LANL Platform Planning and Update

HPC User Forum 2023

Gary Grider LANL

Input from
Jim Lujan, Jason Pruet,
Steve Poole, and
Galen Shipman

08/2023

LA-UR-23-29918

Different Missions/Different Architectures

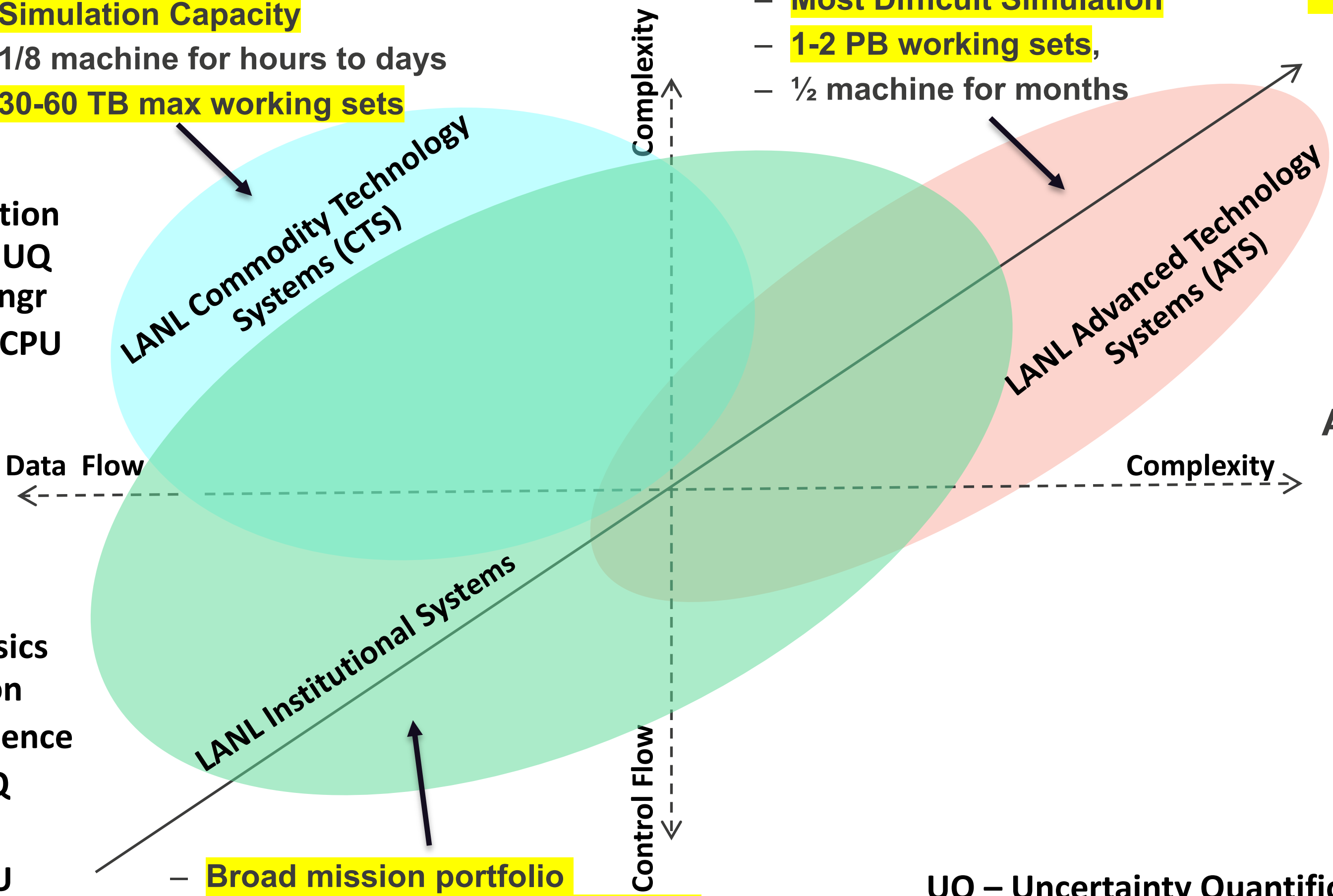
Architecturally Challenging

- Simulation Capacity
- 1/8 machine for hours to days
- 30-60 TB max working sets

- Most Difficult Simulation
- 1-2 PB working sets,
- ½ machine for months

1-2D Simulation
Small/Med UQ
Weapons Engr
Commodity CPU

Full 3D, Multi-Physics, Multi-link scale, Extreme Multi-resolution, Irregular/Sparse Simulation, Advanced Features



Complexity
LANL Commodity Technology Systems (CTS)
LANL Advanced Technology Systems (ATS)
Data Flow
Complexity
LANL Institutional Systems
Control Flow

Single Physics Simulation
Weapons Science
Large UQ
ML/AI
GPU/TPU

Edge/IOT
Dataflow
FPGA, Network Processing

- Broad mission portfolio (Simulation, AI/ML/DL, Analytics)
- 100-200 TB max working sets
- ¼ machine for days to weeks

UQ – Uncertainty Quantification

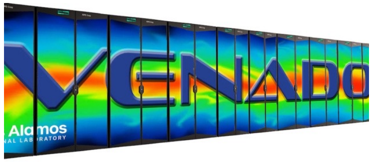
Different NNSA Labs have different mission drivers too! LLNL ATS != LANL ATS, etc.

Concept courtesy Giri Chukkapalli - Nvidia

LANL HPC Systems

Current/Older Systems

CTS - Fire: Penguin/Intel 1104 Broadwell nodes
CTS - Ice: Penguin/Intel 1104 Broadwell nodes
CTS - Cyclone: Penguin/Intel 1118 Broadwell nodes
Viewmaster3: HPE, Visualization



ATS - Trinity – HPE/Intel ~20,000 Intel Haswell/KNL nodes 2PB mem 4 PB flash (20000 sockets)

Moon: Appro/Intel 1600 Ivybridge nodes (systems testbed)

Badger: Penguin/Intel 660 Broadwell nodes
Kodiak: Penguin/AMD 128 Rome + 4 A100 GPU nodes
Snow: Penguin/Intel 368 Broadwell nodes
Trinitite: HPE/Intel 100 Haswell 100 KNL nodes
Darwin: test bed for apps on architectures

Red – Secure (ATS/CTS)
Blue – Open/Institutional
Large machine

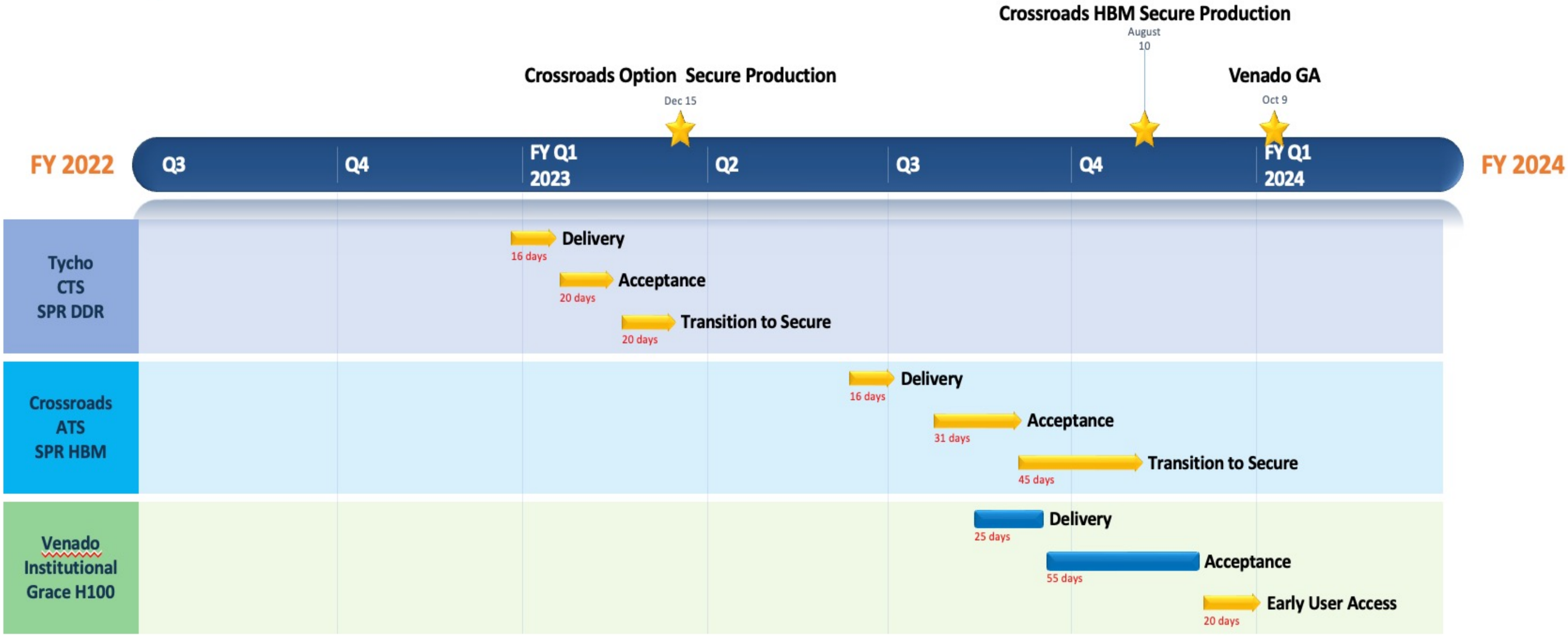
Near Future/New/Recently Refreshed FY22-23

