

# Energy-Efficient Computing: Do We Have A Roadmap?

Peter Kogge

McCourtney Prof. of CS, Univ. of Notre Dame

IBM Fellow (retired)

Cray, IEEE Computer Pioneer Awardee

# Snapshots From My Undergrad EE Days





# Computing Has Reached a “Cambrian Explosion”

turing lecture

DOI:10.1145/3282307

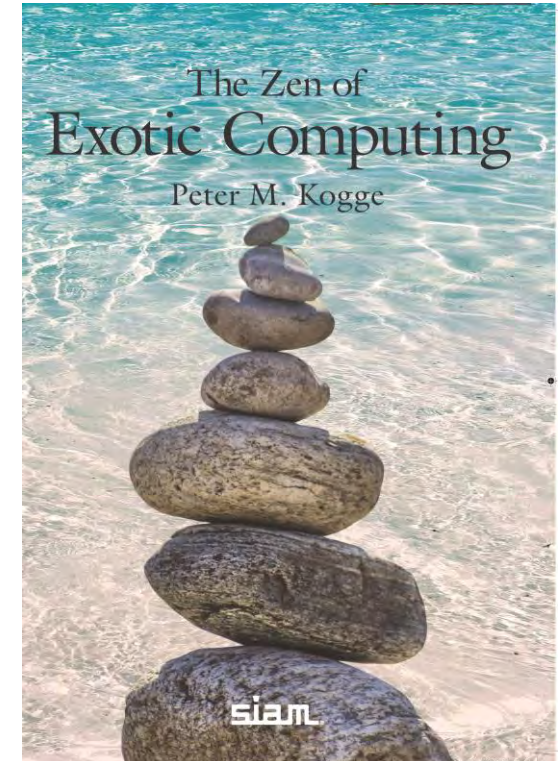
Innovations like domain-specific hardware, enhanced security, open instruction sets, and agile chip development will lead the way.

BY JOHN L. HENNESSY AND DAVID A. PATTERSON

## A New Golden Age for Computer Architecture



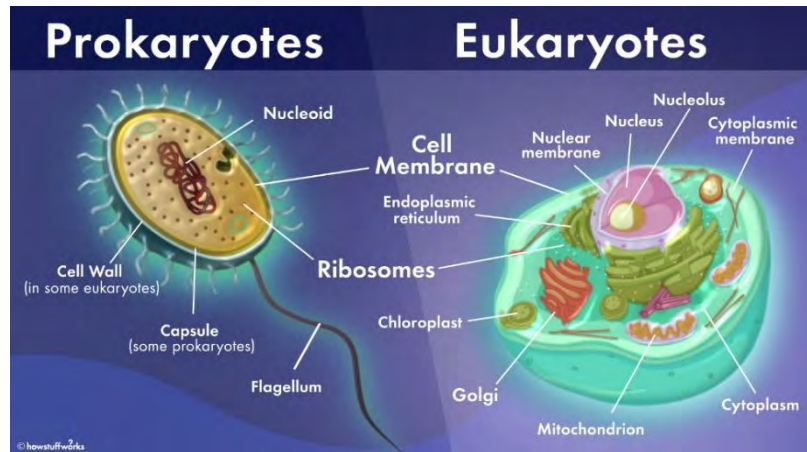
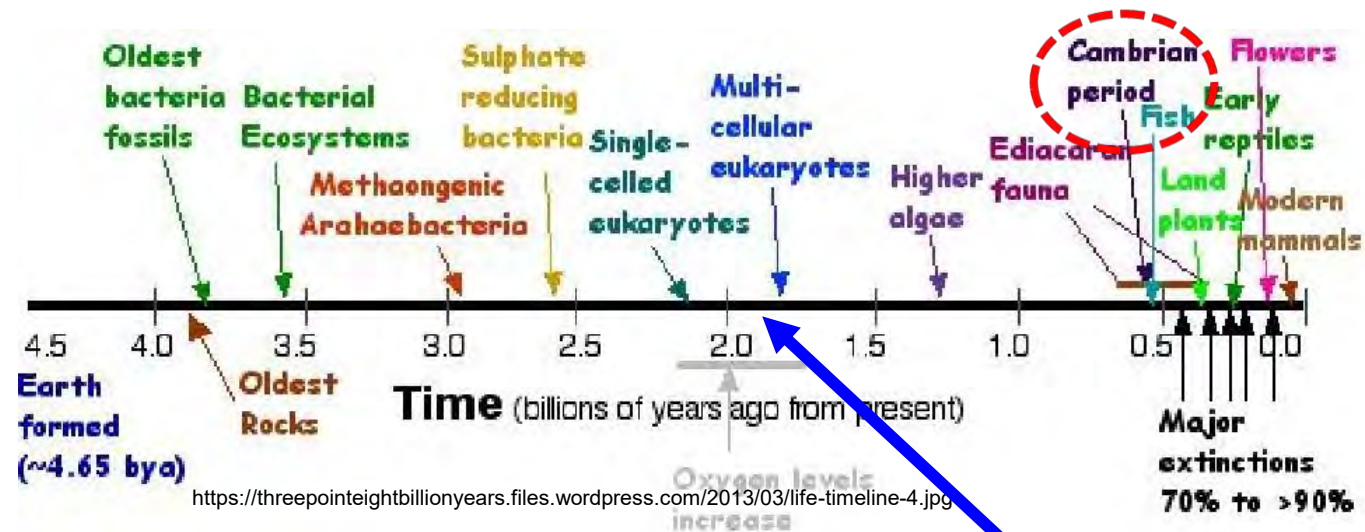
- Cellular Automata
- Map Reduce
- Systolic Arrays
- Petri Nets
- Actor model
- Lambda Calculus
- Combinators
- Logic Computing
- Reversible
- Probabilistic
- Functional
- Data Flow
- Neural Morphic
- Adiabatic
- Analog-Electrical
- Analog-Mechanical
- Genetic
- Ising
- DNA
- Fluidics
- Optical
- Quantum



Coupled with growing irregularity, sparsity, transitory nature of computation, what “consumes” power is changing



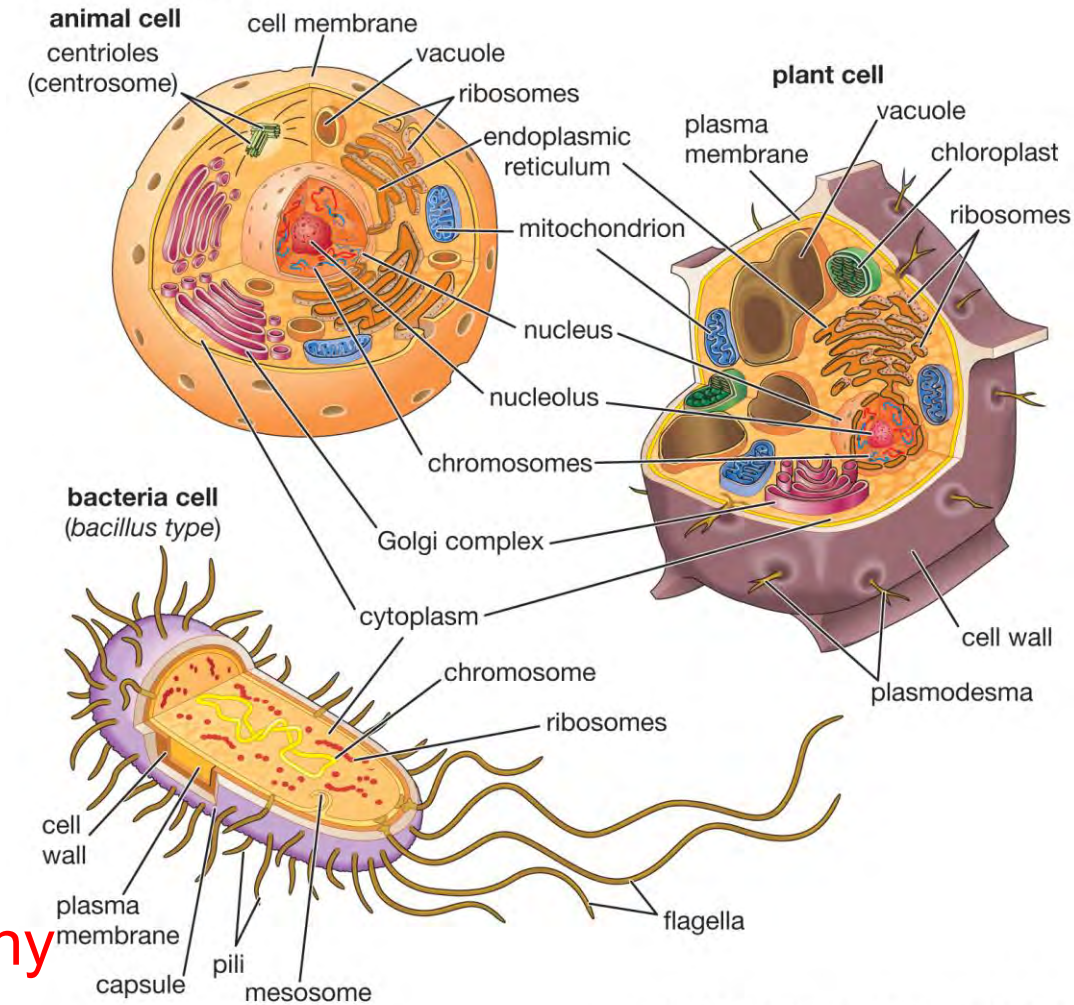
# Evolution of Cells: (A Core as a “Cell”)



## Eukaryotes/Core:

- Nucleus
  - Program store
- Organelles
  - Function units
  - Memory hierarchy
- Plasmodesma
  - Basic I/O

## Some typical cells



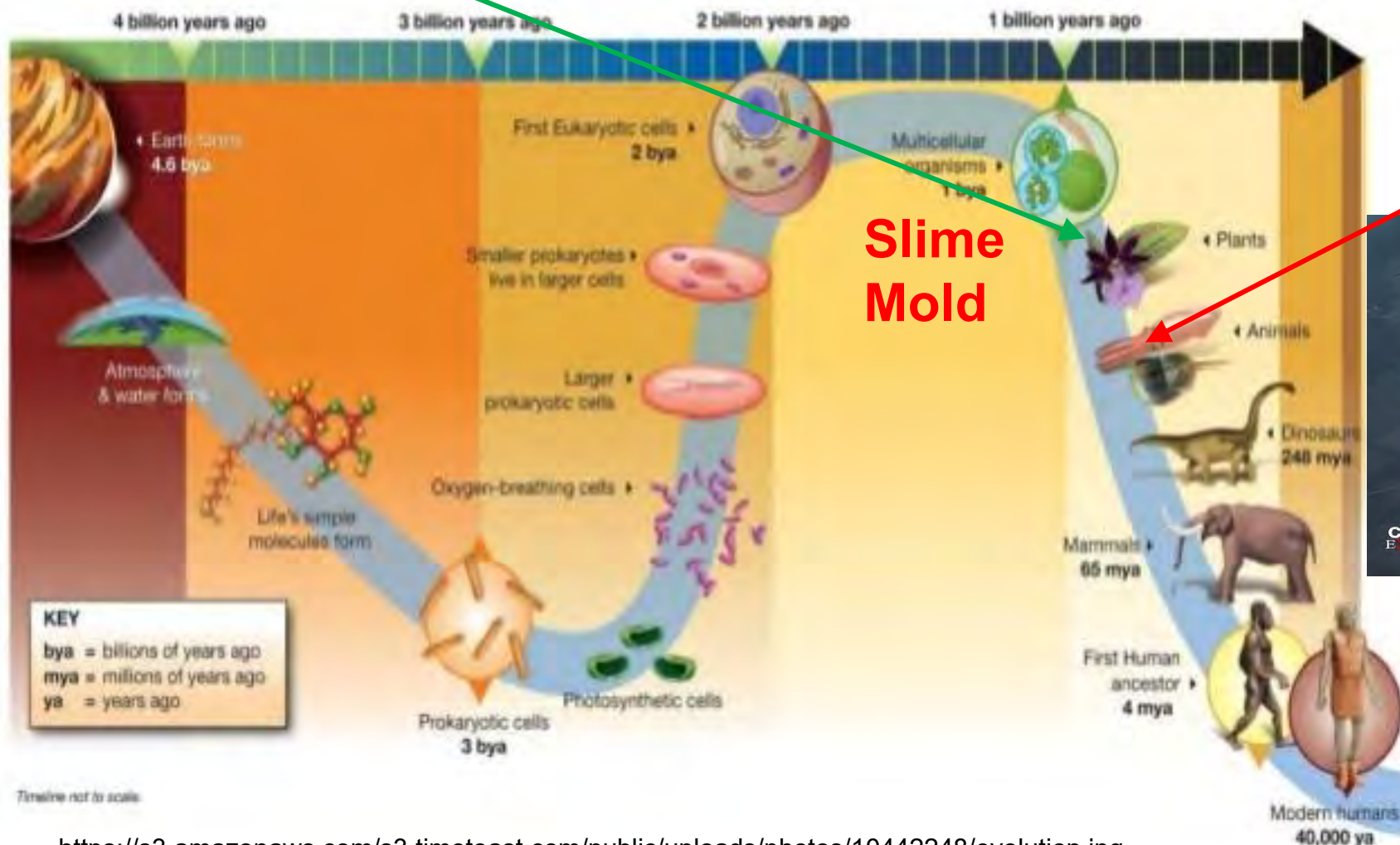
© 2012 Encyclopædia Britannica, Inc.

<https://cdn.britannica.com/85/78585-050-3B7B6E8E/cells-animal-plant-ways-nucleus-difference-organelles.jpg>

From ARCS 2021 Keynote



# Thesis: Our parallel systems have only evolved to **simple plants**, with ***no animals yet***.



But we may be close to a ***Cambrian Explosion*** Of Architectural Diversity



<https://crossexamined.org/wp-content/uploads/2020/05/Blog-2-cover-1.jpg>

<https://s3.amazonaws.com/s3.timetoast.com/public/uploads/photos/10442248/evolution.jpg>

From ARCS 2021 Keynote

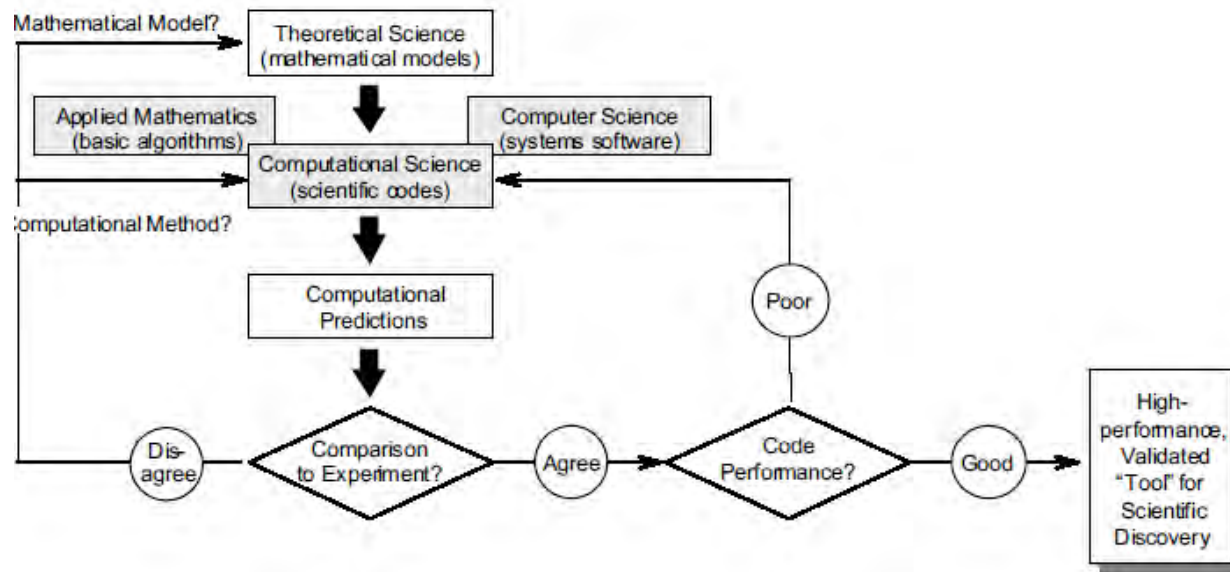
# What is Energy-Efficient Computing?

