

DoE HPC Roadmap: Exascale computing project (2021-2025)



Frontier AMD CPU, AMD GPU; **HIP**

ORNL



Perlmutter AMD CPU, Nvidia GPU; **CUDA**

NERSC



Aurora Intel CPU, Intel GPU; **SYCL**

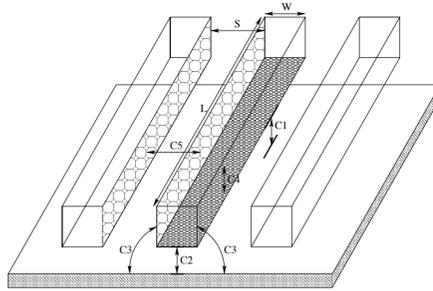
Argonne

HPC computing in the US *will* be accelerated

“Native” programming models are all distinct

- Possible strategies
 1. Abstract the differences (write an interface that can be implemented on them all)
 2. Use someone else’s abstraction (e.g. Kokkos)
 3. Rely on a standard like OpenMP 5.0 *target offload*

Why are we suffering?



Gate Length (nm)	Dielectric Constant κ	Metal ρ ($\mu\Omega\text{-cm}$)	Mid-Level Metal			
			Width (nm)	Aspect Ratio	R_{wire} ($m\Omega/\mu m$)	C_{wire} ($fF/\mu m$)
250	3.9	3.3	500	1.4	107	0.202
180	2.7	2.2	320	2.0	107	0.333
130	2.7	2.2	230	2.2	188	0.336
100	1.6	2.2	170	2.4	316	0.332
70	1.5	1.8	120	2.5	500	0.331
50	1.5	1.8	80	2.7	1020	0.341
35	1.5	1.8	60	2.9	1760	0.348

Simple physics explains computer architecture: model wire as rod of metal $L \times \pi r^2$

- **Charge:** Gauss's law

$$2\pi rLE = \frac{Q}{\epsilon}$$

- **Resistance**

$$R = \rho \frac{L}{\pi r^2}$$

- **Capacitance**

$$C = Q/V = 2\pi L\epsilon / \log(r_0/r)$$

- **Time constant**

$$RC = 2\rho\epsilon \frac{L^2}{r^2} / \log(r_0/r) \sim \frac{L^2}{r^2}$$

RC wire delay depends *only* on geometry: Shrinking does not speed up wire delay!

- “copper interconnect” (180nm) and “low-k” dielectric (100nm) improved ρ and ϵ

Multi-core design with long-haul buses only possible strategy for 8 Billion transistors

- Low number of long range “broad” wires (bus/interconnect)
- High number of short range “thin” wires

Growth in parallelism

Core	simd	Year	Vector bits	SP flops/clock/core	cores	flops/clock
Pentium III	SSE	1999	128	3	1	3
Pentium IV	SSE2	2001	128	4	1	4
Core2	SSE2/3/4	2006	128	8	2	16
Nehalem	SSE2/3/4	2008	128	8	10	80
Sandybridge	AVX	2011	256	16	12	192
Haswell	AVX2	2013	256	32	18	576
KNC	IMCI	2012	512	32	64	2048
KNL	AVX512	2016	512	64	72	4608
Skylake	AVX512	2017(?)	512	64	28	1792

- Growth in core counts
- Growth in SIMD parallelism
- Growth in complexity of memory heirarchy
- Interconnect performance failing to grow as fast as processor and memory performance

