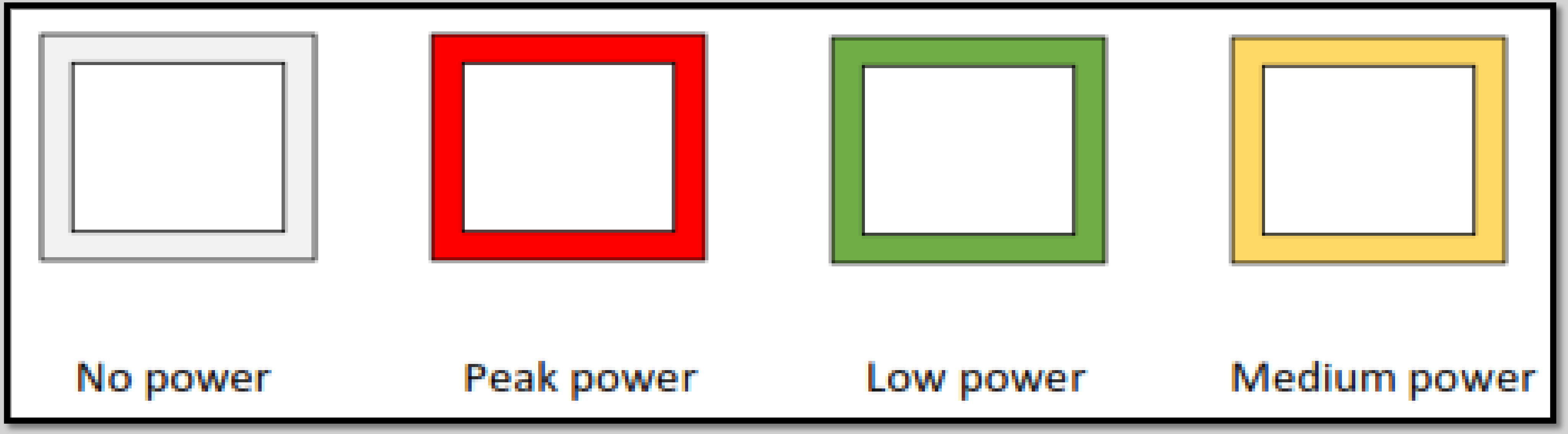
Exploring Hardware Overprovisioning in Power-Constrained, High Performance Computing

Tapasya Patki [1] David Lowenthal [1]
Barry Rountree [2] Martin Schulz [2] Bronis de Supinski [2]

[1] The University of Arizona
[2] Lawrence Livermore National Laboratory

Node Power



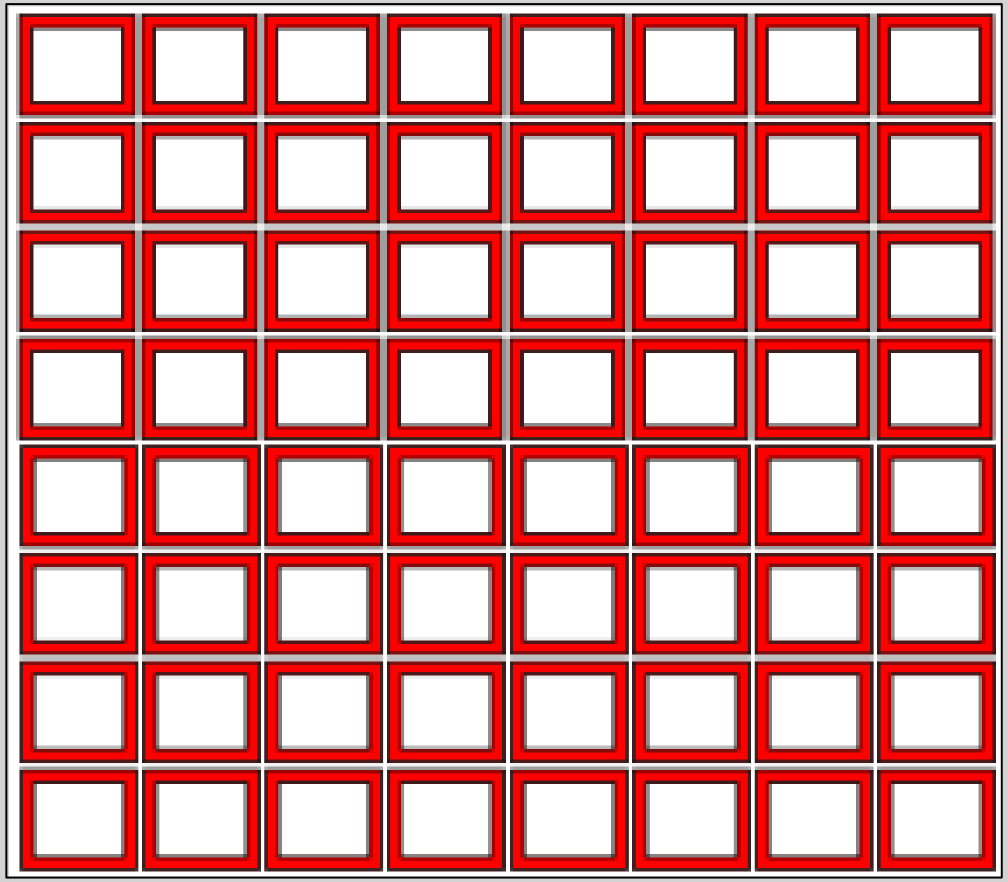
Node Power

— Package: processor die (cores + on-chip caches)