Institutional - Chicoma: 1792 AMD Rome +(188 Milan+ 4 Nvidia A100) node (3772 sockets+ GPUs)_

ATS - Crossroads HPE/Intel ~6144 Intel SPR HBM nodes (12288 sockets)
CTS - Tycho HPE/Intel ~ 2684 Intel SPR DDR nodes (5368 sockets)
Institutional - Venado HPE/Nvidia XXXX Grace-Grace YYYY Grace+H100 nodes
Rocinante HPE/Intel 380 SPR-DDR (760 sockets) 128 SPR-HBM nodes (256 sockets)

TYCHO

VENADO



Crossroads HBM Secure Production
August 10

Crossroads Option Secure Production
Dec 15

Venado GA
Oct 9

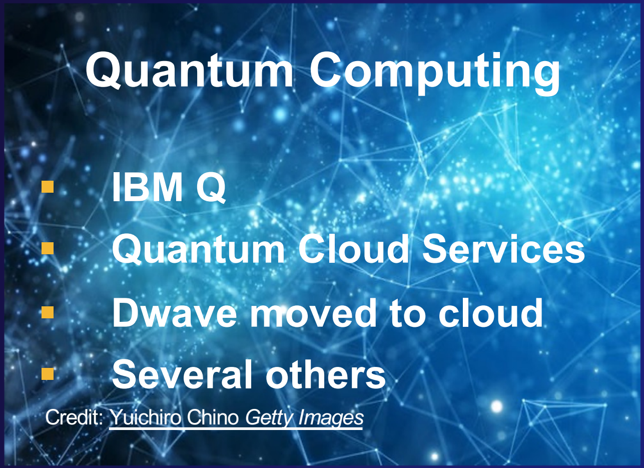
FY 2022 | Q3 | Q4 | FY Q1 2023 | Q2 | Q3 | Q4 | FY Q1 2024 | FY 2024

Tycho CTS SPR DDR
- Delivery — 16 days
- Acceptance — 20 days
- Transition to Secure — 20 days

Crossroads ATS SPR HBM
- Delivery — 16 days
- Acceptance — 31 days
- Transition to Secure — 45 days

Venado Institutional Grace H100
- Delivery — 25 days
- Acceptance — 55 days
- Early User Access — 20 days

Existing Institutional Computing Resources

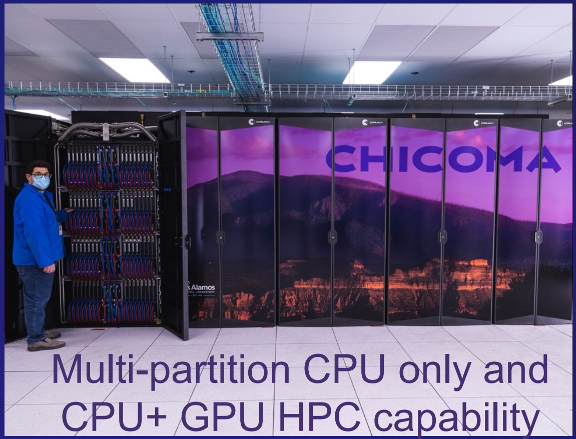


Quantum Computing

- IBM Q
- Quantum Cloud Services
- Dwave moved to cloud
- Several others

Credit: Yuichiro Chino *Getty Images*

Extending reach to a broader base of both non-gate-based and gate-based quantum compute vendors



Multi-partition CPU only and CPU+ GPU HPC capability

1792 Dual Socket Nodes
Rome 64c 2.6GHz
512 GiB/Node

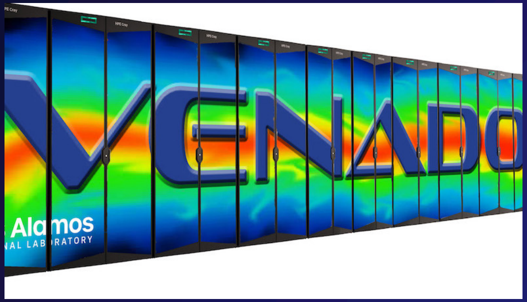
188 nodes
1x CPU, 4x GPU,
Milan 64c 2.0GHz/A100
96 40GB Blades (256 GiB/Node)
22 80GB Blades (512 GiB/Node)

Chicoma

- **Mostly large memory CPU only**
- **Some amount of A100 resource including some large memory A100**
- **Complimentary to Venado (new institutional machine)**

Enable a wide range of open science projects, workloads and applications, from early discovery investigations to large-scale experimentation with customer-provided data sets

An Institutional Heterogeneous System



- CPU-GPU node type: NVIDIA Hopper (H100) GPU + NVIDIA Grace CPU connected with NVLink-C2C to provide a fast coherent, shared memory address space.
- CPU-only node type: NVIDIA Grace CPU Superchip 2 Arm-based CPUs, connected coherently through the high-bandwidth, low-latency, low-power NVIDIA NVLink-C2C interconnect, with up to 144 high-performance Arm Neoverse cores with scalable vector extensions and a 1 terabyte-per-second memory subsystem.

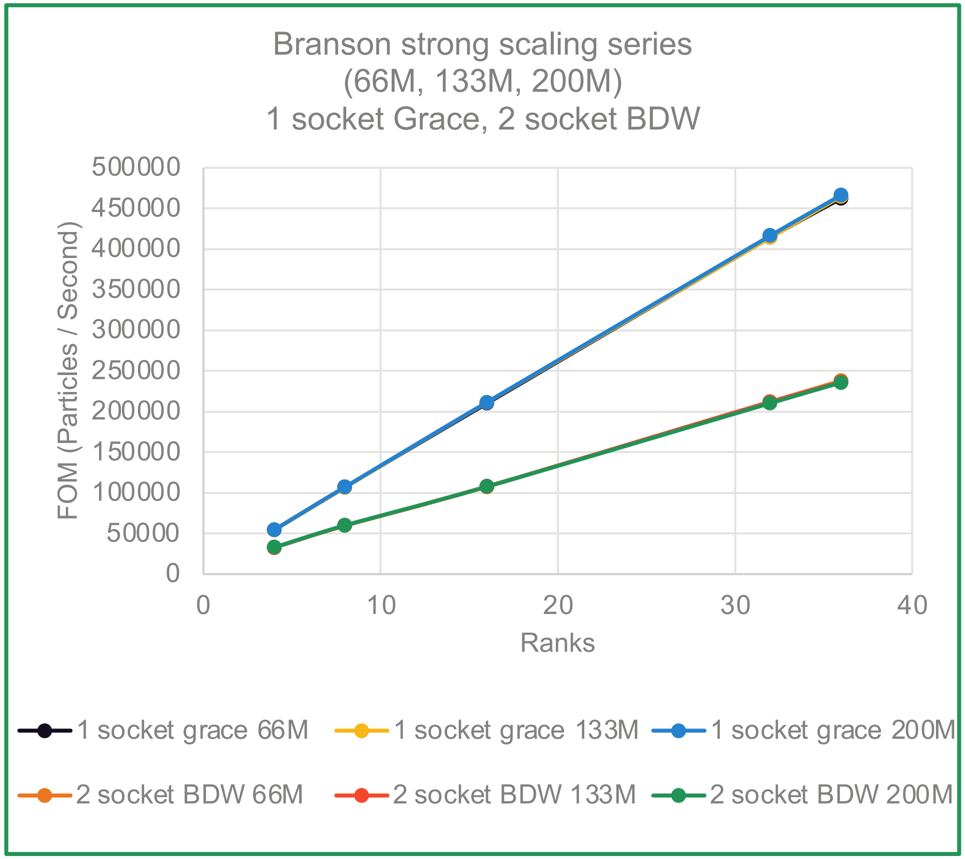
- ~80% cpu+gpu
- ~20% cpu only
- Full retical CPU+GPU with high bandwidth coherent address space
- Latest GPU
- First real HPC class Arm CPU in the US
- CPU-CPU may enable strong scaling studies
- Complementary to Chicoma (which is mostly cpu)

