Tight Coupling for Performance and Productivity

Dr. Robert W. Wisniewski

Senior Vice President, Samsung

SAMSUNG SAIT

Acknowledgments

- SAIT GSP
  - Jared Alvey, Byungwoo Bang, Rohit Bhatia, Eric Borch, Ralph Castain, Sourav Chakraborty, Yong Chen, Jai Dayal, Aditya Deshpande, Praveen Francis, Manisha Gajbe, Alan Gara, Sharon Hall, Angela Highfill-Enriquez, Young-jun Hong, Wei Huang, Hanwoong Jong, Doug Joseph, Jihwan Kim, Jin Kim, Kelly Kim, Kyunghoon Kim, Sang Joon Kim, Taeksoo Kim, Patrick La Fratta, David Lombard, Michelle Kim, Zaid McKie-Krisberg, Thomas Labonte, Sehwan Lee, Seungwon Lee, Seungwook Lee, James Loo, Alfredo Metere, Ron Minnich, Shubham Nema, Quan Nguyen, Dave Poulsen, Ali Raza, Rolf Riesen, Arun Rodrigues, Nick Romero, Kyung Ryu, Samantika Sury, Andy Tauferner, Casey Thielen, Matt Turner, Greg Warren, Jim Wayda, Andy Whitaker, Robert Wisniewski, Matthew Wolf, SeokYoung Yoon, Jaehoon Yu, Elisabeth Zeller, Tina Zou, ++
- Previous Contributors
  - Wooseok Chang, Changkyu Choi, Wonyong Lee, Eunsoo Shim, Sehyun Yang

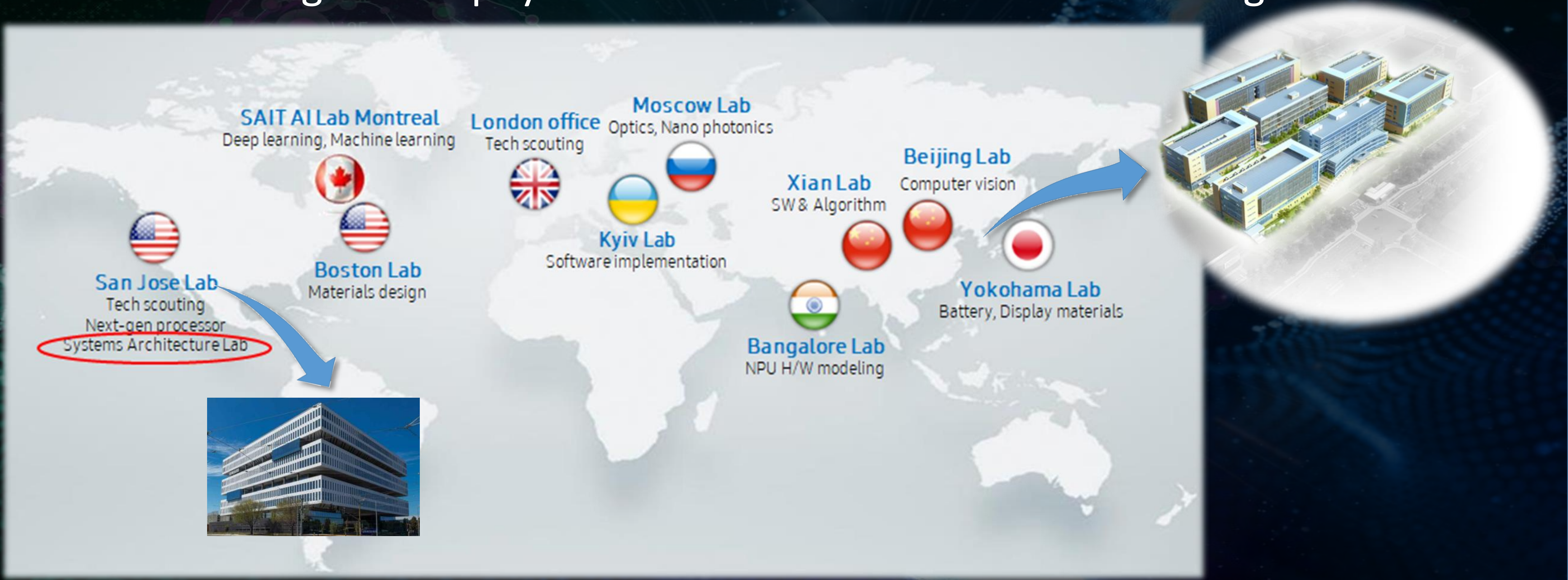
Note

- Ideas and material presented in the following slides are my view of a good technology direction and may or may not represent development or product direction for Samsung. If interested in those details, an NDA discussion would be required.

Global SAIT (Samsung Advanced Institute of Technology) Labs

- SAIT was established in 1987 as a corporate R&D Center

  - Founding Philosophy: "Boundless Research for Breakthroughs"



SAIT AI Lab Montreal
Deep learning, Machine learning

London office
Tech scouting

Moscow Lab
Optics, Nano photonics

Beijing Lab
Computer vision

Xian Lab
SW & Algorithm

Kyiv Lab
Software implementation

San Jose Lab
Tech scouting
Next-gen processor
Systems Architecture Lab

Boston Lab
Materials design

Yokohama Lab
Battery, Display materials

Bangalore Lab
NPU H/W modeling

Systems Architecture Lab

- Vision
  - To develop the most innovative technologies for future HPC and AI systems

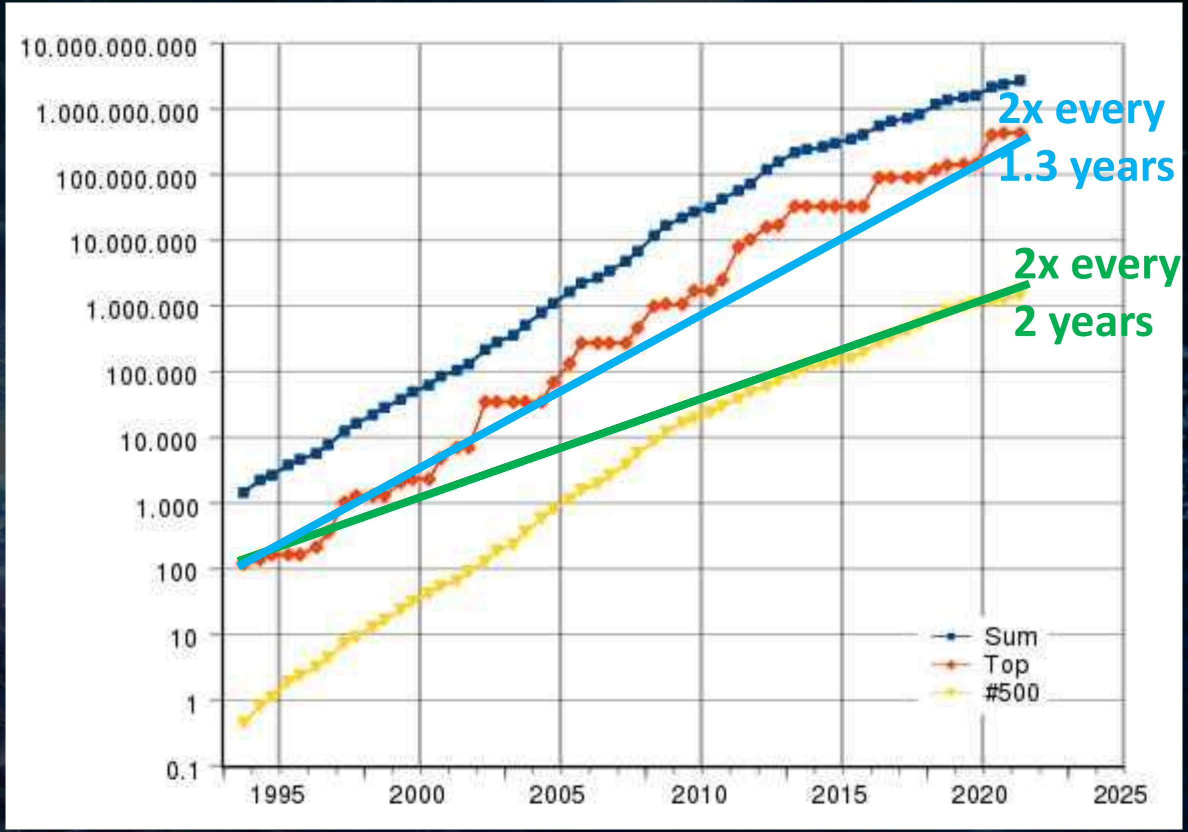
- Strategy
  - To break through the memory wall by significantly increasing the memory byte/flop ratio and reducing the power per bit with memory coupled compute
  - To break through the communication wall with high network byte/flop ratio utilizing memory coupled compute efficiencies and novel fabric technologies

Overview

- An inspiring observation
- Key considerations for HPC and AI systems
- The memory wall
- Tight coupling
- The communication wall
- Putting it all together
- Conclusion

Discontinuities

- Vectors (Cray)
- Microprocessors (Beowulf)
- Multicore, multithread (x86/ Power)
- Massive parallelism (Blue Gene)
- Heterogeneity (GPUs)
- Memory coupled compute
  - The next discontinuity
  - Innovate the future collaboratively



Source: Wikipedia.com based on data from the top500.org

Key Considerations for HPC and AI Systems

- Memory

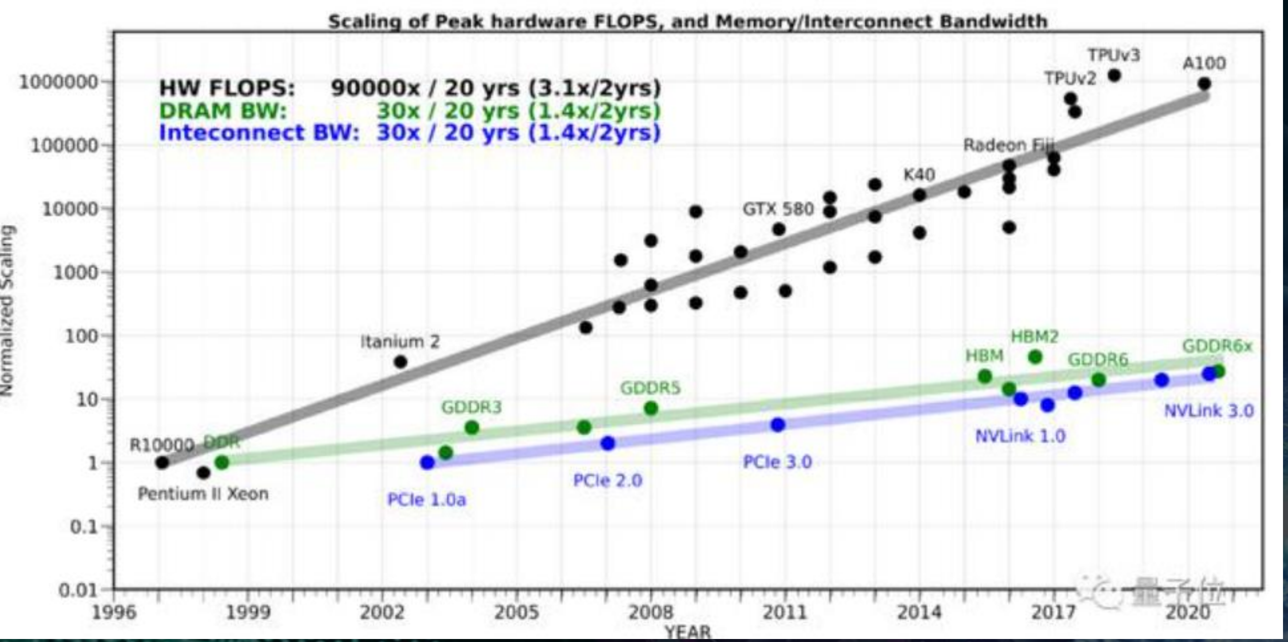
- Efficiently utilizing compute
  - Note: not more compute

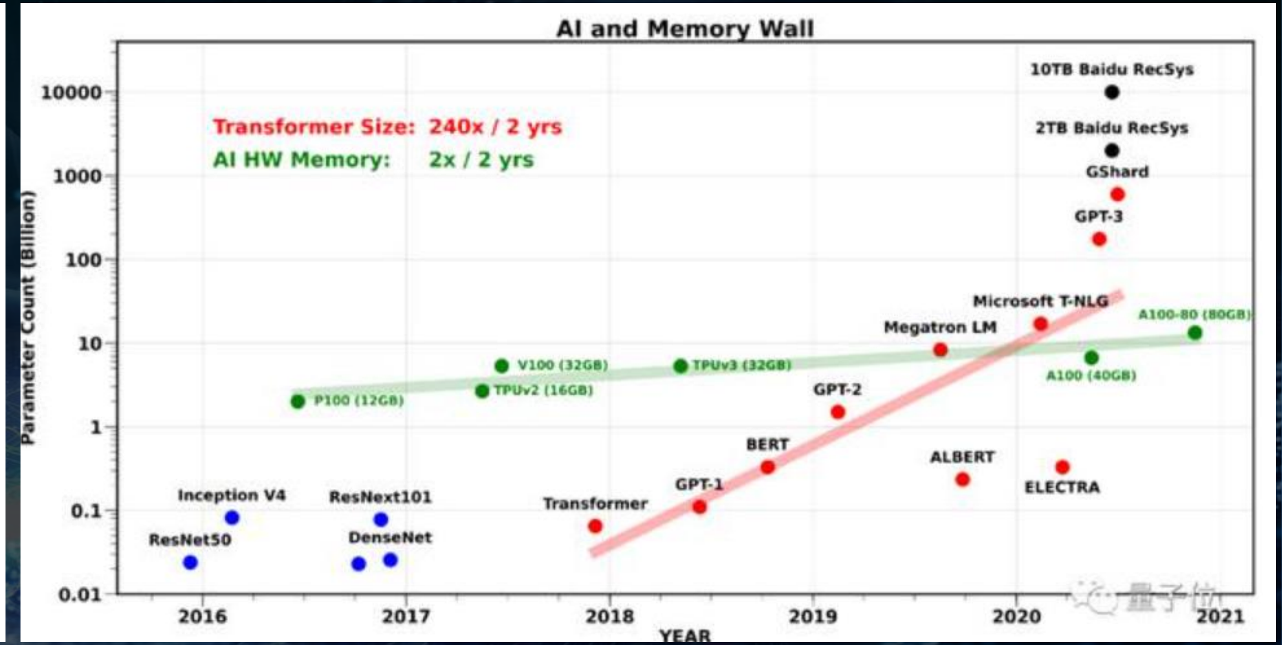
- Communication

The Memory Wall

- Coined in 1995
  - William A Wulf and Sally A Mckee
    - ACM SIGARCH computer architecture news, 1995
  - Observed that processors are getting faster faster than memory is getting faster
    - "each is improving exponentially, but the exponent for microprocessors is substantially larger than that for DRAMs. The difference between diverging exponentials also grows exponentially"
  - DDR
    - DDR2-200 1.6 GB/s released 1999 available in 2000
    - DDR5 32-64 GBs released 2020 available in 2021
  - HBM
    - HBM 128 GB/s adopted and available 2013 used 2015
    - HBM3 819 GB/s January 2022

The Memory and Communication Wall is getting Higher



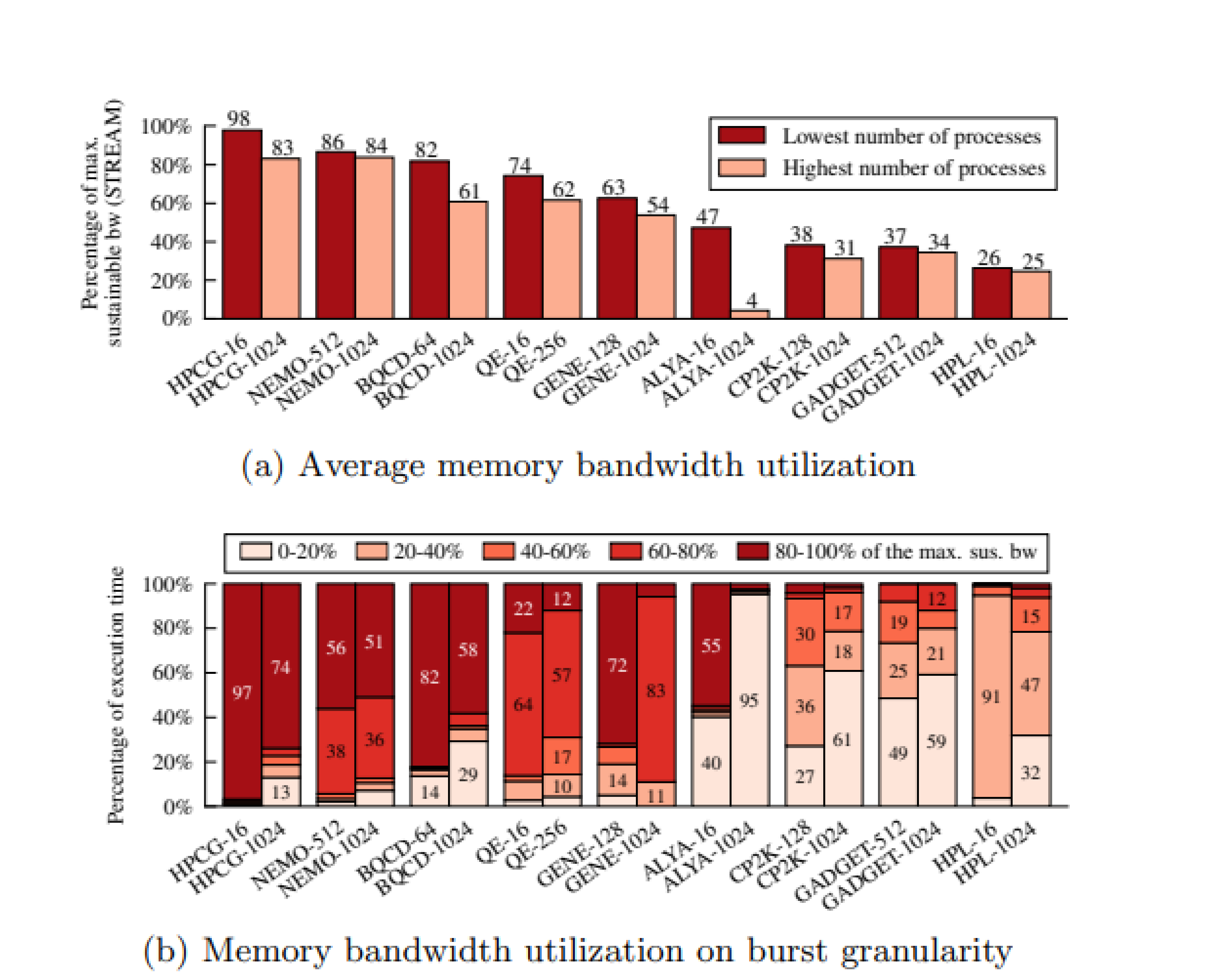


- https://daydaynews.cc/en/science/the-biggest-obstacle-to-ai-training-is-not-computing-power.html

- Modeling and simulation, and some AI apps, are memory bandwidth limited
- AI, and some mod/sim, applications are communication bandwidth limited

Many Applications are Memory Bound

- The increasing divergence between compute and memory has led to an increasing number of applications that are memory bound
- The best component to improve modeling and simulation applications' performance is memory bandwidth

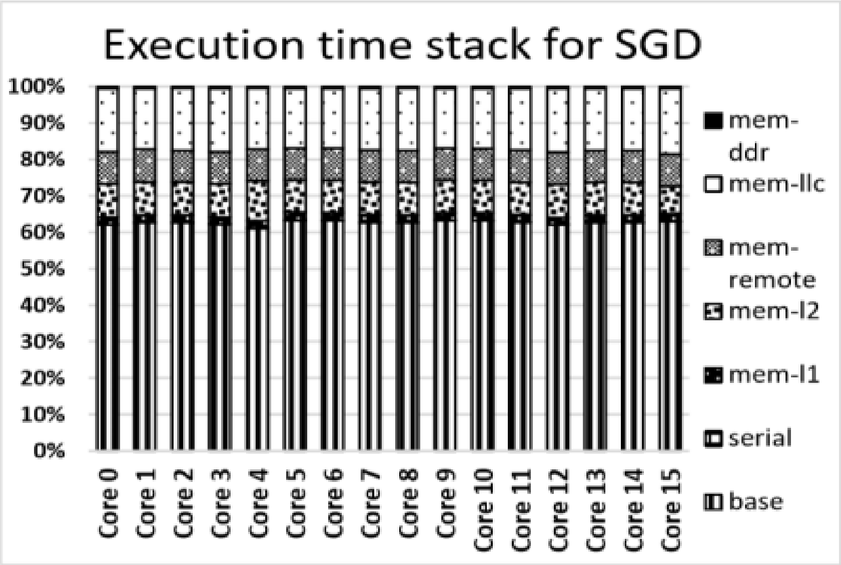


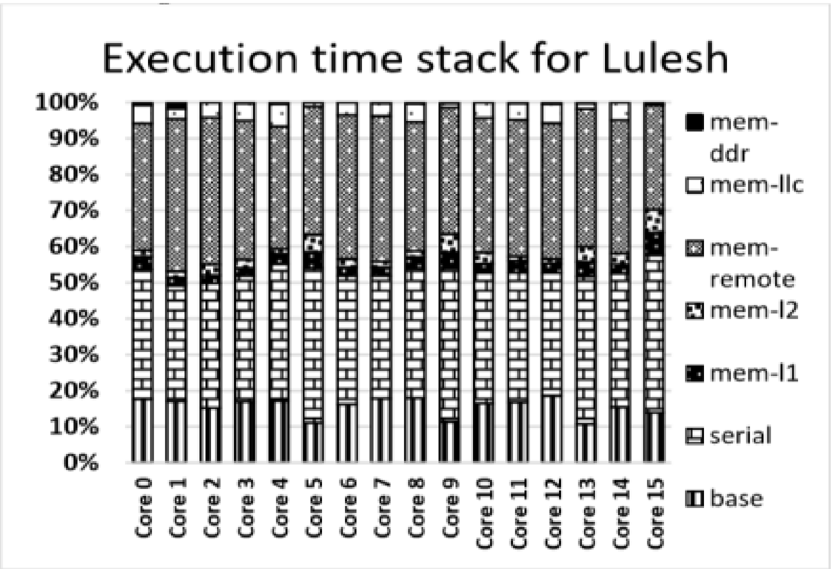
(a) Average memory bandwidth utilization

(b) Memory bandwidth utilization on burst granularity

HPC Benchmarking: Scaling Right and Looking Beyond the Average, Milan Radulović et. al., International Conference on Parallel and Distributed Computing, 2018

Impact of Memory Performance





Stochastic Gradient Descent used in Machine Learning Algorithms

Hydrodynamics code used in Classical HPC

Why are we spending so many cycles communicating data?

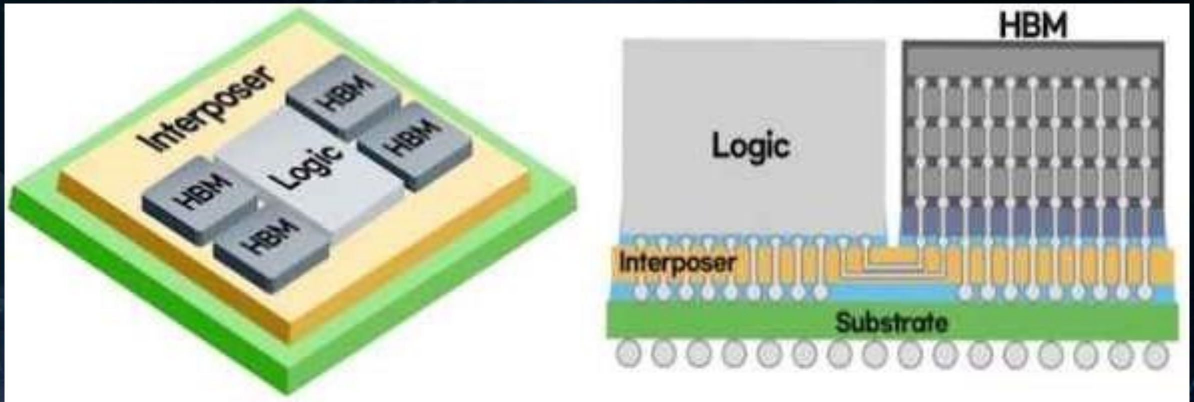
IEEE AICCSA 19: CONCORD: Improving COmmuNication using COnsumeR-Count Detection Farah Fargo, Shobha Vissapragada, Samantika Sury

Addressing the Memory Wall

- Put compute close to memory
  - 2.5D (Processing near memory)
    - Current technology
    - HBM co-packaged with compute
  - PIM (Processing in Memory)
    - Closest possible to memory
    - Current constraints limit functionality
  - 3D
    - Compute closer to memory than in 2.5D
    - Reduces power consumption
    - More efficient packaging than in 2.5D

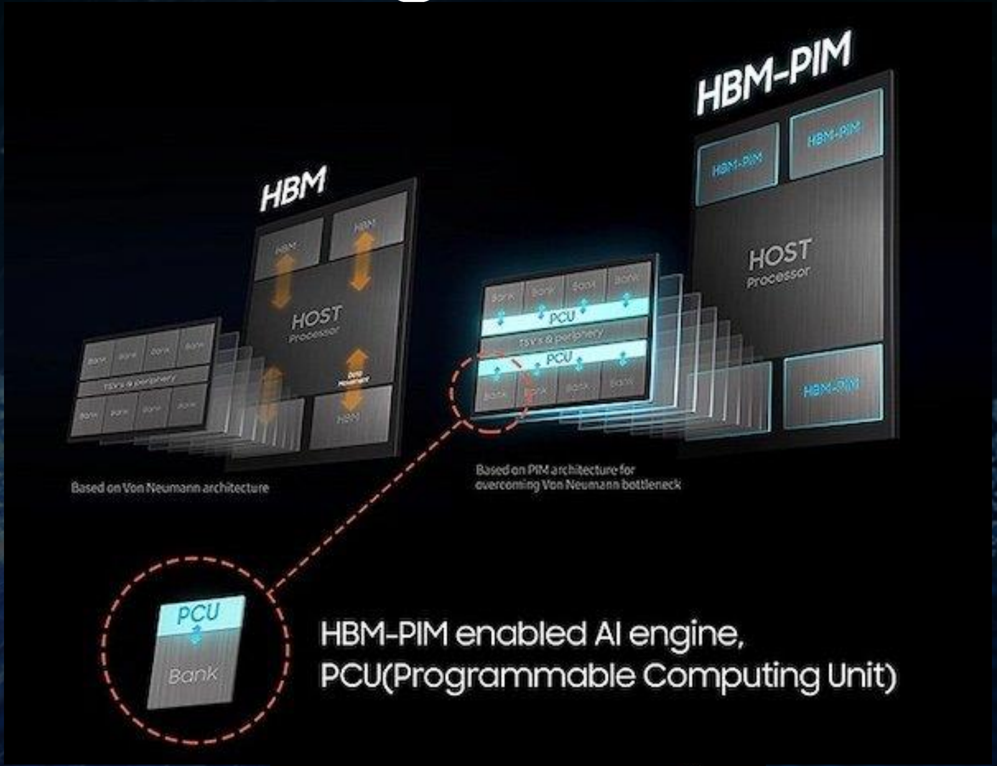
2.5D Opportunities and Challenges

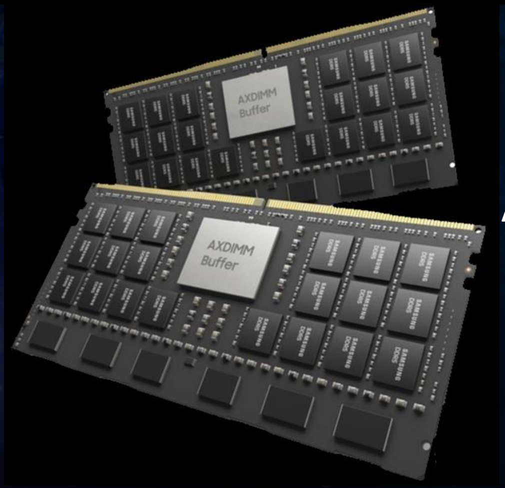
- Significant improvement over DDR
  - Bandwidth is higher
  - Latency on par
- Substrate and connections can be expensive
- Requires off die connection from logic to HBM
  - Off-die signals require more power
  - Takes die area to connect the wires



PIM Opportunities and Challenges

- Most energy efficient compute
  - ALUs on same die as memory cells
  - Data movement is minimal
- The type of operations are constrained
  - ALUs reduce memory are or increase die area
- The operations are synchronous
  - If conforming to JEDEC standard


HBM
HOST
Processor
Based on Von Neumann architecture
HBM-PIM
HBM-PIM
HBM-PIM
HBM-PIM
PCU
PCU
TSVs & periphery
HOST
Processor
Based on PIM architecture for overcoming Von Neumann bottleneck
PCU
Bank
HBM-PIM enabled AI engine,
PCU(Programmable Computing Unit)

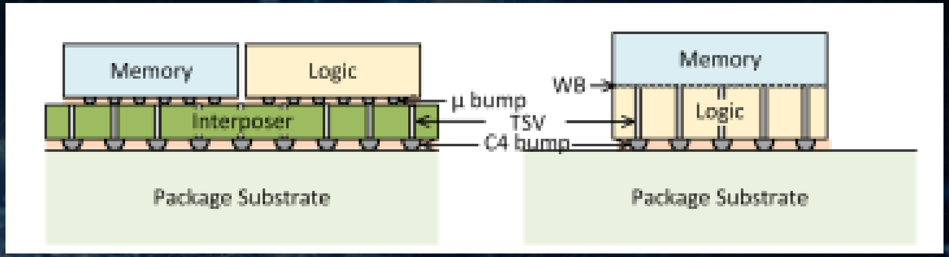


AXDIMM

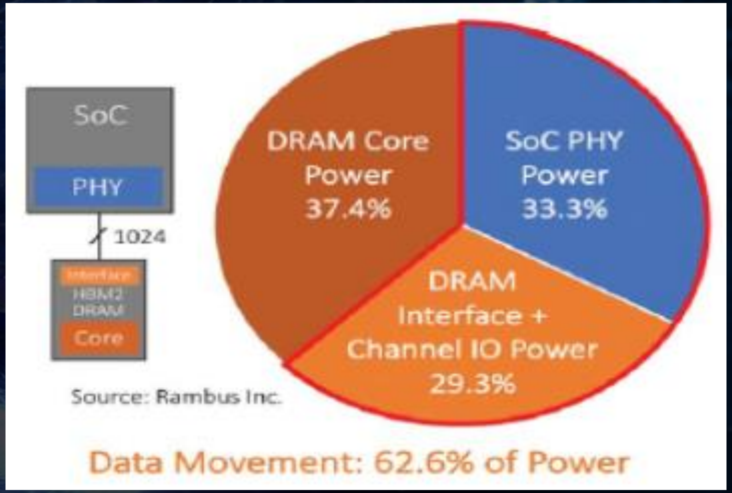
3D

- Improves power efficiency
  - Data moves less
- Reduces latency
  - Data travels less distance
- Allows general purpose logic
- Key decisions
  - What compute
    - Keep the programming model productive
  - How much compute
    - Provides opportunity for high B/F ratio

Closer coupling of compute with memory



e.g. 3D systolic ML accelerators in IEEE Journal on Exploratory Solid-State Computational Devices and Circuits – June 2021



Source: Rambus Inc.

Data Movement: 62.6% of Power

Productively Utilizing Compute - Tight Coupling

- Accelerators are more challenging to use than general purpose cores

- Accelerators have higher efficiency than general purpose cores
  - Performance/power
  - Performance/cost

- A tightly coupled architecture allows more productive use of accelerators
  - Bandwidth
  - Latency
  - Coherency

GPUs and Today's AI have Co-Evolved

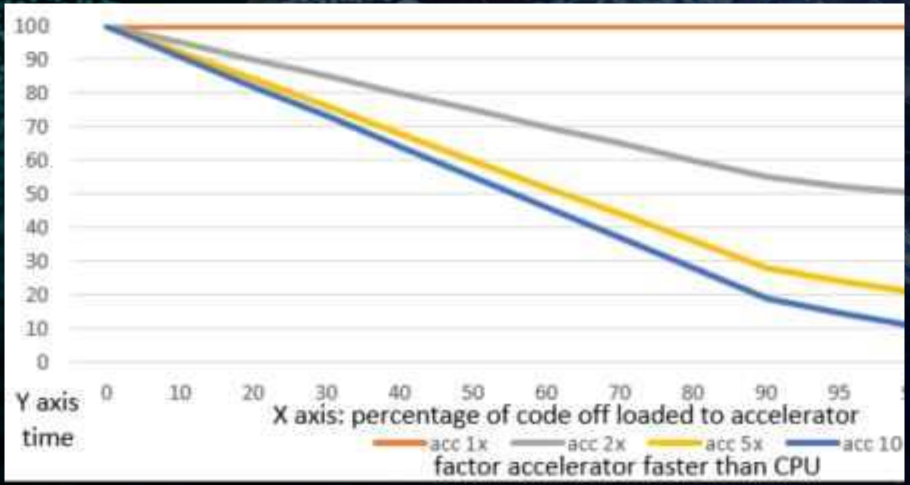
- As AI progressed GPUs included features useful to them
  - Volta (2017) introduced tensor cores a 4x4 matric multiple and accumulate
  - Turing (2018) introduced integer tensor cores
  - Support of BF16
    - Considerations to further optimize for stability and convergence
- As GPUs progressed AI applications were modified to leverage new features
  - On early GPUs there was long latency, low bandwidth, disparate memory regions
    - Codes (especially inference) written to be [off]loaded once to GPU
- We are done – right ?

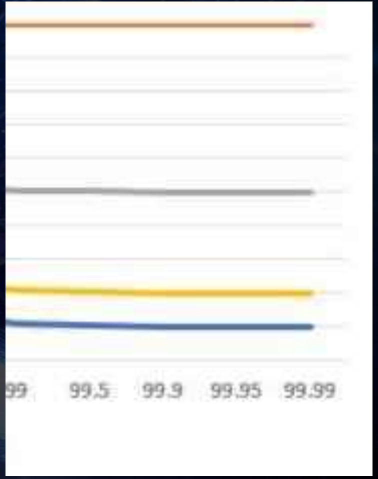
GPUs and Today’s AI Have Co-Evolved

- As AI progressed GPUs included features useful to them
  - Volta (2017) introduced tensor cores a 4x4 matric multiple and accumulate
  - Turing (2018) introduced integer tensor cores
  - Support of BF16
    - Considerations to further optimize for stability and convergence
- As GPUs progressed AI applications were modified to leverage new features
  - On early GPUs there was long latency, low bandwidth, disparate memory regions
    - Codes (especially inference) written to be [off]loaded once to GPU
- GPUs and HPC have not Co-Evolved as closely as GPUs and AI
  - Many key HPC codes were written before GPGPUs
  - Large AI codes are becoming more HPC like

HPC Leveraging Accelerators Still Requires Work

- Portions of HPC applications need or prefer serial cores
  - Many HPC applications remain bulk synchronous
    - Percentage of parallel code may be high
    - Interrupted by code that can not or would be better not run on accelerator
  - Brachy code between loops
  - MPI runtime and communication
- Running higher percentage of code on accelerator improves performance
  - Higher percentage of off-loadable code implies finer-grained parallelism

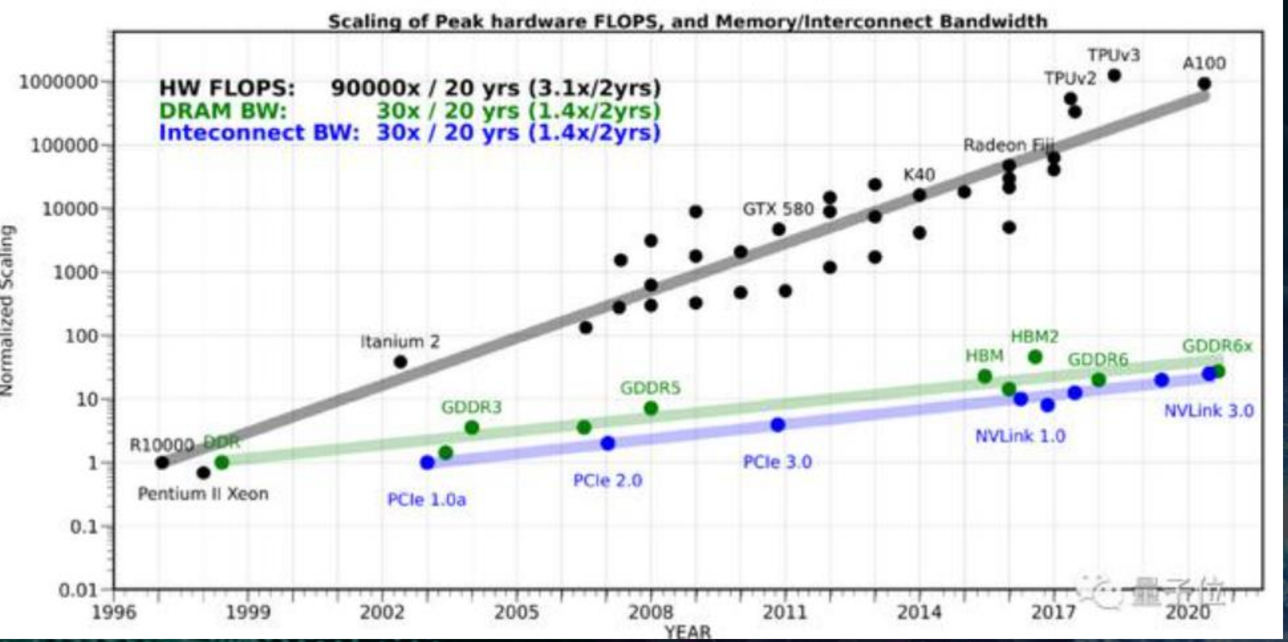


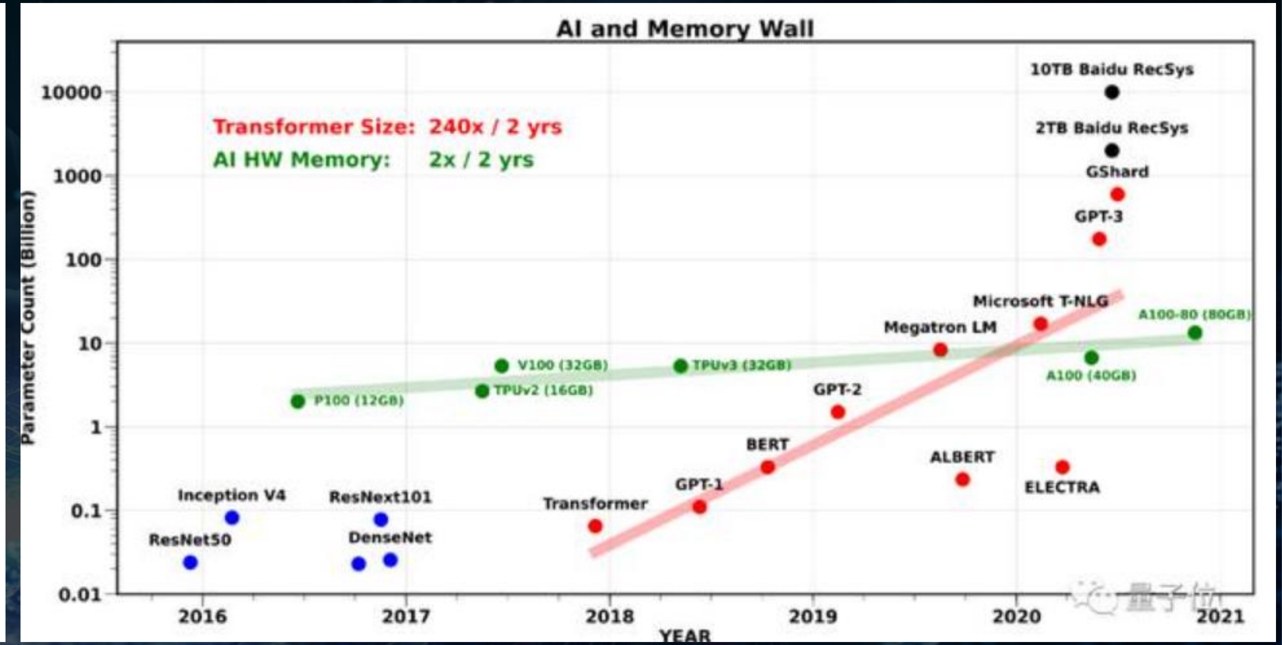


HPC Leveraging Accelerators Still Requires Work

- Portions of HPC applications need or prefer serial cores
  - Many HPC applications remain bulk synchronous
    - Percentage of parallel code may be high
    - Interrupted by code that can not or would be better not run on accelerator
  - Brachy code between loops
  - MPI runtime and communication
- Running higher percentage of code on accelerator improves performance
  - Higher percentage of off-loadable code implies finer-grained parallelism
    - There is a cost to running on accelerator due to
      - Bandwidth for accelerator to access data
      - Latency to launch first line of code
        - Hardware and software components
    - There is a development effort to run code on accelerator

The Memory and Communication Wall is getting Higher





- https://daydaynews.cc/en/science/the-biggest-obstacle-to-ai-training-is-not-computing-power.html

- Modeling and simulation, and some AI apps, are memory bandwidth limited
- AI, and some mod/sim, applications are communication bandwidth limited

Addressing the Communication Wall

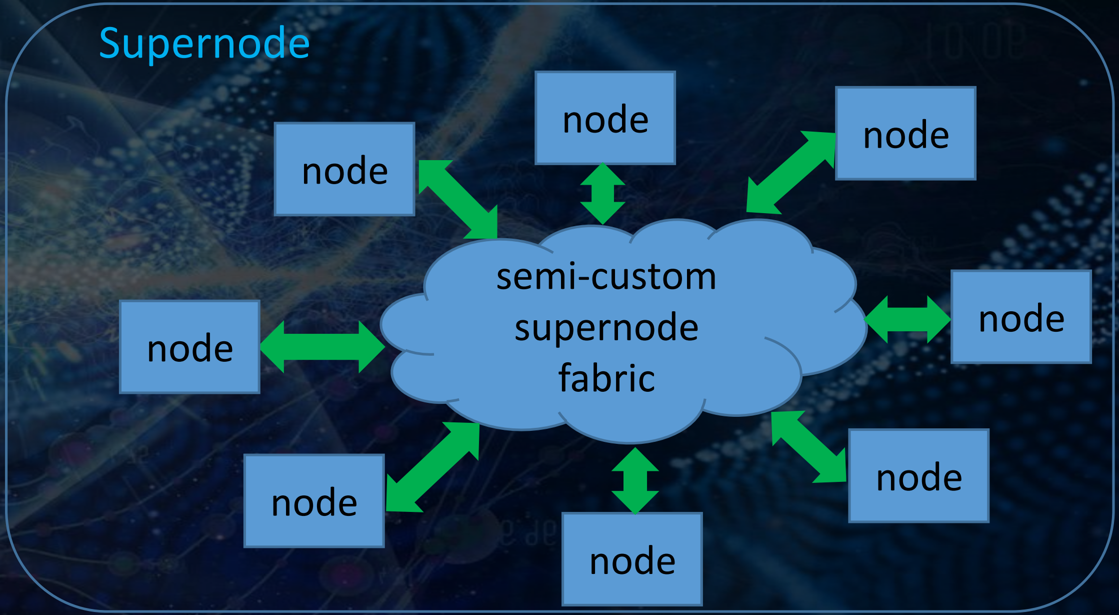
- Closer coupling of compute with memory and communication
  - Cost-efficient performance
  - Power sharing

- 3D packaging → higher communication performance
  - High point-to-point and all-to-all bandwidth

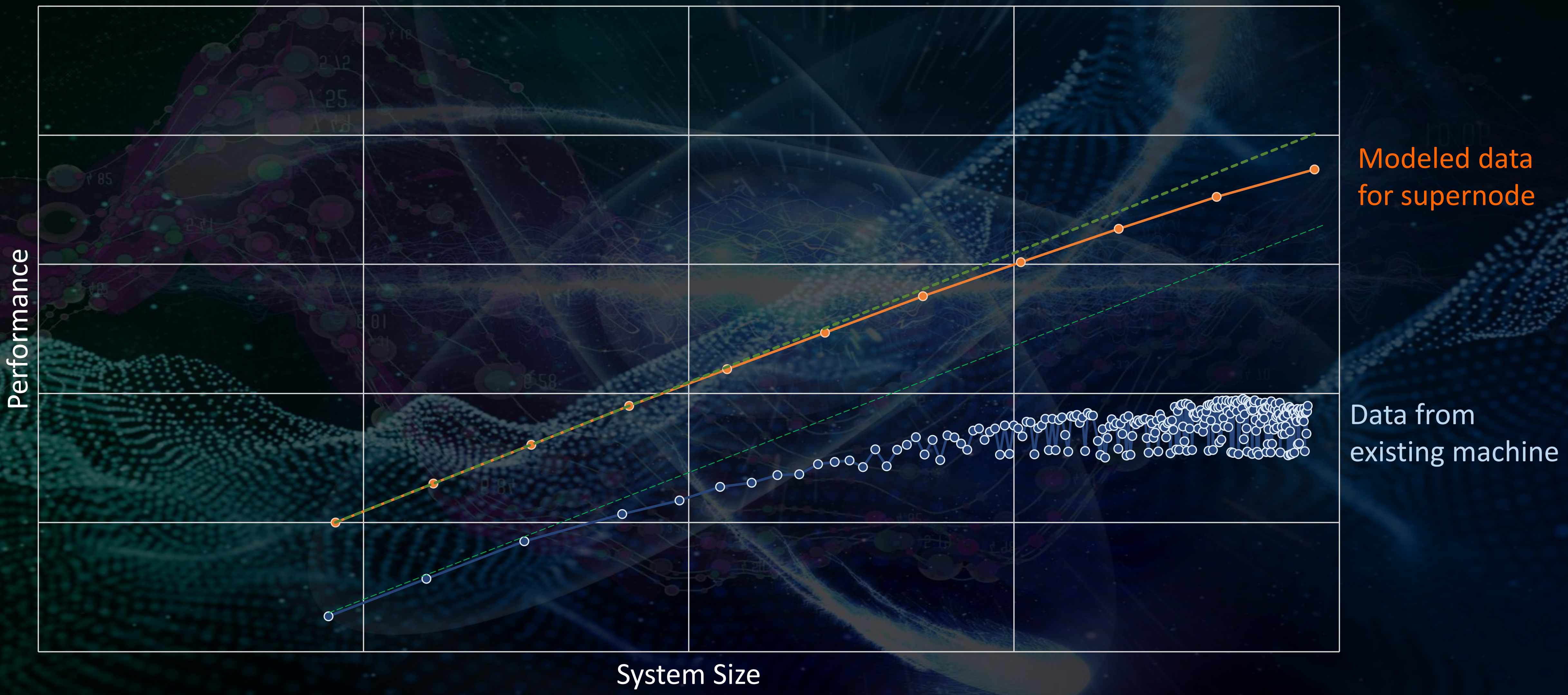
- Large supernodes with productive programming model
  - Valuable to AI models for large reductions and large data exchanges, parallel FFT
  - Utilize a productive programming model

Supernode Desired Properties

- Much larger set of nodes with SMP-like behavior
  - Capabilities that would be advantageous: low latency, high bandwidth, atomics, globally accessible memory between nodes
- Programming model to allow developers to transparently or explicitly leverage above capabilities
- Semi-custom fabric to enhance power properties
- A shared memory model for productive prorgamming



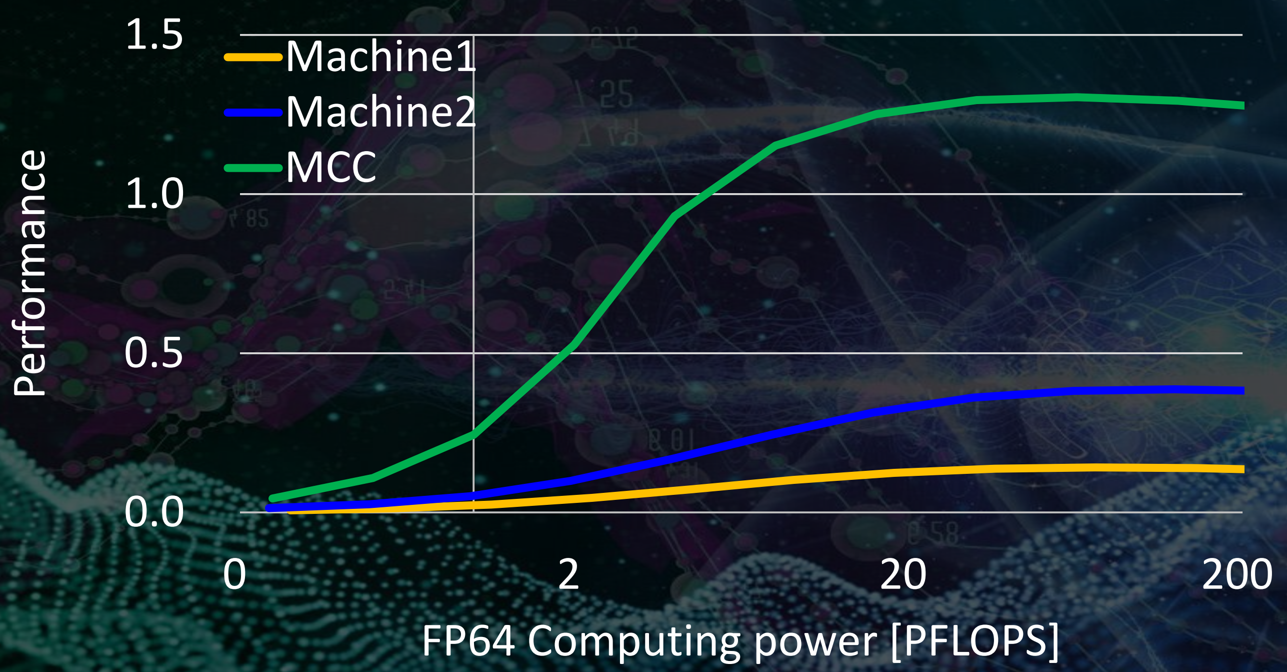
Benefits of the Supernode for Strong Scaling

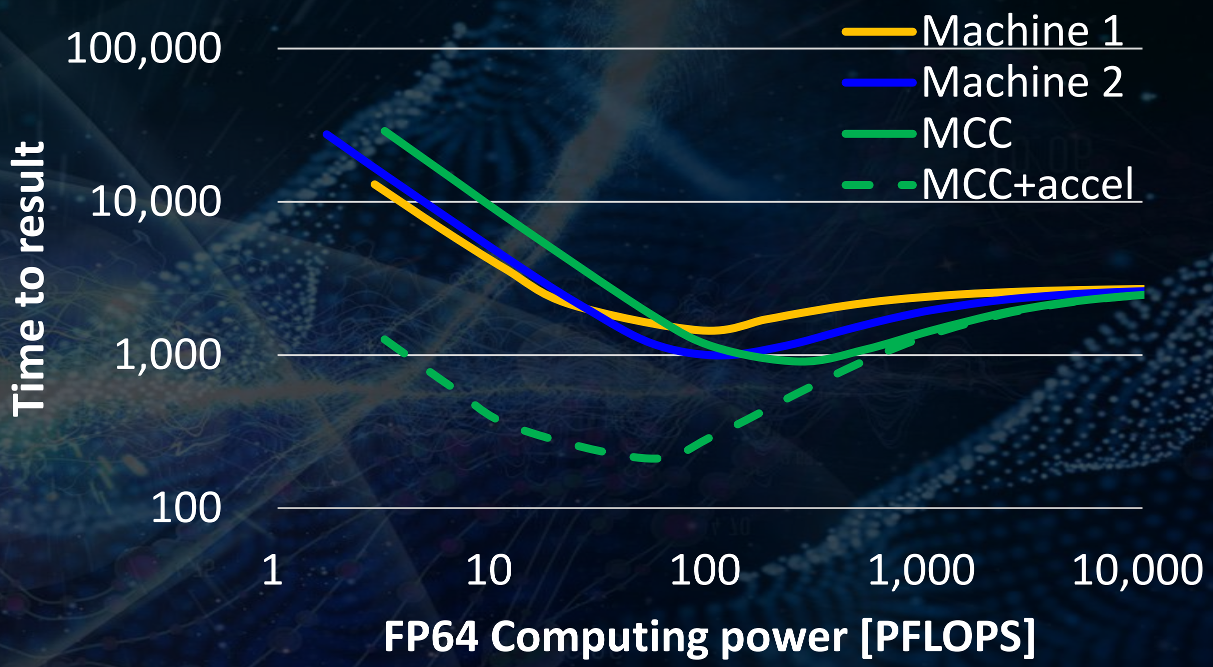


System Overview

- Key innovations to advance HPC and AI
  - Memory Coupled Compute
  - Productive and tight-coupling of mainstream cores with accelerators
  - Supernodes (large and high-performing globally accessible memory)
  - System-level energy-efficiency

Benefits for a Classical HPC and an AI Training Application





- Memory and communication bound classical HPC code
  - Y axis performance: higher is better

- Communication bound BF16 hungry multi-T AI app
  - Y axis time: lower is better

SAMSUNG SAIT

Standard Productive Software Stack

Application	HPC Simulation		AI / Deep Learning		Data Analysis
Framework Library	MATLAB	BLAS	TensorFlow	PyTorch	scikit-learn
Profiler Debugger	Score-P	VAMPIR	Valgrind	GDB	ARM DDT
Parallel Programming	OpenMP	OpenACC	SYCL	Kokkos	MPI
Programming Language	C/C++	Fortran	Python	Julia	Java
Management	SLURM	LSF	Docker	Singularity	Spack
File System	Lustre / DAOS				
Operating System	Linux / Lightweight OS				
Hardware	MCC SoC		Inter-node Connection		

OpenHPC
as base

Innovating the Next Discontinuity

- The time is right to innovate the next discontinuity
  - Vision: In the future memory coupled compute will be ubiquitous
- Tightly coupling memory, compute, and communication will allow future optimizations
- Focusing on exploring technology for AI and HPC systems

Thank You