— DRAM

— Uncore: Off-chip caches, Quick Path Interconnect

Worst-case provisioning



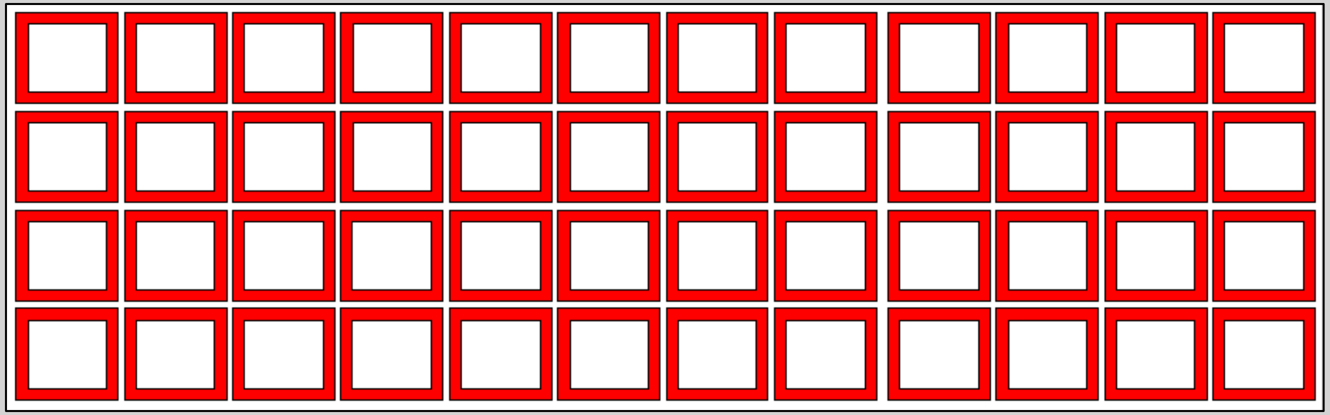
64 node cluster

Node power: Peak (300 W)

Why limit power?

- Tianhe-2: 31 petaflops today; 54 petaflops in 2015 at 17 MW
- Projected power needed for one exaflop: 0.5 GW
- Typical power plant generates 1 GW of power, provides for a million homes
- Cost: $1M per MW per year
- May have physical limitations on power that can be brought into a machine room

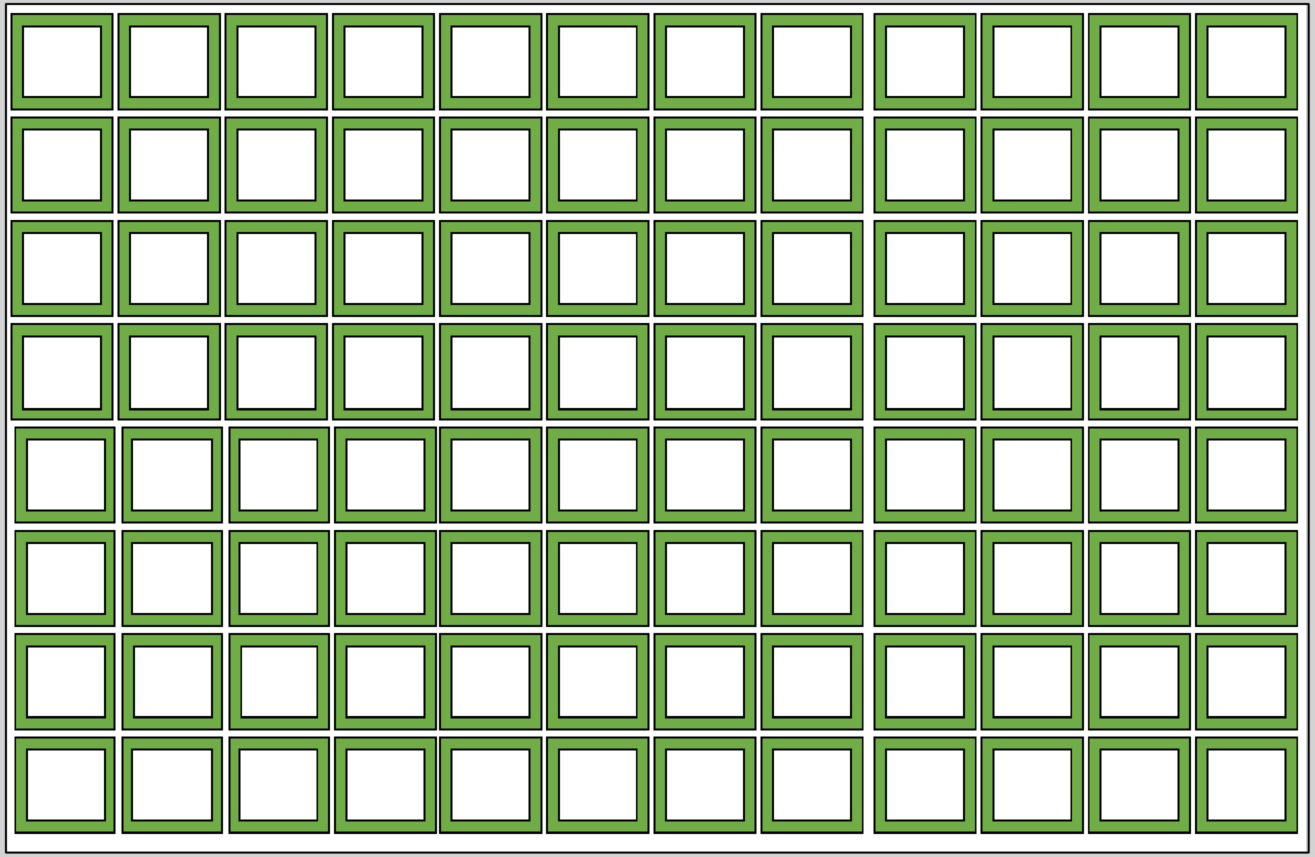
Enforcing a power bound



Node power: Peak (300 W)

