Economic Viability of Hardware Overprovisioning in Power-Constrained High Performance Computing

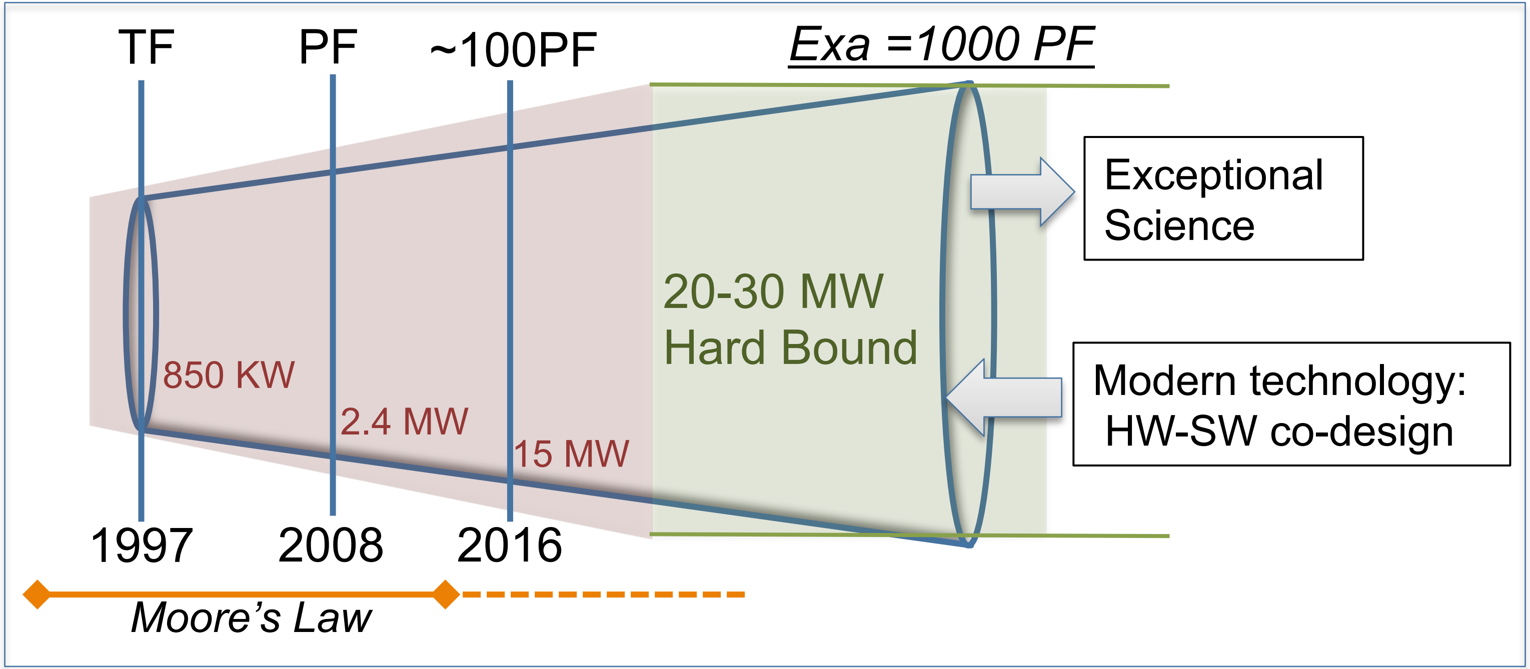
Energy Efficient Supercomputing, SC'16

November 14, 2016

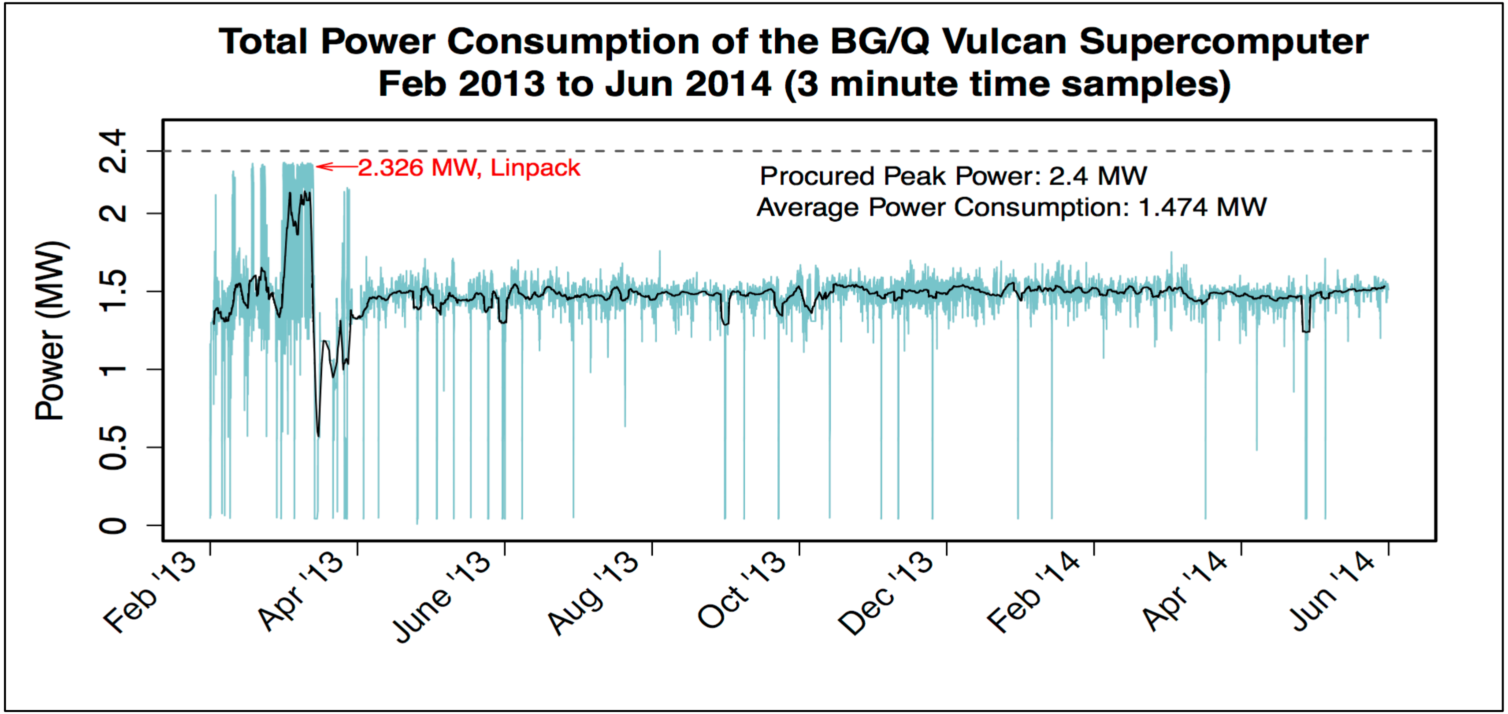
Tapasya Patki, David Lowenthal, Barry Rountree,
Martin Schulz, Bronis R. de Supinski

LLNL-PRES-708937
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