The first large system in the U.S. to be powered by NVIDIA Grace CPU technology

Early Grace Measurements on Branson

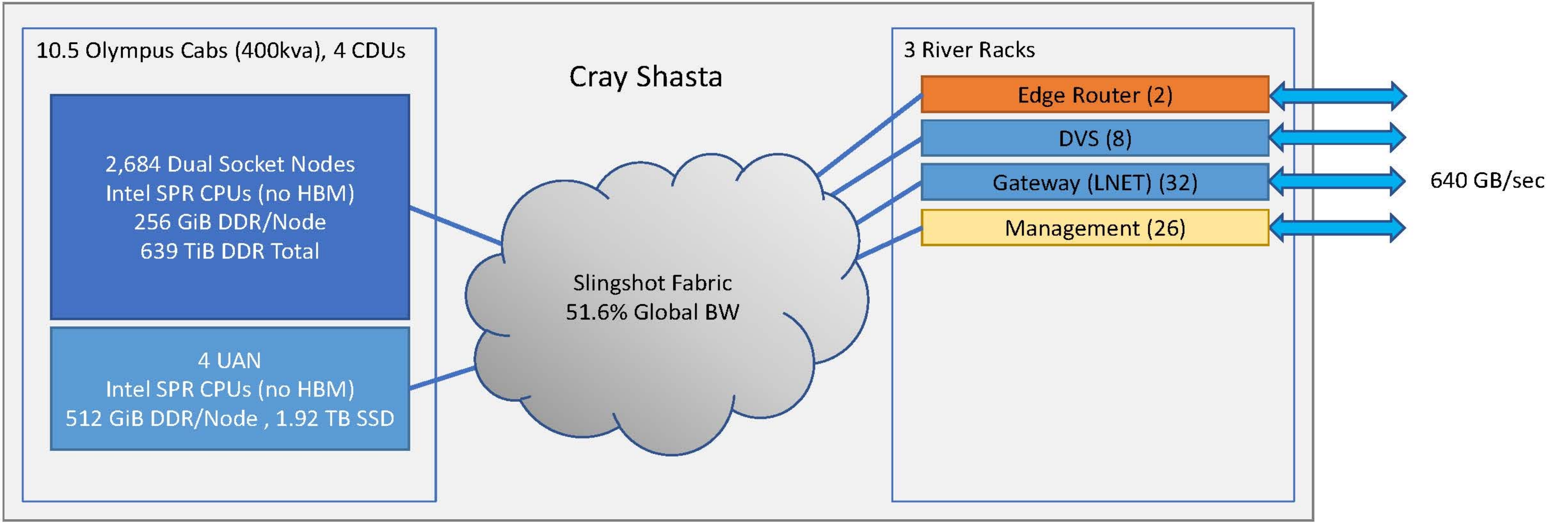
Branson is a proxy application for parallel Monte Carlo transport. It contains a particle passing method for domain decomposition.

1. **Intel Broadwell dual socket: Intel oneAPI-2023.1.0**
2. **Nvidia Grace single socket: GCC 12.3.1 (3.8x).**
3. **Nvidia Grace strong scales well**
4. **Both processors are not sensitive to these problem sizes**


Branson strong scaling series
(66M, 133M, 200M)
1 socket Grace, 2 socket BDW

FOM: Particles / Second

Programmatic Computing CTS Commodity Tech System
Tycho Secure SPR DDR / Rocinante open SPR DDR and HBM



10.5 Olympus Cabs (400kva), 4 CDUs

2,684 Dual Socket Nodes
Intel SPR CPUs (no HBM)
256 GiB DDR/Node
639 TiB DDR Total

4 UAN
Intel SPR CPUs (no HBM)
512 GiB DDR/Node , 1.92 TB SSD

Cray Shasta

Slingshot Fabric
51.6% Global BW

3 River Racks

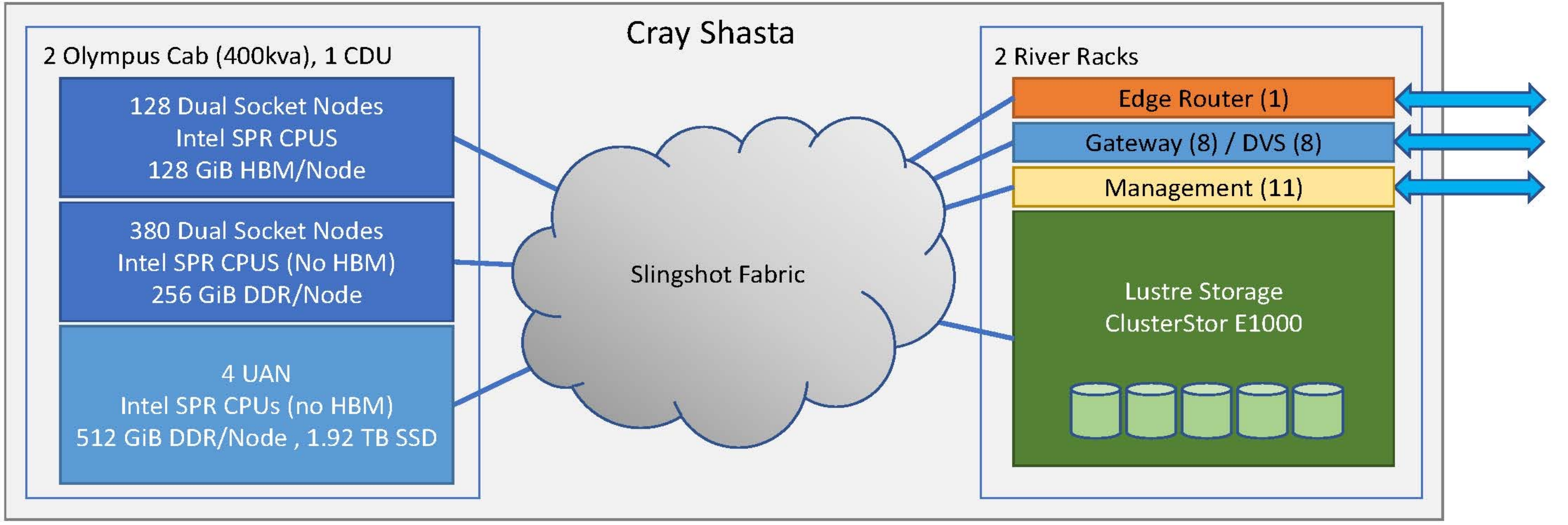
Edge Router (2)

DVS (8)

Gateway (LNET) (32)

Management (26)

640 GB/sec



Cray Shasta

2 Olympus Cab (400kva), 1 CDU

128 Dual Socket Nodes
Intel SPR CPUS
128 GiB HBM/Node

380 Dual Socket Nodes
Intel SPR CPUS (No HBM)
256 GiB DDR/Node

4 UAN
Intel SPR CPUs (no HBM)
512 GiB DDR/Node , 1.92 TB SSD

Slingshot Fabric

2 River Racks

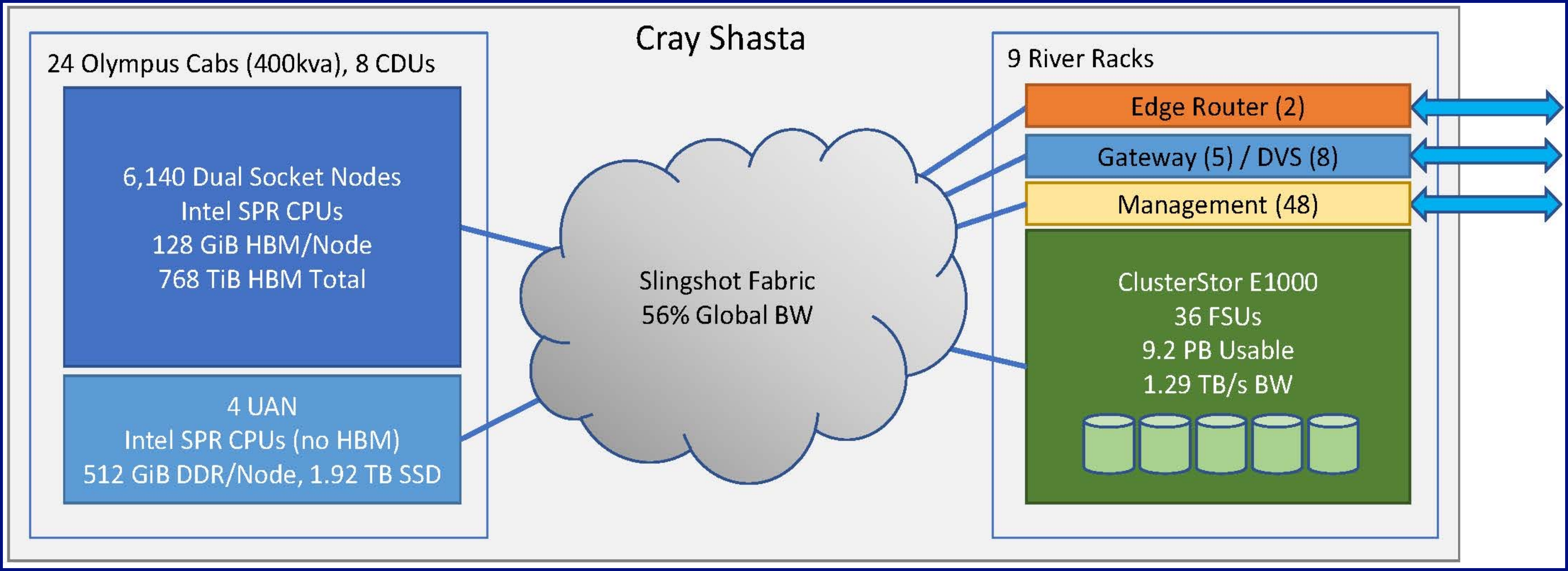
Edge Router (1)

Gateway (8) / DVS (8)

Management (11)