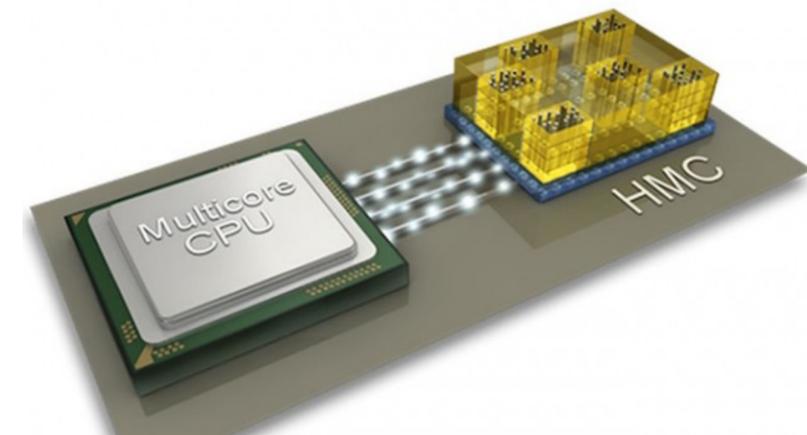
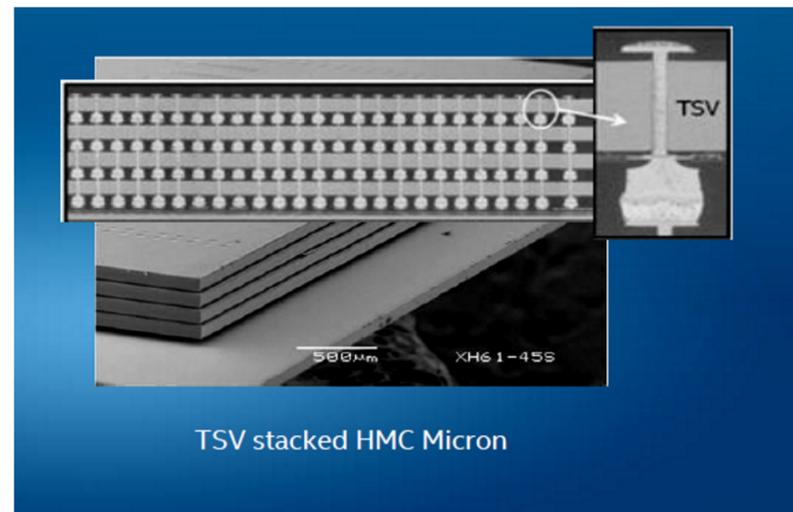
Standard industry solution is to dump it on the progammer!

3D & non-uniform memory : small and fast or big and slow

- Apply to memory buses with through-silicon-via's (TSVs)!
- **2.5D** : Integrate memory stacks on an *interposer* (Intel, Nvidia, AMD)
In package memory: long thin wires → short broad fast wires
- **3D** : Direct bond memory stacks to compute (PEZY, mobile, Broadcom)
3D memory could grow the bus widths almost arbitrarily

Massive replica counts **from silicon lithography** compared to macroscopic assembly

There's plenty of room at the bottom (Feynman); Avagadro's number is big!



Fragmentation of node memory

Aurora: Bringing It All Together

2 INTEL XEON SCALABLE PROCESSORS
"Sapphire Rapids"

6 X^E ARCHITECTURE BASED GPU'S
"Ponte Vecchio"

ONEAPI
Unified programming model

LEADERSHIP PERFORMANCE
For HPC, data analytics, AI

UNIFIED MEMORY ARCHITECTURE
Across CPU & GPU

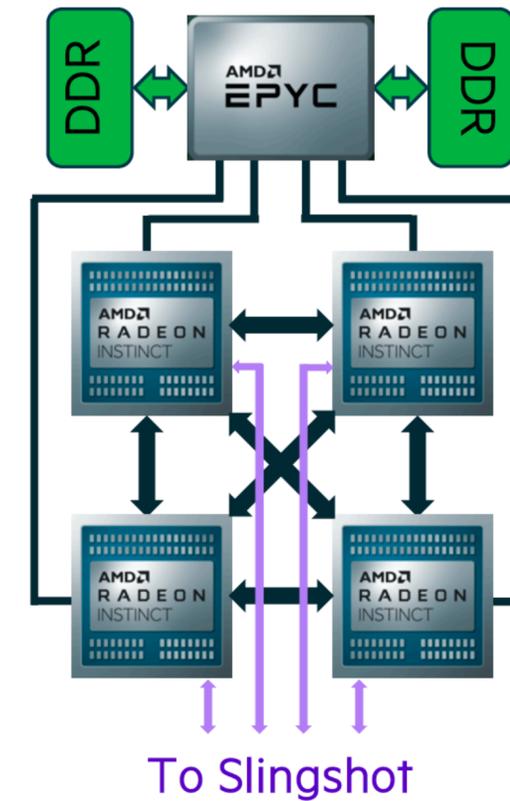
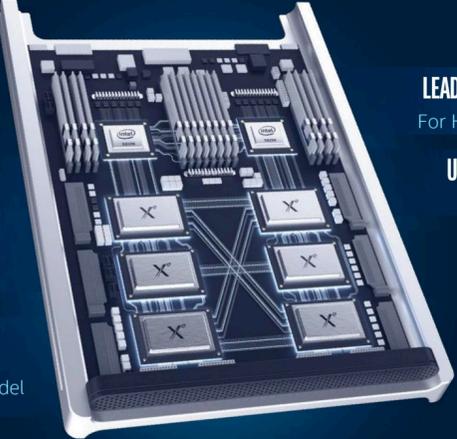
ALL-TO-ALL CONNECTIVITY WITHIN NODE
Low latency, high bandwidth