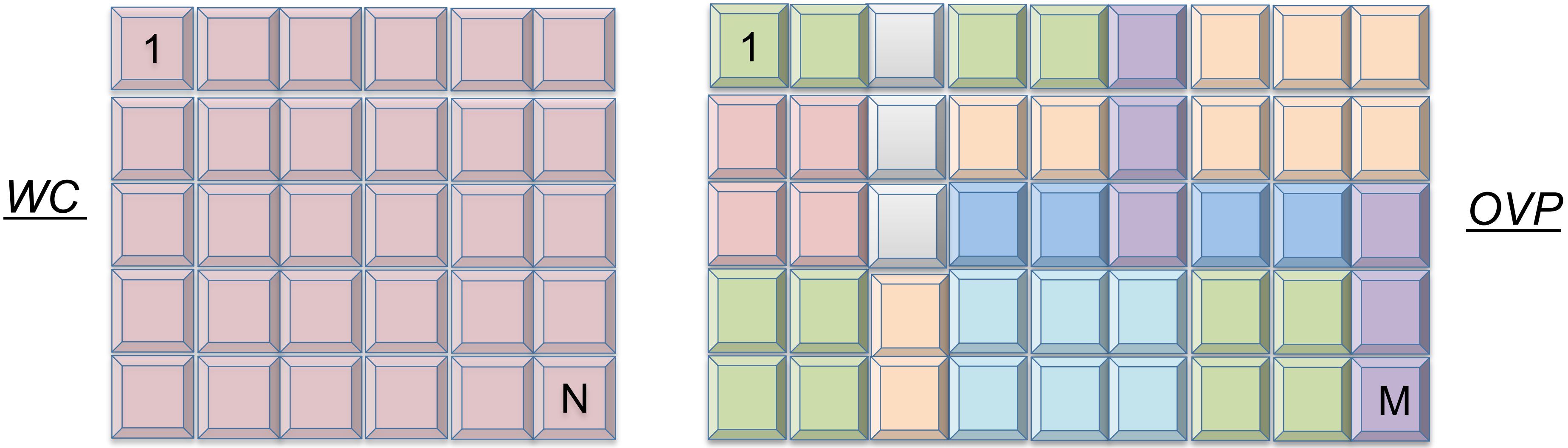
The *holy grail* of large-scale system design: achieve scientific progress with high throughput, high utilization, and low cost

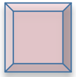


Power constraints make it very challenging to balance throughput, utilization, and cost



Design choices: conservative or liberal? Worst-case power provisioning and hardware overprovisioning




