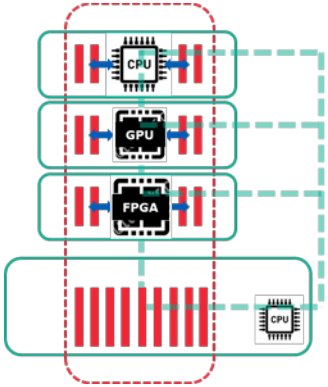




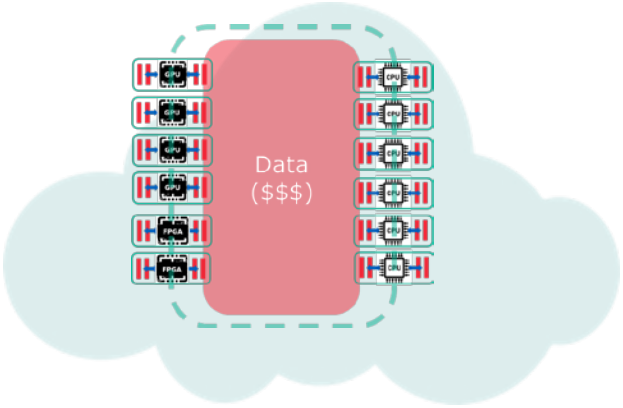
Compute accelerators



- Traditional distributed system:
  - Memory accessible only via general purpose CPU or via RDMA



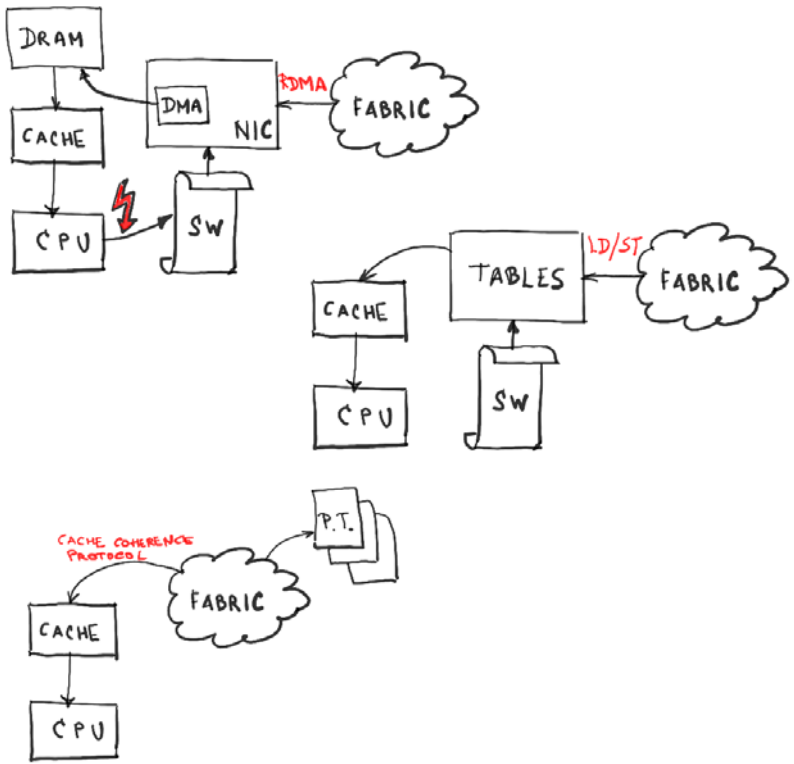
- Distributed system with compute accelerators, with own memory:
  - FPGA, GP-GPU, ASIC
  - Compute accelerators need large memory bandwith and may drive need for memory appliances (e.g. NVLink)



- Data (memory) centric architecture:
  - Large number of compute devices (Cores, FPGA, GP-GPUs, ASIC) assessing large pool of memory