- We Can't Mean Just:
  - Lower energy/flop
  - Lower node power
  - Lower energy per computation
- We ***Must*** Mean:
  - Lower Overall System Energy *At Performance* Comparable or Better Than What Is Achievable Today
  - At the Same Size Problems, Or Bigger
- Will Require **ATSAA**:
  - Architecture/Technology/Software/Algorithm/Application Codesign

Particularly Science specific

# First Observation: Evolution of Challenges

- 2000: DOE Report “Scientific Discovery through Adv. Computing”
  - Beginning of ubiquitous parallelism
  - Challenge: Make full use of terascale
  - **Critical issues:** performance, portability, adaptability
    - Sim codes 5%-10% of peak – and decreases with parallelism



E,g. Computing energy of Iso-octane

- 275M nonlinear equations
- Iterative solution
- 2.5PB between processors
- 15 TB to disk
- 30 PFlops

- 2008 Exascale Study: Eye-opener on energy/power

# Topics

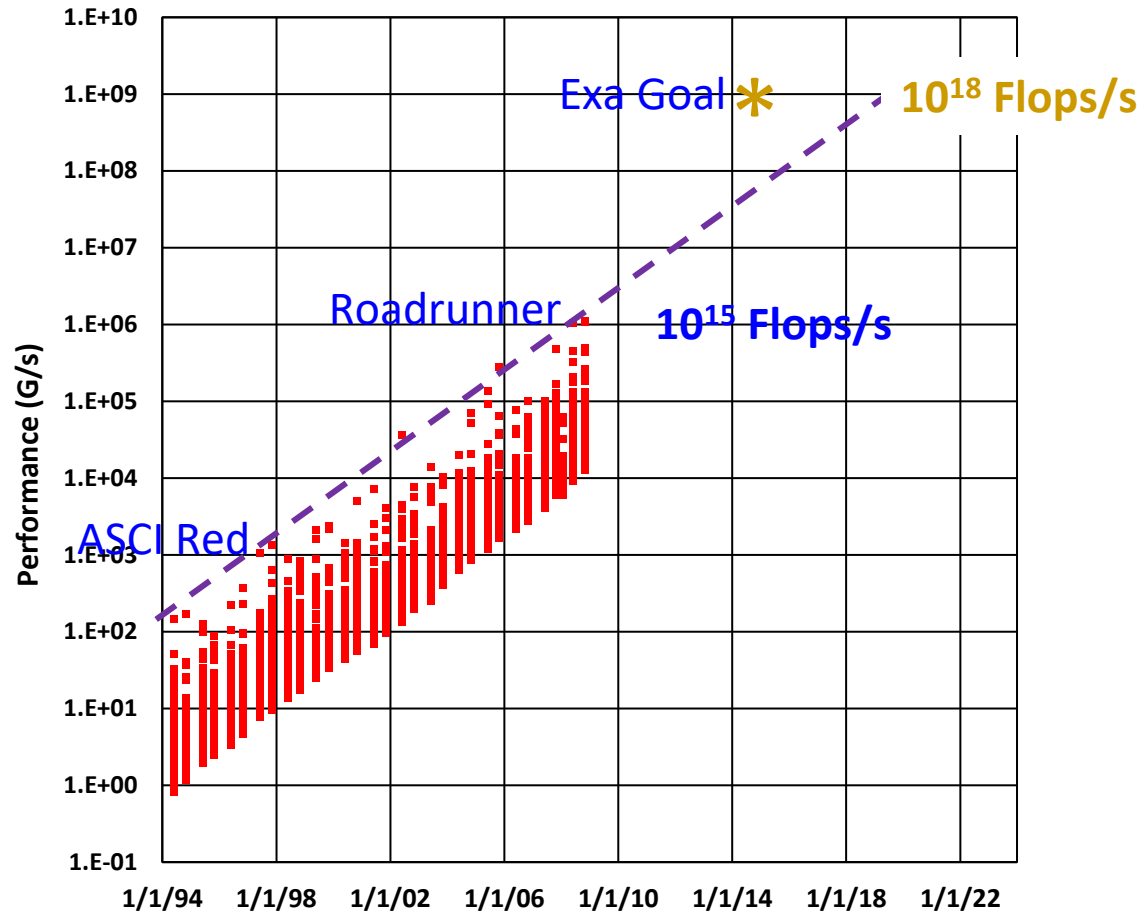
- The Road to Exascale: Lessons Learned
- Current Projections to Zettascale
- Deep Dive on The Past
- Technology Advances
- Non-Obvious Research Directions



# **Review: The Road to Exascale**

Includes content from “Frontier vs the Exascale Report: Why so long?  
and Are We Really There Yet?” by Kogge & Dally, SC 2022

# The HPL World in 2008

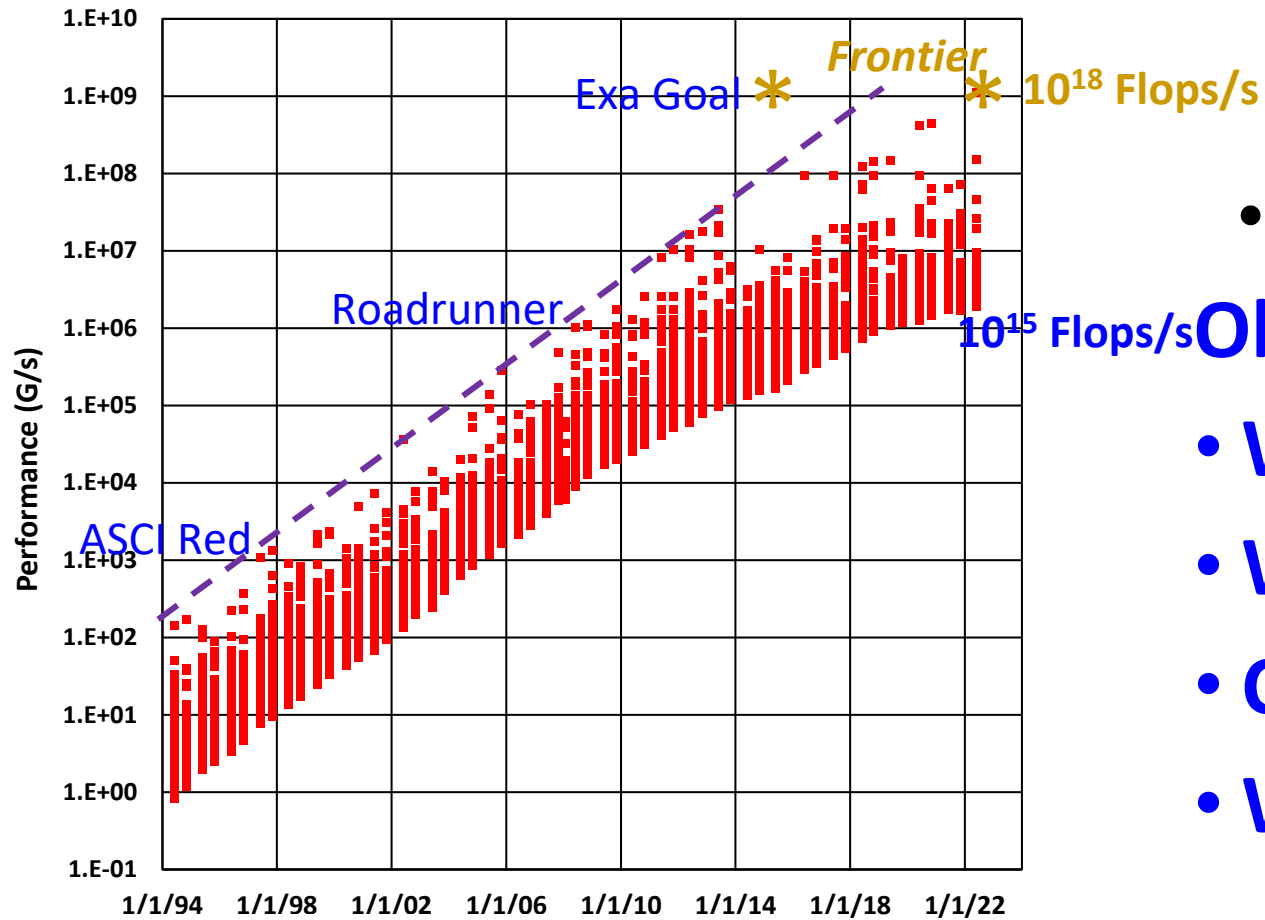


“First Light” for new TOP500 entries

- Roadrunner: 1+ PF/s
- DARPA (Bill Harrod): Exa by 2015?
- 2008 Exascale Report: Yes, but...



# The HPL World in 2022



“First Light” for new TOP500 entries

- 2022: Frontier Cracks 1EF/s
  - 7 years after Report Goal
  - 4 years after extrapolating curve
- Bounding Curve Changed in 2013

## Obvious Questions

- What Is/Was Exascale?
- What Did 2008 Report Predict?
- Comparison to Frontier
- What did Report get Right/Wrong?



# The Exascale Study Report Outline

- **What *should* “Exascale” Mean?**

- The 2008 state of the art
  - Architectures, Runtimes, Programming, Metrics
- 2008 Application Characteristics
  - Computation vs Memory intensive Apps, Scaling, Concurrency
- Technology Roadmaps
  - Logic: Silicon and Non, Memory, Storage, Interconnect, Packaging, Resiliency, Programming Models

- **Strawman Designs**

- Subsystem projections, Evolutionary designs (Heavy and lightweight), Aggressive design

- **Challenges & Research Areas**

- Power, power, power, & power
  - Memory capacity & bandwidth
  - Programmability
  - Reliability
- Remain**
- Practically Solved**

**Blue:** Addressed here

# What Was Exascale?

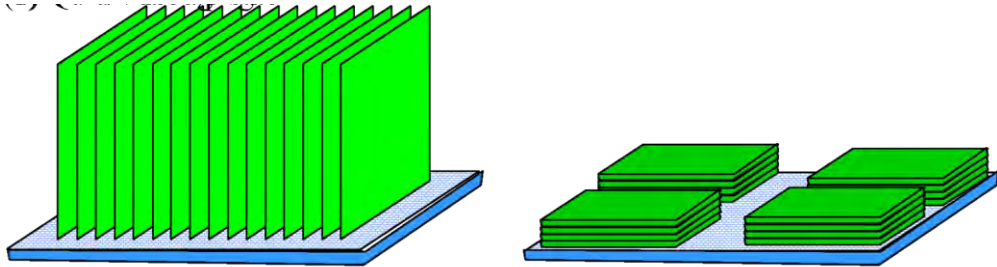
- Report Emphasis: *Try* to change focus from flops
- Goal: **overall** 1000x capability over “Petascale” by 2015
  - In Same Footprint for Supercomputer at max 20MW
  - 1000X in a rack (peta scale)
  - 1000X in a module (tera scale)
- Goal: not just flops but 1000x in:
  - Memory
  - Memory Bandwidth
  - Network Bandwidth
  - ...
- Plus ability to efficiently use massive concurrency

# Technologies Investigated

- Logic: power, area, energy, clock
  - CMOS: hi perf/low voltage
  - Options: hybrid, superconducting
  - Voltage scaling
- Main Memory
  - SRAM, DRAM, NAND, Alternatives
  - Reliability, packaging, power
- Storage Memory
  - Disk, Holographical, Archival
- Interconnect: esp. energy
  - On chip
  - DRAM to Processor (Stacking)
  - Intra/inter module
  - Rack to rack
  - Electrical vs optical
- Packaging and Cooling
- Resiliency & Checkpointing
- Programming Models



# 2015 Aggressive Strawman Design (2013 Tech)



**Node:** 742 simple cores/chip with 4 FPUs @ 1.5GHz

- 32nm CMOS with 30Gb/s SERDES
- 16 Memory channels: each 1 GB *Stacked* DRAM
- 150 Watts w/o routing chip

**Group:** 12 nodes with 12 64-radix router chips

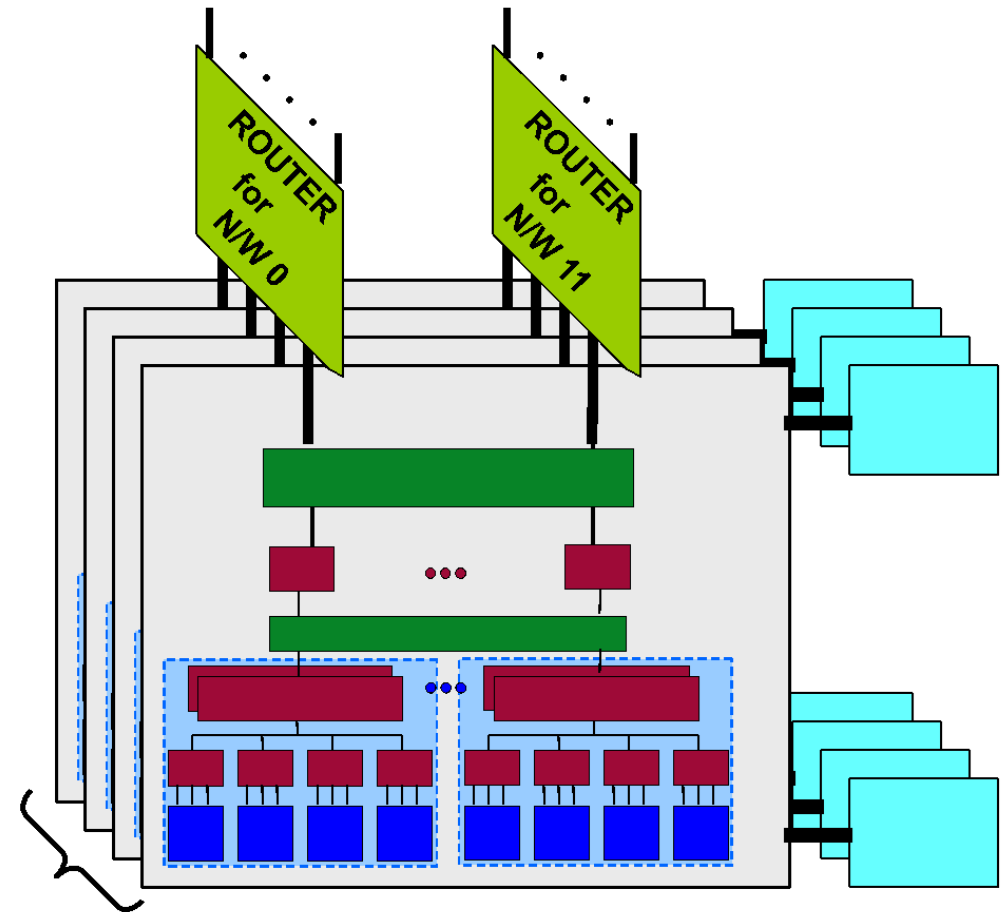
- Includes 16 12GB SATA drives for checkpointing

**Cabinet:** 32 Groups = 384 nodes

- Assumed max power of 120KW

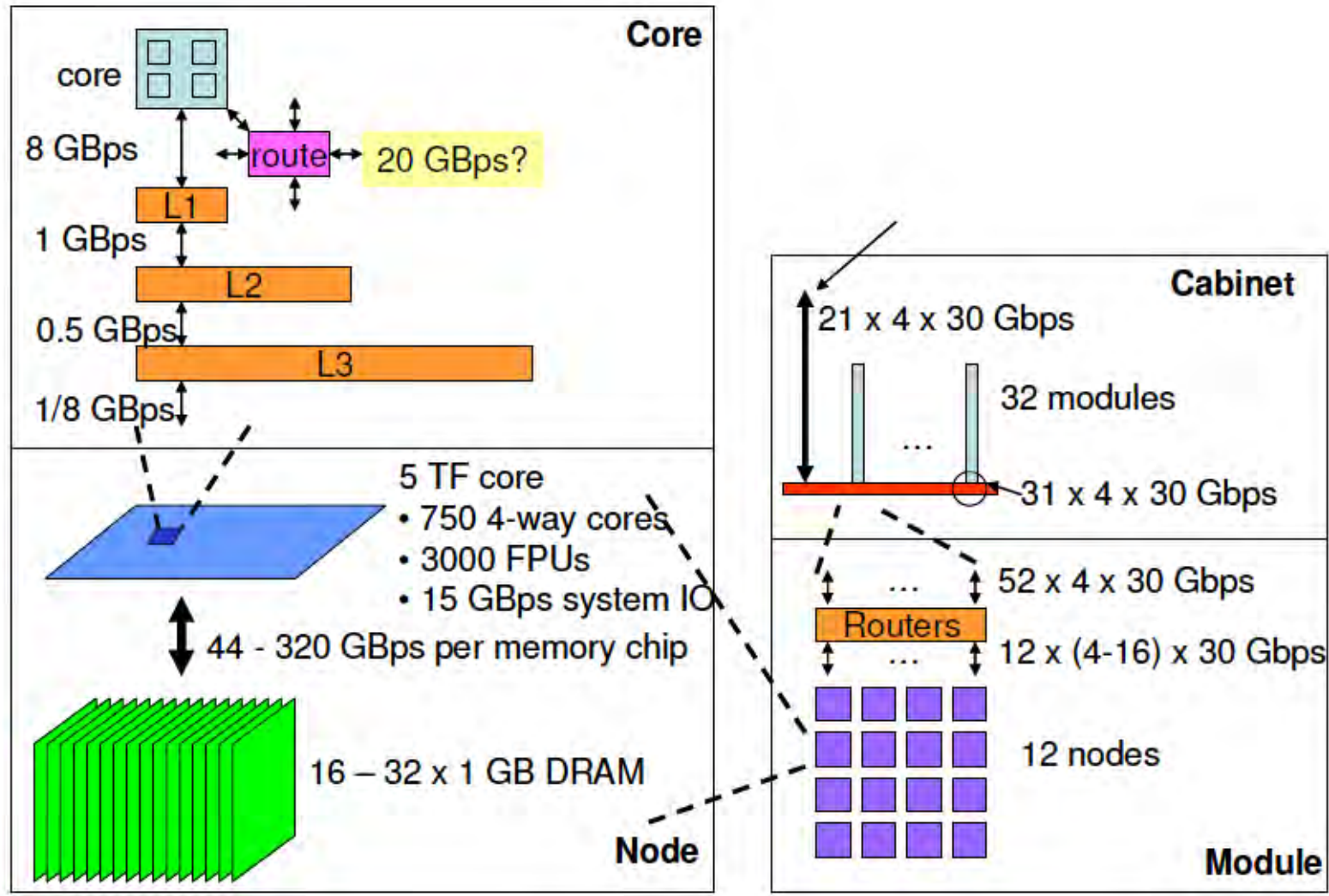
**System:** 583 Cabinets, 67MW

- 3-hop Dragonfly interconnect (optical)
- 166 million cores with 664 million FPUs