Node with peak power

$$M > N \text{ and } P_{wc} = P_{ovp}$$

More hardware under the same power budget

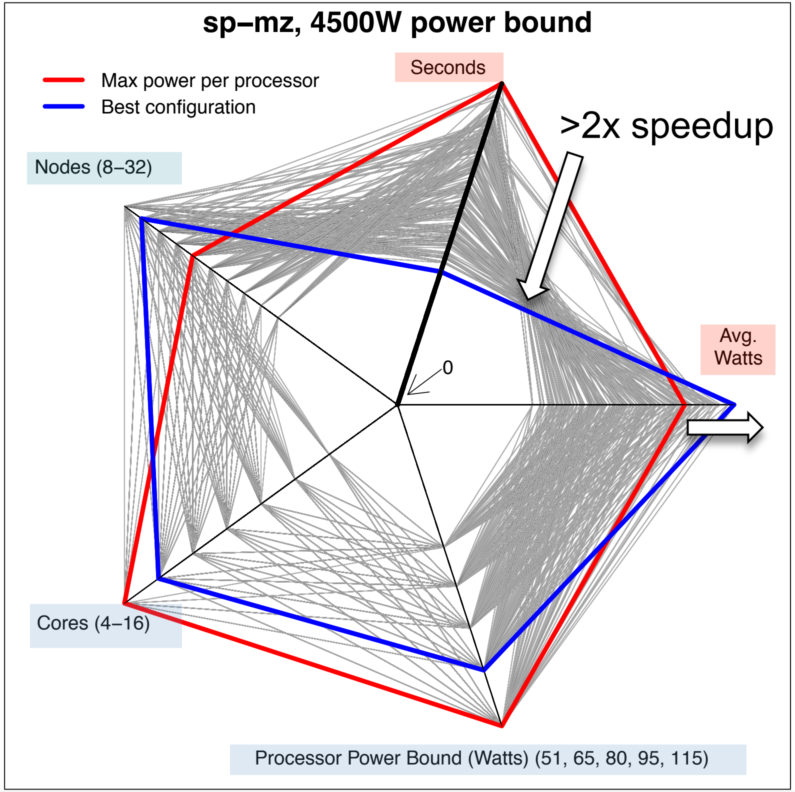
(managed with power capping)

The case for hardware overprovisioning: a simple example

- Intel Sandy Bridge cluster of 32 nodes
  - 2 sockets, 8 cores per socket, 2 DRAM modules
- NAS SP-MZ, CFD solver kernel, malleable
- 350 *configurations*
  - *Nodes*: 14 to 32, *cores per node* (scatter): 4 to 16
  - *Processor power caps* (W): 51, 65, 80, 95, 115
- Peak system power
  - 32 x 2 x (115_{cpu} + 25_{dram}), or ~9000 W

Assumed Budget: 4500 W

The case for hardware overprovisioning: we gain performance with intelligent power distribution, memory tuning and scaling



Considerations:

- Application's time to solution
- Energy = Power * Time
- Underutilizing power is bad for performance as well as energy

Bound: 4500W

Config: (n x c, p)	*Time (s)*	*Power* (W)*
WC: (24 x 16, 115)	7.16	3806
OVP: (30 x 14, 80)	2.94	+ 4459

**Actual Consumption of power across n nodes*

Overprovisioning improves throughput and utilization, but introduces operational safety and infrastructure cost concerns

- Dynamic power management techniques require application models, which may be error prone
- We can cap node and memory power, but we cannot guarantee network, I/O and other power through software
- How many *extra* nodes should we add before we lose the benefit and flip this into a problem of underutilized, idle nodes?
- More hardware implies added costs → focus of this paper

Given a fixed power budget and cost budget, can we build an overprovisioned system that results in a net performance benefit?

- Key intuition: server processors that are a generation older offer features similar to current generation at a much lower price

Feature	*Intel Ivy Bridge, 22nm*	*Intel Sandy Bridge, 32nm*
List Price (USD)	$3300	$1700
PassMark Performance*	17,812 (27% faster*)	13,895
Processors (Cores)	2 (24)	2 (16)
Clock Speed (Turbo)	2.7 (3.5) GHz	2.6 (3.3) GHz
TDP	130 W	115 W

**On a single node, all cores considered*

Let us build a high-end HPC system and a older-generation overprovisioned HPC system with fixed cost and power budgets

Input Parameters	*Description*
Power Bound*, P_{sys}	Power budget allocated to the *computational* components
Maximum Node Power, P_{n_max}	Maximum possible node power for the *high-end node* based on its overall *TDP*
Minimum Node Power, P_{n_min}	Minimum possible node power for the *older-generation node* based on its *idle* power
Cost Ratio*, r_c	Ratio of the *effective* per-node cost of the high-end node to that of the older-generation node (>1.0)
Performance, r_p	Percentage the high-end node is faster by on a single-node (>0%)

Note: *These can incorporate rack and interconnect information.*

A workload scalability model to predict multi-node performance at scale is also needed

- Predict performance of workload on the high-end system at a different node count based on multi-node data from older-generation system
- HPC systems are typically designed with a purpose and target workload
  - RFPs come with specific benchmarks and hardware options
- Orthogonal problem
  - Assume a linear model valid over a limited node range for simplicity

Let us now design our two HPC systems based on the power constraint P_{sys}, and the derived cost constraint, c_{wc}