Worst-case provisioned nodes: 48

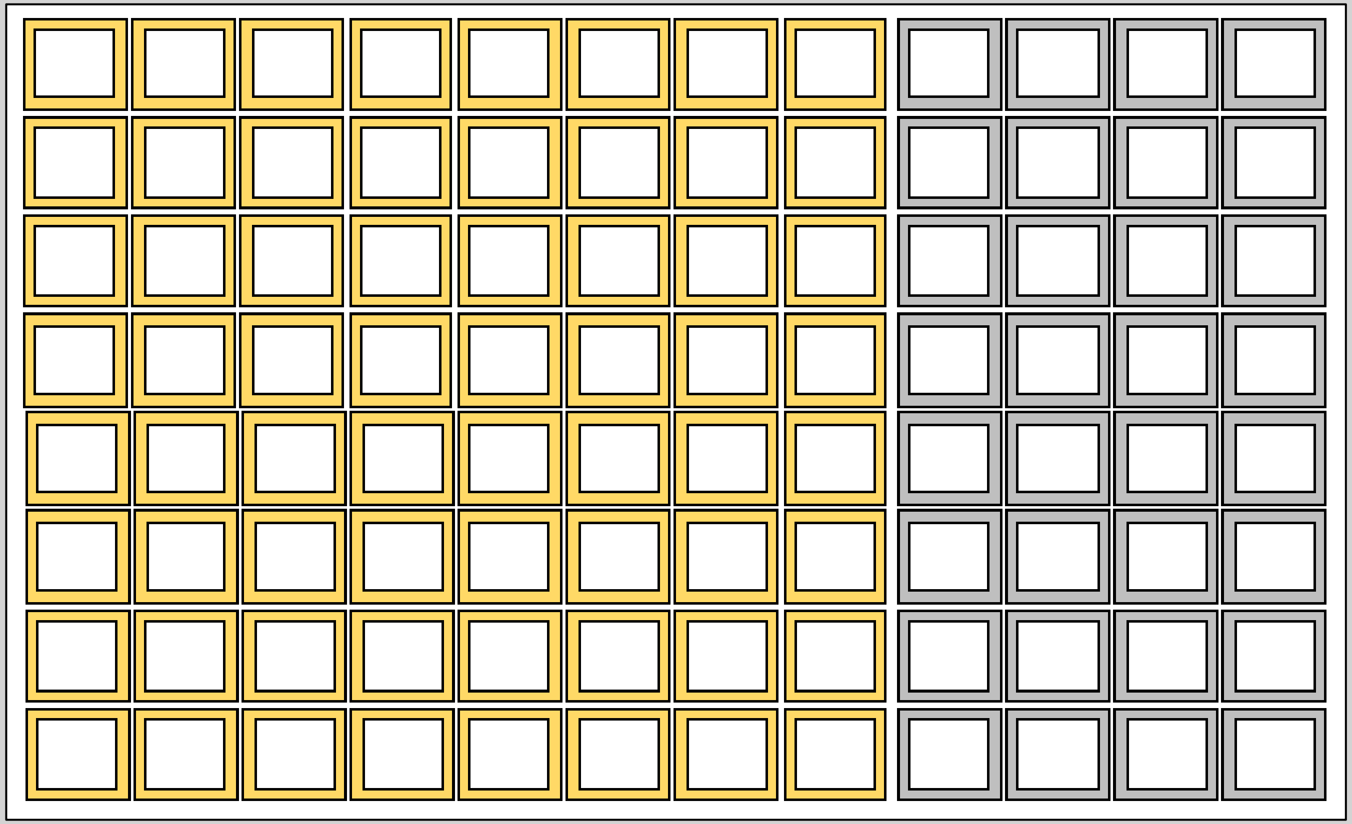
Hardware Overprovisioning



Node power: Low (150 W)

Nodes with overprovisioning: 96

Reconfiguring an Overprovisioned Cluster

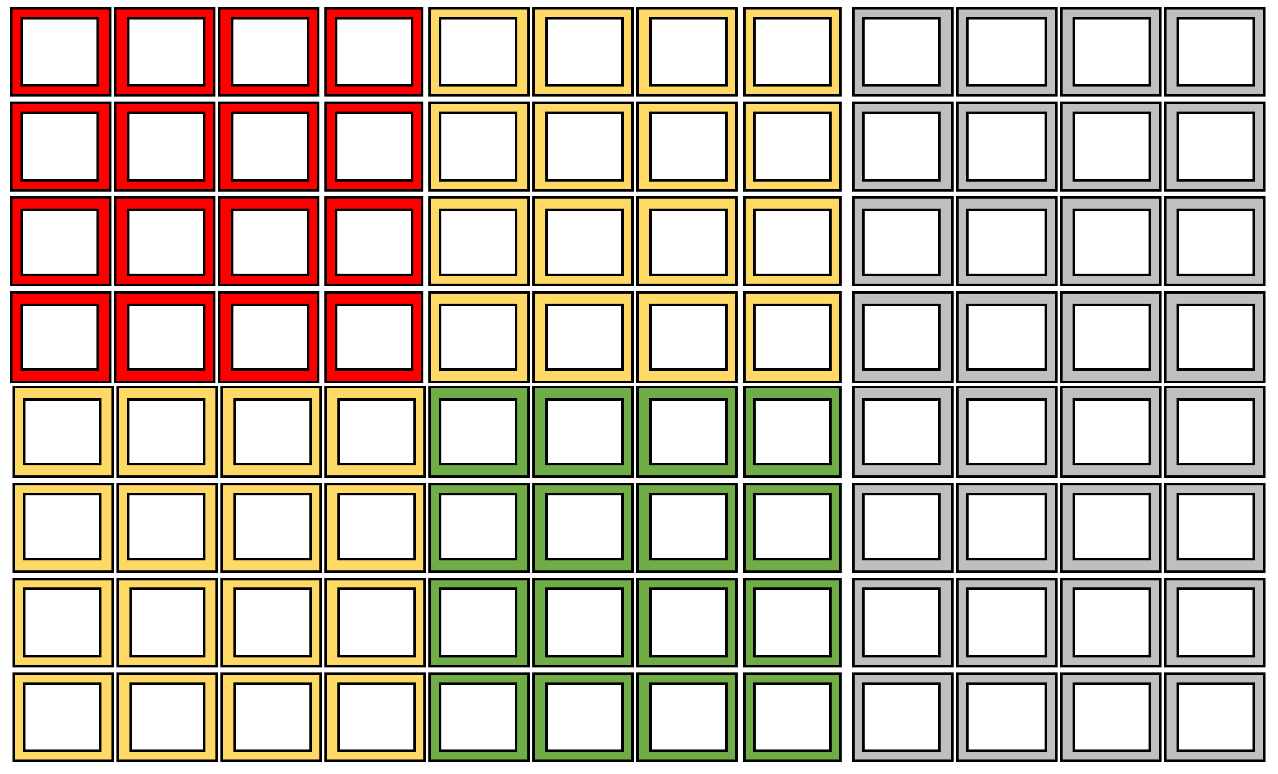


Node power: Med (225 W)

Nodes (reconfigured): 64

- Reconfigure based on application characteristics

Reconfiguring an Overprovisioned Cluster



- **Objective**: Study the impact of overprovisioning on application performance given a power-constrained cluster
- Found a performance improvement of over 62% as compared to worst-case provisioning

Outline

- Hardware Overprovisioning
- Experimental and Application Details
- Baseline Power Results (single-node)
- Multiple-node Results
- Summary