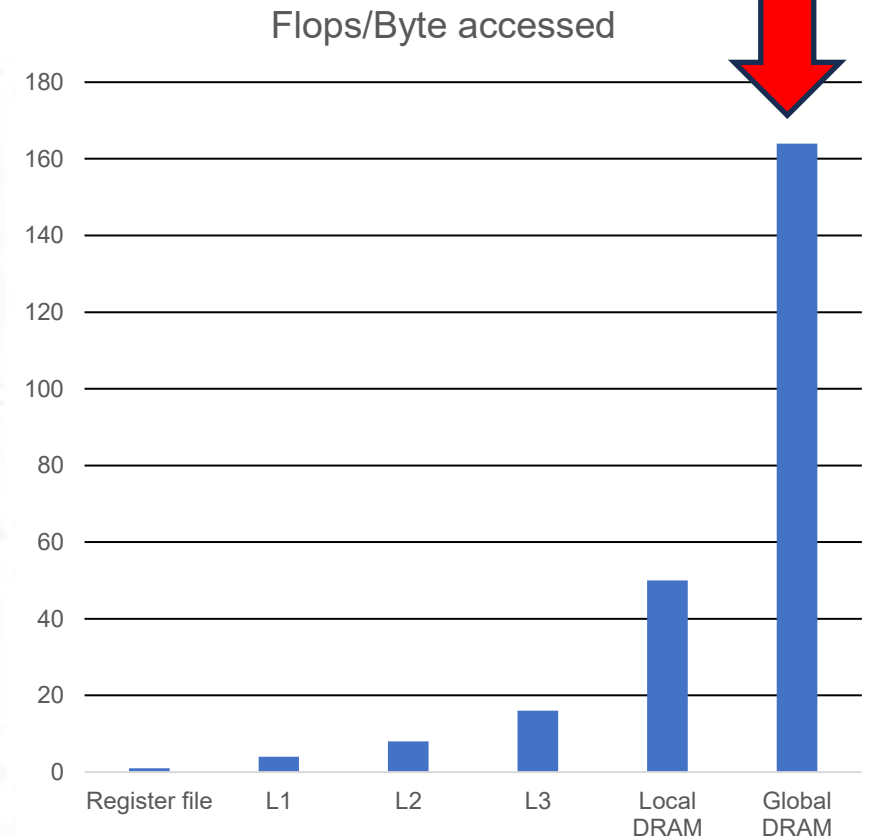


**Est. 14.9 GF/W**  
**Or 67 pJ/flop**

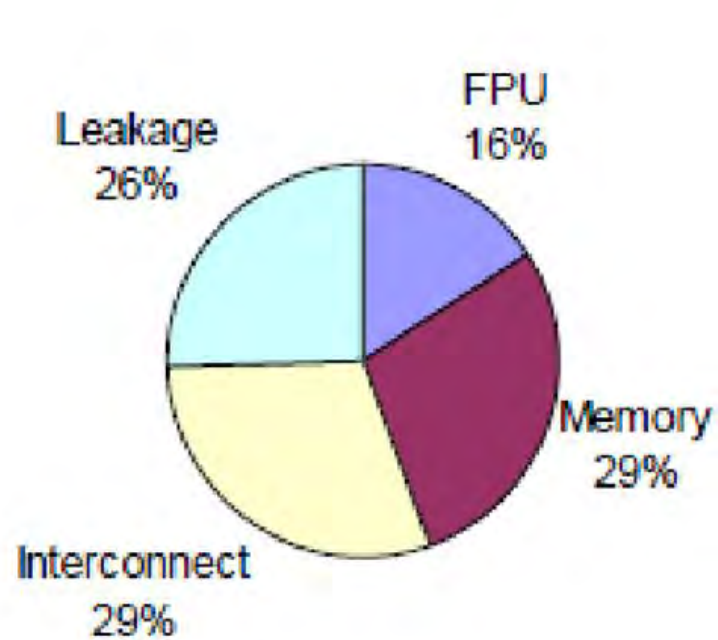
# Power/Energy Constrained Memory Bandwidth



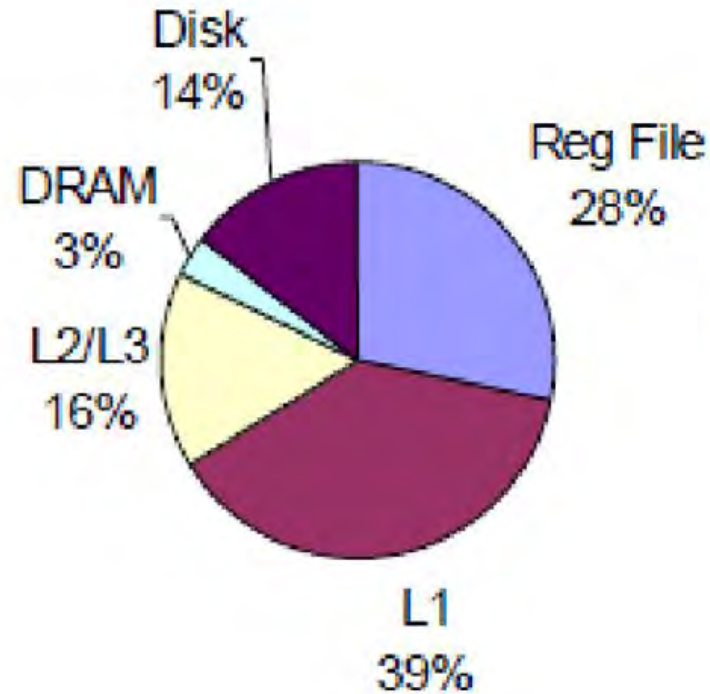
Need to do 164 flops  
For each byte of remote  
Memory accessed



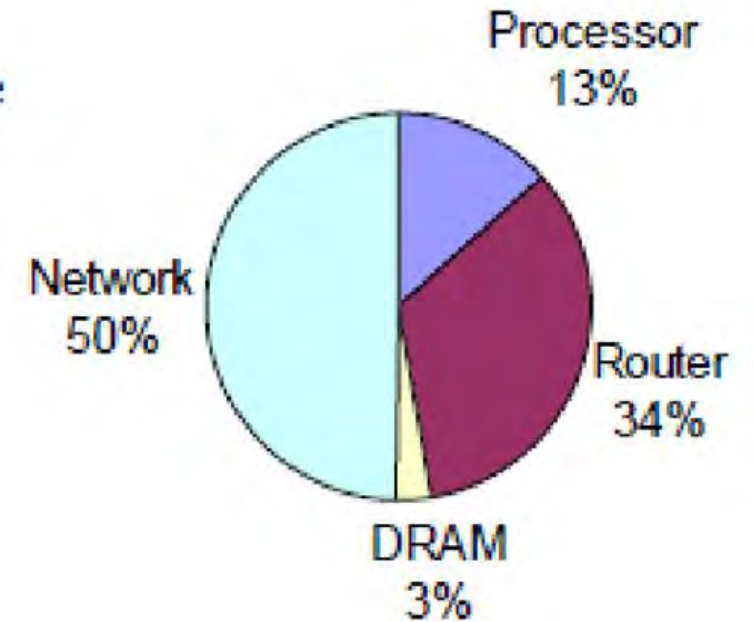
# Where Did the Energy Go?



(a) Overall System Power



(b) Memory Power

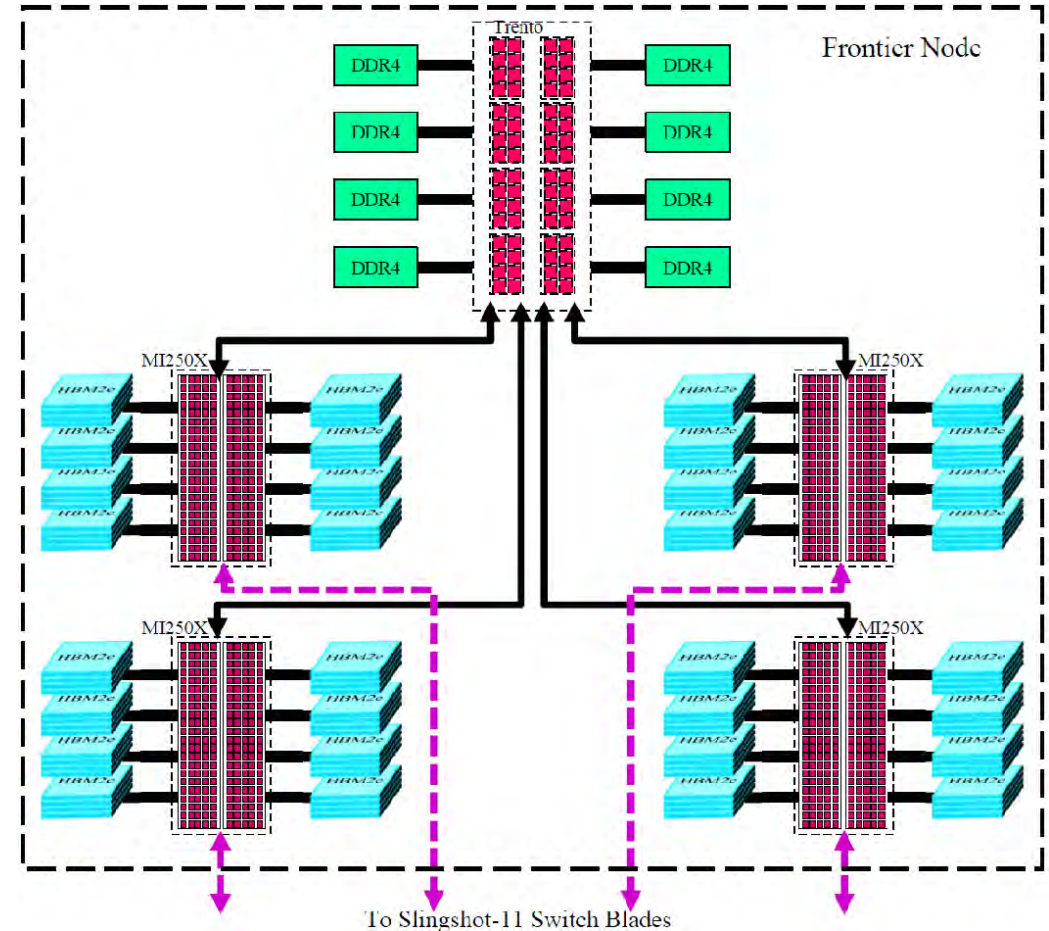


(c) Interconnect Power



# 2022 Frontier Node

- Heterogeneous Processors
  - 64-core 2GHz CPUs
  - Quad GPUs: closer to Strawman
    - But more FPU/core
    - And slightly faster
- Chiplet design
- Mixed memory hierarchy
  - 8 DDR4 DRAM Channels
  - 8 HBM2e stacks/GPU
- Quad network ports



**Measured 52.2 GF/W**  
**~3.3X Strawman**

# Frontier vs Strawman

	Road-Runner	2008 Strawman	Frontier
System Counts			
Nodes/Blade	1	12	2
Blades/Chassis	4	1	8
Chassis/Cabinet	3	32	8
Nodes/Cabinet	12	384	128
Total Nodes	3060	223,872	9,408
Cores/Node	40	742	944
MACs/Node	76	2,968	56,832
Total MACs	232K	665M	535M
Memory Metrics			
Total Memory (TB)	36	3,498	9,408
Total Memory BW (TB/s)	378	157,605	125,239
Network Bandwidth Metrics			
Network ports/node	1	12	4
Total Network ports	3,060	2.7M	37,632
Switch Chips/Cabinet		384	64*
Switch Radix	24	64	64
Total Switch Chips	900	223,872	4,736*
Signal Rates (Gb/s)	4	30	56
Inj. B/W/Node (GB/s)	2	180	100
Bisection B/W (TB/s)	0.192	210	540
* Assuming 8 switch cards/chassis			

- Strawman's **huge #s of nodes**
  - Exploded # of Network ports
  - And thus huge switching costs
- Frontier had fewer, **bigger nodes**
  - Reduced network ports
- **Comparable** Memory Bandwidth
  - Use of wide stacked memory
  - But only 3X capacity
- Essentially **same N/W topology**
  - But 2X better SERDES
  - And 2+X better bisection B/W

# Frontier vs Roadrunner: Did We Get 1000X?

	Road-Runner	Frontier	Growth Ratio
GFlops/s/core	8.4	126	15
GFlops/s/chip	56	23,426	419
TFlops/s/node	0.34	117	349
TFlops/s/cabinet	4	14,993	3,726
TFlops/s/sq. ft.	0.17	151	882
Flops/core/cycle	2.74	208	75
Flops/cycle <sup>1</sup>	3.2E5	6.7E8	2,022
Flops/Mem byte	9.9	119	12.1
Flops/Mem BW byte	2.7	8.8	3.25
Flops/Inj. byte	168	1,171	7
GFlops/watt	0.44	52.2	119
Watts/core	19.24	2.4	1/8
Watts/chip	128	449	3.5
Watts/node	766	2,243	2.9
All cores and all chips included			
<sup>1</sup> Using clock for major compute core.			