- Determine maximum WC nodes based on power budget, derive *cost budget*

Represents OVP nodes →

$$n_{wc} = P_{sys}/P_{n\_max}$$
$$c_{wc} = n_{wc} \times r_c$$

- Determine maximum possible OVP nodes. Note that cost of older-generation node is 1 based on how we defined r_c

$$n_{lim} = P_{sys} / P_{n\_min}$$
$$n_{ovp} = min(n_{lim}, c_{wc})$$

Simple performance prediction based on the workload scalability linear model (slope, intercept)

- For the OVP system, performance on n_{ovp} nodes is:

$$t_{ovp} = m \times n_{ovp} + b$$

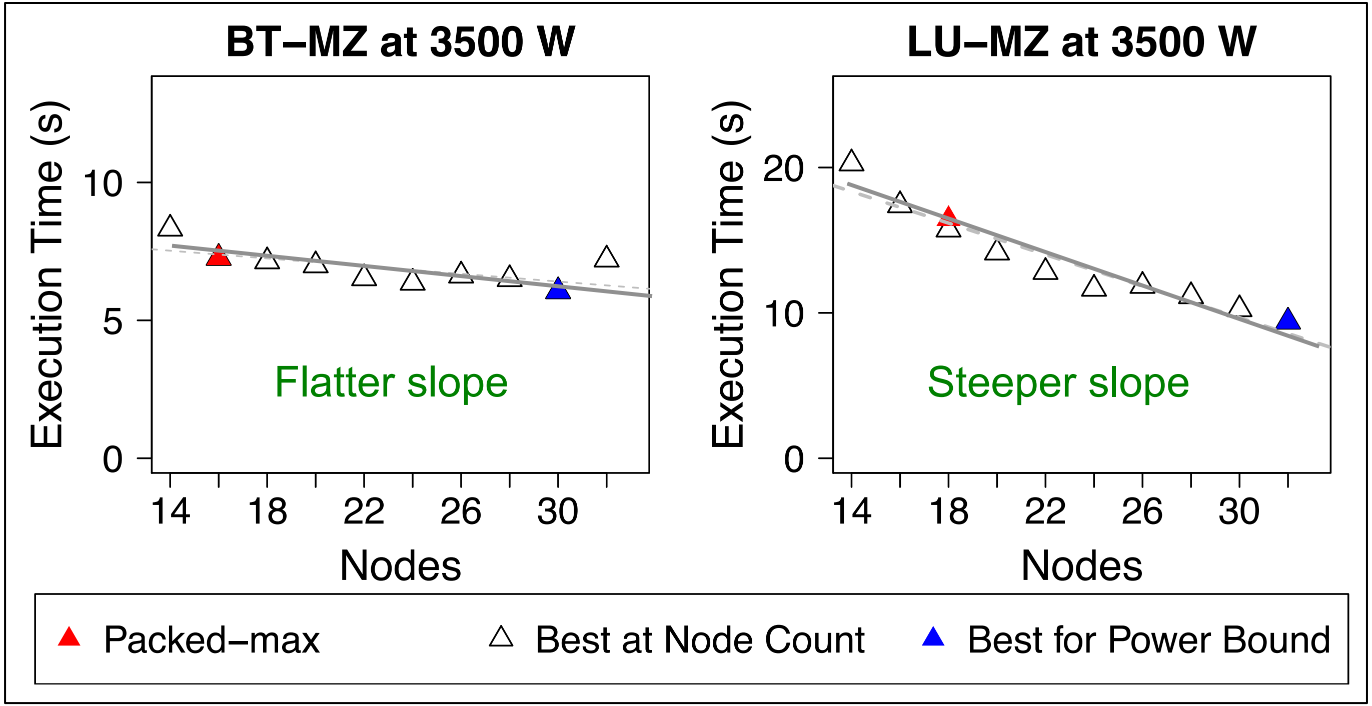
- For the WC system, performance on n_{wc} nodes is:

$$t_{wc} = (m \times n_{wc} + b)\ (1 - (r_p/100))$$

- For overprovisioning to be beneficial, speedup, s_{ovp}, should be greater than 1

$$s_{ovp} = t_{wc}/t_{ovp}$$

Two examples of workload scaling models with the best configuration selected at each node count