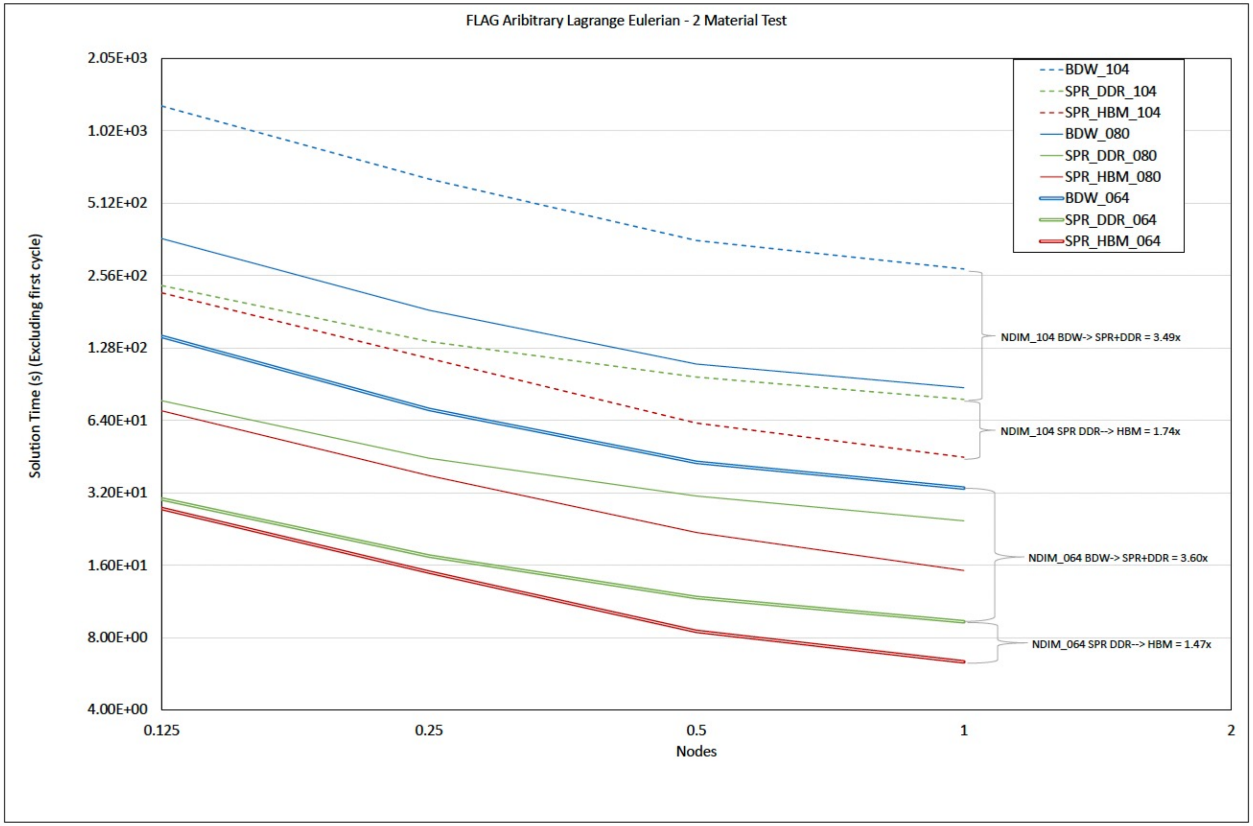
Lustre Storage
ClusterStor E1000

Programmatic Computing ATS Advanced Tech System
Crossroads Secure SPR HBM



- ATS follow on to Trinity
- All Flash File System
- HBM only

SPR DDR/HBM Initial Performance Experience



From FugakuNEXT talk

- Mix of cpu, vector, and matrix
- Memory BW
- Reasonable way to program

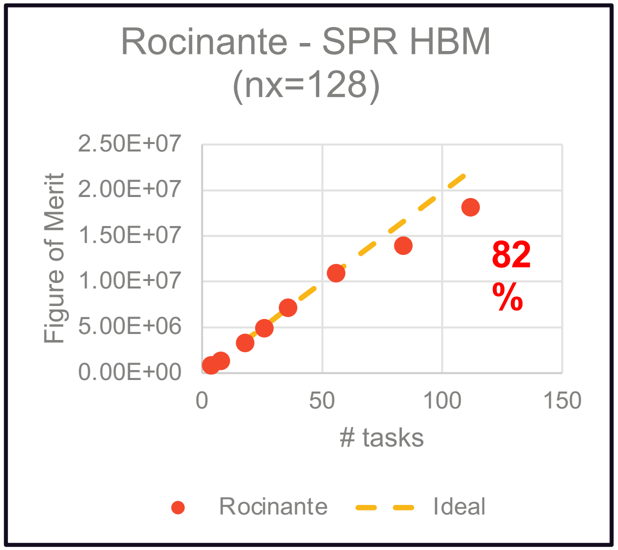
- Everything is memory performance bound except training

Parthenon: **6X** on SPR+HBM -- **4.2X** on SPR DDR5 (**43%** improvement on HBM over DDR5)
UMT: **5.9X** on SPR+HBM -- **3.2X** on SPR DDR5 (**84%** improvement on HBM over DDR5)
SPARTA: **9X** on SPR+HBM -- **4.1X** on SPR DDR5 (**120%** improvement on HBM over DDR5)
AMG2023: **7.6X** on SPR+HBM -- **4.2X** on SPR DDR5 (**105%** improvement on HBM over DDR5)

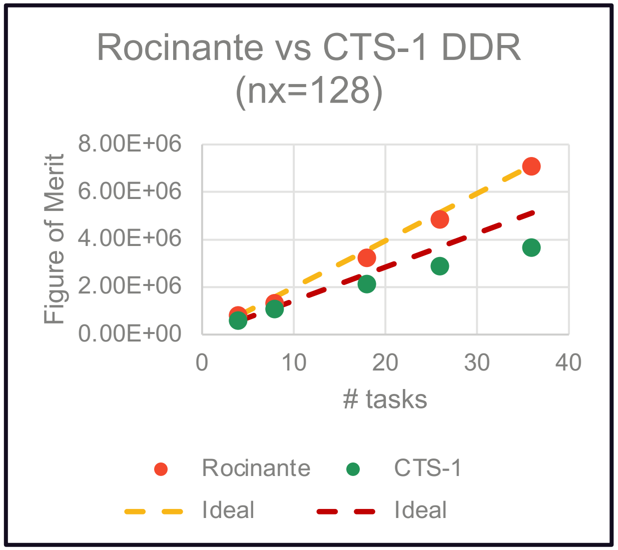
Parthenon-VIBE

The Parthenon-VIBE benchmark solves the Vector Inviscid Burgers' Equation on a block-AMR mesh. Block size of 16^3 balances memory footprint and computational efficiency.

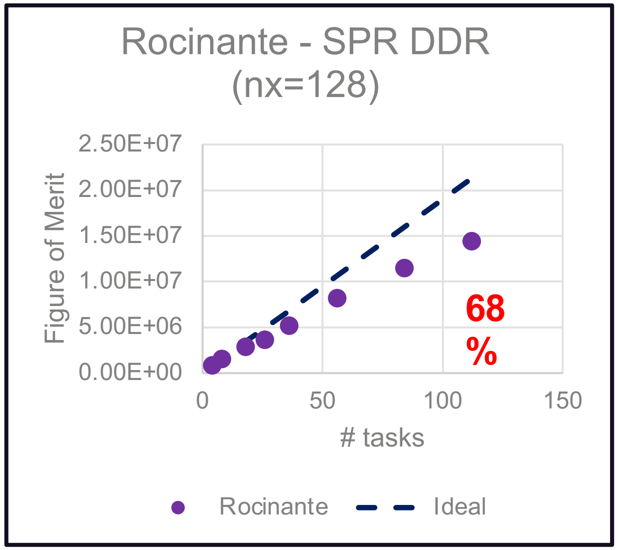
1. **SPR HBM: Intel oneAPI-2023.1.0 (6X), Intel classic-2021.9.0 (4.4X), gnu-12.2.0 (4.3X), cce-15.0.1 (5X).**
2. **SPR DDR: Intel oneAPI-2023.1.0 (4.2X), Intel classic-2021.9.0 (3.6X), gnu-12.2.0 (3.8X), cce-15.0.1 (4X).**
3. Scales best on SPR HBM using Intel oneAPI compiler (shown below) (**82%**).



Rocinante - SPR HBM (nx=128)



Rocinante vs CTS-1 DDR (nx=128)



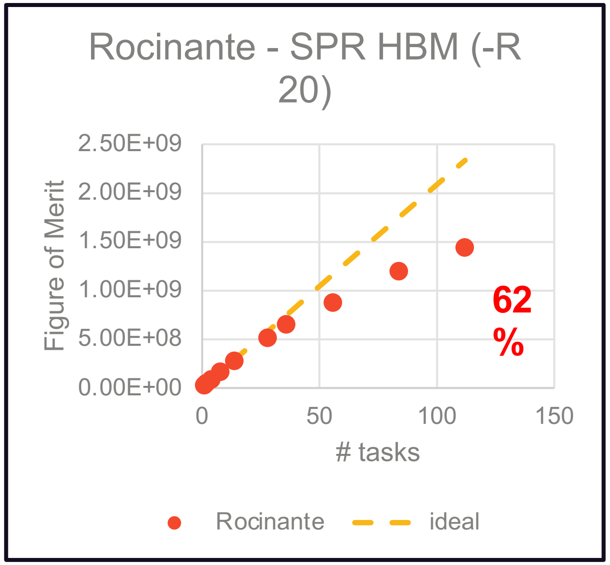
Rocinante - SPR DDR (nx=128)

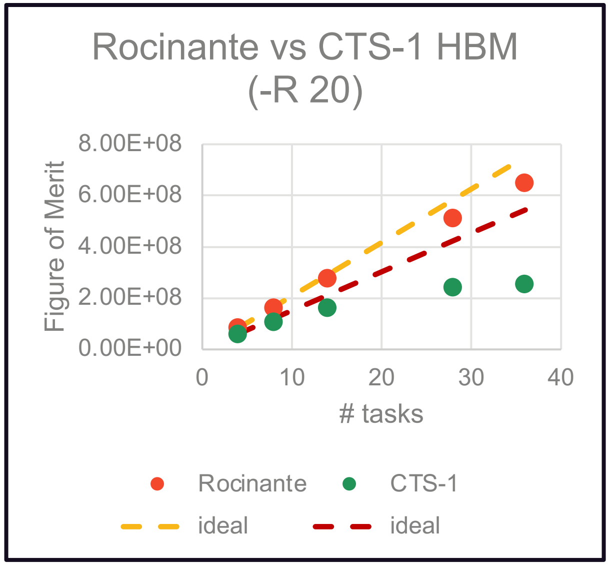
FOM: Cell zone-cycles / wallsecond which is the number of AMR zones processed per second.