UNPARALLELED I/O SCALABILITY ACROSS NODES
8 fabric endpoints per node, DAOS

DELIVERED IN 2021

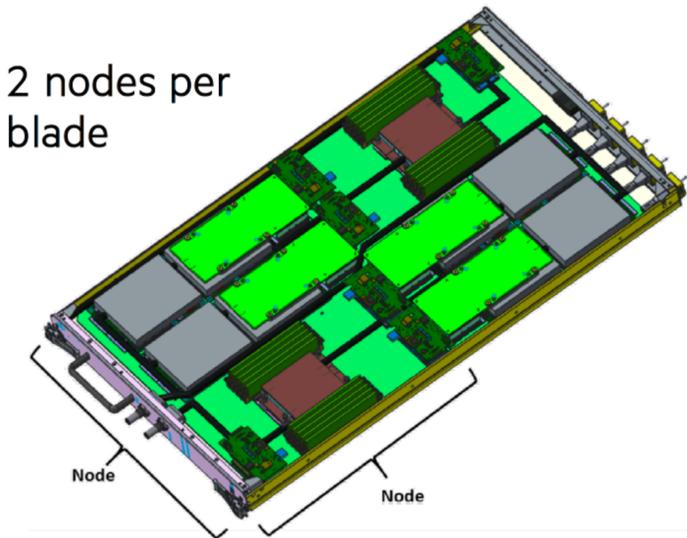
U.S. DEPARTMENT OF ENERGY | Argonne NATIONAL LABORATORY | intel | CRAY

News Under Embargo: November 17, 2019 – 4:00 p.m. Pacific Time



AMD GPU
(ORNL)

2 nodes per blade

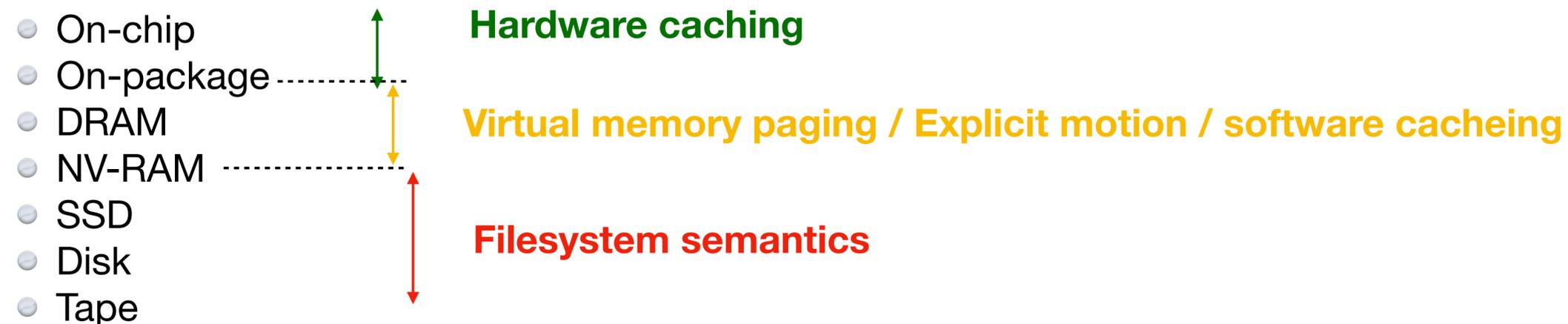


COPYRIGHT 2020 HPE

Explicit message passing (MPI)

What can we say about 10-15 years? Some best guesses:

- Many computer nodes communicating via message passing is the cheapest way to build a large HPC system
 - I bet batch queues, constraints, and associated difficulty of **debugging MPI programmes is not going away.**
- **Computer node organisation will be at least as complex.**
 - **Multiple distinct memories on node**
 - Distinct intra-node and inter-node networks (but can programme with MPI).
- Whether “Accelerator” or “CPU” not, silicon will be highly parallel
 - Programmers will **have to care about “vectorisation”** whether it’s called “coalescing” or “SIMD”
 - Not all algorithms naturally map to this. However ML does, and SGEMM use cements the position of “accelerators”
- Memory & storage will be even more non-uniform:



- **Programmer may remain tasked with placement and motion.**
 - If we’re lucky motion could be moved into “virtual memory”.
 - **OpenMP 5.0** might not work as well as vendor languages in 2021/2, but **longer term it looks the right way to go.**
- For **Machine Learning consider a commercial sector code** like **Google TensorFlow** or **Baidu Paddle.**

Summary

- **HPC systems are increasingly difficult to programme and will remain so**
 - **Underlying physics & locality is driving “compartmentalisation” and “parallelism”**
 - **You canny change the speed of light, Captain!**
- **Exploiting them for science will take real effort**
- **Think about what is required to deliver the science you propose to do in Snowmass**