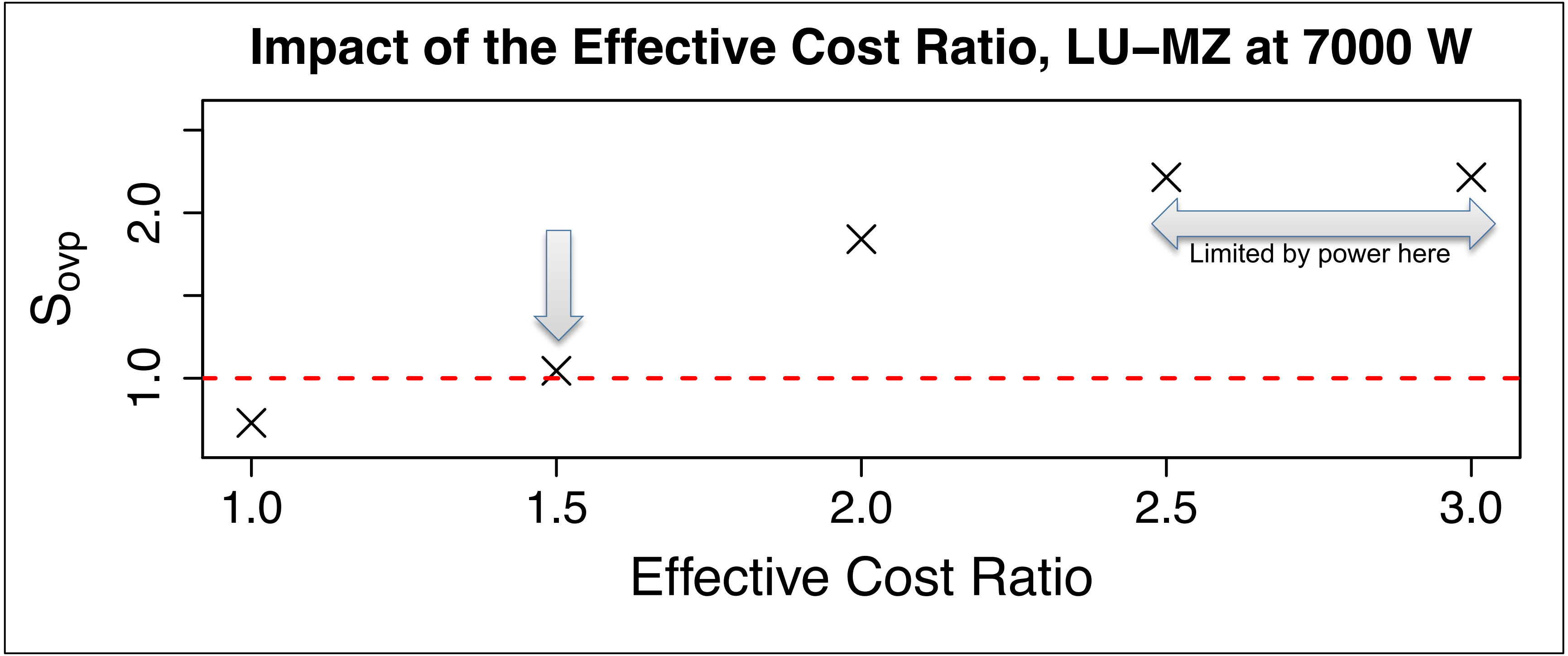
Evaluation Example: we benefit if s_{ovp} is greater than 1

Workload	N_{wc}	N_{ovp}	S_{ovp}
LU-MZ	18	30	1.22
BT-MZ	18	30	*0.83*

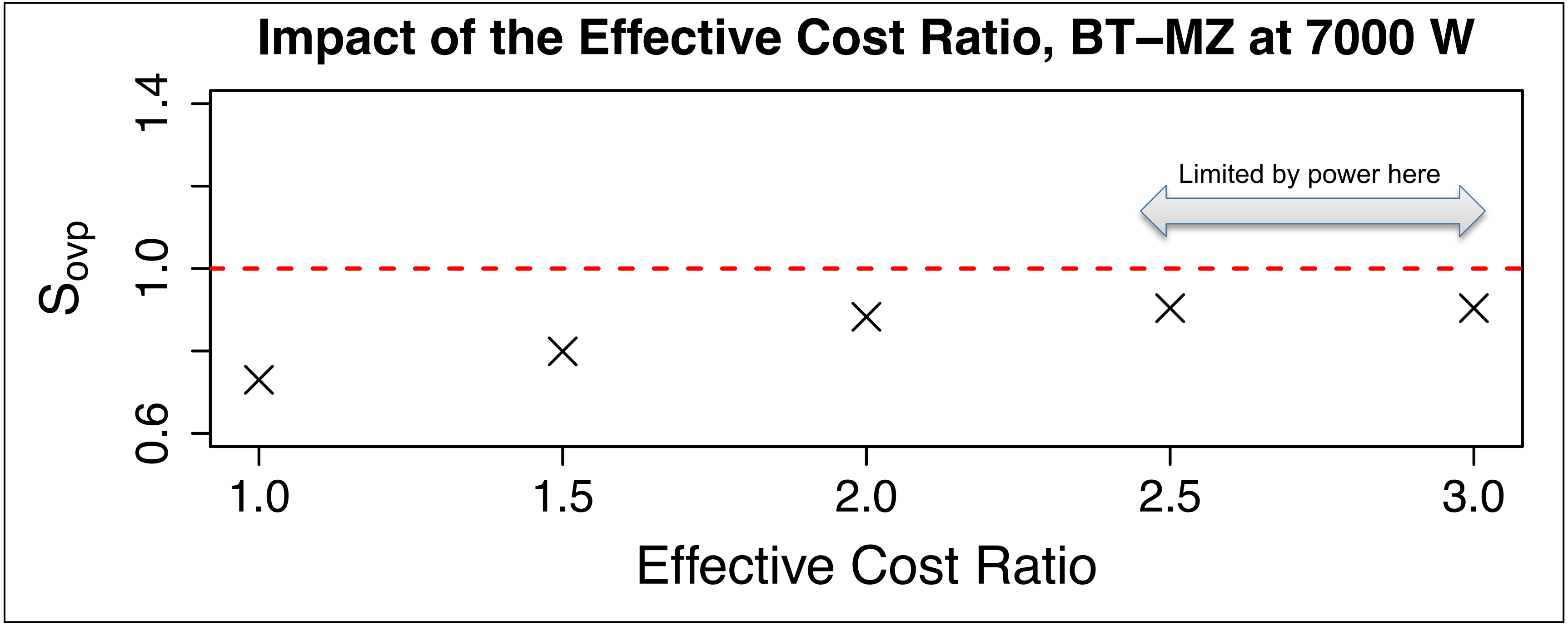
- LU-MZ represents workloads that scale well, BT-MZ otherwise