- Flops/s **exceeded** 1000X / cabinet
  - But huge cabinets
  - Within 3X for chip & node
- **>100X** in flops/s per watt
  - And flops/cycle
- **Miserable increase** in Memory, Memory Bandwidth, N/W Injection Bandwidth

# Report Card

## What We Got Right

- CMOS, flat clocks
- Large # of wide simple cores
- Aggressive memory hierarchy
- Stacked memory
- Near reticle-limited dies
- Energy of movement predominates
- Near billion-way concurrency
- Memory concerns were valid
- Dragonfly with hi radix switches
- N/W signaling rate would improve

## What We Missed

- Exploding Heterogeneous designs
- SIMD width much larger
- Stacked memory: more ports/lower transfer rate
- Machine Learning & short FP
- Massive chips (area and power)
- Reliability not a show-stopper
- New programming models



# What about “Zettascale”

# Industry Projections: Road to 1 Zettaflops/s

- 2018: China NUDT: 2035 is reasonable
  - Moving from exascale to zettascale computing: challenges and techniques. *Frontiers Inf Technol Electronic Eng* **19**, 1236–1244 (2018). <https://doi.org/10.1631/FITEE.1800494>
- 2021: Intel: 2027 may be possible
  - Raja's chip notes lay out intel's path to zettascale. <https://www.servethehome.com/rajas-chip-notes-lay-out-intels-path-to-zettascale/>, 2021
- 2023: AMD: 2035 is reasonable
  - <https://www.hpcwire.com/2023/02/21/a-zettascale-computer-today-would-need-21-nuclear-power-plants/>
- All assume power budget 50-100MW

# NUDT's 2018 Projection for 2035

- Technical Challenges: Manufacturing, *Energy*, Interconnect, *Storage*, *Reliability*, *Programming*
- Expected changes
  - **Architecture:** heterogeneous, mixed precision, near-memory, non-von Neumann
  - **Package driven:** 3D integration, inter-chip interconnection, density per node to 0.5-0.8 Pflops/s @0.8 Tflops/W
  - **Network:** electrical to exceed 50Gb/s, optical CWDM to 112Gb/s per channel, on-die photonics
  - **Storage:** 3D stacks, hybrid multi-layer storage, NVM, Network attached memory
  - **New technologies:** memristor, quantum
  - **Programming:** MPI+X, with variety of intra-node threading models, DSLs

**Table 1 Zettascale metrics**

Metric	Value
Peak performance	1 Zflops
Power consumption	100 MW
Power efficiency	10 Tflops/W
Peak performance per node	10 Pflops/node
Bandwidth between nodes	1.6 Tb/s
I/O bandwidth	10–100 PB/s
Storage capacity	1 ZB
Floor space	1000 m <sup>2</sup>

- 100,000 Nodes
- **1000W per node**
- Only 1.5X more floor space implies 100KW per sq. m of floorspace, or 3.2X more power per rack

# Intel 2021: Zettaflops/s by 2027-28

- **Goal:** same power as ~2Eflops Aurora
- **Architecture:** 16X – “AI Math” + keeping execution units fed from memory
- **Power/Thermals:** 2X – Lower voltages and higher-end cooling (eg liquid cooling but more than rear door heat exchangers)
- **Data Movement:** 3X – higher integration, fewer better SERDES
- **Process:** 5X - multi-tile design with advanced packaging that allows technology mix
- Phase 1: 2022-23 Exascale+
  - “Granite Rapids”
- Phase 2: 2024-25 Pre-Zetta
  - “Falcon” = Xeon + Xe
  - “Lightbender” = Silicon photonics
- Phase 3: 2026-28: Zetta
  - Reduce 50MW Exaflop-class to 50KW
  - Requires all changes from left



# 1 Supercomputer Energy Use Trajectory

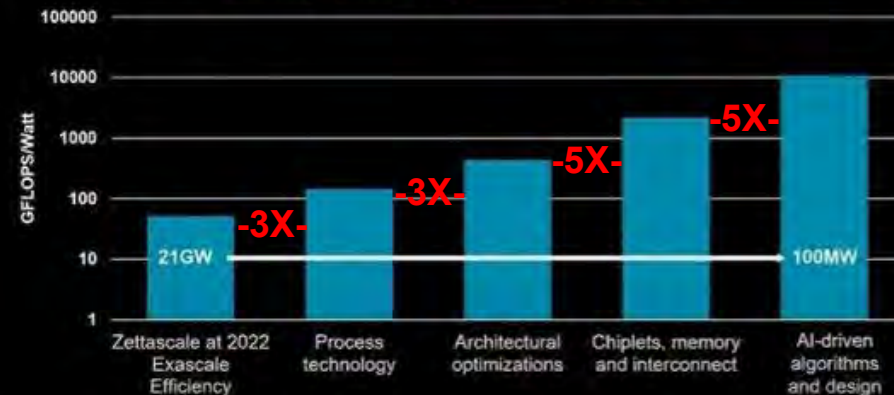
Green500 Supercomputer GFLOPs/Watt and Projection



AMD:  
Need  
to Use  
“AI”

# 3 Achieving Zettascale Computing

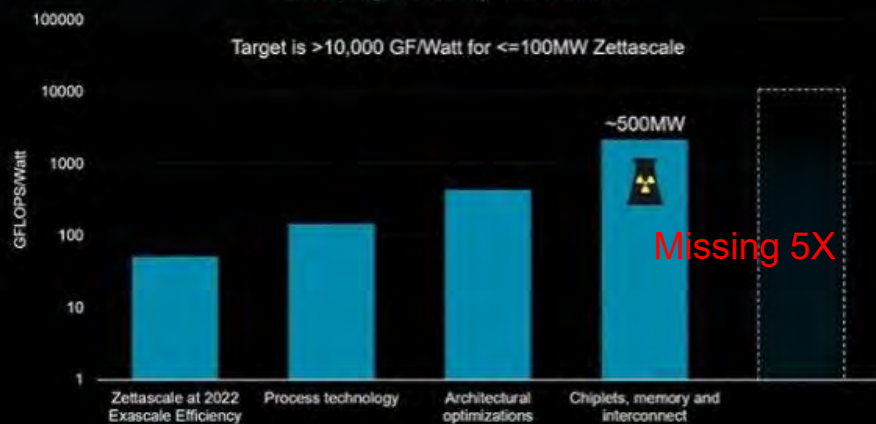
Efficiency Roadmap to Zettascale Leveraging AI



# 2 Achieving Zettascale Computing

Efficiency Roadmap to Zettascale

Target is >10,000 GF/Watt for <=100MW Zettascale



# 4 Domain-Specific Computation Enables Workload Optimization which Drives Performance and Efficiency

• Tailor architecture by application

• Adapt algorithms to use lower precision math formats for significant improvements in energy efficiency

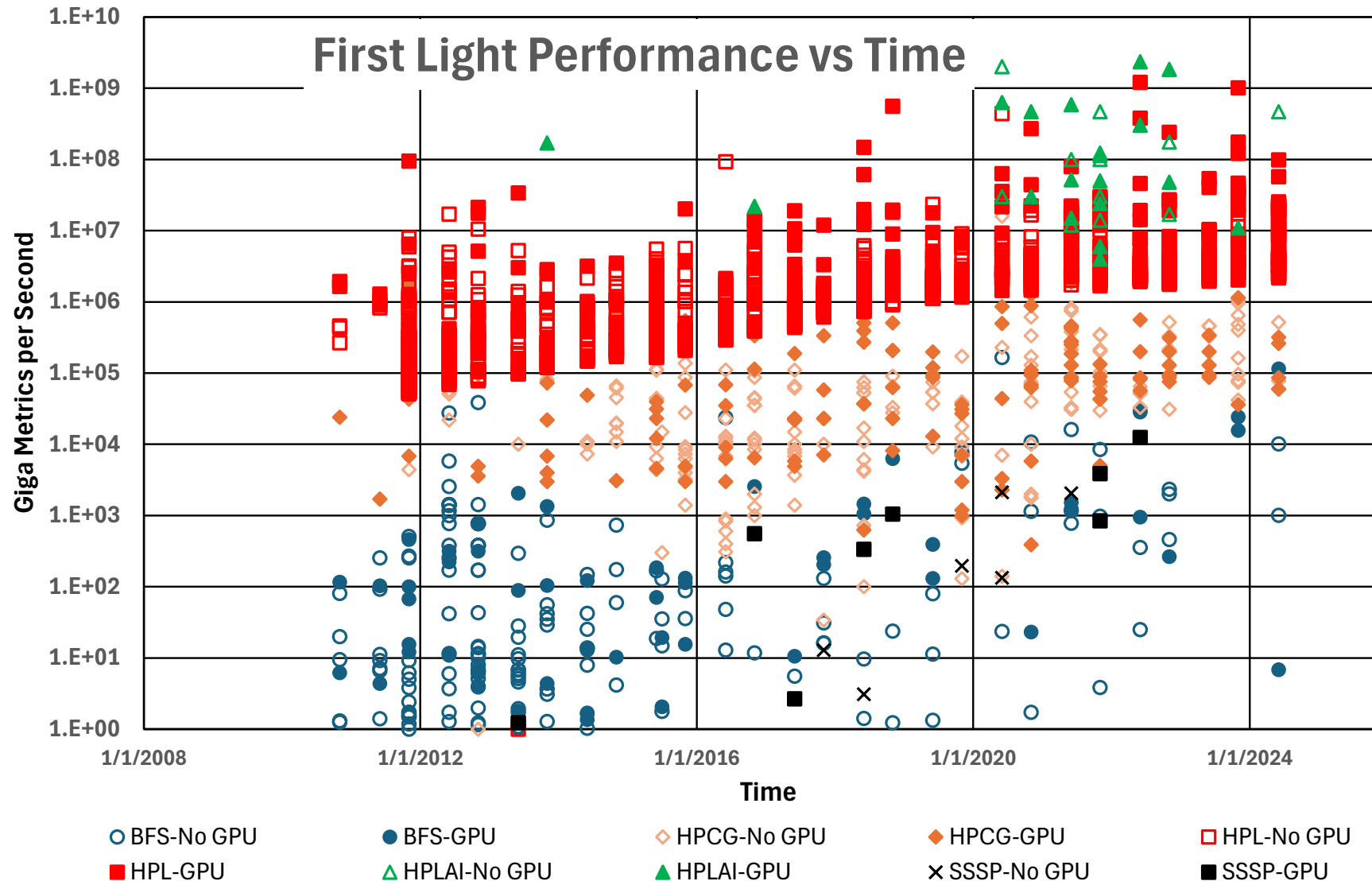


# Clear Takeaways

- Focus seems to solely be on flops/s
- Aggregate power the major challenge
  - Power budgets of 100MW imply need 200X in efficiency
- No silver bullet from “traditional” architecture
- Process improvements small (3X?)
- Bulk of improvement riding on packaging, esp. 3D
- Much hope on “AI” – i.e. mixed precision
- Memory/networks mentioned but not any real thought

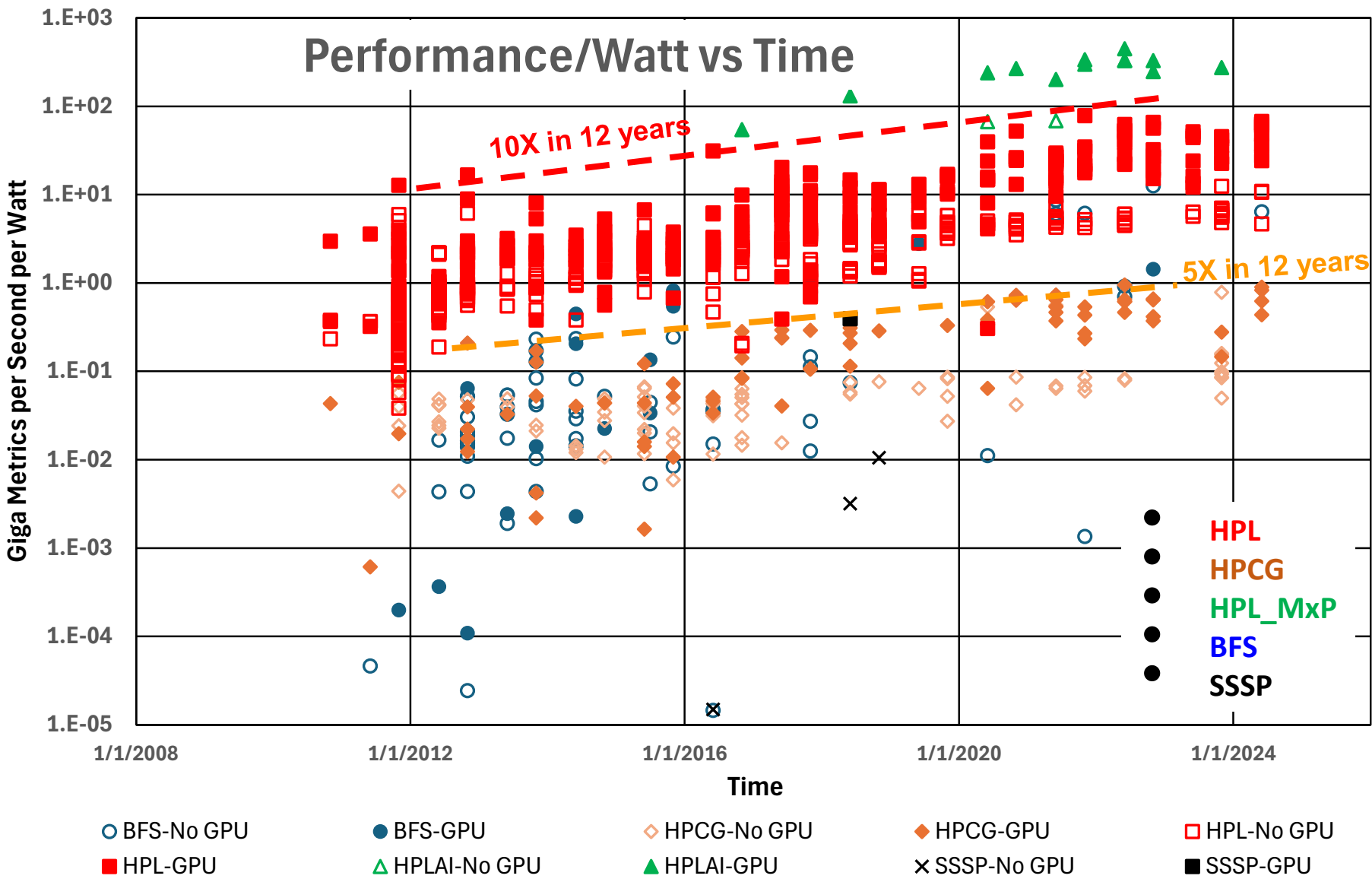
# **Deep Dive on The Past: An Energy Focus**

# The Multi-Faceted World of 2020s



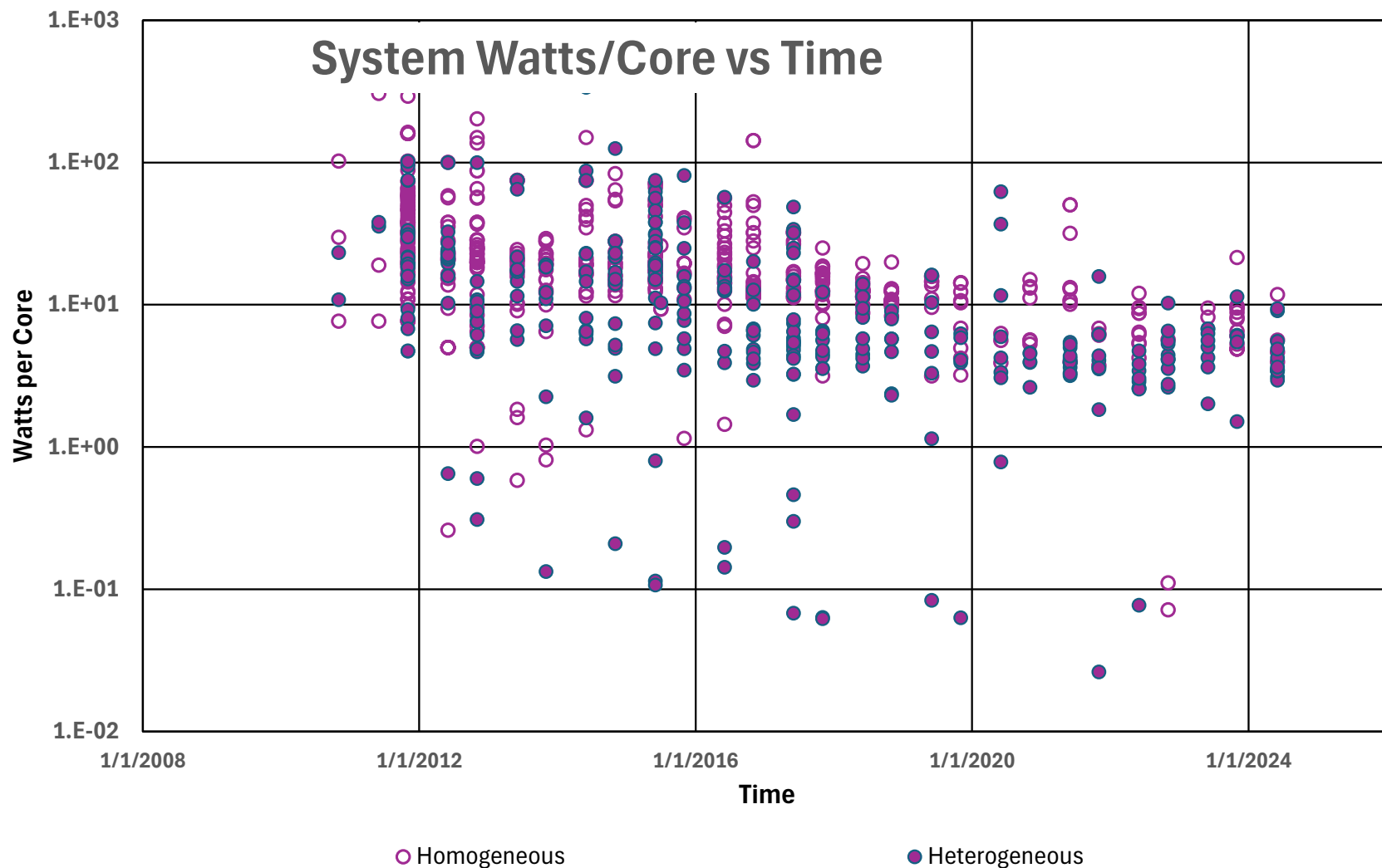
- “First Light”: 1<sup>st</sup> appearance
- Color=Benchmark
  - HPL
  - HPCG
  - HPL\_MxP
  - BFS
  - SSSP
- Fill
  - Solid: with GPUs
  - Hollow: without





