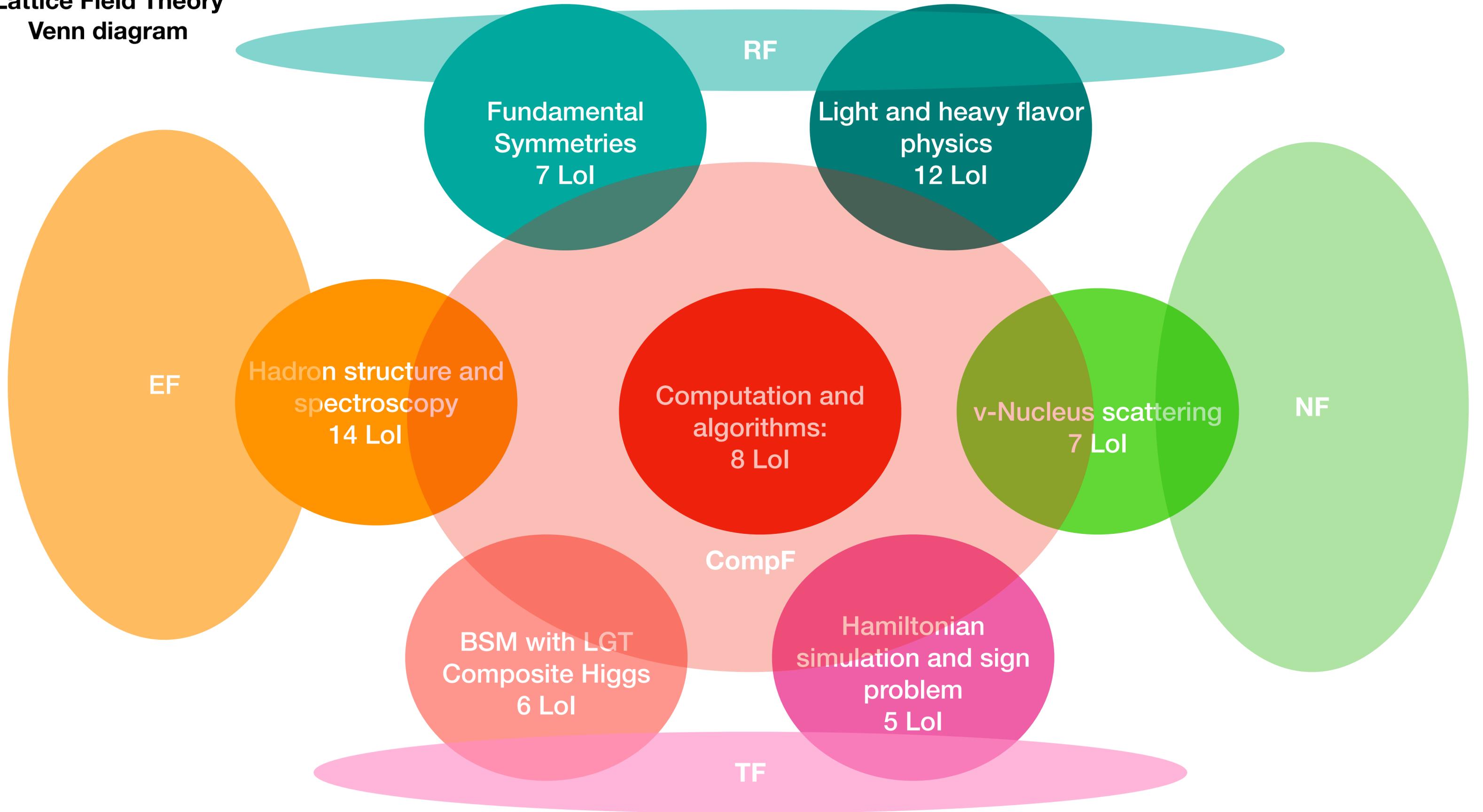


**Lattice Field Theory
Venn diagram**



Lols

SNOWMASS21-CompF2_CompF0-TF5_TF0_Rich_Brower-133.pdf	Multi-scale QCD algorithms Beyond the Exascale
SNOWMASS21-CompF2_CompF4-TF5_TF0-007.pdf	FPGA for HPC - exploring the possibilities of an alternative
SNOWMASS21-CompF2_CompF6-EF5_EF6-NF6_NF8-RF1_RF7-TF5_TF11_Kronfeld-017.pdf	Numerical Lattice Gauge Theory
SNOWMASS21-CompF3_CompF2-TF5_TF0-061.pdf	Machine learning for sampling in lattice quantum field theory
SNOWMASS21-CompF3_CompF2-TF5_TF0_Akio_Tomiya-131.pdf	Machine Learning and Lattice QCD
SNOWMASS21-TF5_TF0-CompF2_CompF0_DeTar-032.pdf	Algorithms and Software in Support of Computational HEP and NP at the Exascale and Beyond
SNOWMASS21-TF5_TF0-CompF2_CompF4_Boyle-030.pdf	Chiral Lattice Fermions and the Computational Frontier
SNOWMASS21-TF5_TF0-CompF4_CompF0-065.pdf	Lattice Quantum Chromodynamics on FPGA hardware

- USQCD whitepaper cover letter
- “Conventional” algorithms(multiscale solvers)
- “Conventional” HPC software development

- (2) Machine learning (black-box) RG / hot-cold / trivialising map evolution proposals

- (2) FPGA with OpenCL - based on Korcyl work
 - I’d personally say just use SyCL, and I asked that question at APLAT parallel session !
 - In principal such a reconfigurable dataflow is interesting, but data motion the bigger challenge