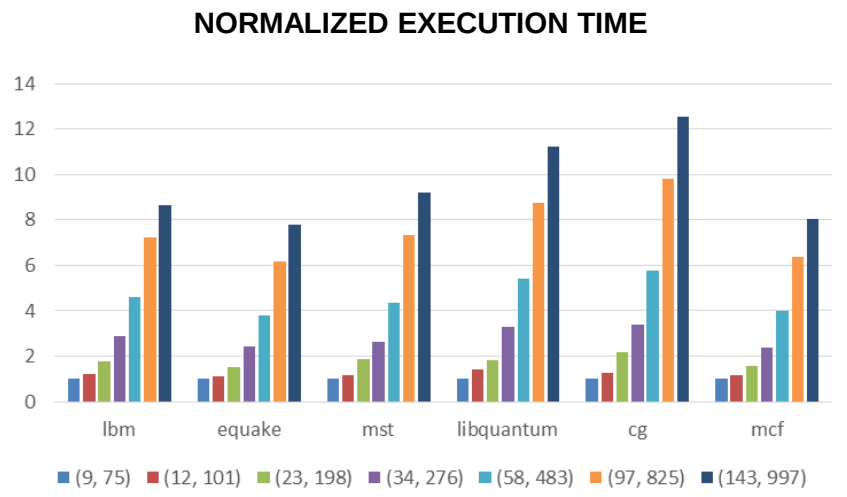
The emergence of memory fabric

- Memory fabric may mean different things to different people:
  - Page fault trap leading to RDMA request (incurs context switch and SW overhead)
  - Global address translation management in SW, leading to LD/ST across global memory fabric
  - Coherence protocol scaled out, global page management and no context switching



Latency requirements

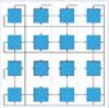
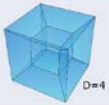
- It is not trivial to determine the threshold latency, for which CPU load operation still makes sense:
  - As opposed to using block IO, which allows more efficient CPU utilization
- Based on slowing down DRAM, and executing large number of benchmarks, we determine soft threshold of approximately:
  - 500 ns – 1 us



Impact of latency on execution time
(Intel Xeon, DRAM frequency settings changed in BIOS)

Persistent memory scale-out

- Exabyte scale memory system requirements:
  - Assume exabyte scale, or at least many 10's of Petabytes distributed in large number of nodes
- Assuming 256 Gb/die, the amount of persistent memory per node can be estimated as 8-16 TB:
  - Limited by total power budget – SoC and media
- Number of nodes required is quite large:
  - For 1 PB: 64 nodes
  - For 64 PB: 4096 nodes
- Assuming 3D torus, the number of hops (one direction) is:
  - For 1 PB: 64 nodes; 8 hops
  - For 64 PB: 4096 nodes; 48 hops
- Path for improvement is:
  - Increasing dimensionality of the network topology – expensive
  - Improving power-density envelope of the memory technology in order to pack more capacity per node

Topology	Block diagram	#neighbors	throughput	latency	cost	Comment
1D torus		2	O(n)	$O(n^{0.5})$	O(n)	
2D torus		4	O(n)	$O(n^{0.333})$	O(n)	
3D torus		6	O(n)	$O(n^{0.25})$	O(n)	Latency is more likely $O(n^{0.333})$ due to loading
Hypercube		nD/2	O(n log n)	O(log n)	O(n log n)	Latency is more likely $O(n^{0.333})$ due to loading

End-to-end latency requirements

- Assuming that at least 8 hops across the cluster are required for remote memory access:
  - 8 x switch_latency+raw_memory_latency <500 ns – 1000 ns
- Assuming raw memory latency is 100ns:
  - Switch_latency <(50 – 100) ns
- Larger clusters and larger number of nodes will lead to significantly tighter latency requirement