Input Parameters	*Values*
P_{sys}	7000 W
P_{n_max}	380 W
P_{n_min}	180
Cost Ratio, r_c	1.7
Performance, r_p	27%
LU-MZ model, *(m,b)*	(-0.542, 25.93)
BT-MZ model, *(m,b)*	(-0.069, 8.50)

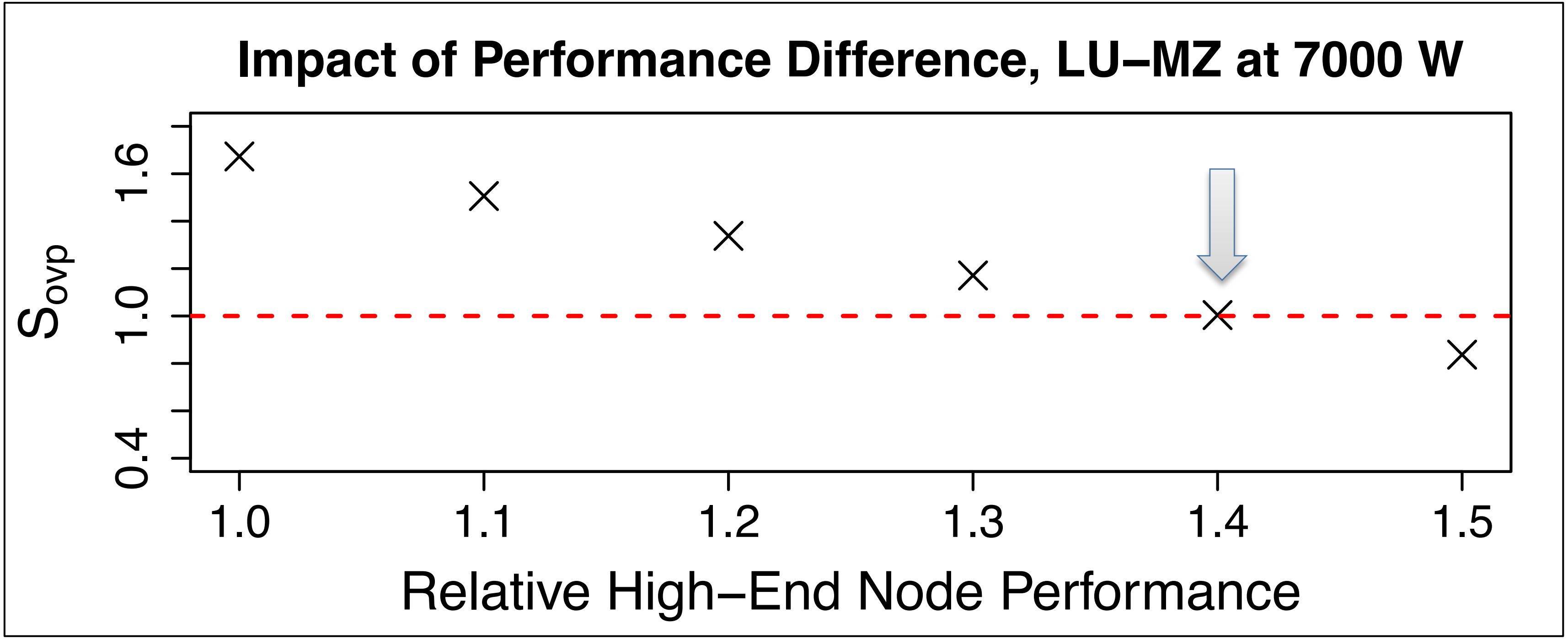
Significant benefit for workloads such as LU-MZ (Cost Ratio: better when the crossover is toward the left)