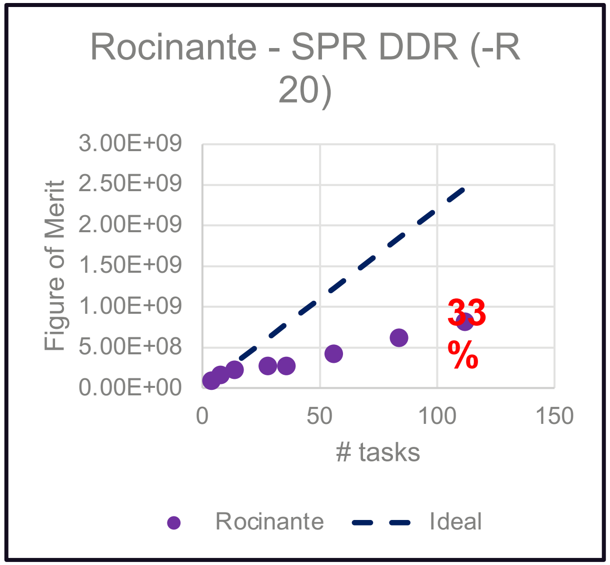
UMT

UMT (Unstructured Mesh Transport) is **an LLNL** proxy application that solves a thermal radiative transport equation using discrete ordinates (Sn).

1. **SPR HBM: Intel classic-2021.9.0 (5.9X), Intel oneAPI-2023.1.0 (5.8X).**
2. **SPR DDR: Intel classic-2021.9.0 (3.2X), Intel oneAPI-2023.1.0 (3.2X).**
3. **Scales best on SPR HBM using Intel classic compiler (shown below) (62%).**







FOM: Number of unknows (cells, corners, directions, energy bins) solved per second.

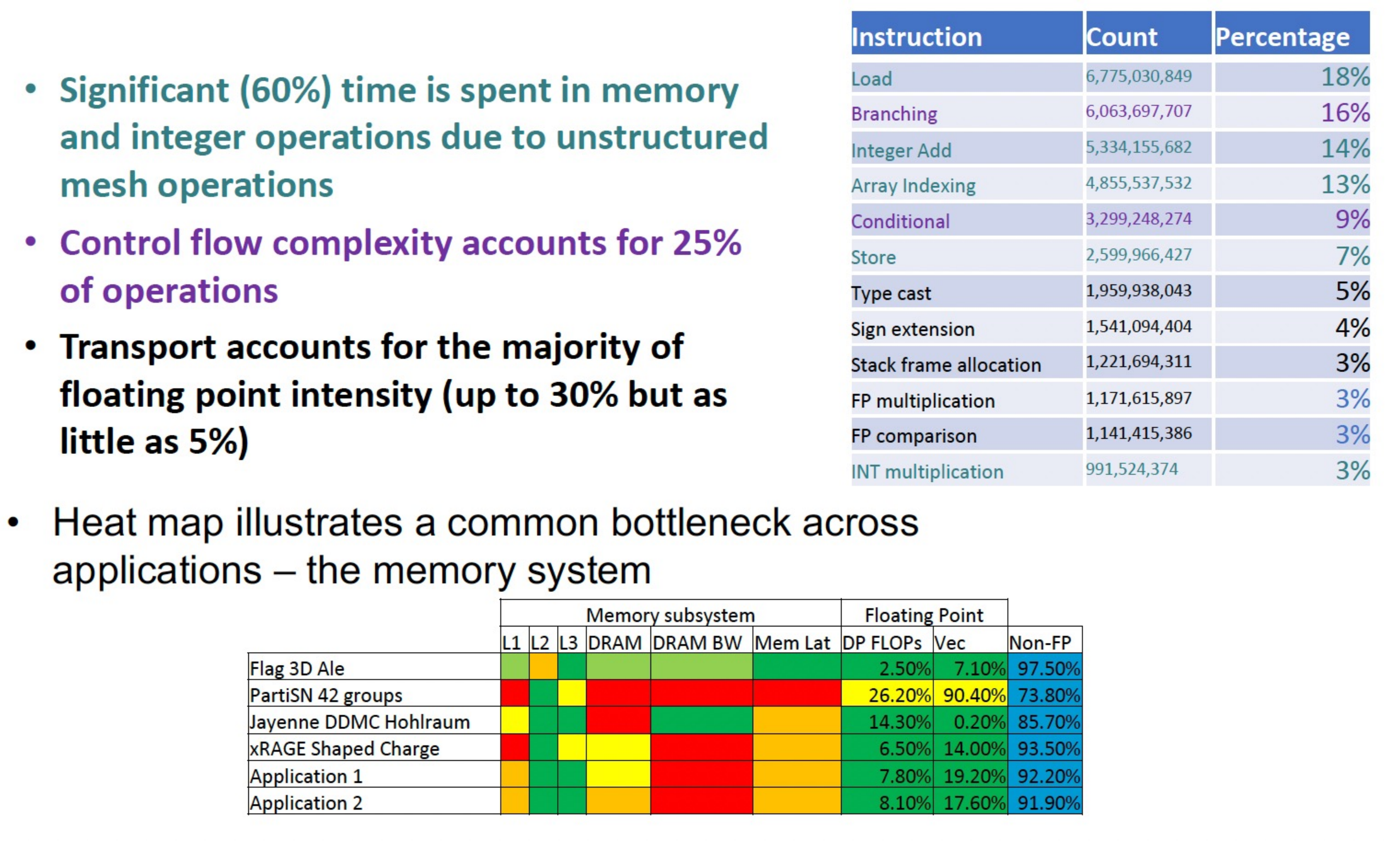
https://github.com/lanl/benchmarks
https://lanl.github.io/benchmarks

Benchmark Overview

Benchmark	Description	Language	Parallelism
Branson	Implicit Monte Carlo transport	C++	MPI + Cuda/HIP
AMG2023	AMG solver of sparse matrices using Hypre	C	MPI+CUDA/HIP/SYCL OpenMP on CPU
MiniEM	Electro-Magnetics solver	C++	MPI+Kokkos
MLMD	ML Training of interatomic potential model using HIPYNN on VASP Simulation data. ML inference using LAMMPS, Kokkos, and HIPYNN trained interatomic potential model	Python C++ C	MPI+Cuda/HIP
Parthenon-VIBE	Block structured AMR proxy using the Parthenon framework	C++	MPI+Kokkos
Sparta	Direct Simulation Monte Carlo	C++	MPI+Kokkos
UMT	Deterministic (Sn) transport	Fortran	MPI+OpenMP and OpenMP Offload

3D multi-physics AMR Problem Background

The focus of the LANL ATS platform march



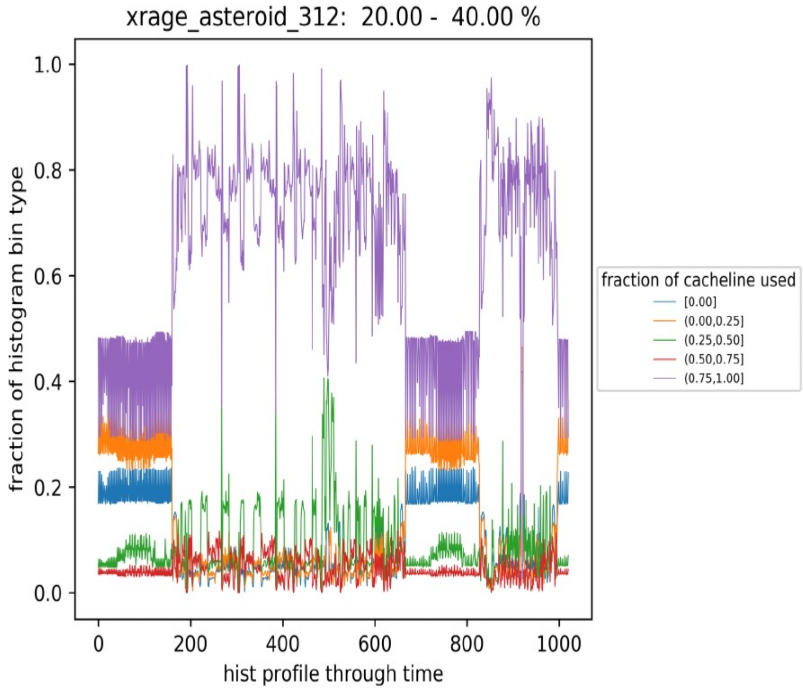
- Significant (60%) time is spent in memory and integer operations due to unstructured mesh operations
- Control flow complexity accounts for 25% of operations
- Transport accounts for the majority of floating point intensity (up to 30% but as little as 5%)

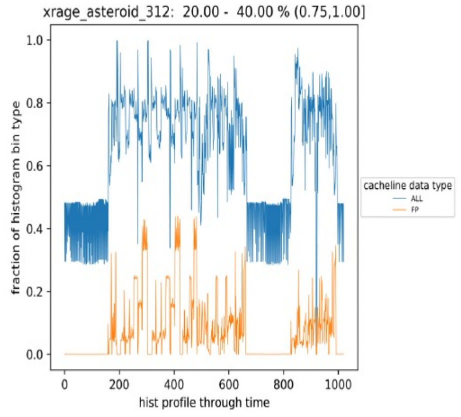
Instruction	Count	Percentage
Load	6,775,030,849	18%
Branching	6,063,697,707	16%
Integer Add	5,334,155,682	14%
Array Indexing	4,855,537,532	13%
Conditional	3,299,248,274	9%
Store	2,599,966,427	7%
Type cast	1,959,938,043	5%
Sign extension	1,541,094,404	4%
Stack frame allocation	1,221,694,311	3%
FP multiplication	1,171,615,897	3%
FP comparison	1,141,415,386	3%
INT multiplication	991,524,374	3%

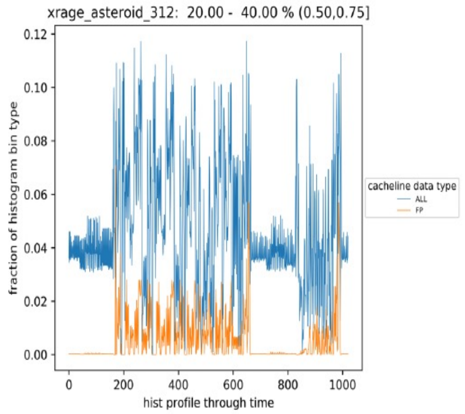
- Heat map illustrates a common bottleneck across applications – the memory system

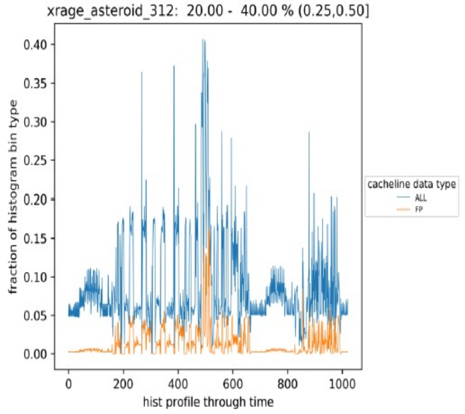
	Memory subsystem						Floating Point		
	L1	L2	L3	DRAM	DRAM BW	Mem Lat	DP FLOPs	Vec	Non-FP
Flag 3D Ale	🟩	🟧	🟩	🟩	🟩	🟩	2.50%	7.10%	97.50%
PartiSN 42 groups	🟥	🟩	🟨	🟥	🟥	🟥	26.20%	90.40%	73.80%
Jayenne DDMC Hohlraum	🟨	🟩	🟩	🟥	🟩	🟧	14.30%	0.20%	85.70%
xRAGE Shaped Charge	🟥	🟩	🟨	🟨	🟥	🟧	6.50%	14.00%	93.50%
Application 1	🟧	🟩	🟩	🟨	🟥	🟧	7.80%	19.20%	92.20%
Application 2	🟧	🟩	🟩	🟧	🟥	🟧	8.10%	17.60%	91.90%

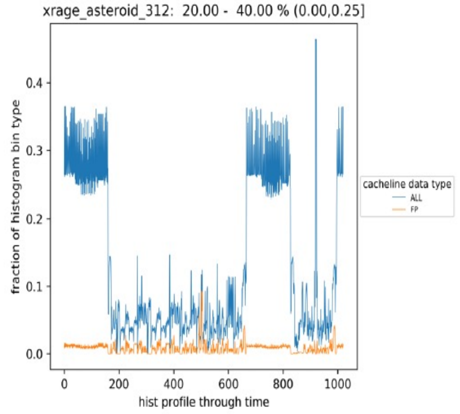
It’s all about memory access, NOT ABOUT FLOPS
High branching and horrible memory access favored CPU in Crossroads Bids


xrage_asteroid_312: 20.00 - 40.00 %
fraction of histogram bin type
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
hist profile through time
fraction of cacheline used
[0.00]
(0.00,0.25]
(0.25,0.50]
(0.50,0.75]
(0.75,1.00]


xrage_asteroid_312: 20.00 - 40.00 % (0.75,1.00]
fraction of histogram bin type
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
hist profile through time
cacheline data type
ALL
FP


xrage_asteroid_312: 20.00 - 40.00 % (0.50,0.75]
fraction of histogram bin type
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
200
400
600
800
1000
hist profile through time
cacheline data type
ALL
FP

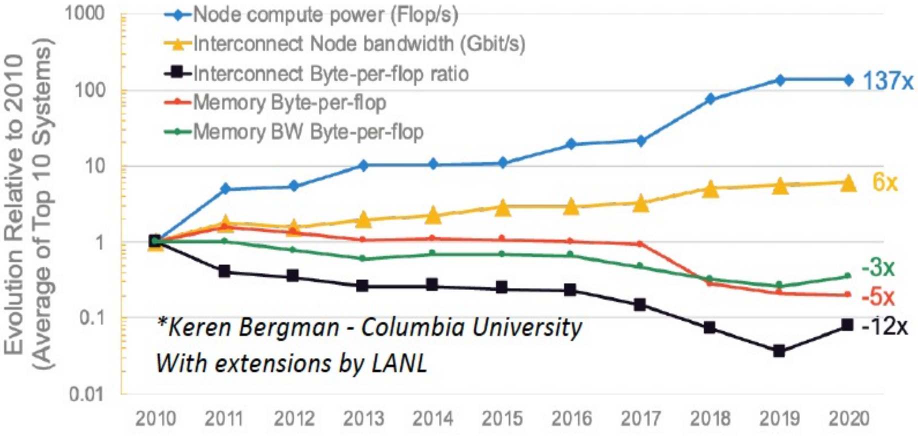

xrage_asteroid_312: 20.00 - 40.00 % (0.25,0.50]
fraction of histogram bin type
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
200
400
600
800
1000
hist profile through time
cacheline data type
ALL
FP


xrage_asteroid_312: 20.00 - 40.00 % (0.00,0.25]
fraction of histogram bin type
0.4
0.3
0.2
0.1
0.0
0
200
400
600
800
1000
hist profile through time
cacheline data type
ALL
FP

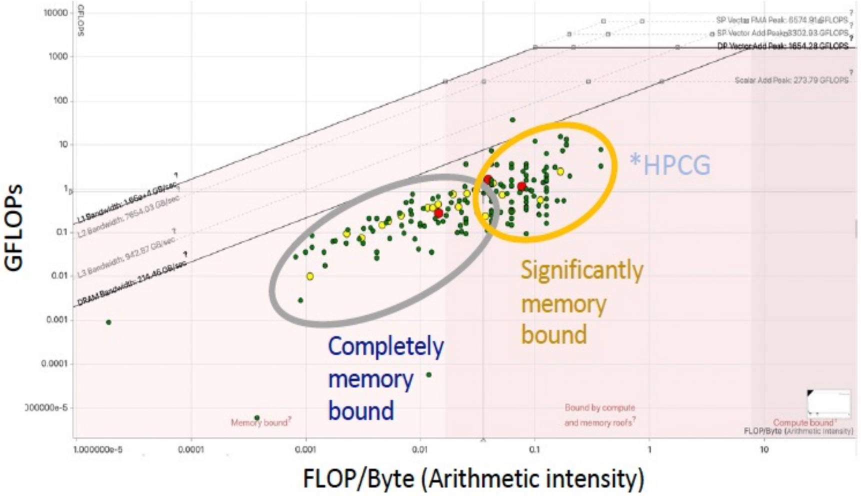
- HPC systems are becoming less balanced

- Amdahl's law makes the massively parallel core path difficult
  - Branching exacerbates this

- Less than 1% of the flops are useful
  - Memory capacity/bandwidth, and branching efficiency are **much** more important
  - HPCG is an upper limit on arithmetic intensity in many codes