 - ASICs even more efficient than FPGA

Status and Future Perspectives for Lattice Gauge Theory Calculations to the Exascale and Beyond

Bálint Joó,^{1,*} Chulwoo Jung,^{2,†} Norman H. Christ,³ William Detmold,⁴
Robert G. Edwards,¹ Martin Savage,⁵ and Phiala Shanahan⁴

(USQCD Collaboration)

¹Theory Center, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606

²Physics Department, Brookhaven National Laboratory, Upton, NY 11973

³Department of Physics, Columbia University, New York, NY 10027

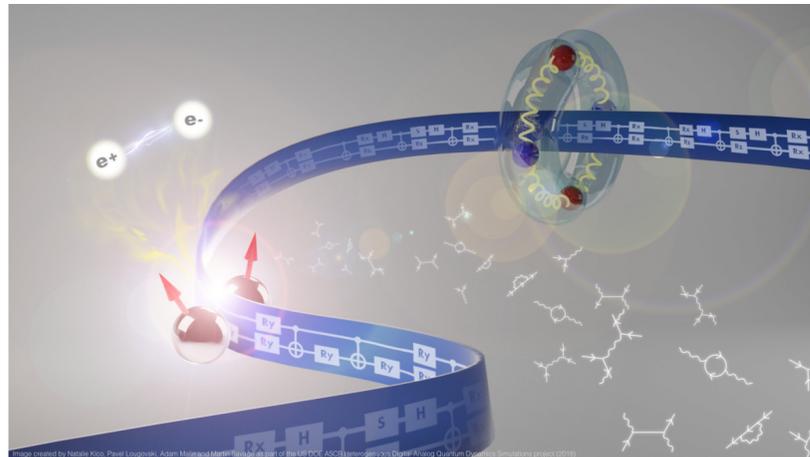
⁴Department of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139

⁵Institute for Nuclear Theory, University of Washington, Seattle, WA 98195-1550

(Dated: November 26, 2019)

Abstract

In this and a set of companion whitepapers, the USQCD Collaboration lays out a program of science and computing for lattice gauge theory. These whitepapers describe how calculation using lattice QCD (and other gauge theories) can aid the interpretation of ongoing and upcoming experiments in particle and nuclear physics, as well as inspire new ones.



* Editor, bjoo@jlab.org
† Editor, chulwoo@bnl.gov

- Excellent and significant contribution & prior work
 - Certainly needs updated, but very useful
 - Reflect proliferation of programming models & difficulty / implications for manpower