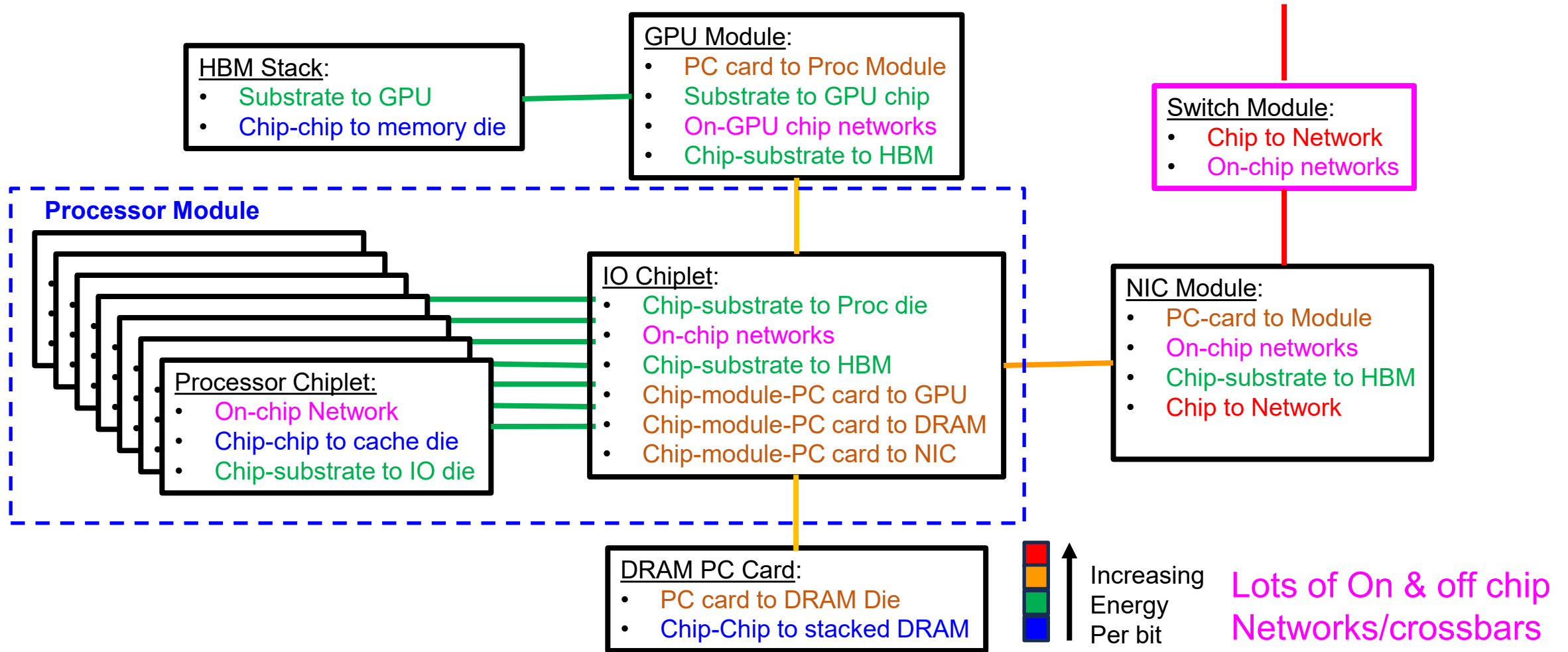
CAGR

- **HPL: 1.2**
- **HPCG: 1.13**
- **BFS: Flat**



- Seems to have been ~ constant since ~2018
- “System” power/core >> chip power per core
- We need to look at “system”

# Today's Node Design – Movement Energy



# Technology Advances

A Quick Subset



# Lower Operating Voltage – No Free Lunch

- A.k.a in prior decades as Dennard Scaling
- Lower voltage implies *lower energy per operation* (good)
- Implies lower clock rates
- Implies more parallelism needed
- Requires more paths for interconnect
- And more data movement
- And *more power spent in data movement* (bad)

# And What About Memory?

	Petaflops/s	Exaflops/s	Zflops/s
	Roadrunner	Frontier	At same ratio
Capacity (Bytes)	0.036PB	0.009EB	0.0025ZB
Bytes per flops/s	0.032	0.0075	0.0025
Bandwidth (Bytes/s)	0.378PB/s	0.125EB/s	0.04ZB/s
Bandwidth per flops/s	0.344	0.105	0.04
Flops per Byte	3	10	25

- Assume characteristics scale as they did from Peta to Exa
- Relative capacity dropping
  - Can problem size scale with flops?
- Is being able to access a word of memory every 400 flops acceptable?
- With all the deep cache hierarchies how much extra data transfers between memories are there?
  - All of which consume power

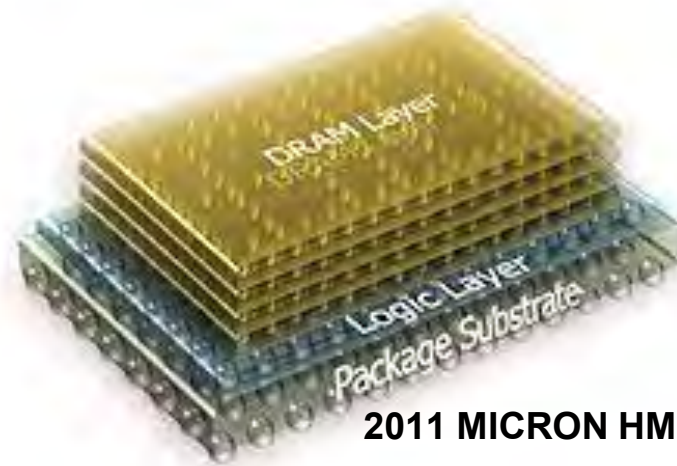
# At Best Modest Gains in PHY/SERDES Energy

	2008 Projection	Recent		
Data Movement	Energy	Bandwidth	Energy	Citation
Chip-chip vertically on Thru Silicon Via	0.01pJ/b			
Long range on chip	0.018pJ/b/mm			
Chip-chip copper	2pJ/b	224Gb/s	3pJ/b	ISSCC 2024 Paper 7.3
Routed Interconnect	2pJ.b + 1pJ/b per switch			
Optical	1.5pJ/b + 0.1pJ/bit for routing	224Gb/s	1.04pJ/b	ISSCC 2024 Paper 7.2

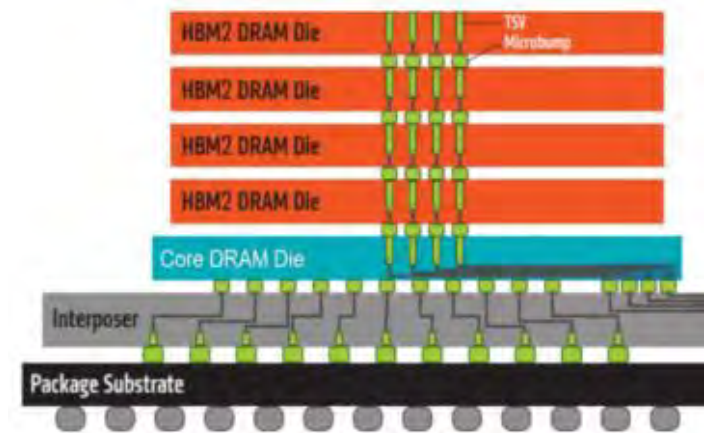
# Stacking: Reduce Die-Die Energy Cost



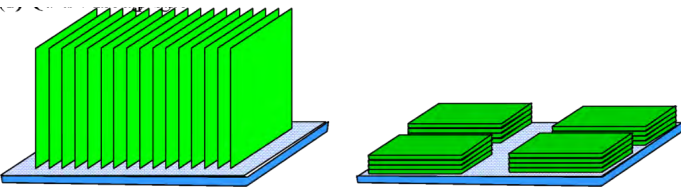
1990s Irving Sensors Die Stack



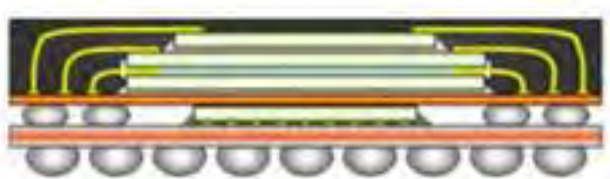
2011 MICRON HMC



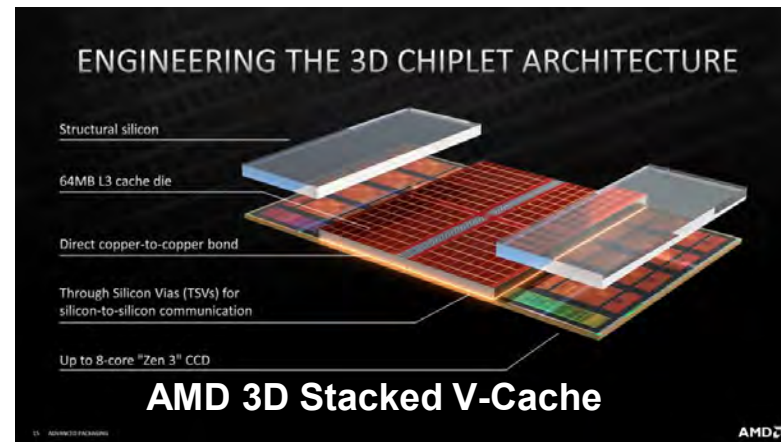
Modern HBM



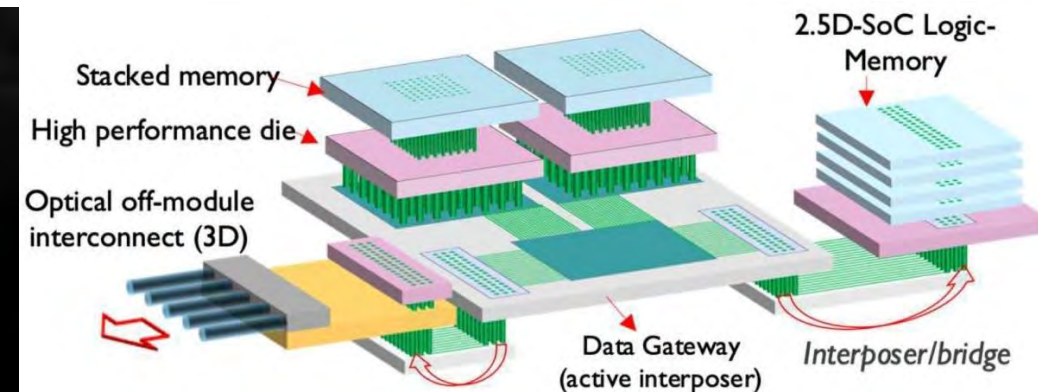
2008 Exascale Strawman



Apple 2012 iPhone chip stack



AMD 3D Stacked V-Cache

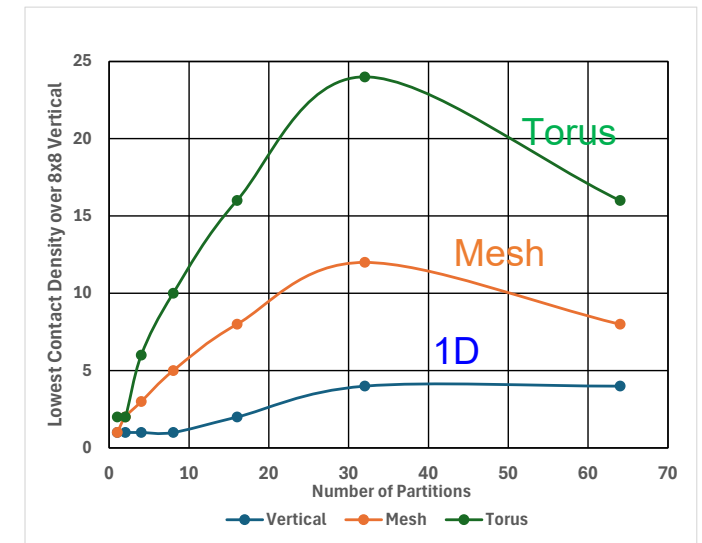
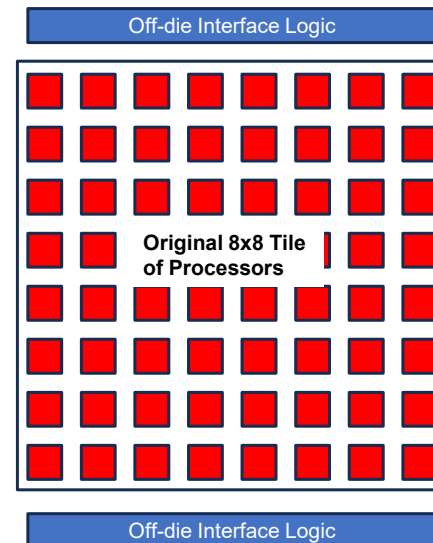
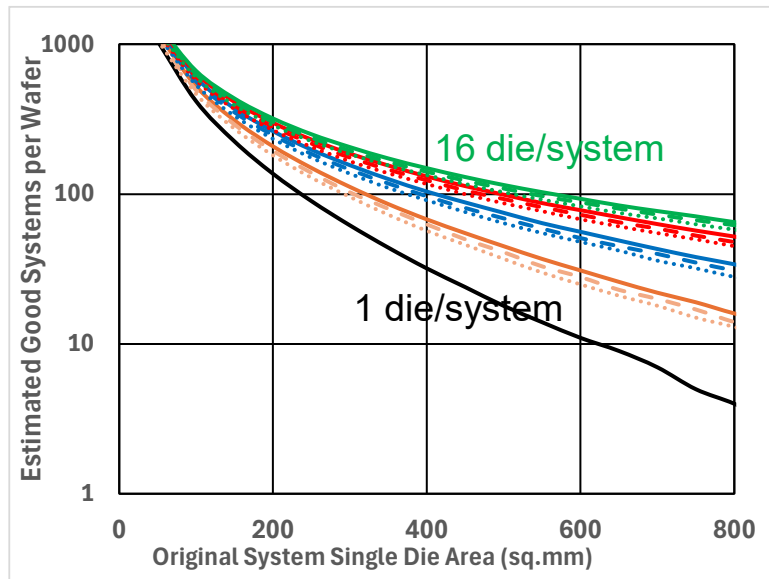


<https://scx2.b-cdn.net/gfx/news/2021/benefits-of-3d-soc-des.jpg>

# Chiplets



- **Chiplet**: Breaking large tiled die into multiple smaller die
  - And place on substrate not PC
  - With lower energy per bit transferred
- Reason: Increase yield => Lower cost
  - Also promote “mix & match”
- Side-effect: Increased off-die contacts
  - Each will consume energy

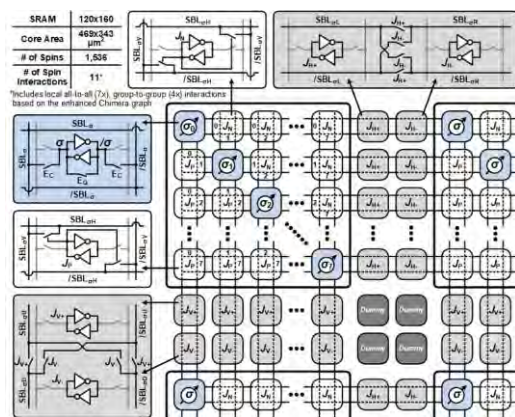




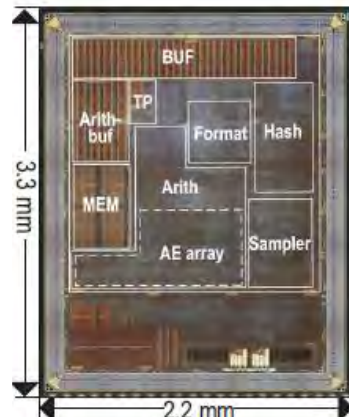
# ISSCC 2024 – New Architectures Sessions & Forums