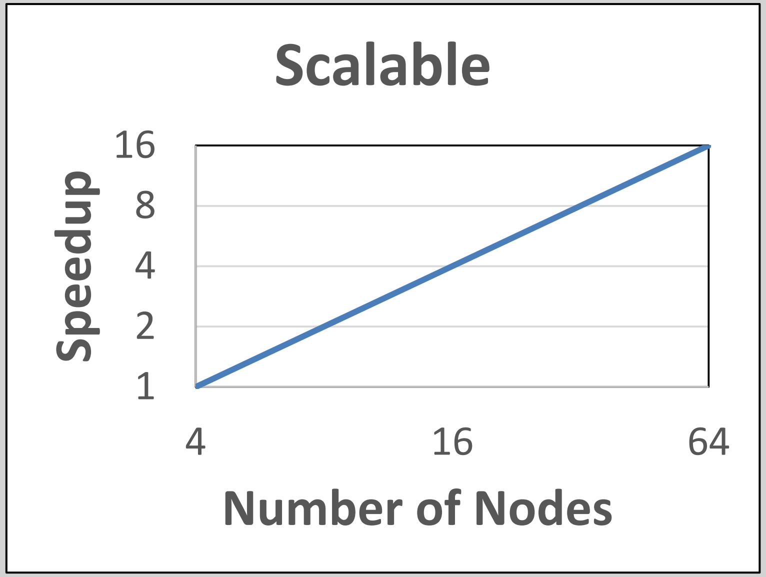
Power-constrained supercomputing

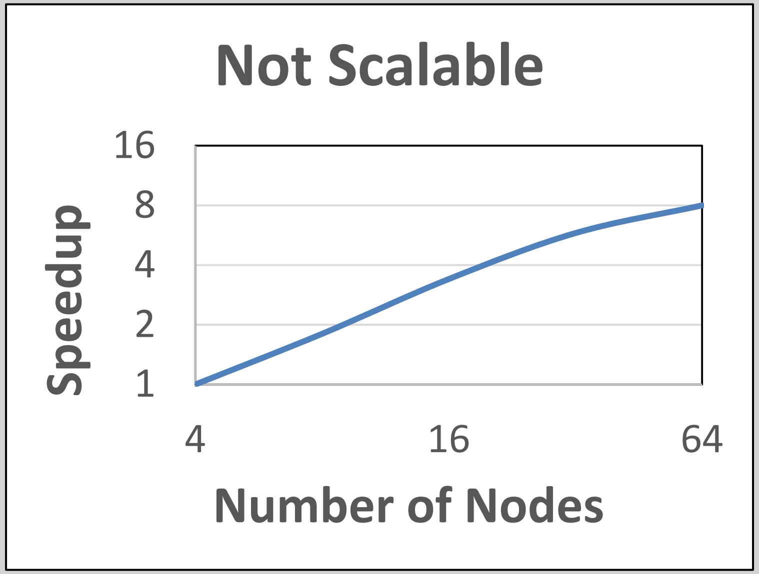
- DoE’s goal: one exaflop by 2020 with 20 MW
- Worst-case provisioning
  - Guarantee full power to a restricted number of nodes
- Overprovisioning
  - Limit power to a larger number of nodes

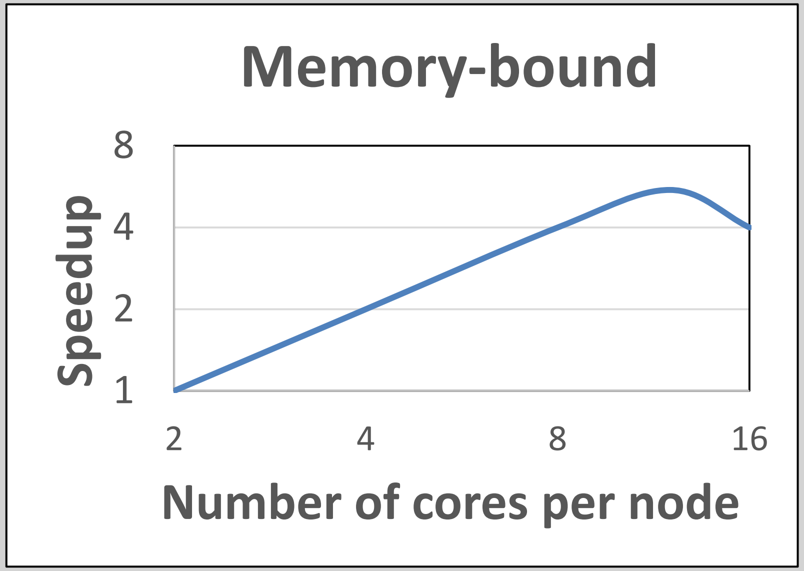
Why overprovision?

- Has been successful in the architecture community and in data centers
  - Intel TurboBoost, AMD TurboCORE
- Better performance under a power bound
  - One size doesn't fit all
- Can reconfigure based on application characteristics

Why reconfigure?







- More nodes at lower power per node
- Fewer nodes at higher power per node
- Fewer cores per node to avoid contention

Intel's Running Average Power Limit (RAPL)

- Sandy Bridge: on-board power measurement and capping
- Domains:
  - Package (PKG)
  - Power Plane 0 (PP0)
  - Power Plane 1 (PP1)
  - DRAM
- Models:
  - Client (062A): PKG, PP0 and PP1
  - Server (062D): PKG, PP0 and DRAM

Intel's Running Average Power Limit (RAPL)

Power capping

- Specify a power bound and a time window
- Hardware ensures that the average power over the time window does not exceed the specified bound
- Implemented using MSRs

`librapl`

- Safely access MSRs from user-space
- Gather power and CPU frequency data per process for MPI applications
  - Use MPI Profiling layer
- `librapl` is currently in use at UA, LLNL, Purdue, UIUC, NCSU, Virginia Tech, and Marquette U.
- https://github.com/tpatki/librapl

Experimental and Application Details

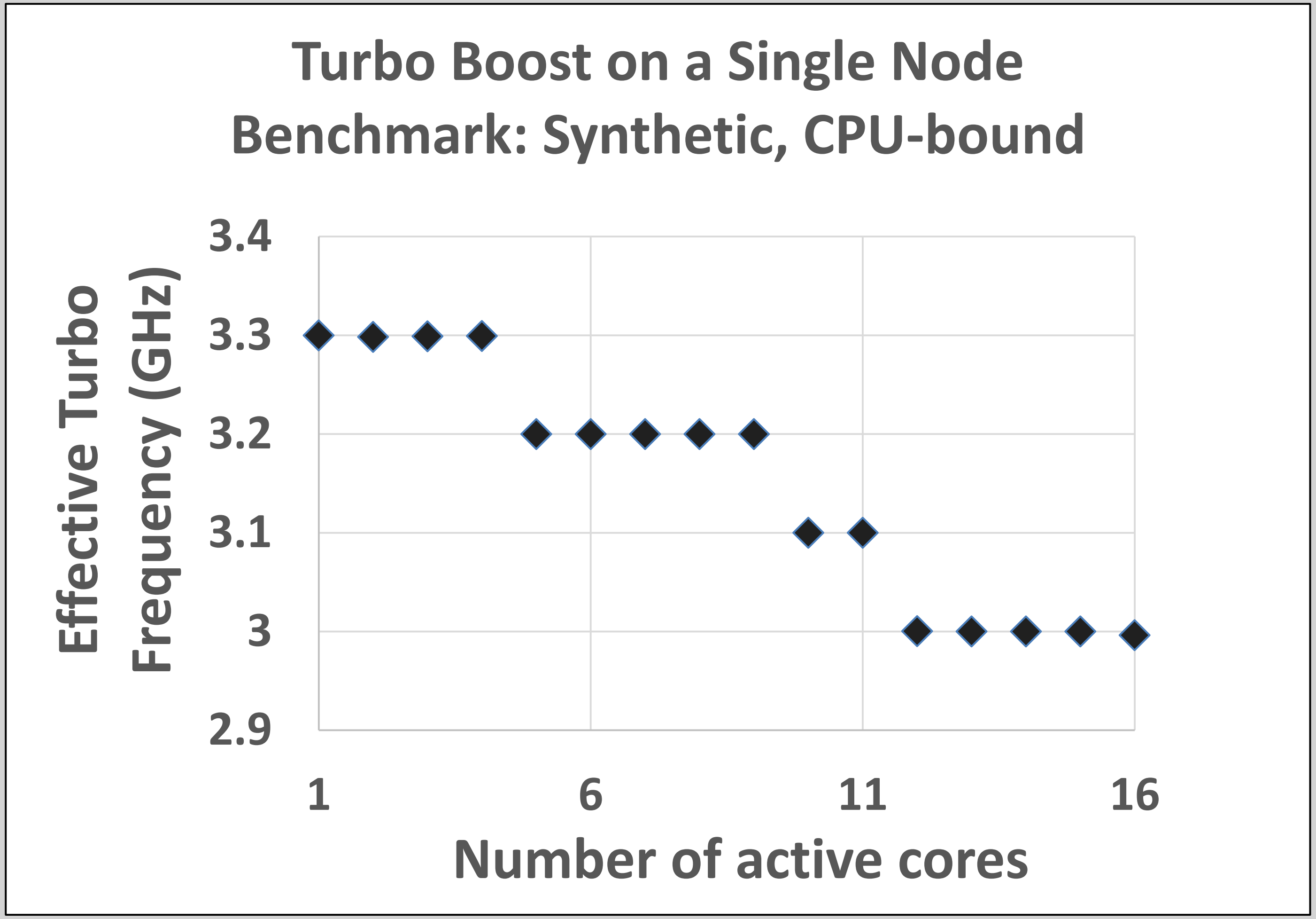
- Sandy Bridge Server cluster, 32 nodes
- 2 sockets, 8 cores per socket, 2.6 GHz / 3.3 GHz (Turbo)
- Emulated overprovisioning using RAPL PKG capping
- Hybrid: MPI + OpenMP
- Thermal limit: 115 W, Minimum power cap: 51 W (PKG)
- 8 to 32 nodes, 4 to 16 cores per node, increments of 2

Experimental and Application Details

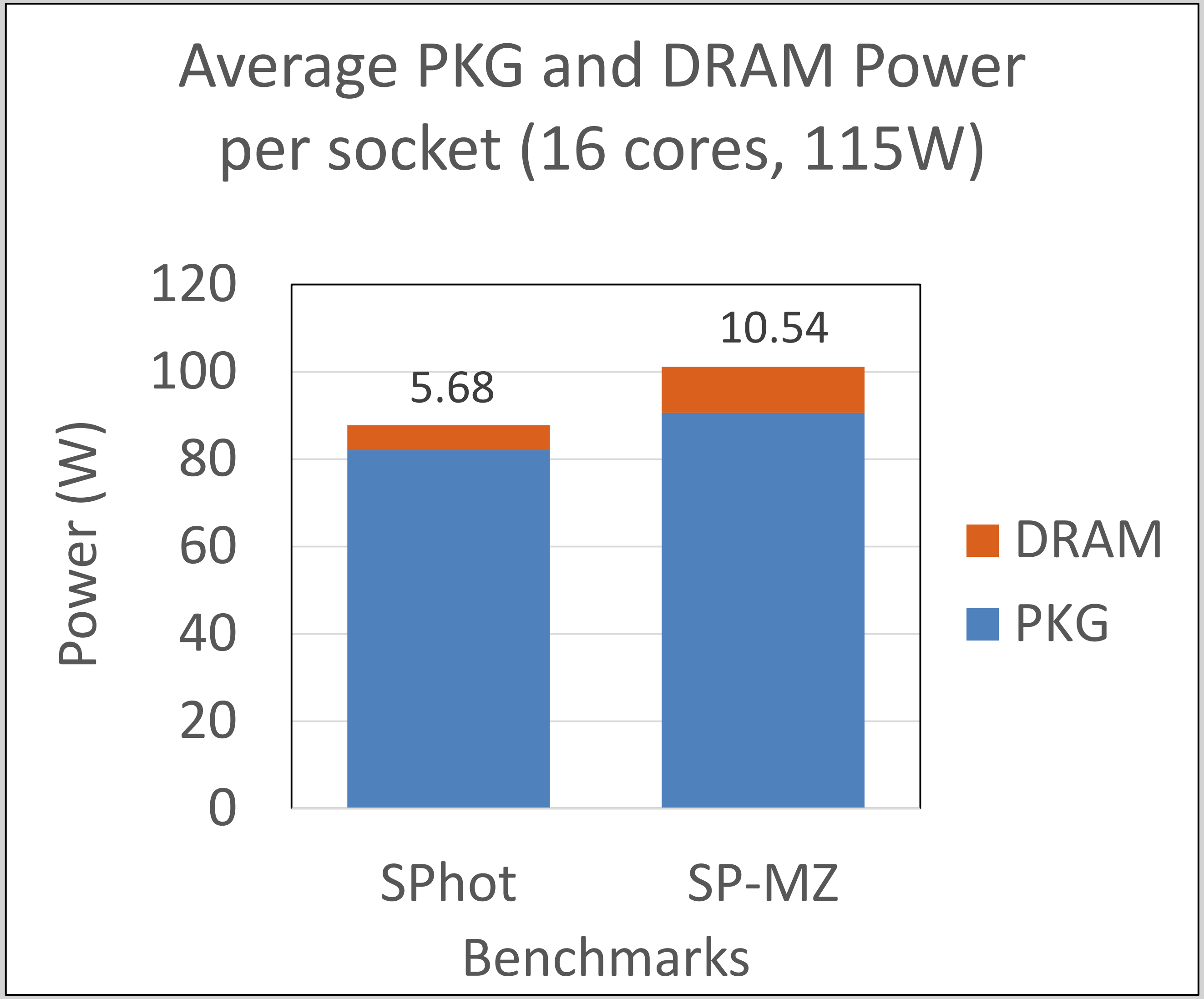
- HPC Applications
  - SPhot
  - NAS-MZ (BT-MZ, SP-MZ and LU-MZ)
- Synthetic Benchmarks
  - CPU-bound and memory-bound; scalable and not-scalable

Baseline Results: Intel Turbo Boost

- Turbo frequency depends on the number of active cores
- All nodes engage in Turbo mode in a similar manner
  - — uniform applications and consistent room temperature



Baseline Results: Power Profile (Turbo mode)

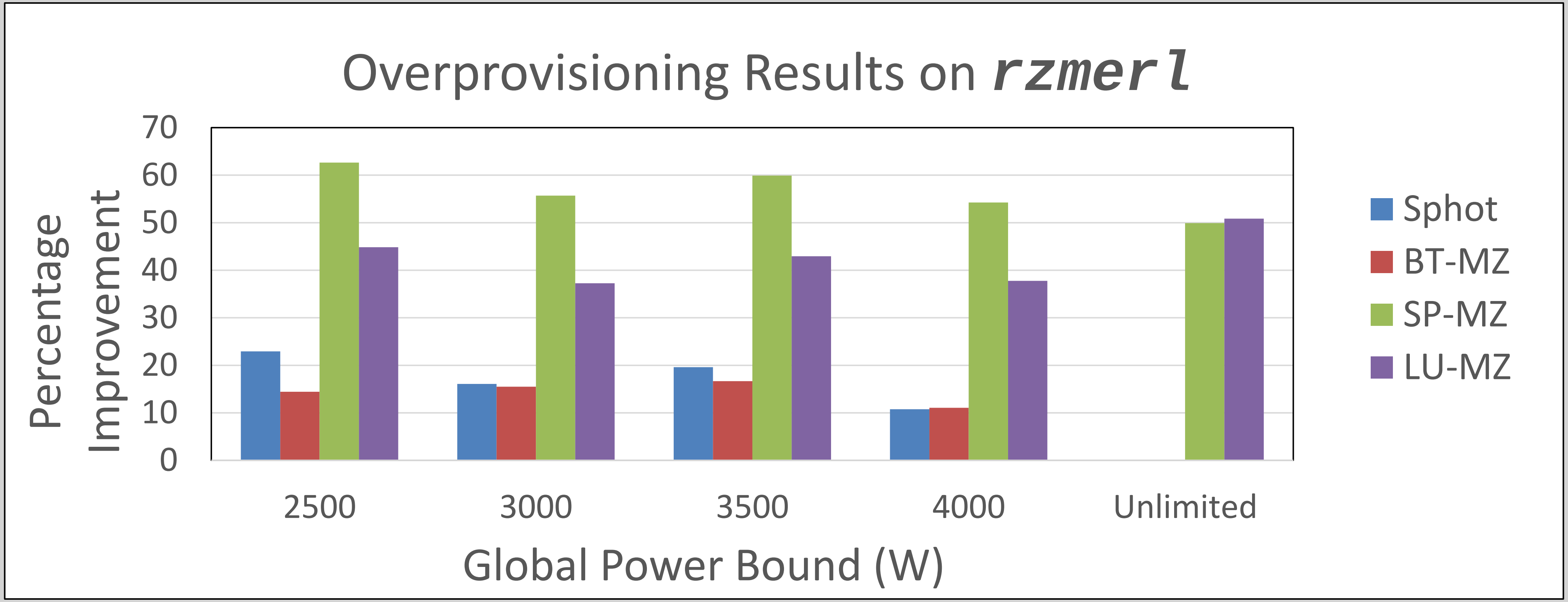


- Some applications are more memory intensive than others
- Some applications don't use all the allocated power

Multiple-node Results: Configurations

- Configuration: Number of nodes, number of cores per node, PKG power cap per socket , **`(n x c, p)`**
- Special Configurations
  - **`Packed`**: Use all cores on a node before adding another node
  - **`Spread`**: Use 4 cores on a node, spread evenly across available set of nodes
  - **`Max/Min`**: To denote 115 W / 51 W of PKG power, based on thermal specifications

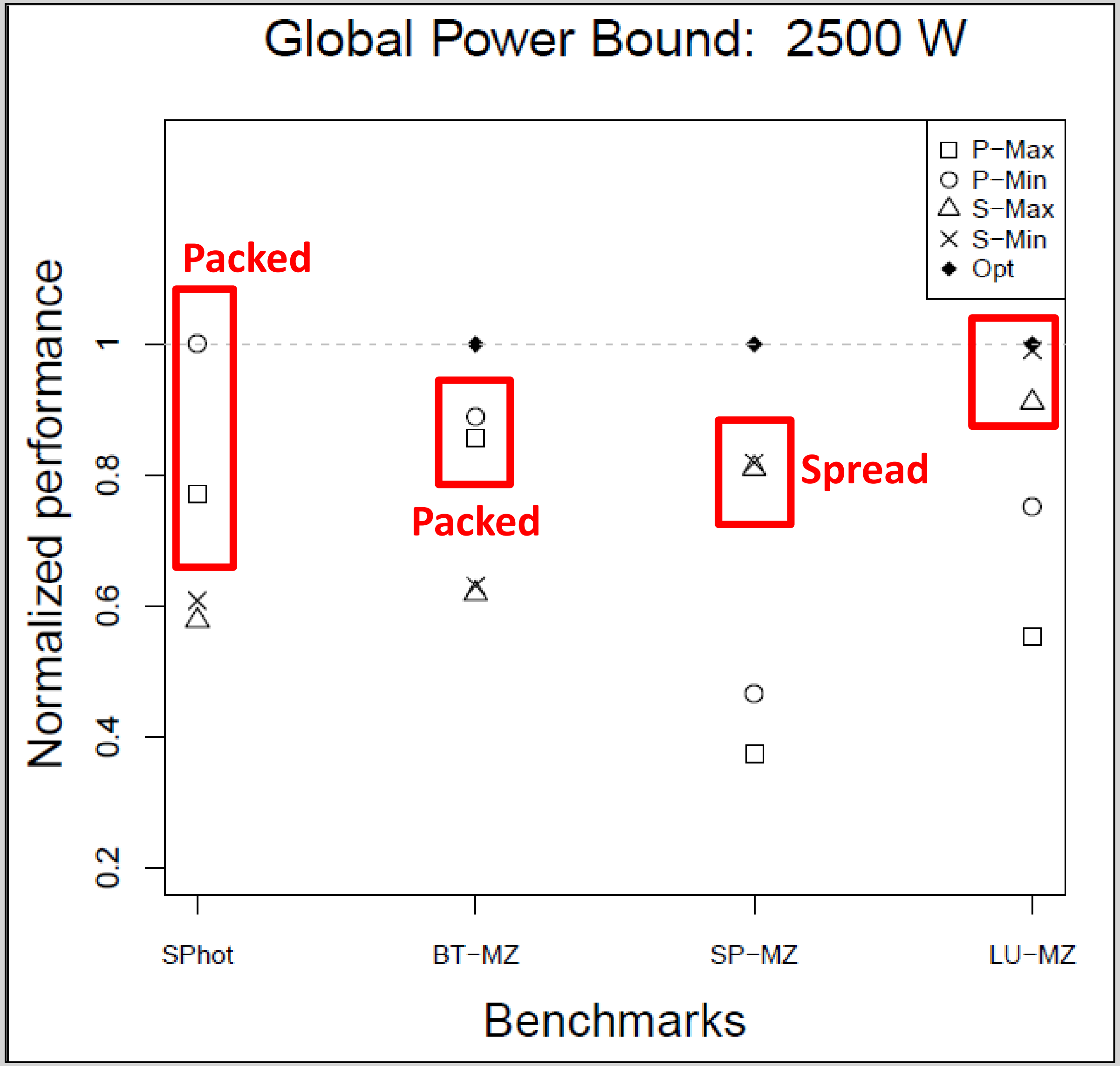
Multiple-node Results: Overprovisioning



- Compare `packed-max` to `optimal` under a power bound
- Maximum improvement: 62%; Average: 32%

Multiple-node Results: Comparing Configurations

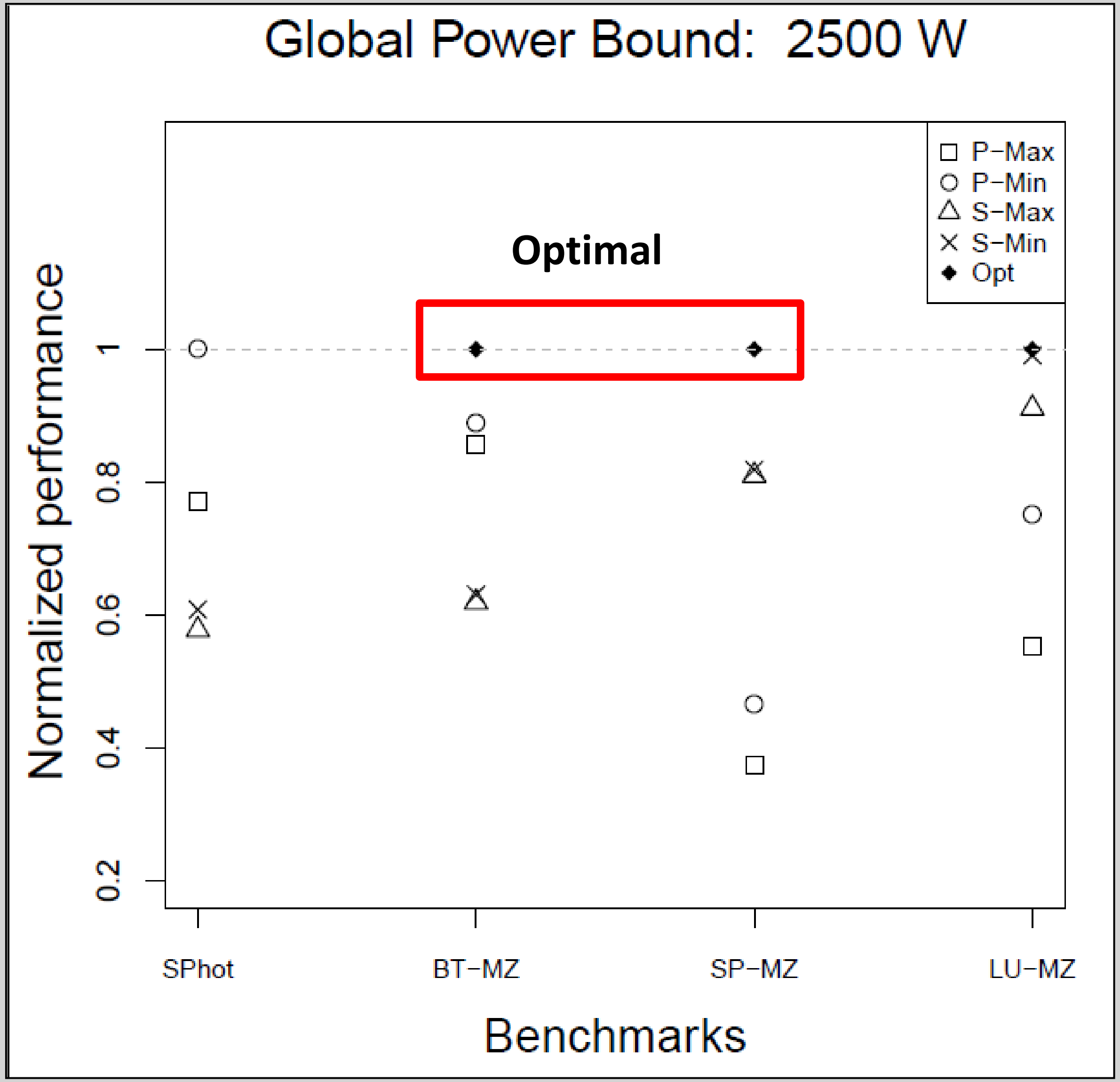
- Some applications prefer `packed` over `spread`
- Significant performance difference between `packed` and `spread`, `max` and `min`



Multiple-node Results: Comparing Configurations

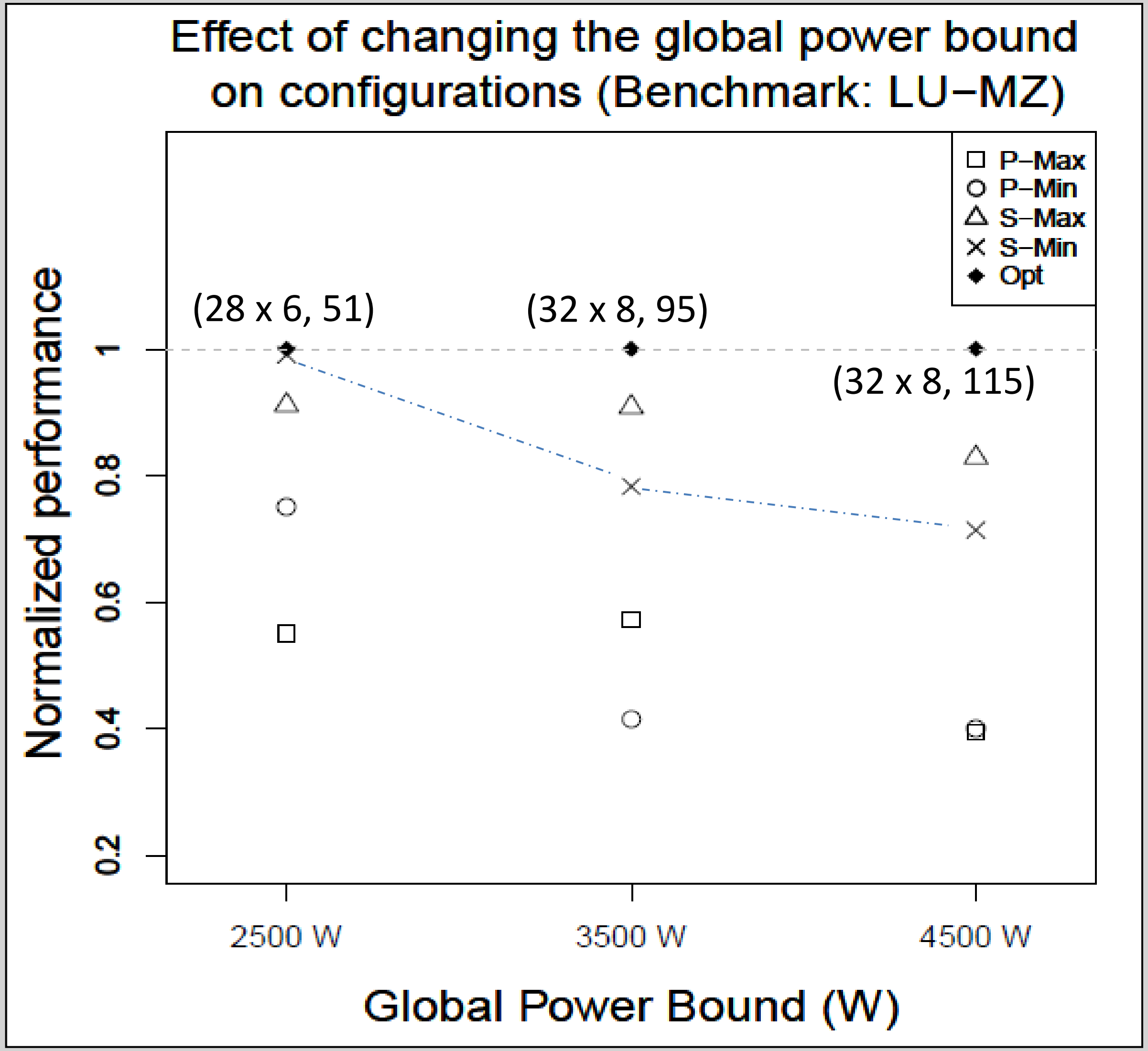
- Best configuration is not always one of the canonical ones

- Depends on application characteristics



Multiple-node Results: Comparing Configurations

- Optimal configuration depends on the global power bound



Multiple-node Results: Comparing Configurations

Global Power Bound: 2500 W

SPhot

	Configuration (n x c, p)	Time (s)
P-Max	(12 x 16, 115)	74.27
P-Min	(22 x 16, 51)	57.24
S-Max	(24 x 4, 115)	99.18
S-Min	(32 x4, 51)	94.19
Opt	(22 x 16, 51)	57.24

SP-MZ

	Configuration (n x c, p)	Time (s)
P-Max	(12 x 16, 115)	13.88
P-Min	(20 x 16, 51)	11.16
S-Max	(22 x 4, 115)	6.40
S-Min	(28 x4, 51)	6.34
Opt	(22 x 8, 80)	5.19

- Maximum improvement of 42.2% for SPhot, 62.6% for SP-MZ
- Fewer total cores at lower power can give better performance (192 vs 176 cores for SP-MZ)

Multiple-node Results: Comparing Configurations

Bmark: SP-MZ

Global Bound (W)	Optimal Configuration (n x c, p)	Time (s)
2500 W	(22 x 8, 80)	5.19
3500 W	(26 x 12, 80)	3.65
Unlimited	(32 x 14, 115)	2.63

- Optimal configuration depends on the global power bound

Multiple-node Results: Take-away

- Significant time difference between `packed` and `spread`; `max` and `min` configurations
- Optimal configuration:
  - —Not always one of the canonical configurations
  - —Depends on application characteristics
    - CPU-bound applications prefer packed configurations
    - Memory-bound applications prefer fewer cores per node
    - Applications that scale well prefer lower power per node and more nodes
  - —Depends on the global power bound enforced

Summary

Hardware Overprovisioning

- Limit power to a larger number of nodes
- Reconfigure based on application characteristics
- Performance improvement of up to 62% on real applications

Future work

- Software and tools to automatically achieve good performance on hardware overprovisioned systems

Acknowledgments

We would like to extend our thanks to:

- Livermore Computing and their support staff for providing us with the appropriate permissions required to access the MSRs.
- Lawrence Livermore National Laboratory, Department of Energy
- National Science Foundation

Thank You!

Questions?