- Somewhat light on machine learning

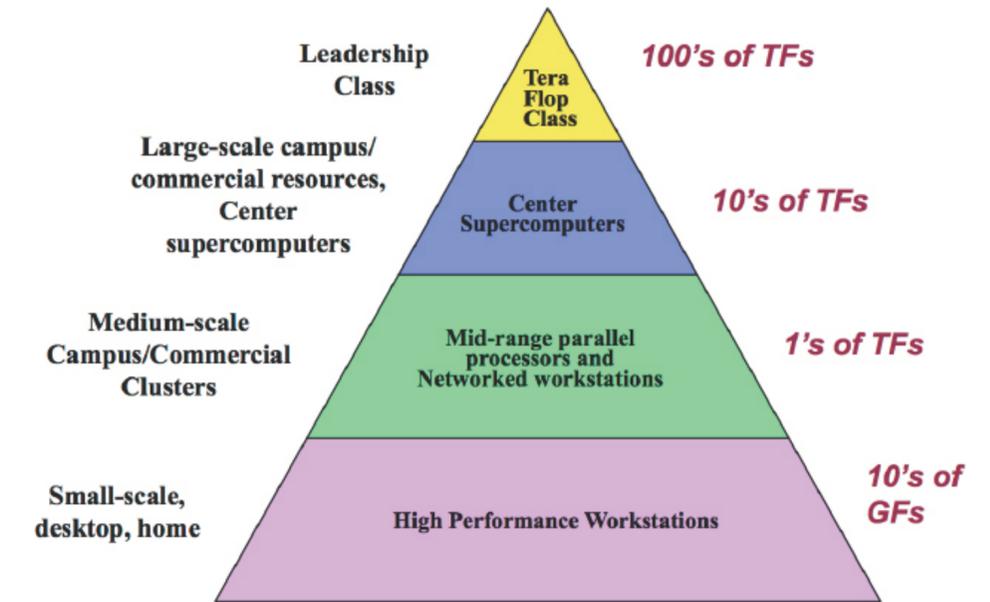
Recommendations beyond ECP

- Key recommendations from ECP/Transition ASCAC report
- Recommendation A.3. “Collaborative application support” “Transition ECP application into SciDAC-like ... joint [funding from ASCR and application home organizations](#)”. Past SciDAC program was critical to create components of current LQCD codes.
- Recommendation C.2. “[Retain the current workforce](#)”
- Recommendation C.3. “[Strengthen ties to universities and the ecosystem](#)” Here “ecosystem” refers to software technology experts and vendor system engineers. Universities are the pipeline for providing the young talent needed to sustain the software effort.
- Recommendation C.4. “[Create career paths for scientific software professionals](#)” For Lattice QCD, lab professionals at ANL, BNL, Fermilab, and JLab have been the backbone for expertise and continuity in LQCD software development.
- See recommendations “ECP/Transitions” ASCAC subcommittee chaired by Prof. Roscoe Giles of Boston University provide a crucial template to sustain lattice field theory beyond 2023.
- https://science.osti.gov/-/media/ascr/ascac/pdf/meetings/202001/EXASCALE_Transition_Presentation.pdf?

Themes I think we need to pass a message on

- **USQCD cluster computing - defend the middle tiers of the Branscomb pyramid**

- On going software development matched to hardware evolution
- On going algorithm development matched to hardware evolution
 - **Critical to our science - don't assume it will be there unless we ask for it!**
- **Careers:**
 - *“The health of the field requires a similar cohort of individuals at the best universities, reflecting the intellectual vigor and potential of this area to contribute to DOE scientific goals. The creation of such positions can be stimulated by DOE-funded joint, five year, tenure-track appointments. Theoretical particle physics is one of the last area of physics to recognize the importance of computation in forefront research and continued effort is urgently required to overcome this historical bias, and create a vibrant pool of skilled young faculty, and around them their PhD students and research groups.”*



- **Machine learning research programme to see whether it can improve gauge evolution**

What is the right set of white papers & “studies” to cover?

- Conventional computing - Software/Algorithms/ASCR access/Clusters/Careers
 - relationships to USQCD white paper?
- Machine learning and novel algorithms
- (Quantum Computing ; CompF6 really though)

- ECP software and algorithm submission, RBC submission
 - (conventional) software & algorithms central to our scientific exploitation of new HPC systems of increasing complexity.
 - Importance of 5 year tenure track joint faculty/lab positions for DOE to influence academic profile to deliver Snowmass programme
 - Importance of long term lab software development and algorithm development positions to retain skills and **staff**.