Memory fabric protocol innovation platform I

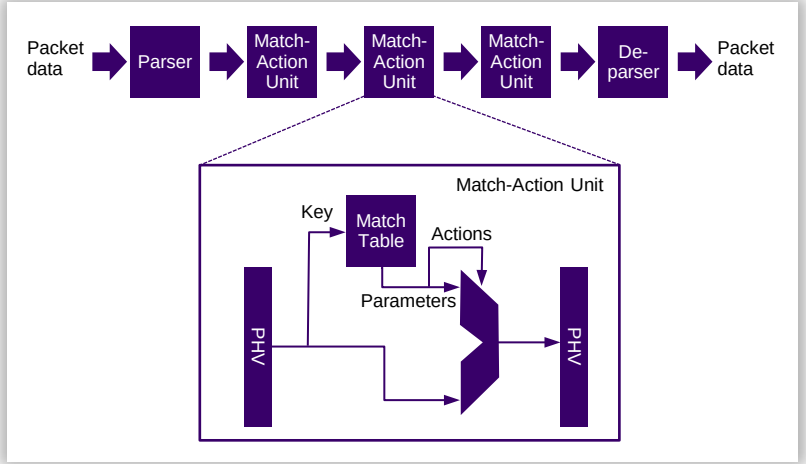
- Routable and switchable fabric:
  - Require uniform fabric from card, to node to cluster level, with uniform switching and routing solutions
- 25Gb/s IP is available today:
  - With a clear roadmap to at least 56Gb with PAM4
  - Same fabric can be used on a node level to enable heterogeneous compute
- Replace L2 Ethernet with custom protocol on top of 802.3 L1
  - Keep L1 from 802.3
  - Rest is programmed with P4 allowing implementation of multiple protocols
  - Supports innovation required for RAS
  - FPGA or ASIC switch

802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46–1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1522 octets →						
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 oct. →