- 15: Embedded Memories & Ising Computing
- 17: Emerging Sensing and Computing Technologies
- 20: Machine Learning Accelerators
- 29: ICs for Quantum Technologies
- 30: Domain-Specific Computing and Digital Accelerators
- 34: Compute-In-Memory
- F1: Efficient Chiplets and Die-to-Die Communications
- F2: Energy-Efficient AI-Computing Systems for Large-Language Models
- F3: Digitally Enhanced Analog Circuits

# Examples of Novel In-Memory Accelerators

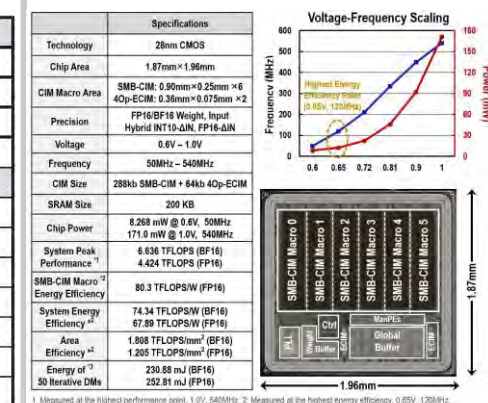


SRAM ISING In-memory Optimizer (ISSCC 2024 15.5)



Post Quantum Crypto-processor (ISSCC 2024 16.2)

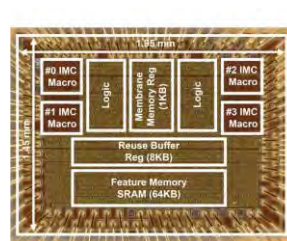
Chip Specifications	
Technology	TSMC 28nm HPC
Supply voltage	0.9V
Package	FCBGA
Die size	2.2 mm × 3.3 mm
Cryptography Core	
Core area	3.2 mm <sup>2</sup>
SRAM	228.5KB
Logic gates	2.1M (NAND2 equiv.)
Hash function	SHA3-256/384/512
PRNG	CHACHA20/AES/SHAKE
Crypto-fields	Zq/Binary/Complex
Power	91~420 mW@0.9V



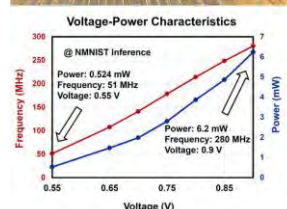
74TF/s BF16 Compute-In-Memory for Diffusion Models (ISSCC 2024 20.2)

Technology		28nm CMOS
Chip Area		3.80mm <sup>2</sup>
SRAM Size		513KB
# of PE		144
Bit Precision		INT8, 16, 32
Max Freq.		500MHz
Supply Voltage		0.55~0.9V
PINN Power		147mW@0.9V
FEM Power		98mW@0.9V
Peak Perform. (GOPS)		336 (0.9V, 16b)
PINN Efficiency (TOPS/W)		1.14 (0.9V, 16b)
FEM Efficiency (TOPS/W)		1.01 (0.9V, 16b)

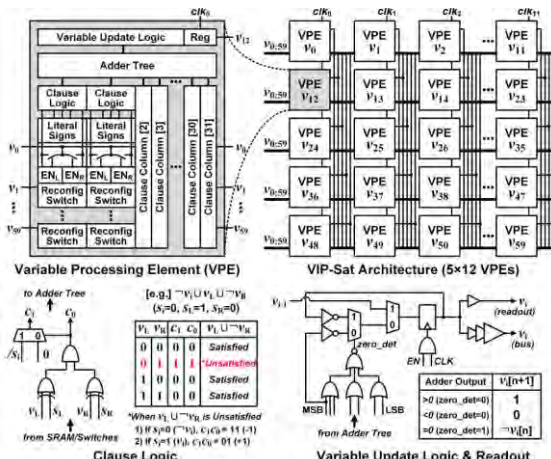
Compute-In-Memory PINN (ISSCC 2024 20.4)



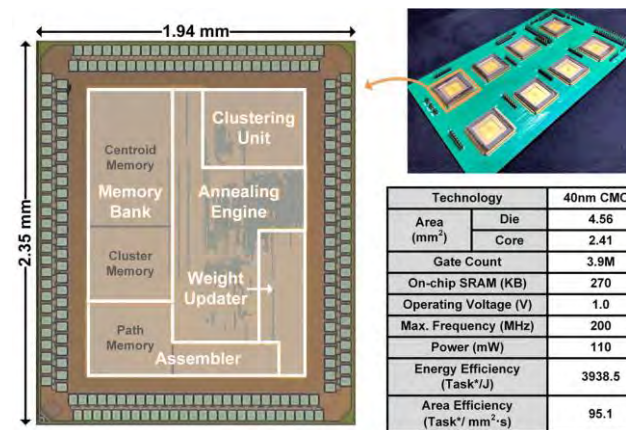
Technology		22 nm
Core Size		1.26 mm × 1.77 mm
Supply Voltage		0.55 V-0.9 V
Frequency		51-280 MHz
Static Power		174 μW @0.55V
Running Power		524 μW @0.55V
Supported Synapse Number		2 M
Power Density		0.26 mW/Synapse
Network Density		18.18 Synapses/Byte
Energy Efficiency		3.78 pJ/SOP @0.55V
Inference Energy		25.9 pJ @0.55V Gesture
		3.8 pJ @NNMIST



Compute-In-Memory Spiking Neural Net (ISSCC 2024 30.2)



Compute-In-Memory SAT Solver (ISSCC 2024 30.3)



In-Memory Annealing Processor (ISSCC 2024 30.4)

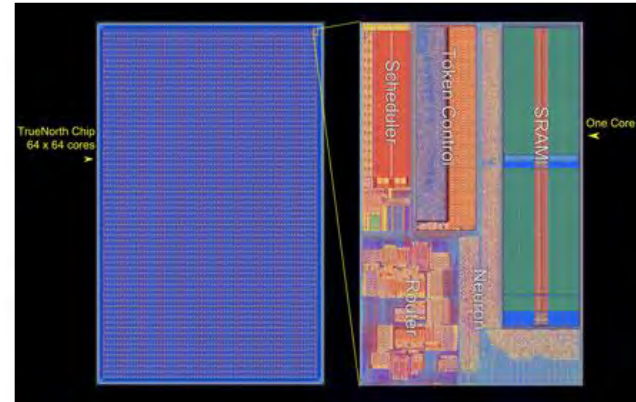
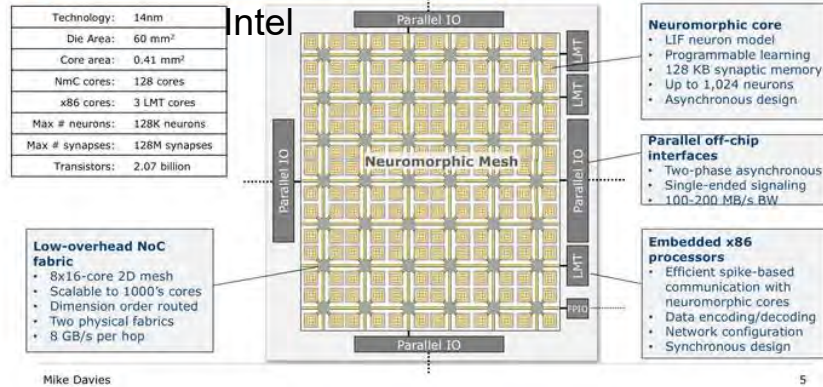
Chip Summary	
Technology	22nm CMOS logic process (Ultra low leakage)
Memory device	Foundry provided 1T1R ReRAM
ReRAM-CIM Capacity	16Mb (16 sub-banks)
Input precision	FP16 / BF16
Weight precision	FP16 / BF16
Output precision	FP32
Macro area (inc. test mode)	8.2mm <sup>2</sup>
Supply voltage	0.7 - 0.8V
Throughputs (TFLOPS) <sup>1,2,3</sup>	0.86 (BF16) 0.78 (FP16)
Computing density (TFLOPS/mm <sup>2</sup> ) <sup>1,2,3</sup>	0.104 (BF16) 0.095 (FP16)
Energy efficiency (TFLOPS/W) <sup>1,2</sup>	31.2 <sup>1</sup> - 65.5 <sup>2</sup> (BF16) 28.7 <sup>1</sup> - 60.4 <sup>2</sup> (FP16)
Inference Accuracy (CIFAR-100) <sup>4</sup>	69.48% (Top-1), 91.59% (Top-5)
Inference Accuracy (ImageNet) <sup>5</sup>	71.56% (Top-1), 90.17% (Top-5)

Compute-in-Memory with 31.2TFLOPS/W(ISSCC 2024 34.8)

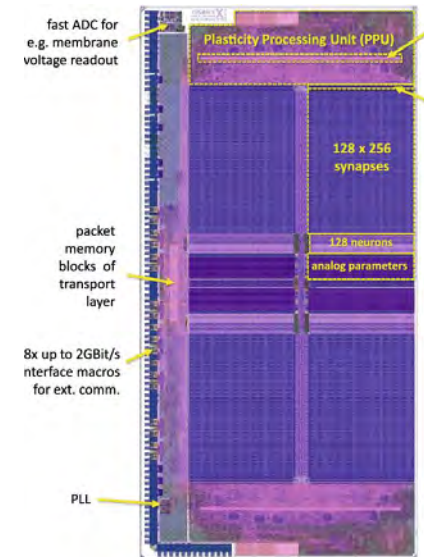
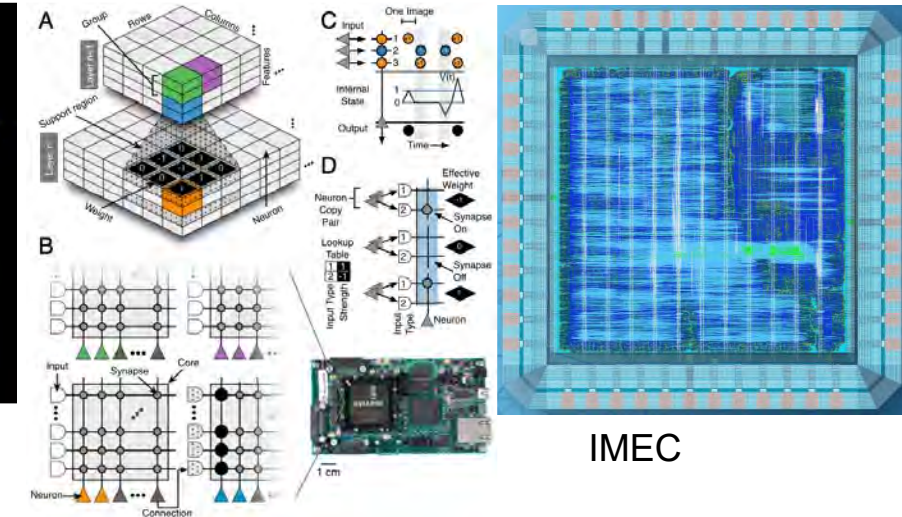


# Spiking Neural Net Accelerators

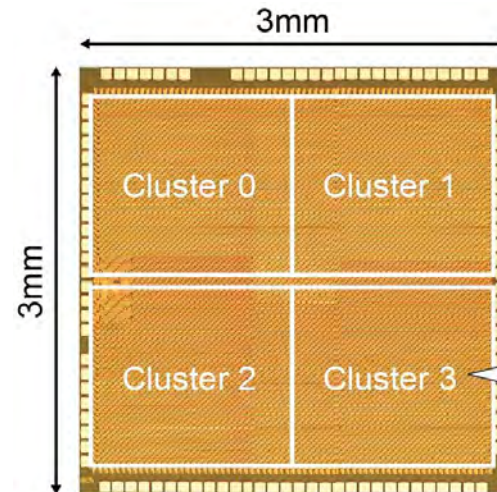
## Loihi Chip Architecture: Fine-Grained Mesh



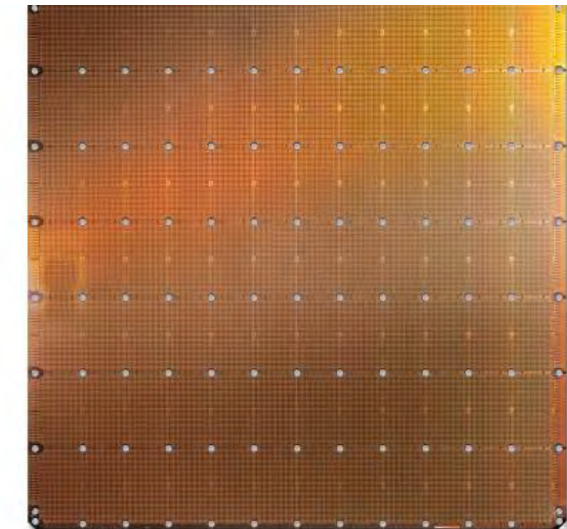
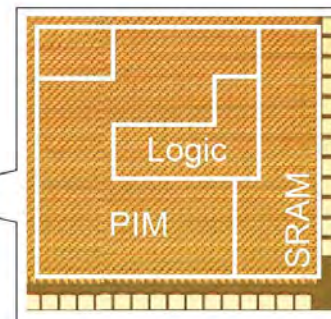
IBM TrueNorth Chip



BrainScale S-2 Chip



Renesas 8.8TOPS/W



Cerebras Trillion Transistor Chip

# **My Suggested Research Directions**

# My Thoughts

- *Need to consider more than HPL*
- Solution has to be heterogeneous & hybrid
  - Integration of “AI” beyond just mixed precision
  - Need to seriously reconsider “Probabilistic” algorithms
- More specialized computation will move:
  - Closer to CPUs on die/module
  - Further from CPUs into memory and network
- Levels of parallelism will seriously explode
  - Places more stress on networking
  - We cannot afford deep S/W stacks to start computations “over there”
- Memory system given short shrift




# Three Major Research Components

- Promote “Benchmarking for Understanding”
- Architect to reduce inefficiencies due to “border crossings”
- Develop flexible “hybrid” algorithmic techniques

# #1: Benchmarking for Understanding

## Reasons for Benchmarking

- Today • **Competitive benchmarking**: “my system is better than yours”
- Internal • **Technical benchmarking**: “I get best performance with parameters xyz”
-  • **Projective benchmarking**: “by looking at previous generations the next Moore’s Law increment tech level should increase performance by X”

## Benchmarking “Crimes”

- Selective benchmarking,
- Using wrong benchmarks,
- Improper comparison of results,
- Improper interpretation of results,
- Measurement omissions

See G. Heiser, “Systems benchmarking crimes,” <https://gernot-heiser.org/benchmarking-crimes.html>

# Current Benchmarks (From Upcoming Paper)



Inadequate for  
“Understanding”