*Keren Bergman - Columbia University
With extensions by LANL

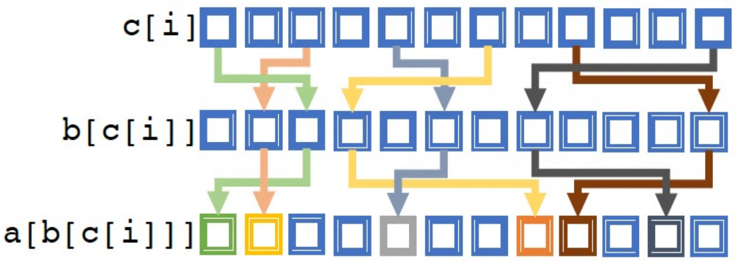


*HPCG

Completely memory bound

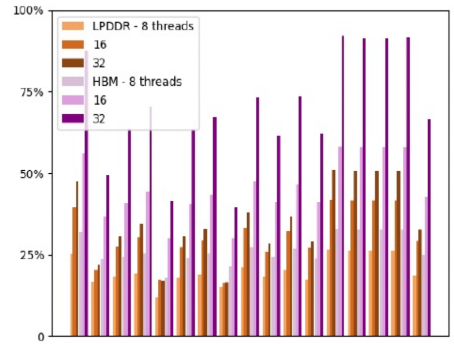
Significantly memory bound

Ultimately, we want to move only the data we need for computation



Hardware simulation of representative workloads shows HBM2e (ATS-3) will significantly help our workloads

- Brute force: still moving all the arrays for indirection across the bus



Instead of a hammer (bandwidth) we would like to explore adding more intelligence in the memory controller to support complex S/G

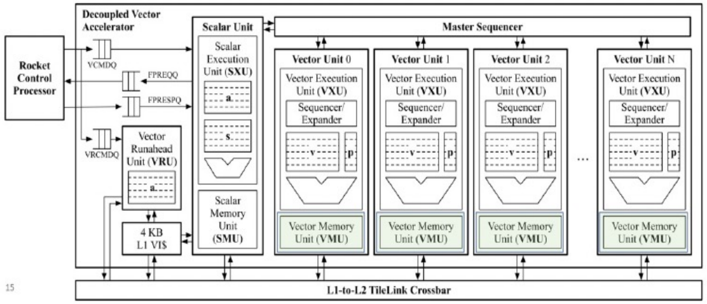
<u>Pseudo assembly code</u>

Single indirection array:

```
vec_ind_load(register vec_dst, register cnt,
             register Baddr,
             register Caddr)
```

Two indirection arrays:

```
vec_ind2_load( register vec_dst, register cnt,
             register Aaddr, register Baddr,
             register Caddr)
```

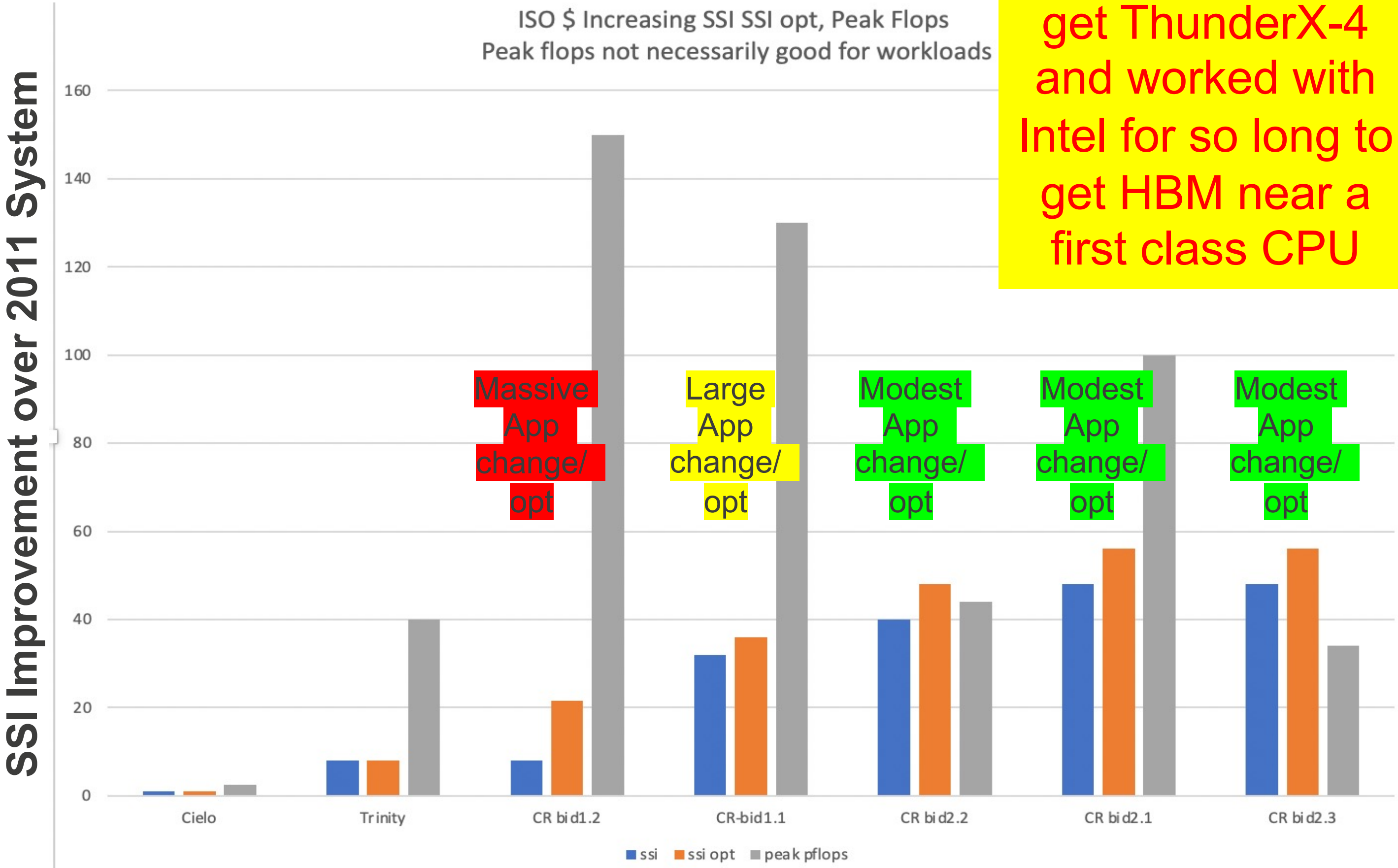


**Hwacha (UC-Berkeley)*

Buying Crossroads for complex apps/workflows was much harder than buying for Peak Ops or HPL

This is why we worked so hard to get ThunderX-4 and worked with Intel for so long to get HBM near a first class CPU

- SSI apps are provided along with workflows
- **Vendors can change apps but must honor the problem**
- Measure SSI, SSI-Opt and changes/ implications on real code base.



Massive App change/ opt

Large App change/ opt

Modest App change/ opt

Modest App change/ opt

Modest App change/ opt

2011 AMD CPU System	2016 Intel CPU System	Cpu+ GPU Trendy App Driven	CpuHBM +Accel	CpuHBM Tailored roadmap	CpuHBM+ MMU Tailored processor	CpuHBM Tailored roadmap

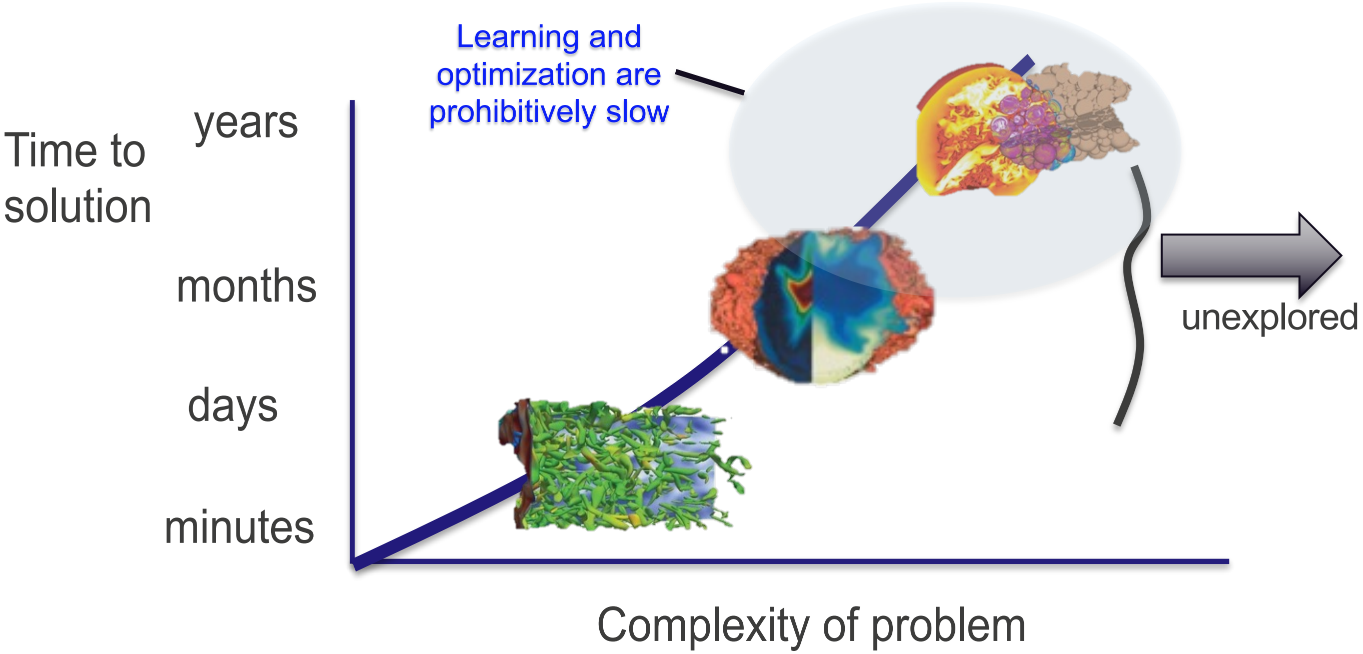
-------------------------------Crossroads RFP bids ---

Blue No App Change
Orange App Change/Optimization
Gray Peak Flops

Was it worth getting higher BW on CPU's? What is next?