P4 example: Gen-Z switch prototyping

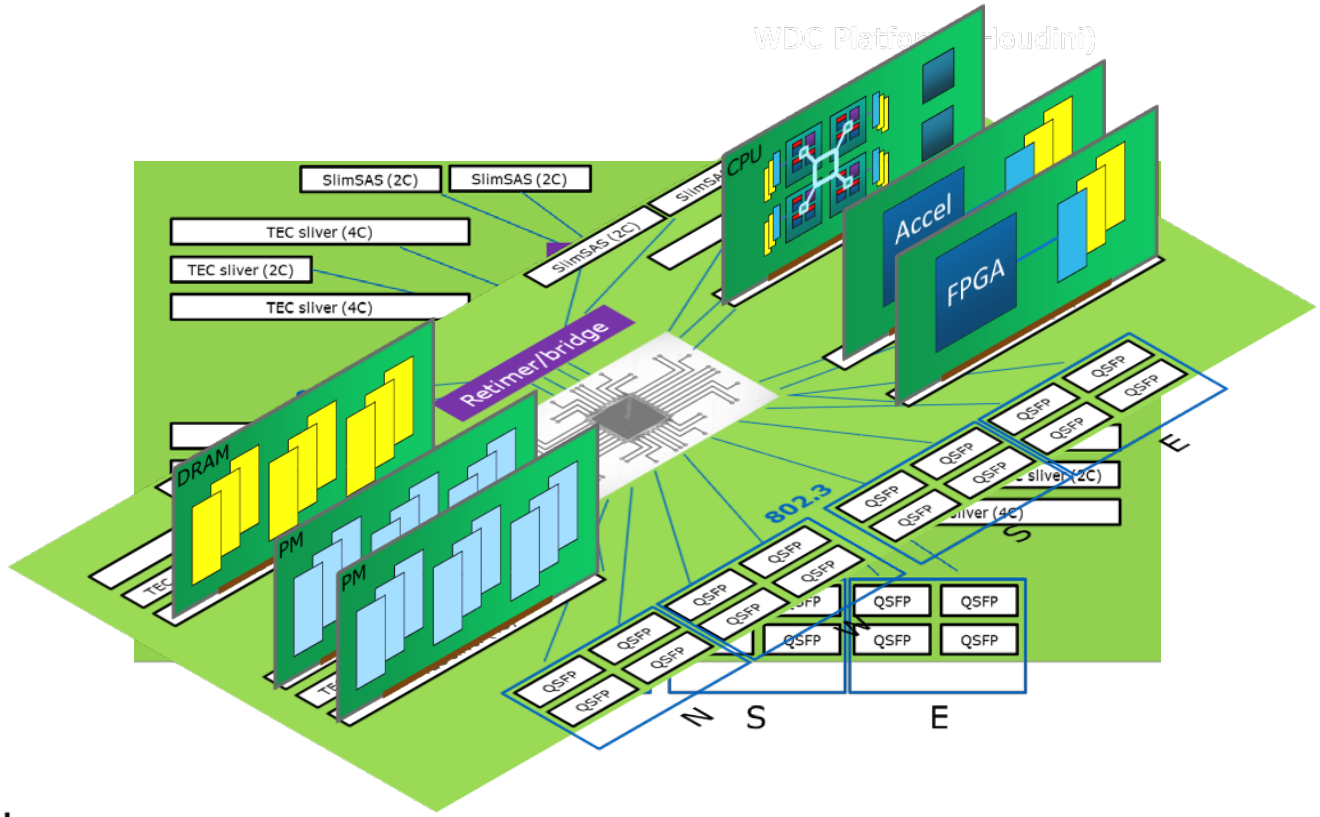
- BarefootTofino ASIC (or FPGA):
  - 256-lane 25 GT/s Ethernet switch, 6 Tbit/s aggregate throughput
  - Supports P4 HDL, successor to OpenFlow enabling protocol innovation
  - Describe Gen-Z packet format in P4
  - Match-Action Pipeline (a.k.a. “flow tables”) enables line-rate performance




```
/* -*- P4_16 -*- */

#include <core.p4>
#include <v1model.p4>

/*
 * Define the headers the program will recognize
 */

/*
 * This is a custom protocol header for the Gen-Z packets.
 * We'll use ethertype 0x1234 for is (see parser)
 */

const bit<16> P4GENZ_P2P_CORE_ETYPE = 0x1234;
const bit<1>  P4GENZ_P2P_END2END_ETYPE = 0x0;
const bit<1>  P4GENZ_P2P_LINKLOCAL_ETYPE = 0x1;

/*
 * Defined the headers of Gen-Z packets
 *
 */

header p4genz_p2p_corereq_t {
    bit<5>    p4genz_pcrc;
    bit<1>    p4genz_oc_0;
    bit<1>    p4genz_o;
```

```
parser MyParser(
    packet_in              packet,
    out   my_headers_t    hdr,
    inout my_metadata_t   meta,
    inout standard_metadata_t standard_metadata)
{
    state start {
        packet.extract(hdr.ethernet);
        transition select(hdr.ethernet.etherType) {
            P4GENZ_P2P_CORE_ETYPE : check_p4genz;
            default               : accept;
        }
    }


    //check L of Gen-Z packet
    state check_p4genz {
        transition
select(packet.lookahead<p4genz_p2p_corereq_t>().p4genz_l) {
            P4GENZ_P2P_END2END_ETYPE  : parse_p4genz;
            default                   : accept;
        }
    }

    state parse_p4genz {
        packet.extract(hdr.p4genz_p2p_corereq);
        transition accept;
    }
}
```

Memory fabric protocol innovation platform II



Conclusions

- RDMA networking today can be used to build memory fabric solutions
  - Typical measured latency in the 1.6-1.8 us range
- From performance perspective, achieving 500 ns fabric latency across exabyte (~10's of PB) scale cluster puts significant pressure on persistent memory power and density, and memory fabric switch latency
- Protocol innovations are required to fulfill latency, throughput and RAS requirements

Western Digital.

Creating environments for data to thrive

Research authored by Zvonimir Z. Bandic, Yang Liu, Chao Sun, Qingbo Wang, Martin Lueker-Boden, Damien LeMoal, Dejan Vucinic for Next Generation Platform Technologies at Western Digital Corporation