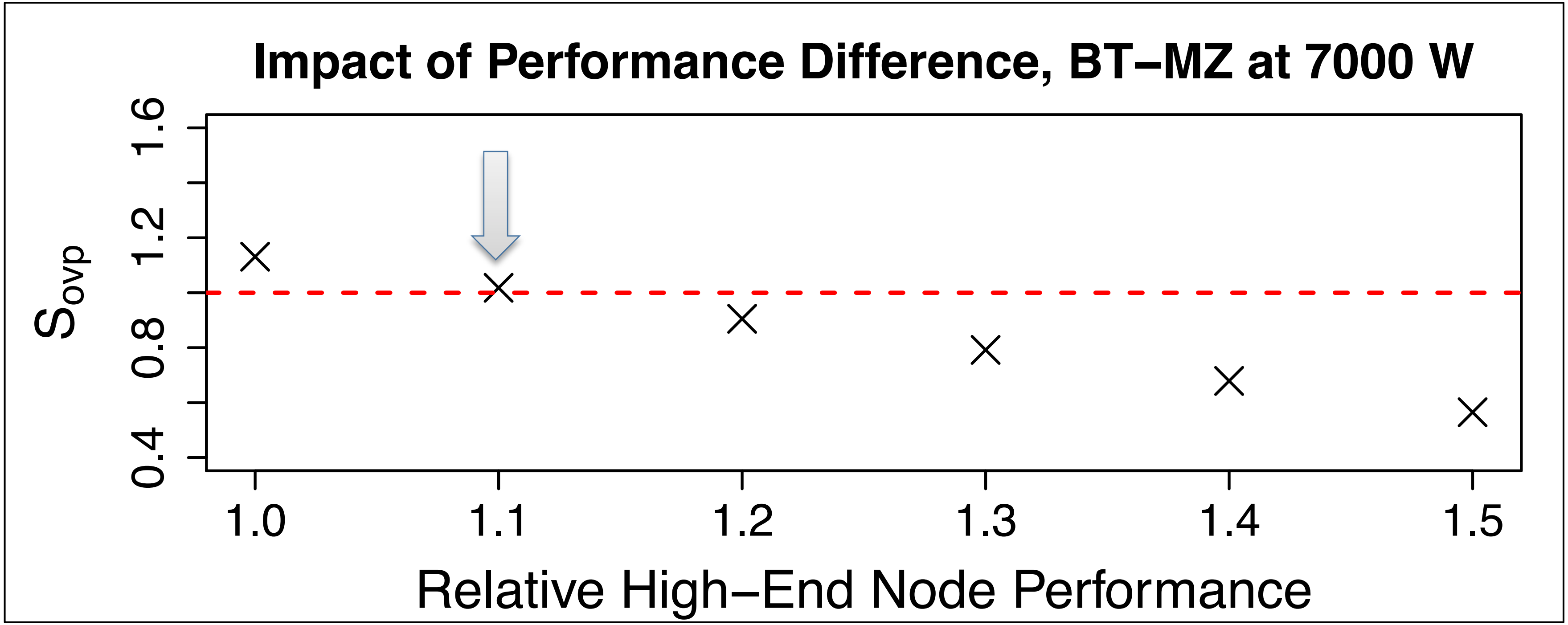
No win with workloads such as BT-MZ
(Cost ratio: better when the crossover is toward the left)



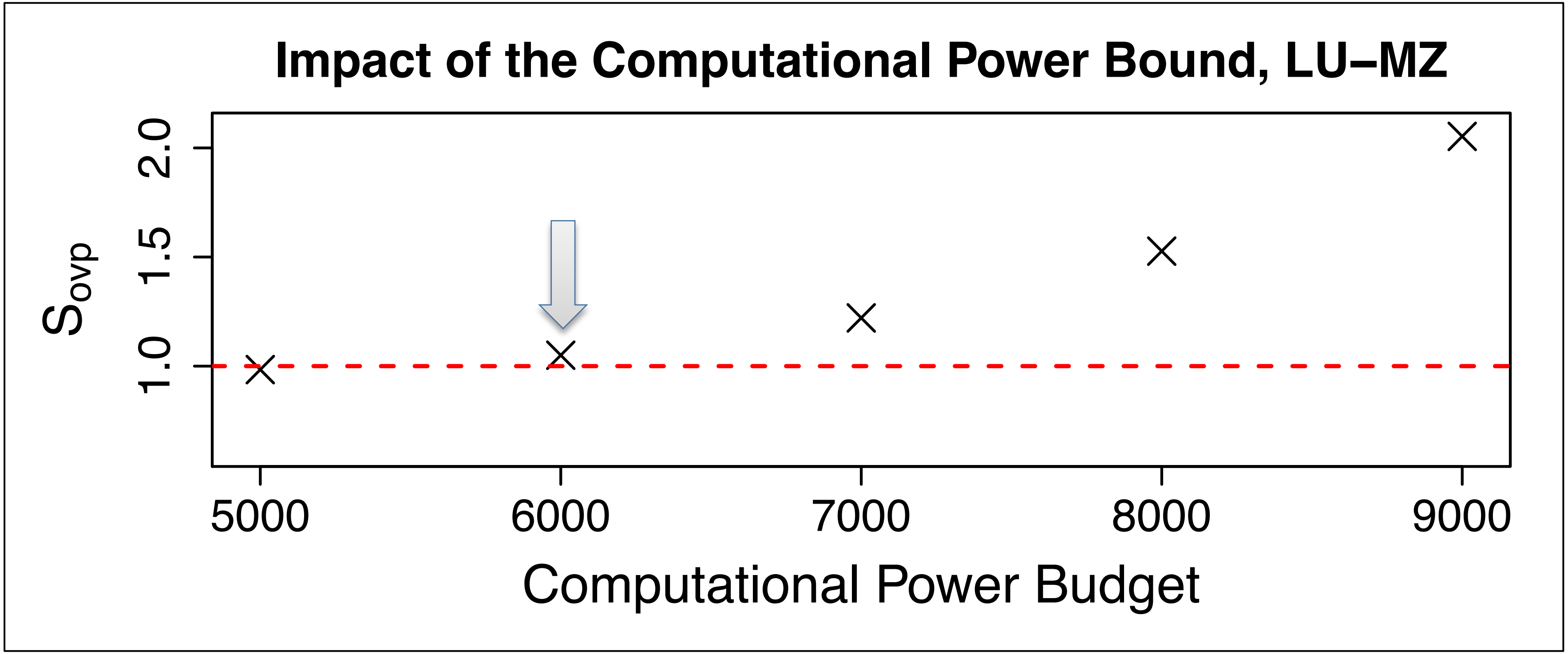
Significant benefit for workloads such as LU-MZ (Node performance: better when the crossover is toward the right)



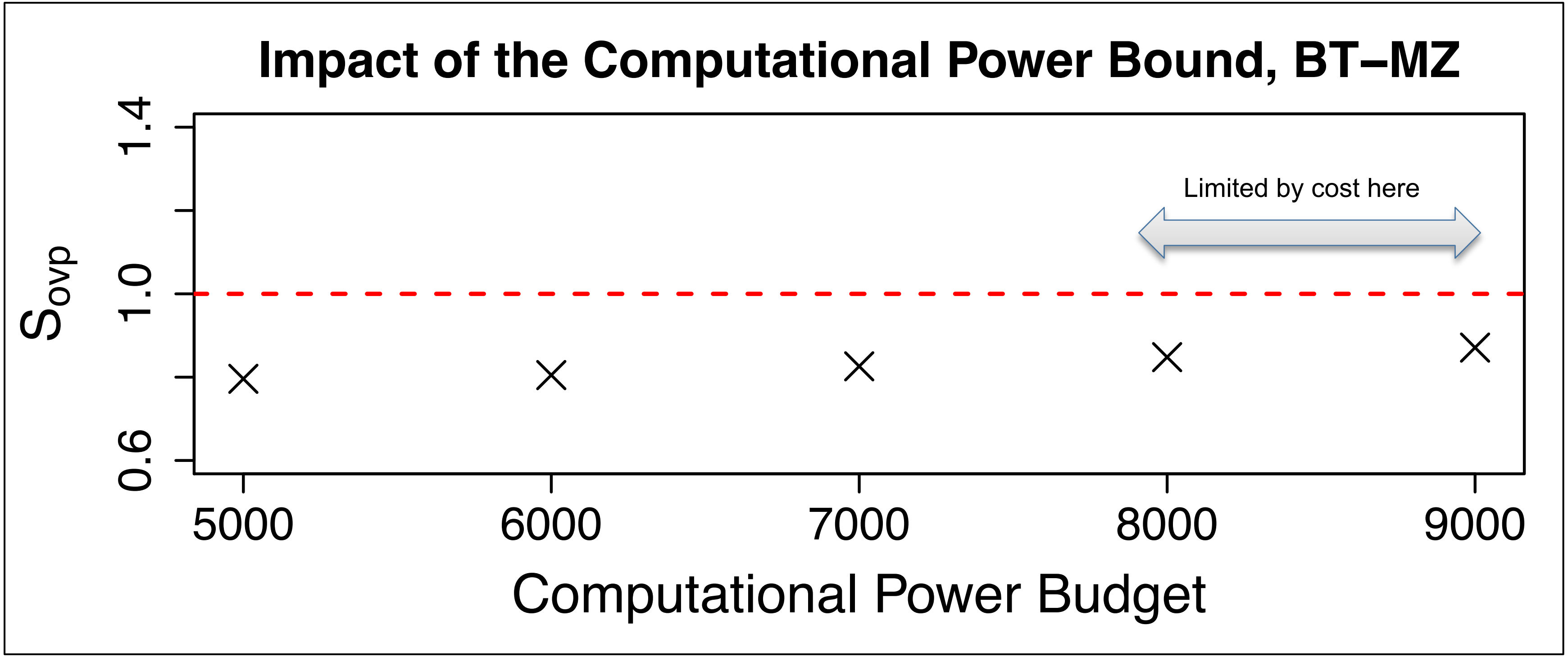
No win with workloads such as BT-MZ
(Node performance Better when the crossover is toward the right)



Significant benefit for workloads such as LU-MZ (Power budget: better when the crossover is toward the left)



No win with workloads such as BT-MZ (Power budget: better when the crossover is toward the left)



Summary

- Design choices: worst-case and hardware overprovisioning
  - Careful cost-benefit analysis is necessary for large-scale design
- An overprovisioned system can be built without additional cost using older-generation nodes with similar features
- Net benefit depends on several factors
  - Relative cost
  - Relative single-node performance
  - Expected workload characteristics
- More research is needed for throughput and utilization analysis

Lawrence Livermore National Laboratory