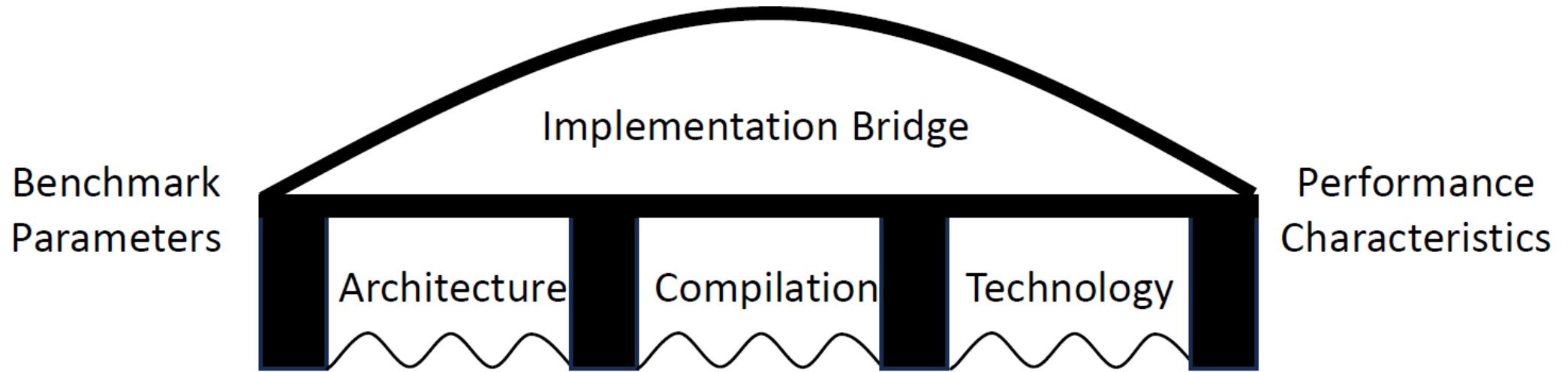
Name	Year	Reference	Area	Architecture	Type	Code	Metric
Gibson Mix	1959	[13]	general	single core	technical	none	Instr/s
Whetstone	1972	[14]	numeric	competitive	synthetic	single core	Whetstone Instrs/s
Linpack	1976	<a href="https://www.top500.org/">https://www.top500.org/</a>	linear algebra	parallel	competitive	application	flops/s
Dhrystone	1984	<a href="http://www.roylongbottom.org.uk/">http://www.roylongbottom.org.uk/</a>	non-numeric	single core	competitive	synthetic	Dhrystone Instrs/s
TPC	1988	<a href="https://www.tpc.org">https://www.tpc.org</a>	database	general	competitive	app suite	varied
SLALOM	1990	[15]	finite element	any	competitive	kernel	problem size
SPEC	1988	<a href="https://www.spec.org/benchmarks.html">https://www.spec.org/benchmarks.html</a>	general	any	technical	app suite	varied
SLALOM	1990	[15]	finite element	any	technical	suite	composite
NAS	1991	[16]	CFD	parallel	competitive	suite	varied
HPL	1993	[17]	linear algebra	parallel	competitive	application	flops/s
NAS	1994	<a href="https://www.nas.nasa.gov/software/hpl.html">https://www.nas.nasa.gov/software/hpl.html</a>	fluid dynamics	parallel	technical	suite	flops/s
Cinebench	2000	[18]	Rendering	general	competitive	application	composite
HPC Challenge	2004	<a href="https://hpcchallenge.org/hpc/">https://hpcchallenge.org/hpc/</a>	memory-intensive	general	technical	kernel suite	varied rates
BFS	2010	<a href="https://graph500.org">https://graph500.org</a>	graphs	parallel	competitive	kernel	TEP/s
[19] Firehose	2015	<a href="https://stream-benchmarking.github.io/firehose/">https://stream-benchmarking.github.io/firehose/</a>	streaming	pipelined	technical	kernel suite	datums/s
HPCG	2017	<a href="https://www.hpcg-benchmark.org/">https://www.hpcg-benchmark.org/</a>	linear algebra	parallel	competitive	kernel	flops/s
IO500	2018	<a href="https://www.io500.org/">https://www.io500.org/</a>	I/O	parallel	competitive	suite	composite
HPL-MxP	2019	<a href="https://hpl-mxp.org/results.md">https://hpl-mxp.org/results.md</a>	linear algebra	parallel	competitive	kernel	flops/s
MLPerf	2019	<a href="https://mlcommons.org/benchmarks/inference-dataset-1/">https://mlcommons.org/benchmarks/inference-dataset-1/</a>	machine learning	general	technical	suite	varied

TABLE I

A SHORT HISTORY OF BENCHMARKING CLASSICAL SYSTEMS. WHERE POSSIBLE, REFERENCES ARE TO WEBSITES CONTAINING MULTIPLE REPORTS.

# The Benchmarking Bridge

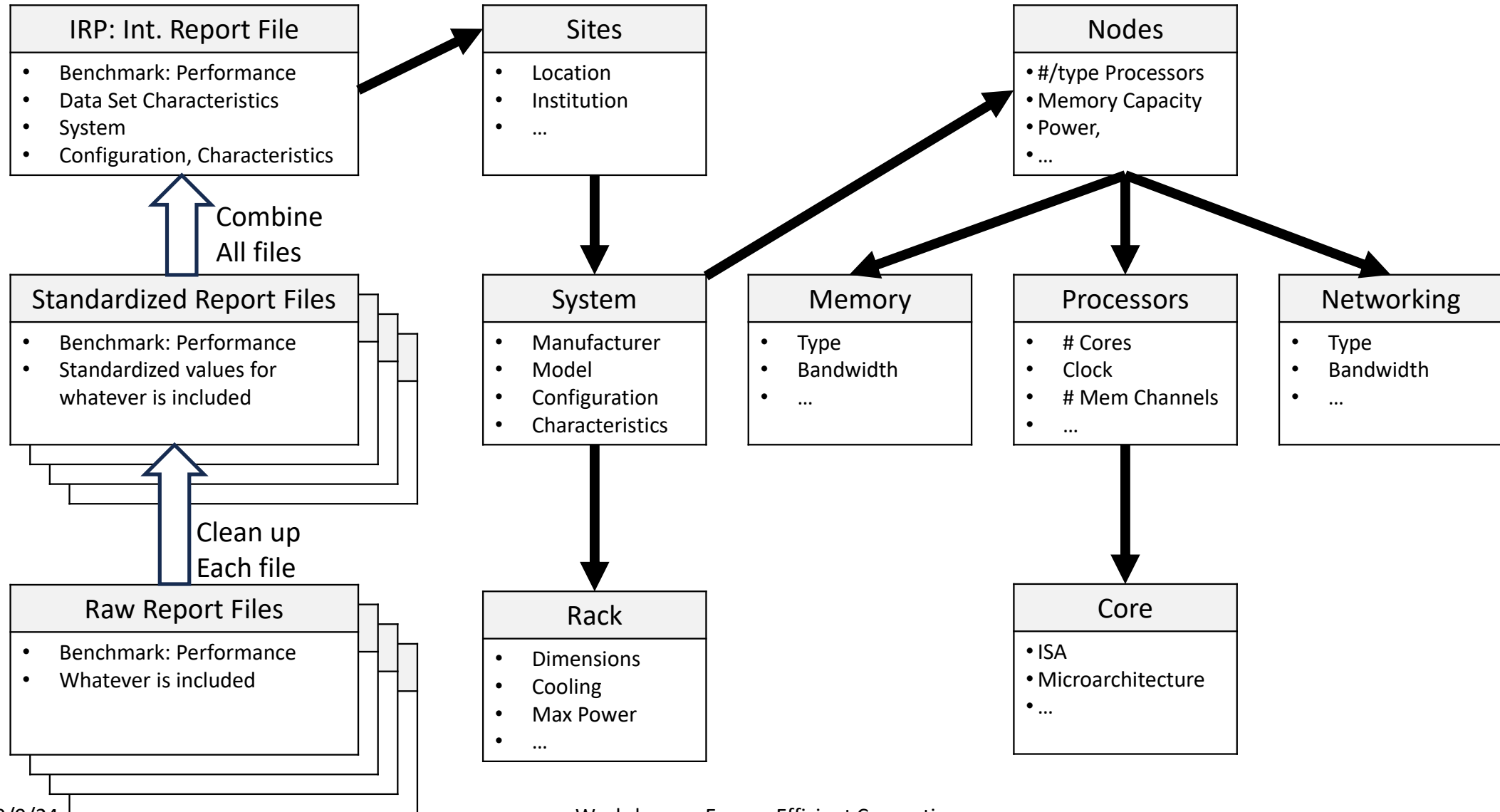
---



Goal: Institutionalize benchmarking process:

- To enable deep understanding how computational infrastructure affects performance
- As a function of benchmark characteristics

# We Need to Capture Cross-Benchmark Data



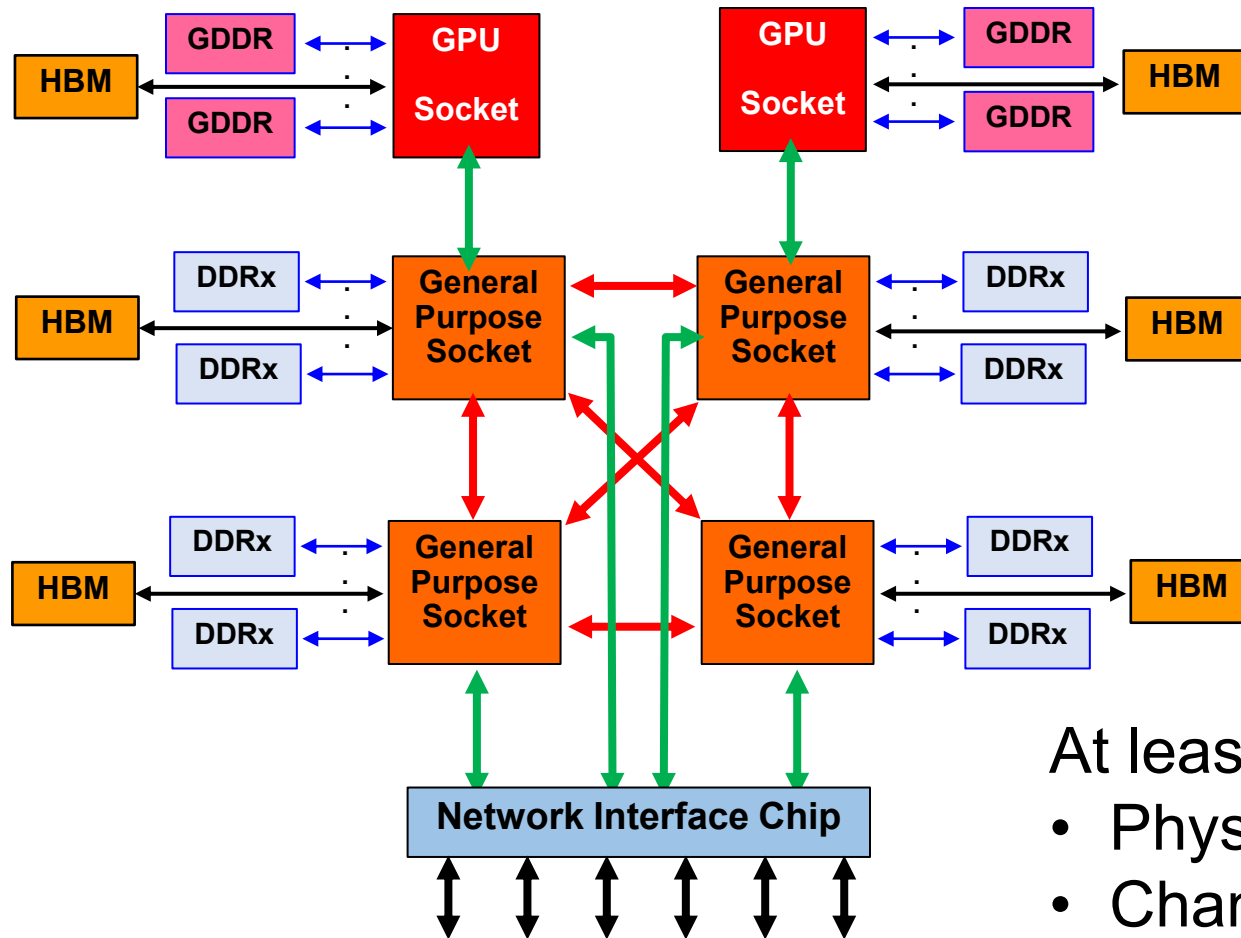


# Expanding Scalability Studies beyond Kernels

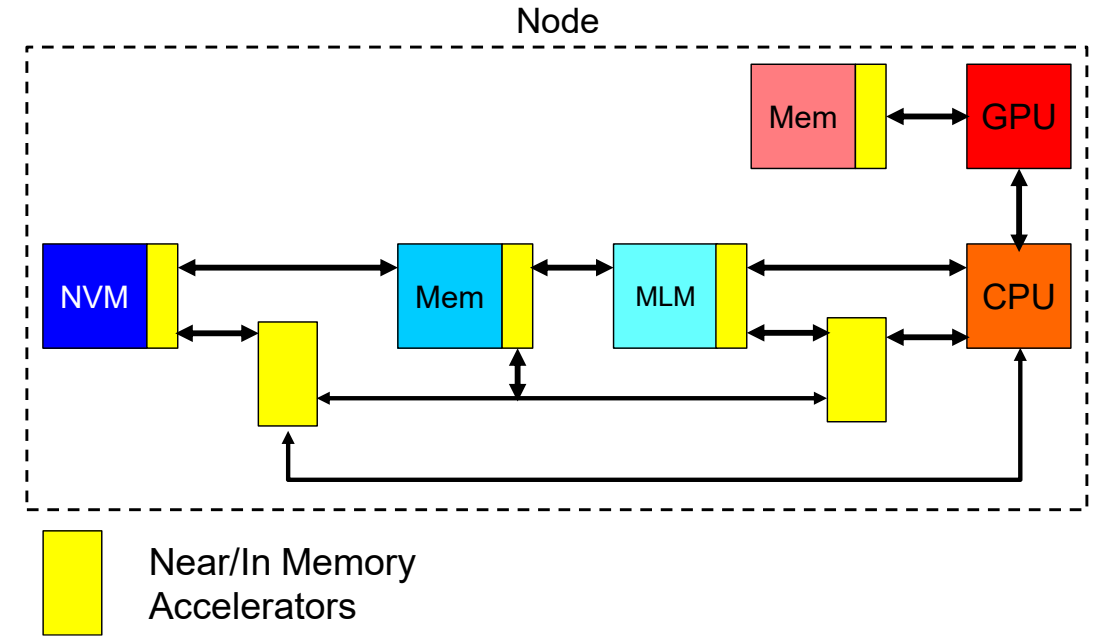
- Build on ECP approach
  - Define kernels and **WORKFLOWS:**
    - Compositions of kernels
    - Including sparsity & irregularity
    - With multiple data set sizes
    - To be measured over multiple system sizes
- i.e. Not just the highest performance but how does app/infrastructure scale – both strong and weak
- Example: current AGILE program from IARPA
    - Workflow 1: Knowledge Graphs – Groups, Relationships, and Interests
    - Workflow 2: Detection – System and Event Patterns
    - Workflow 3: Sequence Data – Identification and Clustering
    - Workflow 4: Network of Networks – such as Cyber-Physical Systems
  - Each with several variations
  - Include instrumentation to track major energy-costly events

# #2: “Border Crossings”: The New Bottleneck

Today



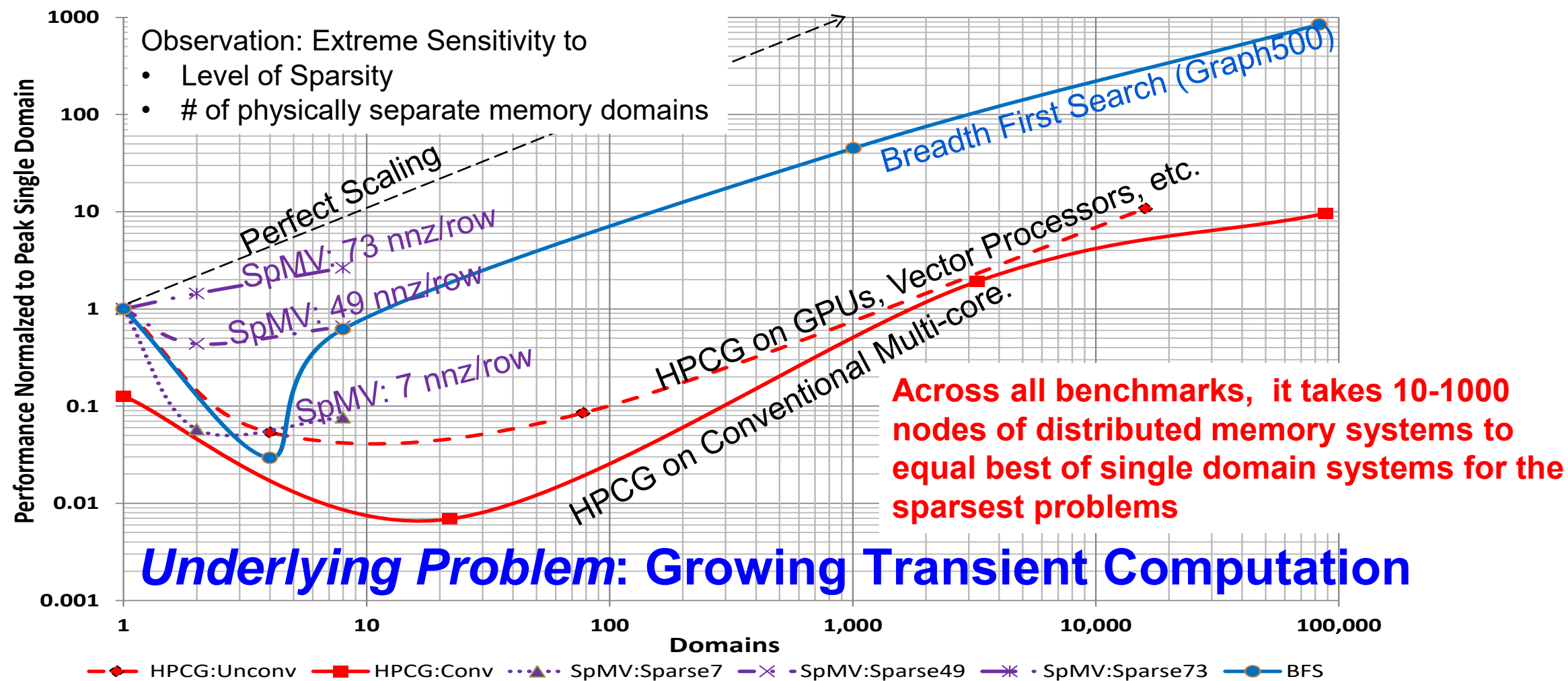
Tomorrow



At least **three kinds** of computational “crossings”

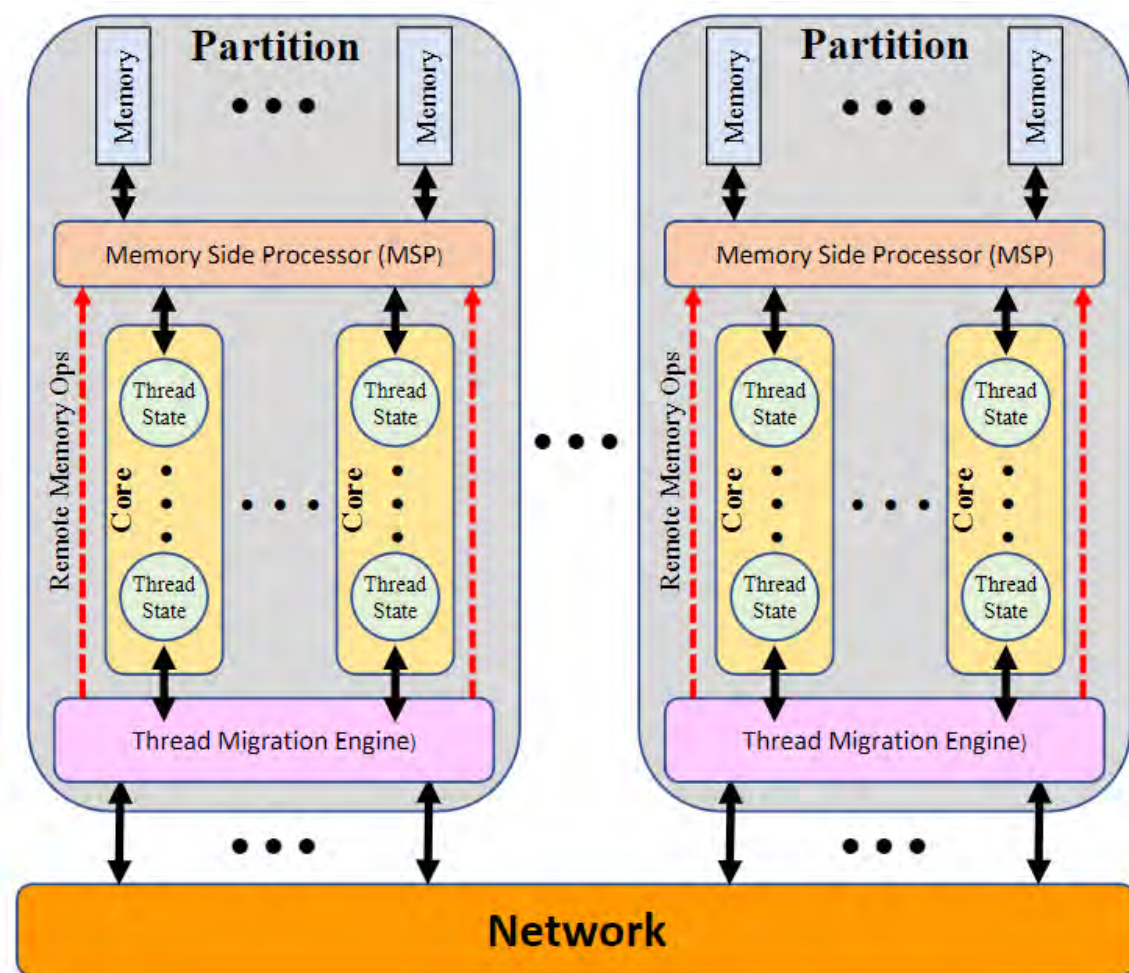
- Physical separation of computational units
- Change in ISA/Execution model
- Significantly different engine technologies

# Example Effects of Physical Separation



Bylina et al., "Performance Analysis of Multicore and Multinodal Implementation of SpMV Operation", 2014. [www.graph500.org](http://www.graph500.org). <http://www.hpcg-benchmark.org/>

# An Example of A New Architectural Technique

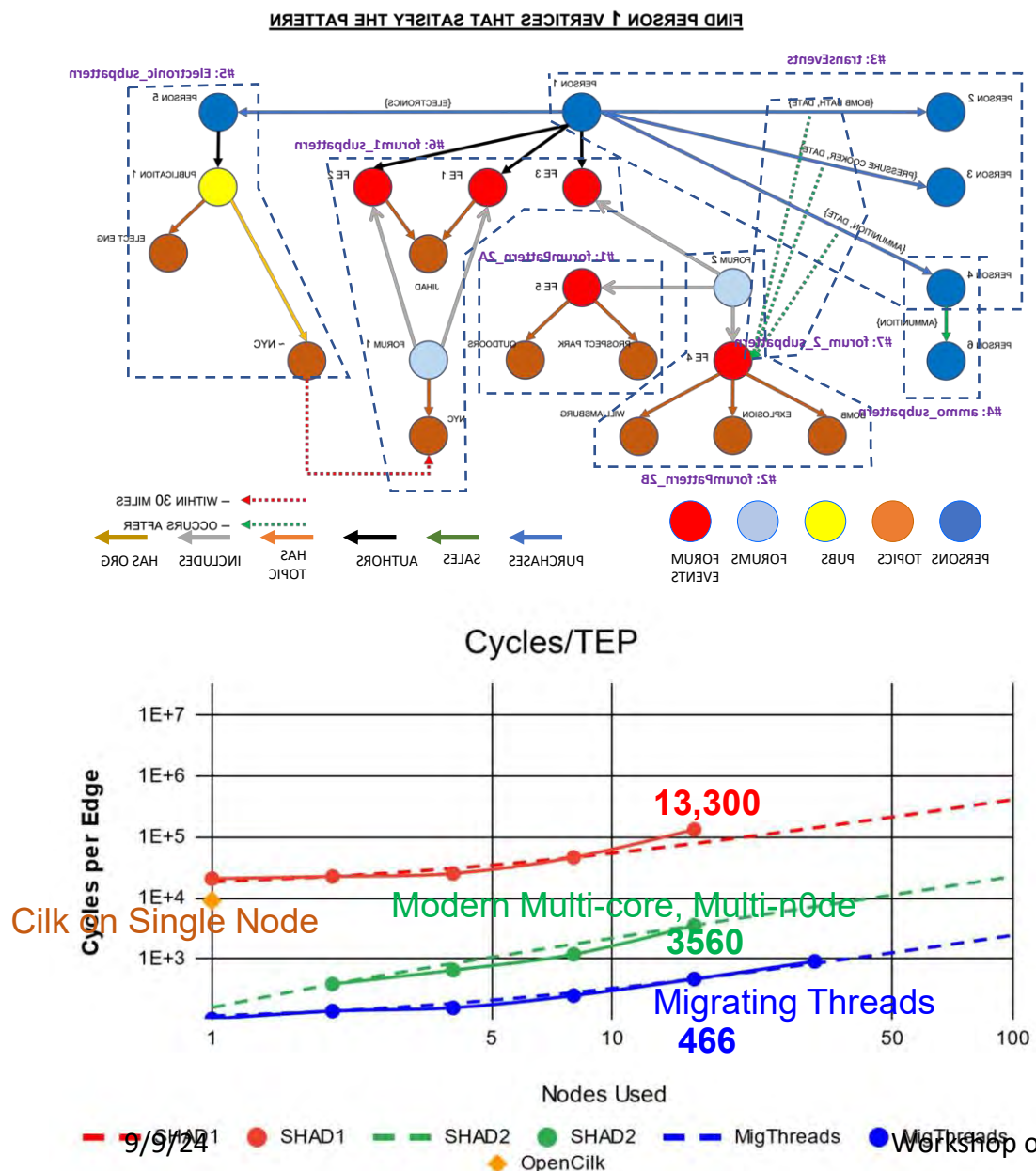


- *Migratory Threading*
- All memory on a single logical space
- On nonlocal reference: H/W
  - Suspends thread
  - Packages State
  - Sends to correct node
  - Unpacks State
  - Restarts thread
- Cheap spawns



16 node Lucata Pathfinder at GaTech

# Example of Usefulness: Subgraph Isomorphism



- Find instances of complex subgraph in a very large graph
- Multiple parallel implementations
- Mig. Threads need no comm s/w
- Takeaway: Best of conventional must use **87+%** of its cycles in handling inter-node comm
- We've seen similar results in ML, SpMV, ...

Prog. Model	Processor	Avail. Nodes	Cores/Node	Core Clock
SHAD-1	AMD EPYC 7763	16	128	2.45GHz
SHAD-2	E5-2680 v2	16	20	2.8GHz
Cilk	Xeon Silver 4208	1	16	2.1GHz
Mig. Thread	FPGA Custom	16	16	225MHz

See Kogge, McMahon, Dysart, HPEC 2024



# #3: Innovation in “Hybrid” Algorithms

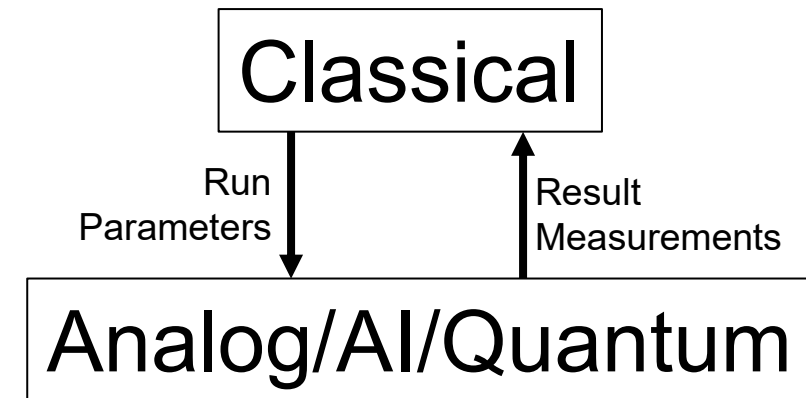
- Hardware is going heterogeneous
  - Codes will bounce between different “cores”
- Algorithms are rapidly leveraging new models of computation
  - Probabilistic, AI, Quantum, Analog in Memory...
- We need to understand algorithms that do both
  - EFFICIENTLY without wasting processing cycles

# History (and Future) of Algorithms

Technique	Repeatable Results	Bounded Precision	Execution Time
<b>Direct Solvers</b>	Yes	Within roundoff	Predictable
<b>Iterative Solvers</b>	Yes with same seed	Usually testable	Iterative improvement
<b>Monte Carlo</b>	Yes with same seed	Math bounds	Iterative improvement
<b>ML-based</b>	Yes after training	Weak	Fixed (Inferencing)
<b>Quantum</b>	No	Distribution	Needs multiple runs

As we move down

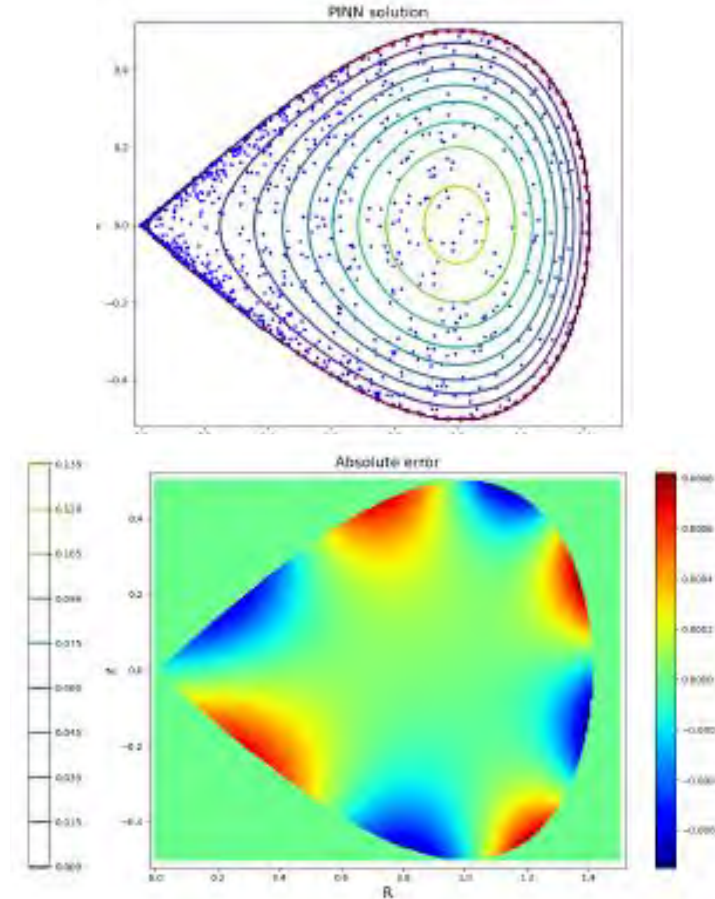
- Solutions get faster
- Likely to be more hybrid, heterogeneous
- “Accuracy” becomes less well-defined
- Need to understand “costs” of hybridization



# ML/AI in Science Apps

- Use as universal function approximators
  - Learning: *Discover* equation (learn from solvers or experimental data)
  - Inferencing: Replace conventional solvers
- PINNs: Physics Inspired Neural Nets
  - Designed with classes of ODEs/PDEs in mind
  - Train to find coefficients
- Issue: Unboundable accuracy
- Techniques exist to architect NN for dynamically adjustable time/accuracy

Example of Solovlev Equilibrium

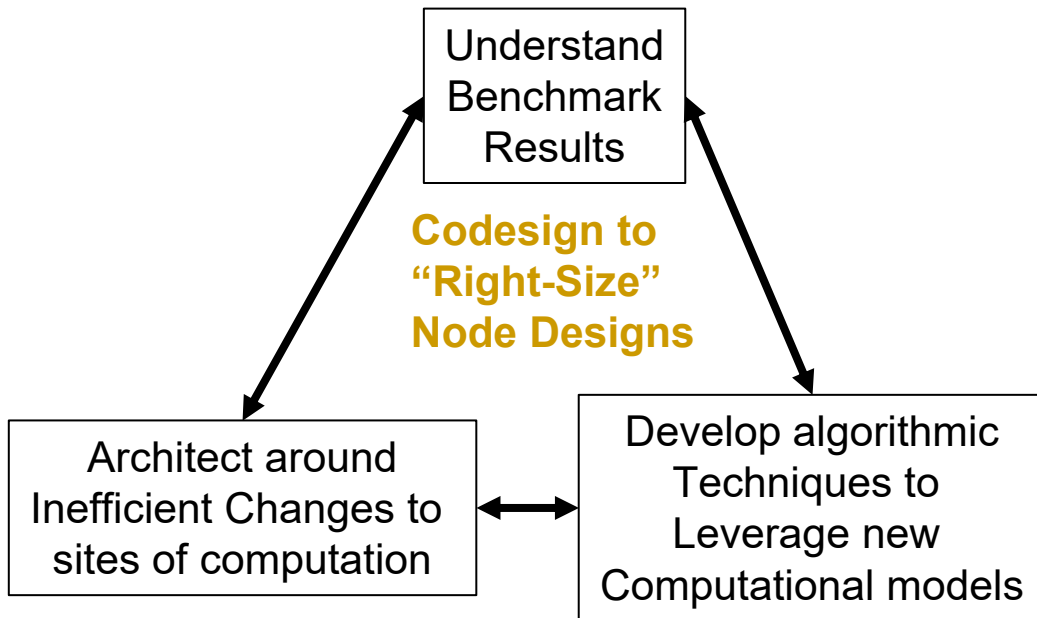


"A hands-on introduction to Physics-Informed Neural Networks for solving partial differential equations with benchmark tests taken from astrophysics and plasma physics",  
<https://arxiv.org/pdf/2403.00599v1>

See for example: <https://towardsdatascience.com/solving-differential-equations-with-neural-networks-afdcf7b8bcc4>

Also "*Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.*" Journal of Computational physics (2019)

# Summary: My Suggested Research Directions



- Much of new technology will be researched by industry
  - Rapid escalation in heterogeneity
- Research focus should be on improving energy efficiency **for Science apps** using alternative models
  1. Understand where are energy-inefficient events
  2. Develop architectures that avoid inefficient border crossings
  3. Design algorithmic techniques that reflect changing nature of computing