Power Bounds and Large Scale Computing

Friday, March 1, 2013

Bronis R. de Supinski[1]
Tapasya Patki[2], David K. Lowenthal[2],
Barry L. Rountree[1] and Martin Schulz[1]

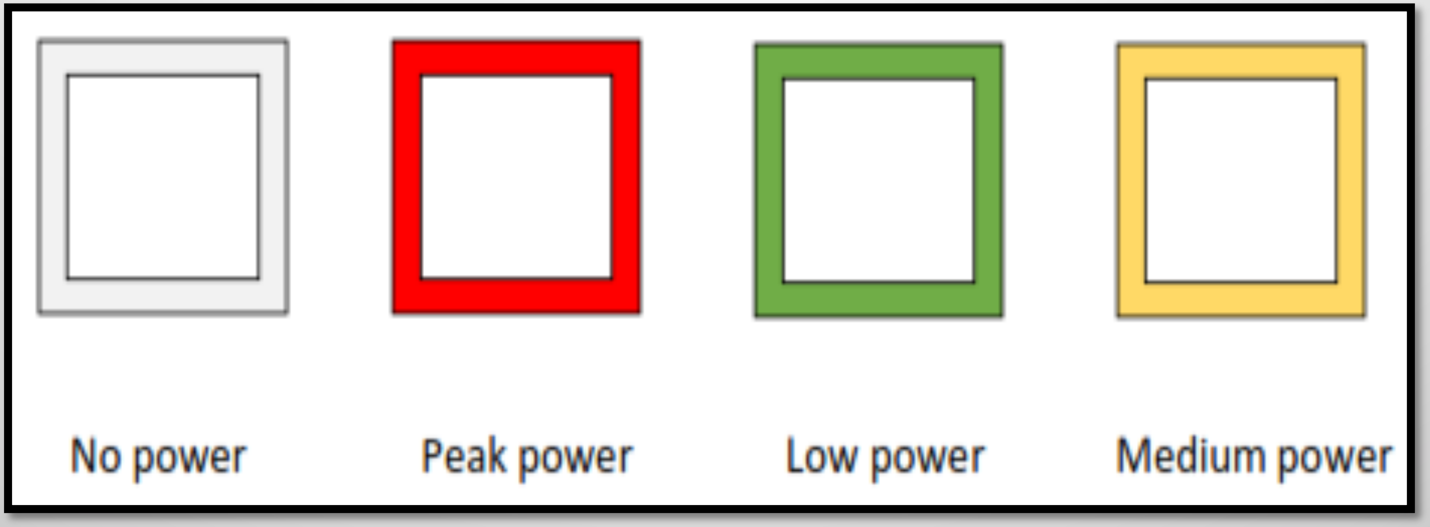
[1] Lawrence Livermore National Laboratory
[2] University of Arizona