- Given sparse, irregular, branchy and STRONG SCALED
- Given crossroads bids/ process allowing vendors to optimize
- On production code, >4X on SPR-DDR from Broadwell, believe >6X with SPR-HBM
  - it follows with BW due to sparse/indirection
  - Why not more with HBM *(see Bandwidth Limits in the Intel Xeon Max (Sapphire Rapids with HBM Processors, ISC 2023 IXPUG Workshop*, John *McCalpin, TACC – Intel first gen HBM integration)*
- How often 6X-9X between generations with little to no code change?
- Codes are changing for dense structures/weak scaling portions, but sparse/indirection and branchy behavior dominates
  - if we changed the dominant parts of the code – change it to what?
- We are buying *PU's to access high bandwidth memory tech, until we can engineer a more elegant solution for sparsity etc.
- Need deeper codesigned hdwr especially for broader sparsity

Impossible to possible to routine!



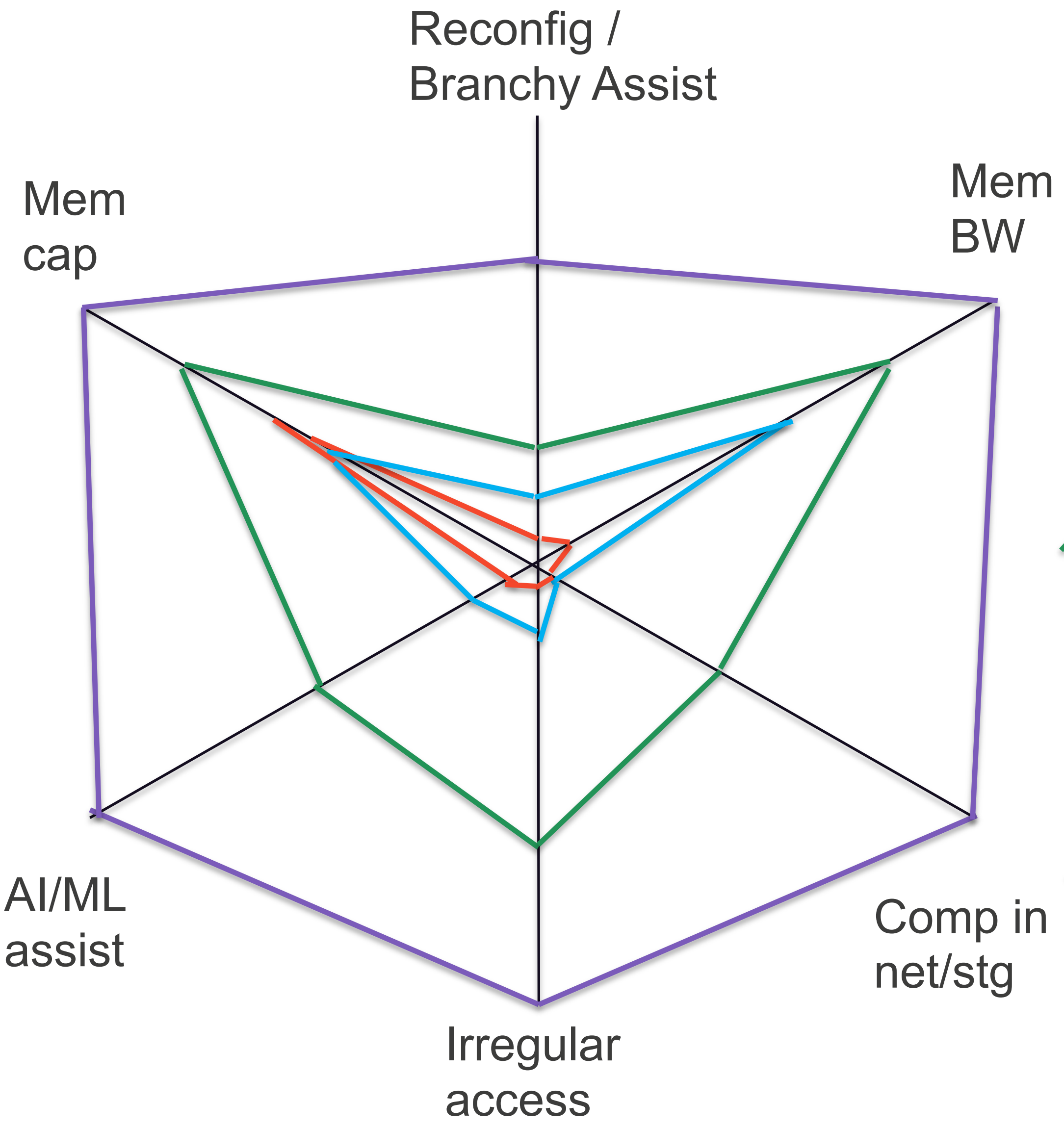
Runs that currently take 1 PB DRAM and 10k nodes / ½ million cores for 6-12 months need to run in 6-12 days

Need to move from 1% efficient to 30% efficient in 5-7 years

Leaning in on Tailoring of Architectures Activities to gain efficiency

- ATS1/Trinity 2PB Dram/Burst Buffer, big enough to run slowly
- ATS3/Crossroads Memory BW, months to weeks
- ATS5-> Irregular access acceleration, weeks to days

LANL ATS Saga (Notional)



ATS1-Trinity

- Can we run complex 3D at all?
- Irregular/branching favors CPU
- Huge memory for the time
- Required Flash/Burst Buffer
- Answer yes but slowly (6mos)

ATS3-Crossroads

- Can we speed it up significantly
- Irregular/branching favors CPU
- Massive Mem BW jump
- Answer yes 4-6X faster

ATS5

- Can we speed it up significantly
- Demo first gen mem access accel
- Provide AI/ML assist (follow trend)
- Provide compute in net/stg
- Answer yes NX faster (opportunity is large due to irregular mem access)

ATS7

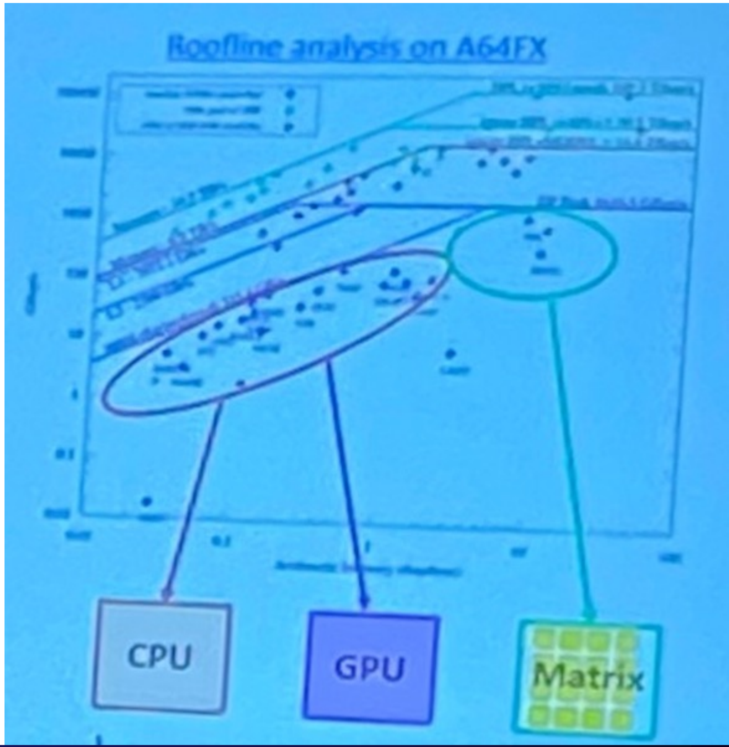
- Leverage course reconfig for branching
- Advance compute in net/stg
- Master AI/ML assist (follow trend)
- Master Irregular access
- Potential Quantum assist

There is a reason LANL's ATS systems aren't all flops, something we knew more than a decade ago!

- **To quote those who quote Jack https://www.nextplatform.com/2022/12/13/compute-is-easy-memory-is-harder-and-harder/**
  - If an exascale machine costs $500 million, but you can use 5 percent of the flops to do real work, it's like paying $10 billion for what is effectively a 100 petaflops machine running at 100 percent utilization…We have to get these HPC and AI architectures back in whack.
  - A BW divergence of 100X or 200X is a performance and economic crime.
- **Recent RIKEN Talk**
  - Transformer based training is matrix bound with small floats
  - Inference is memory bound (GEMV)
  - Almost all Science apps are memory bound
  - FugakuNEXT Plans – Breakthrough Bandwidth Monster (need >> 10 TB/s connected to CPU/GPU/Matrix in proper proportion

RIKEN Study: If matrix was free in workloads across ALCF, K, and Fugaku – it would gain 7-33% usable capability

To Zeta or not to Zeta



Thanks for your time!

Ultra-Scale Systems Research Center

The Efficient Mission Centric Computing Consortium