Snowmass is an important opportunity for us to say what is required for us to deliver our science

Comp F2 plan to issue a Survey to gather requirements

- ASCR computing will deliver many of our needs, *in principle*
 - USQCD clusters: must make the case clearly for local computing
 - Human effort (in addition to theory post-docs):
 - Programming it will *not* be automatic nor easy
 - Hardware barriers
 - Poor communication
 - Fragmented memory models
 - Multiple programming models
 - Algorithms will need to become much more tolerant of poor communication and use greater locality
 - Parallel MCMC streams for high statistics?

Why we shouldn't take our classical computing and algorithms for granted

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**

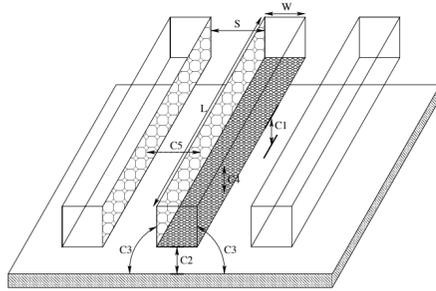
Argonne

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*
- When can we rely on OpenMP 5.0?
 - Best guess: not now, but probably on Snowmass timescale

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

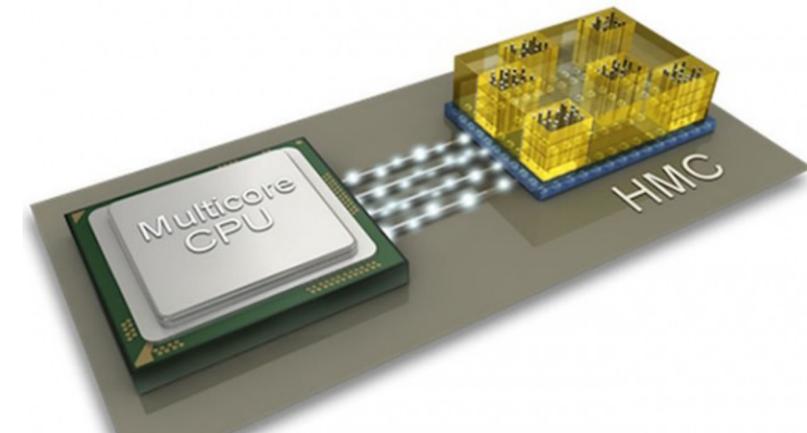
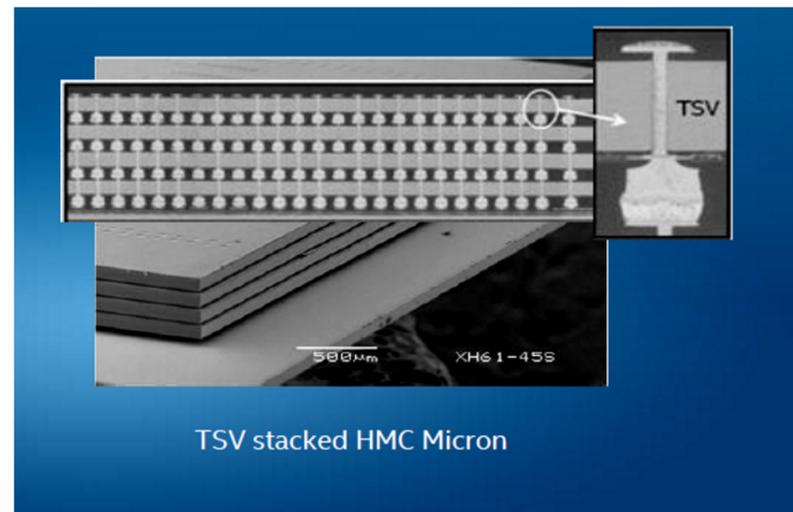
- Low number of long range “broad” wires (bus/interconnect)
- High number of short range “thin” wires

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

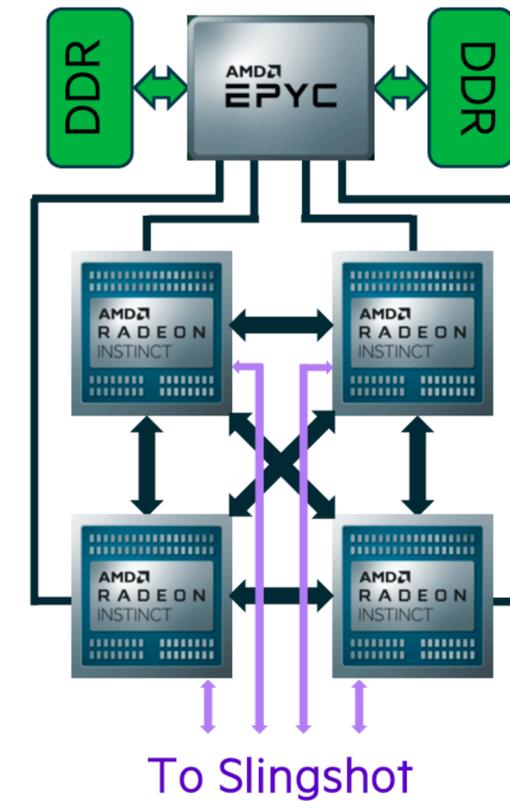
ALL-TO-ALL CONNECTIVITY WITHIN NODE
Low latency, high bandwidth

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

DELIVERED IN 2021

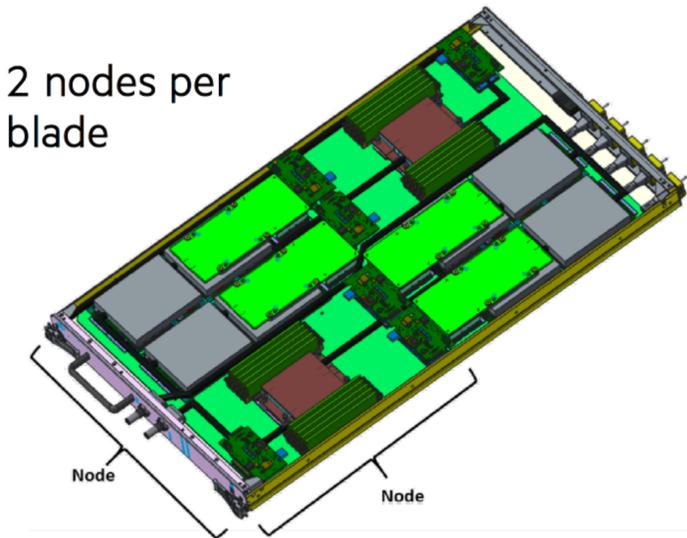
U.S. DEPARTMENT OF ENERGY | Argonne NATIONAL LABORATORY | intel | CRAY

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



AMD GPU
(ORNL)

2 nodes per blade



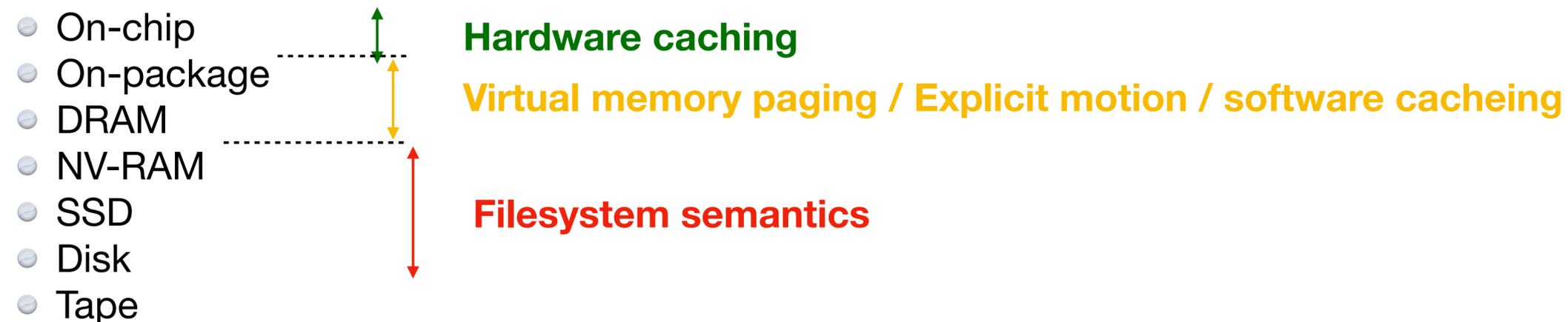
COPYRIGHT 2020 HPE

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.**
 - **Network bandwidth is the hardest thing to address**
- **Computer node organisation will be at least as complex.**
 - **Multiple distinct memories on node**
 - Distinct fast intra-node and slower 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 & 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.**