LLNL-PRES-622092

Future supercomputers will require power bounds

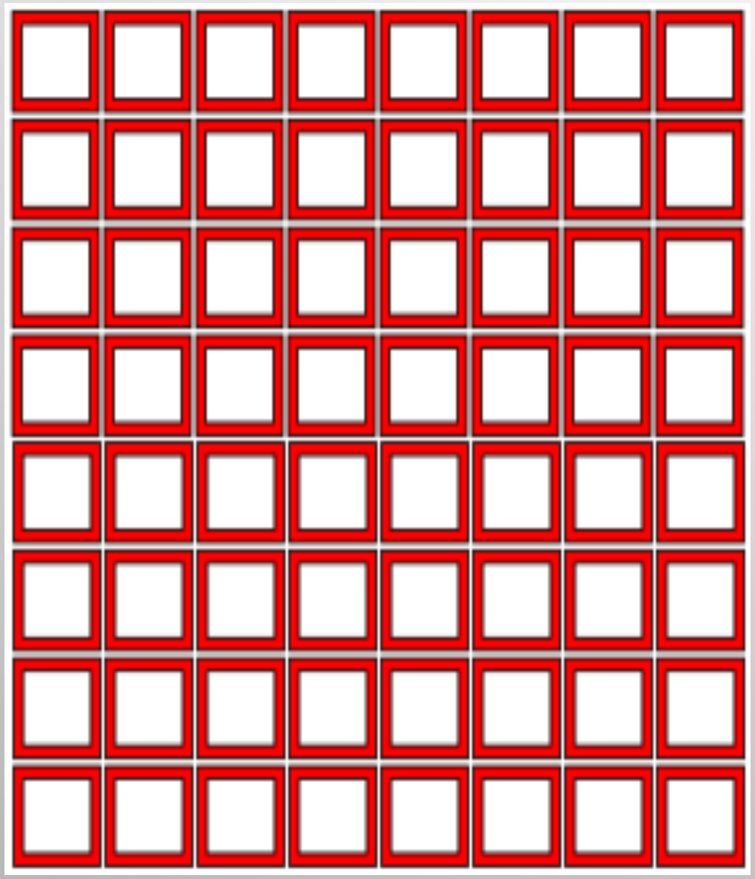
- Power is the limiting factor for exascale
  - Sequoia: 20PFlop, 9.6MW system
  - 50x performance improvement is straightforward
  - How to do with only a 2x power increase is nontrivial
  - Power is expensive ($1M per MW per year)
  - Infrastructure limits power available
- Two possible mechanisms to limit power
  - Worst-case provisioning
  - Overprovisioning with enforced power bound

Node power vocabulary



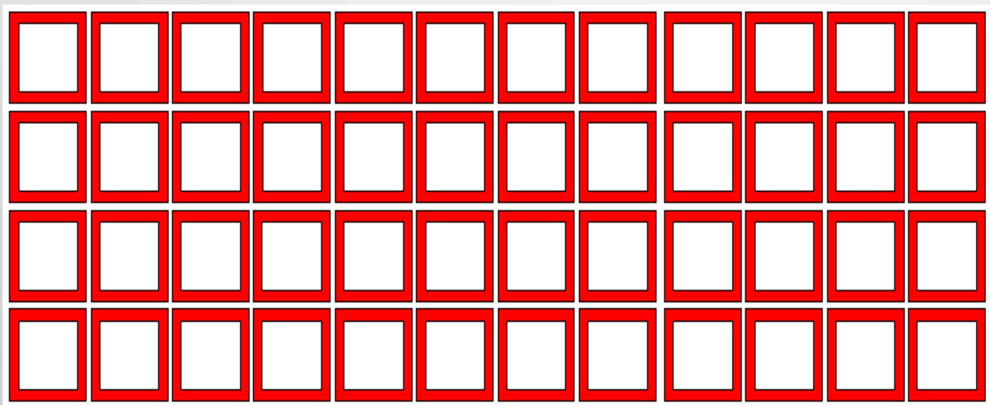
- Power domains
  - Package: processor die (cores + on-chip caches)
  - Uncore: Off-chip caches, QPI
  - DRAM

Example of worst-case provisioning



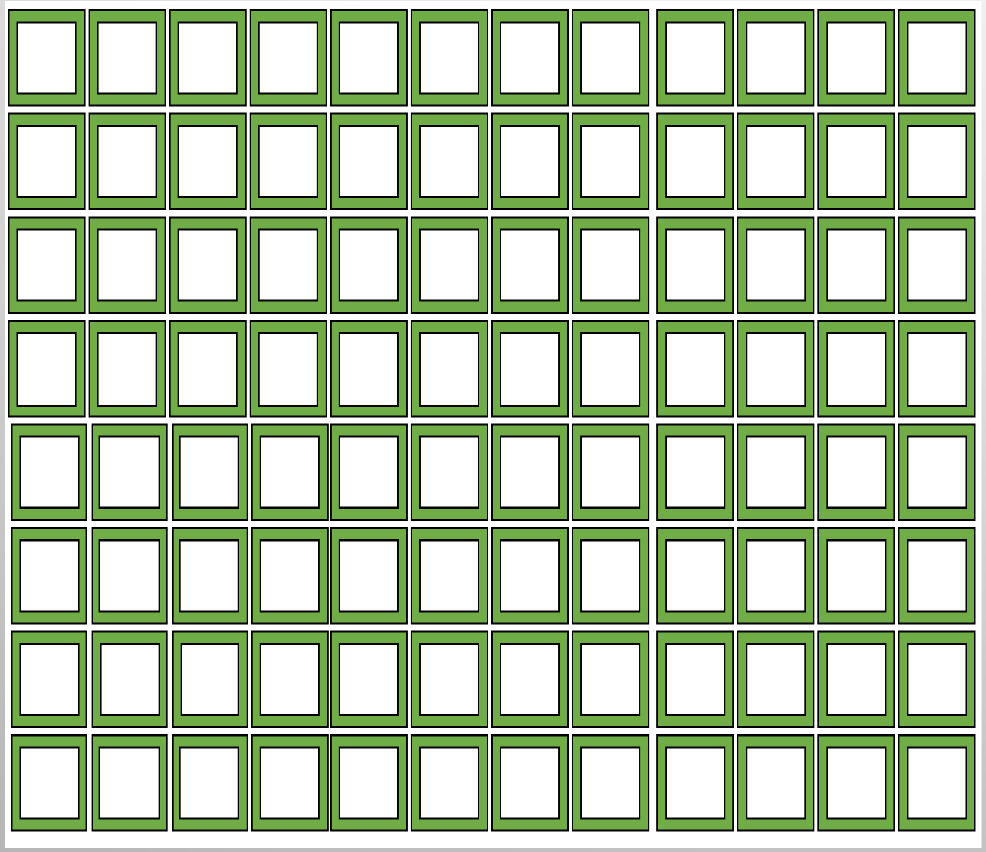
- Assume 64 node cluster
- Assume 300W peak node power
- Total power
  - 19.2KW worst case
  - Less in practice
    - Perhaps close for LINPACK
    - Much less for real applications

What if facility only supports 14.4KW?



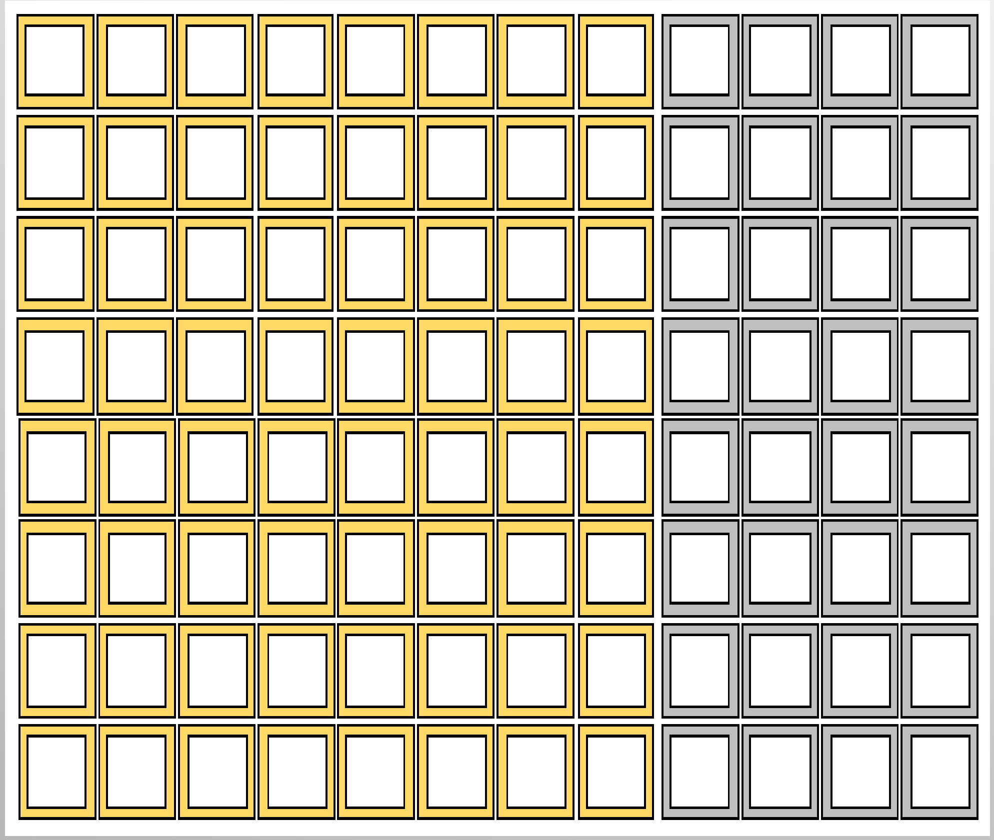
- Assume 300W peak power nodes
- Worst-case provisioning: 48 nodes
- Can we do better?

Overprovisioning at 14.4KW



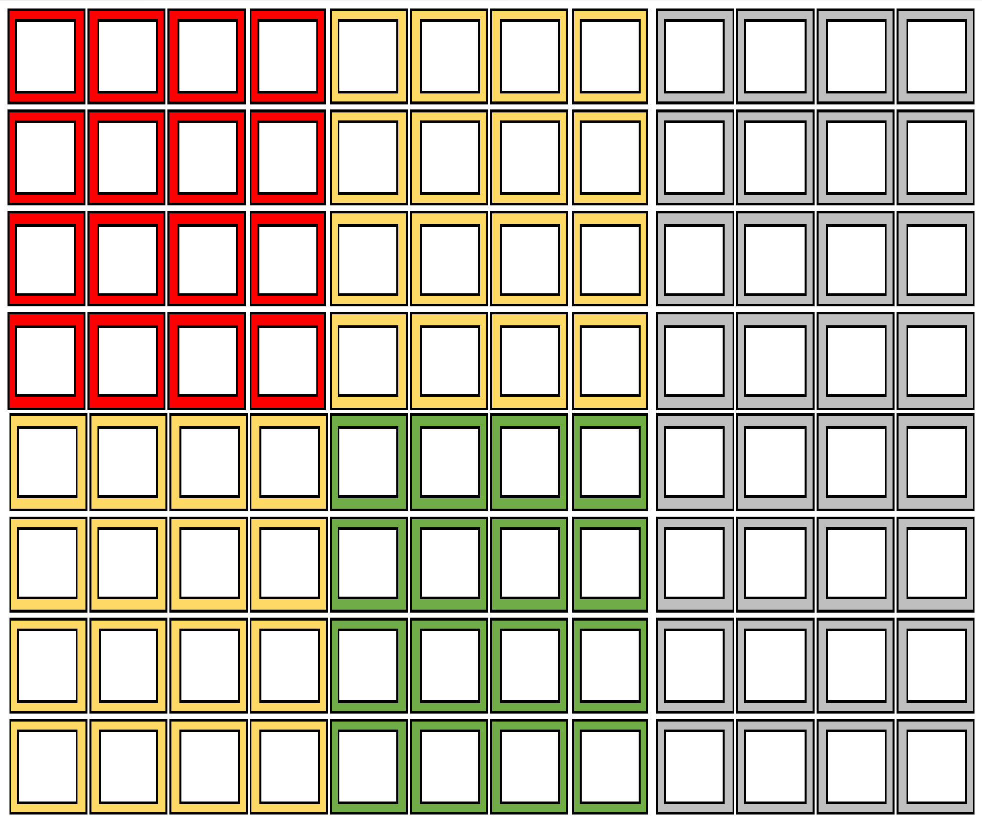
- Assume application power requirements per node of 150W
- Could use 96 nodes and remain within the facility power bound

Suppose application requires 225W



- Must limit to 64 active nodes
- Both cases can use more nodes than worst-case provisioning
- Solution is to reconfigure based on application characteristics

How would overprovisioning work?

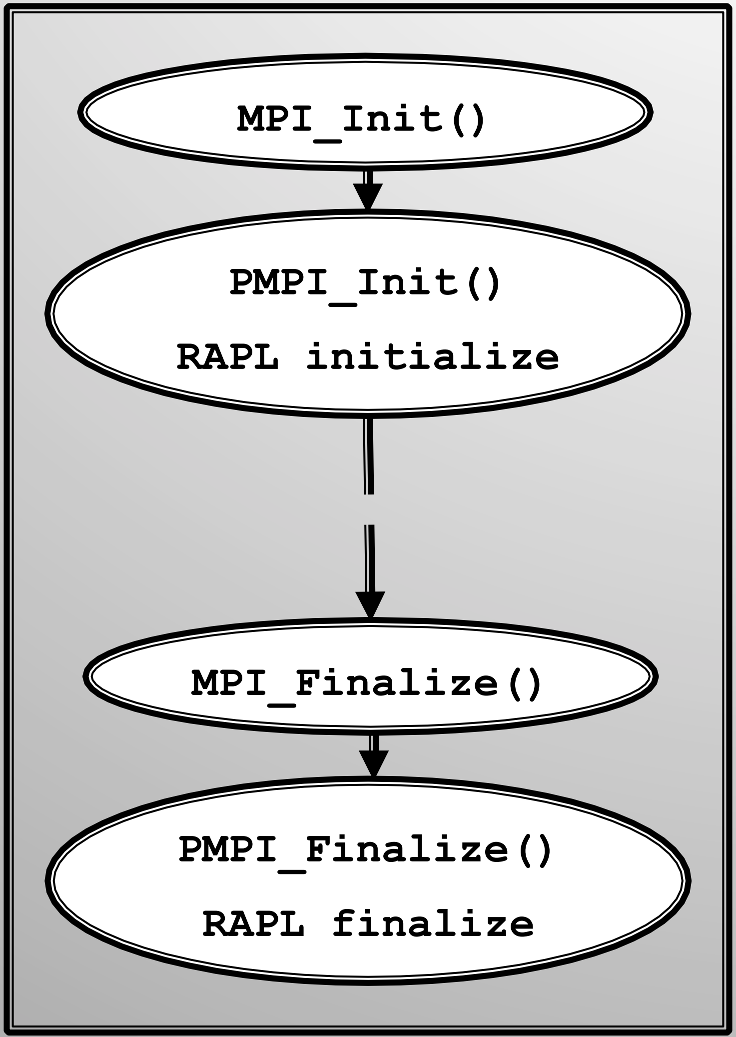


- Limit power
  - Per job
    - Statically
    - Dynamically
  - Per system dynamically
- We study impact of overprovisioning on performance given a power-constraint

Is overprovisioning a radical idea?

- Already provided in current processors
  - Intel Nehalem, Sandy Bridge w/Turbo Boost
  - AMD Phenom II w/TurboCore
- Power capping with Intel's Running Average Power Limit (RAPL)
  - Domains (vary between server and client models)
    - Power Plane 0 (PP0) and Power Plane 1 (PP1)
    - DRAM and Package (PKG)
  - Can specify a power bound for a specified time window
    - Hardware ensures average power below bound over window
    - Implemented in Machine Specific Registers (MSRs)

Our `librapl` provides safe user-space MSR access



- Uses MPI profiling interface
- Sets up MSRs
- Gathers power and CPU frequency data per process
- In use at several sites
- Download at: https://github.com/tpatki/librapl

Our RAPL-based experiments emulate overprovisioning

- 32 node Sandy Bridge cluster
  - 2 sockets, 8 cores per socket, 2.6 GHz/3.3 GHz (Turbo)
  - Use RAPL PKG capping to emulate overprovisioning
  - Thermal limit is 115W; 51W minimum PKG power cap
- Assume hybrid MPI + OpenMP
  - ASC Purple SPhot
  - NAS-MZ: BT-MZ, SP-MZ and LU-MZ
  - Synthetics
    - CPU-bound and memory-bound
    - Scalable and not-scalable

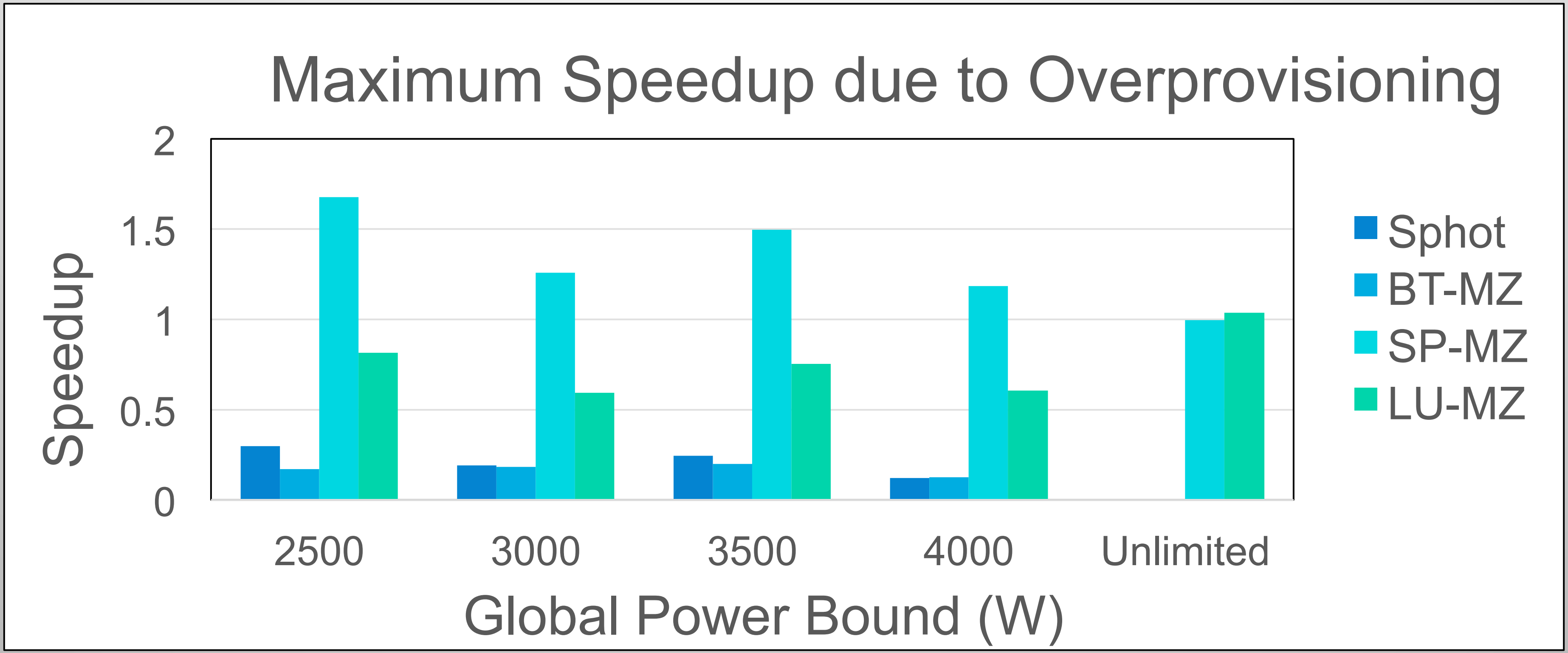
Baseline results for Intel Turbo Boost

- Single node tests for all applications
  - