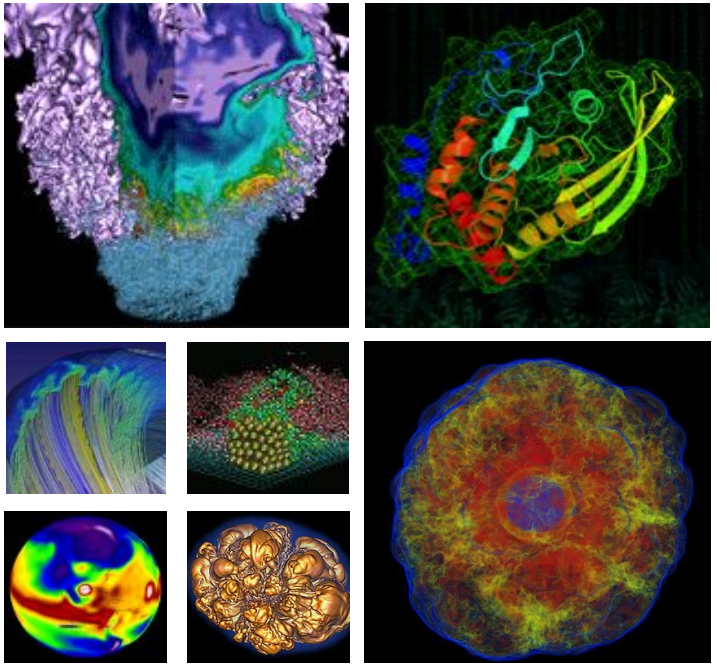
A Metric for Evaluating Supercomputer Performance in the Era of Extreme Heterogeneity



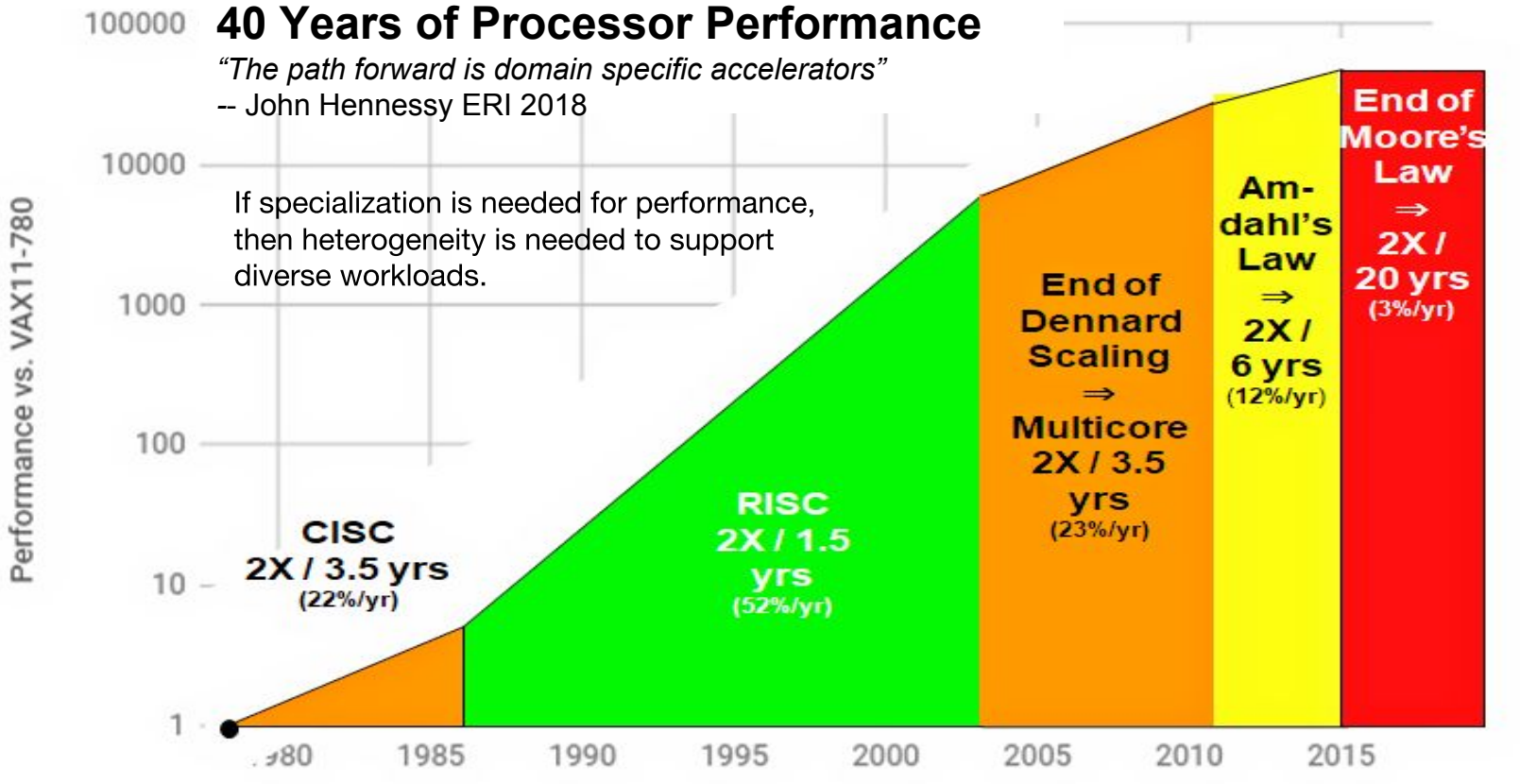
Brian Austin, Chris Daley, Doug Doerfler,
Jack Deslippe, Brandon Cook, Brian Friesen,
Thorsten Kurth Charlene Yang & Nicholas Wright

PMBS Workshop / SC'18

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

The Dawn of Specialization & Heterogeneity

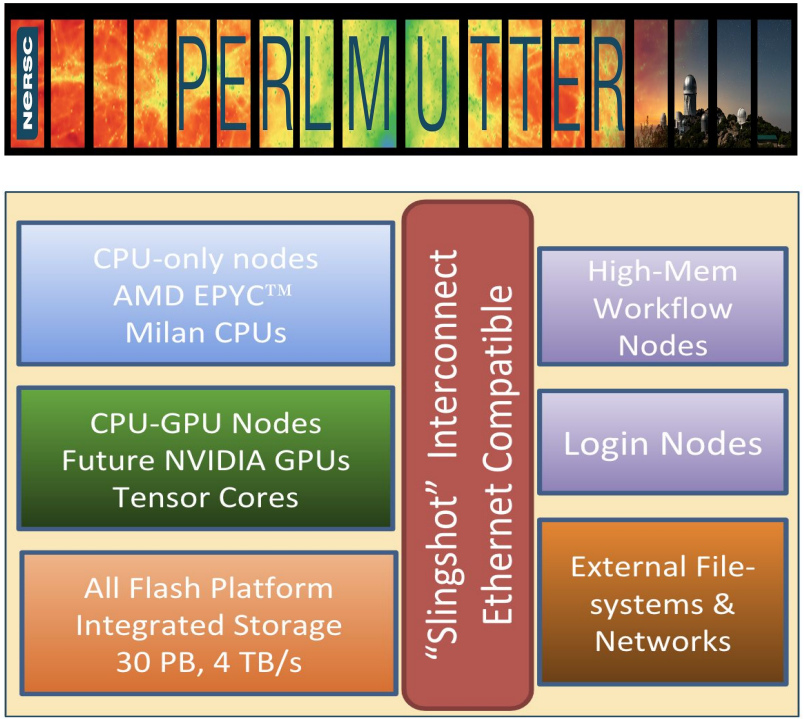


Heterogeneity is already a reality in HPC

- Heterogeneity within nodes:
  - AES accelerators
  - CPU & GPU sockets
  - Domain Specific Architectures
    e.g. ANTON, Google TPU, Nvidia Tensor Units,
    Xilinx FPGA integrated AI engines, …
- Heterogeneity across nodes:
  - NERSC's Cori and NNSA's Trinity: Intel Haswell & Intel KNL partitions
  - NCSA's Blue Waters: AMD CPU & AMD/Nvidia GPU partitions

Perlmutter: A System Optimized for Science

- **Cray Shasta System providing 3-4x capability of Cori**
- **GPU-accelerated and CPU-only nodes meet the needs of large scale simulation and data analysis from experimental facilities**
- **>4,000 node CPU-only partition provides same capability as all of Cori**
- **GPU nodes: 4 NVIDIA GPUs each w/Tensor Cores & NVLink-3 and High-BW memory + 1 AMD "Milan" CPU**
  - Unified Virtual Memory support improves programmability
- **Cray "Slingshot" - High-performance, scalable, low-latency Ethernet- compatible network**
  - Capable of Terabit connections to/from the system
- **Single-tier All-Flash Lustre based HPC file system**
  - 6x Cori's bandwidth
- **Delivery in late-2020**



How do we measure and quantify the performance of a heterogeneous system?

HPC performance metrics

- **Why do performance metrics matter?**
  - **HPC centers need to specify performance metrics during procurements**
  - **Track evolution of performance**

Kernel Benchmarks	Application Benchmarks
Simple: Well defined & understood	Complex: Direct workload relevance
HPL (Linpack) Top500 = FLOPs Green500 = FLOPs/Watt HPCG, HPCC, NAS	DOE/APEX - Sustained System Improvement (SSI) DOE/CORAL - Scalable Science and Throughput Benchmarks NNSA's Cielo - Capability Improvement (CI) NERSC - Sustained System Performance (SSP) NCSA - Sustained Petascale Computing (SPP)

No prior metric explicitly addresses a heterogeneous system composed of a mixture of node types

GPU Readiness of the NERSC Workload

- **NERSC workload is extremely diverse:**
  - Over 600 applications in use
- **Some applications are used more heavily than others:**
  - In 2014, 13 codes accounted for 50% of CPU cycles
- **Variable GPU "readiness"**
  - Some applications are more amenable to GPUs than others
  - Some applications have already been written and tuned for GPUs.

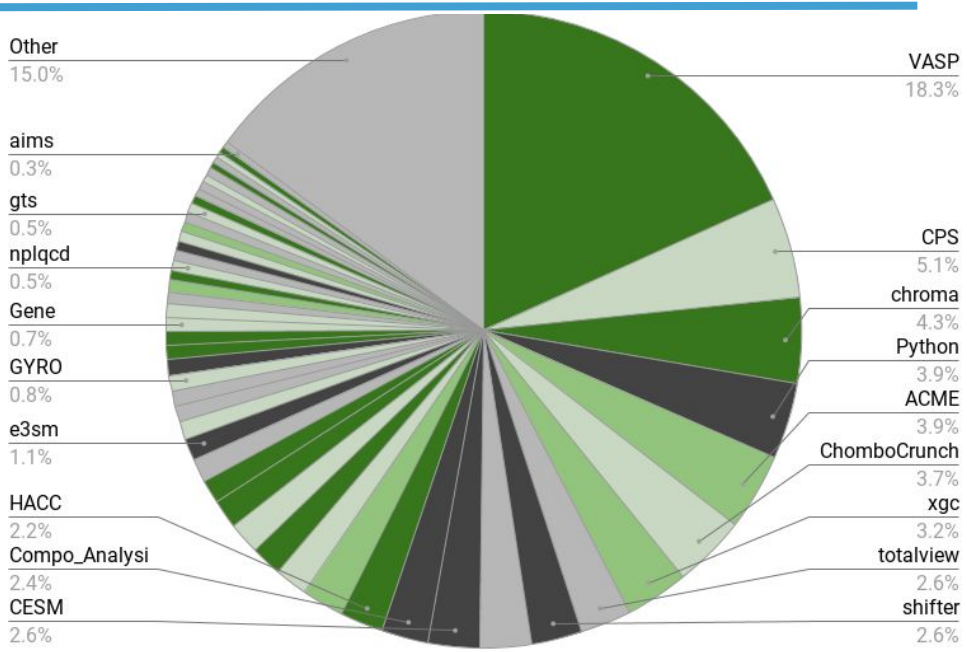


TABLE I
GPU READINESS CATEGORIZATION

GPU Status	Description	Fraction
Enabled	Most features are ported with good performance.	46%
Proxy	Kernels in related codes have been ported.	19%
Unlikely	A GPU port would require major effort.	11%
Unknown	GPU readiness cannot be assessed at this time.	24%

Benchmark Suite Construction

Use a suite representative of the NERSC Workload

- Include the workload's key algorithms
- However, the benchmark suite is limited in size to manage time and expertise
- Include applications that may be underused now but are likely to represent future workloads
- Codes with license restrictions not used for ease of use and reproducibility

Application	Description
Quantum Espresso	Materials code using DFT
MILC	QCD code using staggered quarks
StarLord	Compressible radiation hydrodynamics
DeepCAM	Weather/Community Atmospheric Model 5
GTC	Fusion PIC code
"CPU Only" (3 Total)	Representative of applications that cannot be ported to GPUs

Evolution of the SSI metric

- **Sustained System Performance: Capacity**
  - $SSP = \#Nodes \times \langle Perf\_per\_node \rangle$
- **Capability Improvement: Weak Scaling**
  - $CI = \langle Job\_Size \times Speedup \rangle$
- **Sustained System Improvement: Weak Scaling & Capacity**
  - **Strong Scaling:** $Speedup > 1.0$
  - $SSI = \langle \#Nodes \times Job\_Size \times Perf\_per\_node \rangle / \langle ref \rangle$

Adding heterogeneity

SSP averages the total system throughput of its benchmarks.

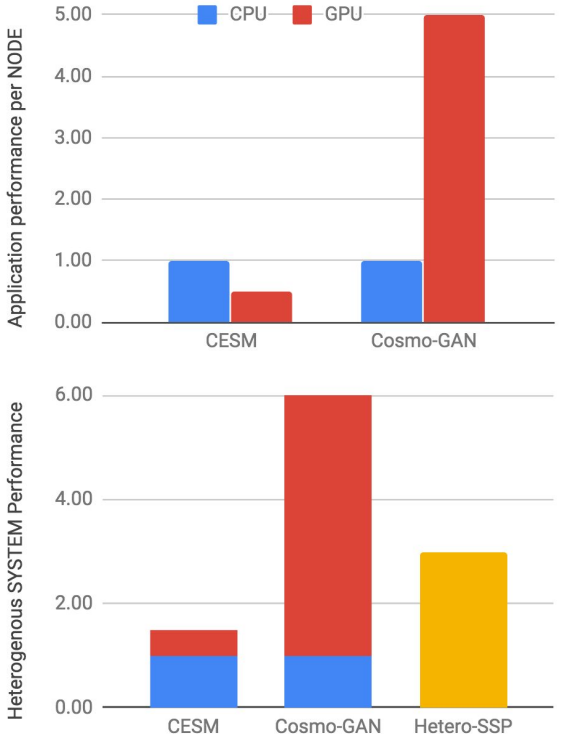
$$SSP = \#nodes \times \langle\, Perf\_per\_node \,\rangle$$

Hetero-SSP averages the total system throughput of its benchmarks.

Total system throughput of a benchmark is the sum over partitions.

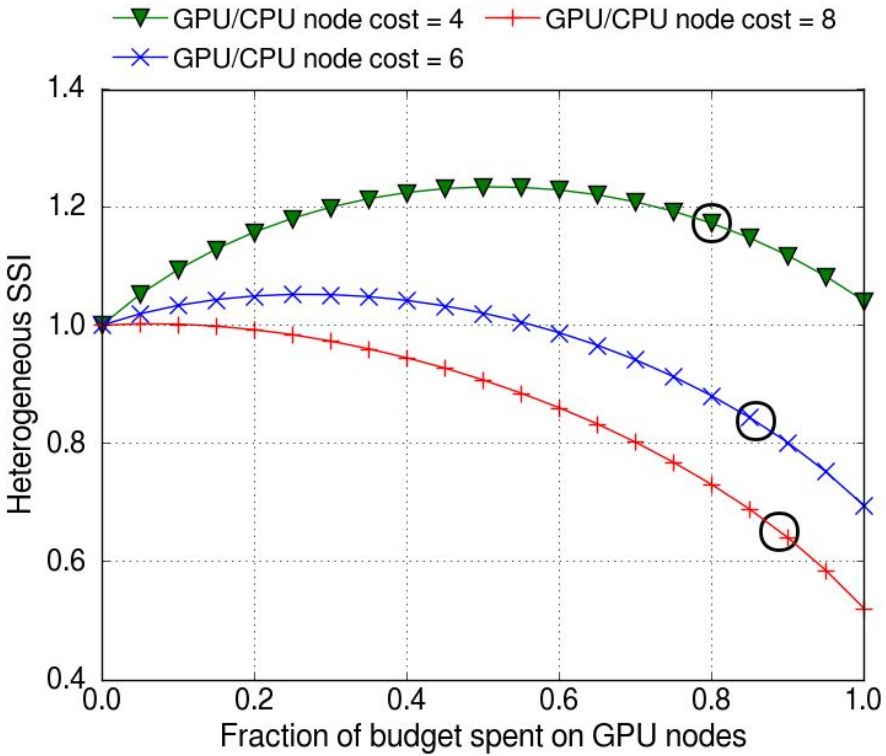
Sum before averaging!

$$hetero\text{-}SSP = \langle \sum_{\text{partitions}} \#Nodes \times Perf\_per\_node \rangle$$



Heterogeneous system design: Price sensitivity

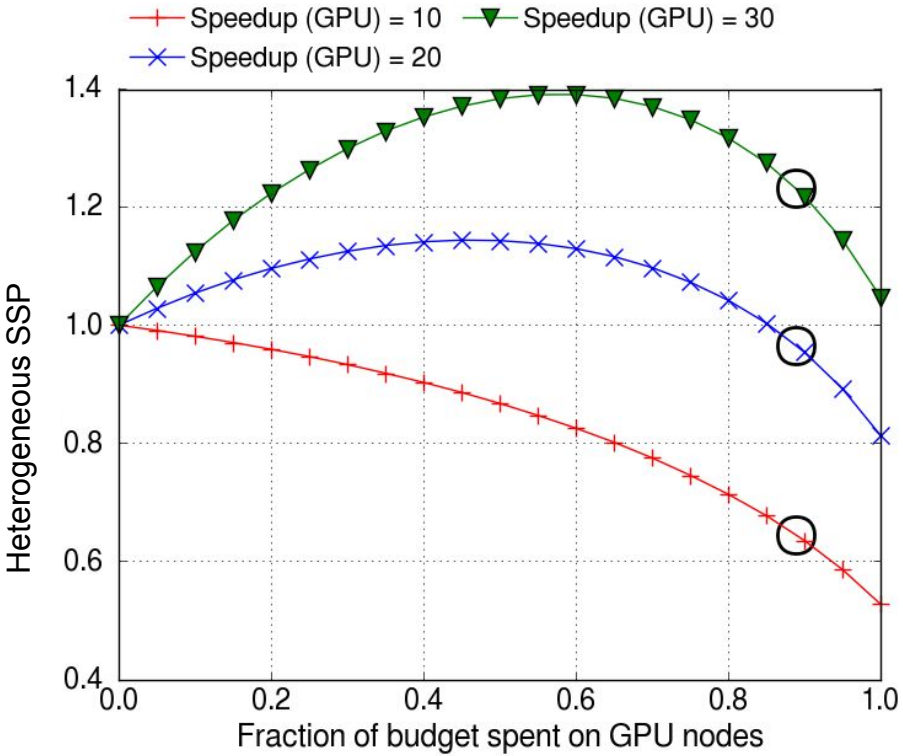
- Explore an isocost design space
  - Examine various GPU/CPU node cost differentials
  - Assume a GPU partition SSP advantage = 10x
  - Vary the budget allocated to GPUs
- Cost ratio = 4:1
  - *SSP improves 1.23x when 52% of the budget is used for GPUs*
- Cost ratio = 6:1
  - *Slight justification to use up to 50% of budget on GPUs*
- Cost ratio = 8:1
  - *No justification for GPUs*



Circles represent a system with 50% CPU nodes + 50% GPU nodes

Heterogeneous system design: Performance sensitivity

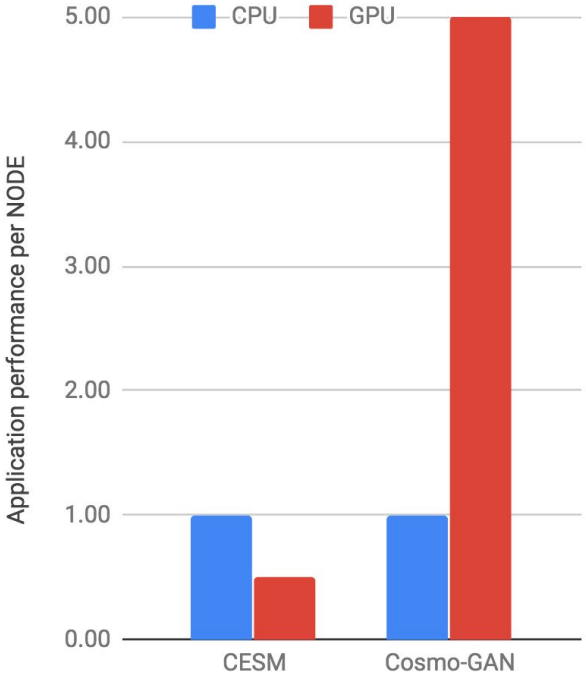
- Explore an isocost design space
  - Examine various GPU / CPU *performance* gains
  - Assume 8:1 GPU/CPU node cost differential.
  - Vary the budget allocated to GPUs
- Avg SSP of GPU node = 30
  - ***SSI improves 1.4x if 50%-60% of budget is used for GPUs***
- Avg SSP of GPU node = 20
  - ***SSI improves 1.15x if 40%-50% of budget is used for GPUs***
- Avg SSP of GPU node = 10
  - ***No justification for GPUs***

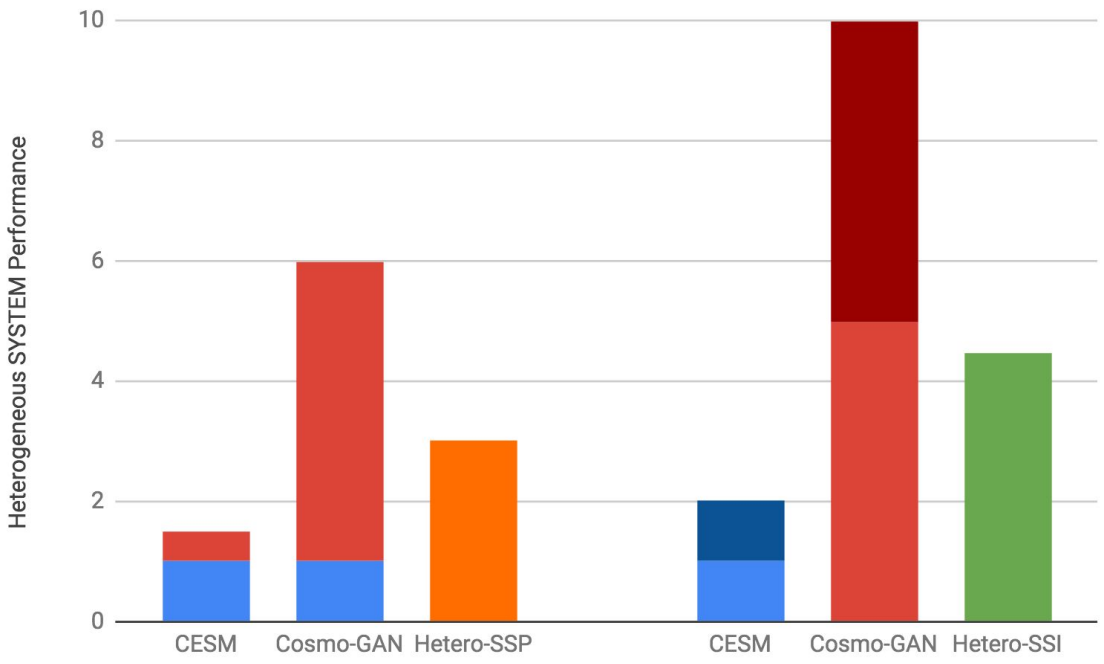


Circles represent a system with 50% CPU nodes + 50% GPU nodes

Specialization increases throughput on heterogeneous systems.

Specialization: run each code on the hardware that suits it best





Hetero-SSI incorporates specialization explicitly.

Partition fractions express specialization.

$f_{i,p}$: fraction of partition p devoted to benchmark i

$$hetero\text{-}SSP = \langle \sum_{p} \quad \#Nodes \times Perf\_per\_node \rangle$$

$$hetero\text{-}SSI = \langle \sum_{p} f_{i,p} \times \#Nodes \times Perf\_per\_node \rangle / \langle ref \rangle$$

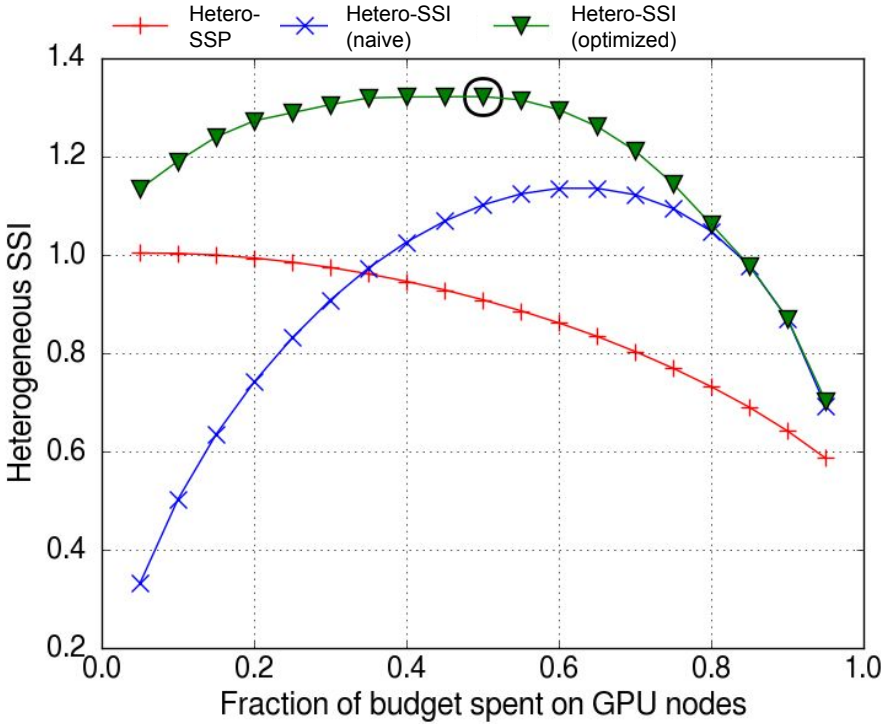
Optimize $\{f_{i,p}\}$ to maximize hetero-SSI;

no oversubscription: $\Sigma_p f_{i,p} \leq 1$

$$hetero\text{-}SSI \geq hetero\text{-}SSP / hetero\text{-}SSP^{ref}$$

Hetero-SSI measures the benefits of specialization

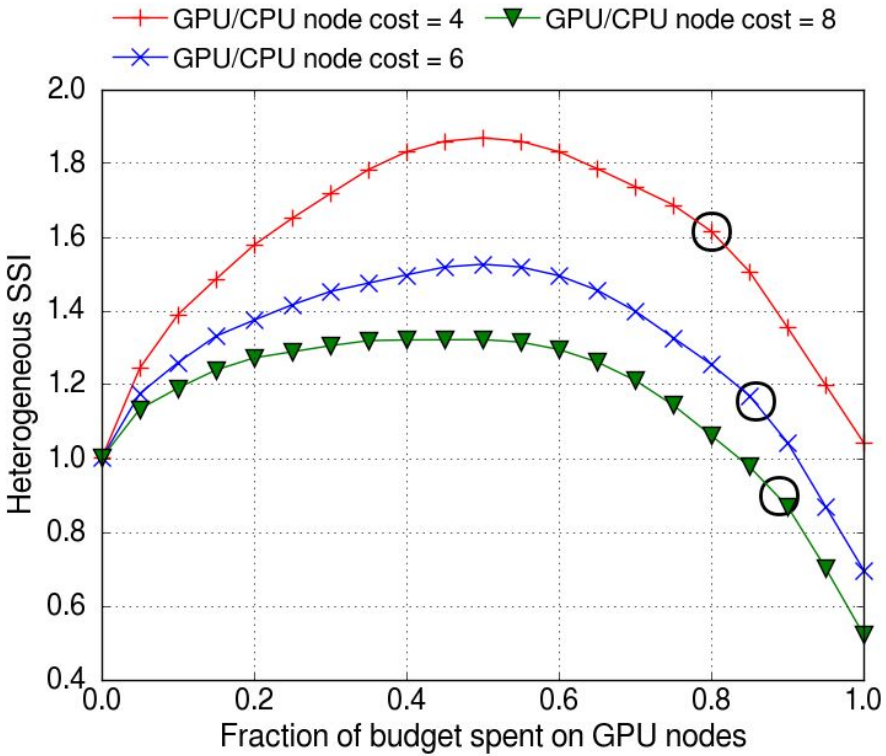
- Explore an isocost design space
  - Assume 8:1 GPU / CPU node cost differential
  - Vary the GPU node budget
- Hetero-SSP: no specialization
  - *No justification for GPUs*
- Hetero-SSI (naive):
  - GPU apps share GPU nodes
  - CPU apps share CPU nodes
  - 15% SSI gain when GPU budget = 70%
- Hetero-SSI (optimized):
  - All apps run on either/both partitions
    - *Best* GPU apps share GPU nodes
    - Other apps share CPU nodes
  - 35% SSI gain when GPU budget = 50%
  - Specialization provides 50% SSI increase



Circle denotes the system with maximum SSI

Specialized heterogeneous system design: Price sensitivity

- Isocost design space with various GPU/CPU node cost differentials
- Cost ratio = 4:1
  - *Hetero achieves 90% SSI gain when GPU budget = 52%*
- *In contrast to a naive approach, specialized SSI gains can be realized even with a high cost differential*



Circles represent a system with 50% CPU nodes + 50% GPU nodes

Conclusion

- Specialization and heterogeneity are already present in today's HPC systems.
  - Hardware specialization is a response to the slowing of Moore's
  - Heterogeneity results from using specialization across diverse workloads.
- HPC performance metrics must adapt to account for the realities of heterogeneity.
  - We have introduced heterogeneous extensions to the SSP and SSI metrics
- Heterogeneity is not inherently beneficial; real gains come from running each application on the most appropriate hardware (specialization)
  - Hetero-SSI incorporates specialization explicitly, which captures 50% more performance in some scenarios.
- Carefully designed performance metrics like hetero-SSI enable quantitative optimization of heterogeneous HPC systems.

Thank You

Evolution of the SSI metric

Sustained System Performance (SSP)

$$SSP = N\langle \frac{F_i / t_i}{n_i} \rangle$$

Capability Improvement (CI)

$$CI = \langle c_i \frac{t_i^{ref}}{t_i} \rangle \sim \langle \frac{F_i / t_i}{F_i^{ref} / t_i^{ref}} \rangle$$

Sustained System Improvement (SSI)

$$SSI = \langle N \frac{c_i / t_i}{n_i} \rangle / \langle N^{ref} \frac{1 / t_i^{ref}}{n_i^{ref}} \rangle$$

- **Runtime improvement:** $t_i \leq t_i^{ref}$
- **Weighted mean**

N: total nodes
i: benchmark enumerator
n_i: nodes used
F_i: flop count
t_i: walltime
c_i: capability factor ~ F_i / F_i^{ref}

Geometric Mean

- Preferred, not prescribed
- FOM independence
- Historically consistent with SSP
- $\langle x_i \rangle / \langle y_i \rangle = \langle x_i / y_i \rangle$

Adding heterogeneity

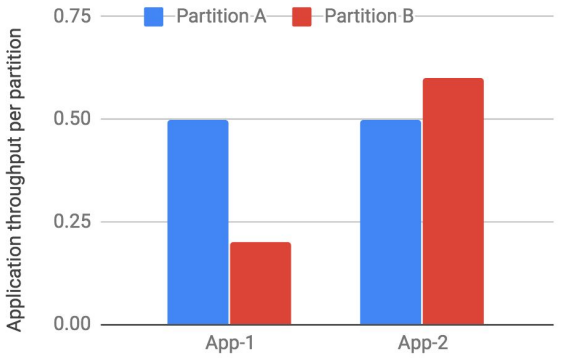
SSP averages the total system throughput of its benchmarks.

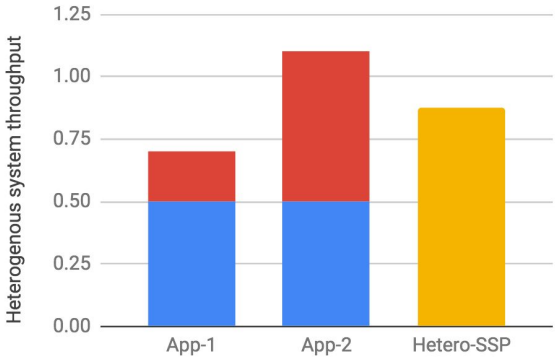
Hetero-SSP averages the total system throughput of its benchmarks.

Total system throughput of a benchmark is the sum over partitions.

$$SSP = N\left\langle \frac{F_i / t_i}{n_i} \right\rangle$$

$$hetero\text{-}SSP = \left\langle \Sigma_p N_p \frac{F_{i,p} / t_{i,p}}{n_{i,p}} \right\rangle$$





Hetero-SSI incorporates specialization explicitly.

Partition fractions express specialization.

$f_{i,p}$: fraction of partition p devoted to benchmark i

$$hetero\text{-}SSP = \langle \Sigma_p N_p \frac{F_{i,p} / t_{i,p}}{n_{i,p}} \rangle$$

$$hetero\text{-}SSI = \langle \Sigma_p f_{i,p} N_p \frac{c_i / t_{i,p}}{n_{i,p}} \rangle / \langle ref \rangle$$

Optimize $\{f_{i,p}\}$ to maximize hetero-SSI

No oversubscription: $\Sigma_p f_{i,p} \leq 1$

$hetero\text{-}SSI \geq hetero\text{-}SSP / hetero\text{-}SSP^{ref}$

Applicable to any number of partitions.

Simplifies to SSI with one partition.

Thank You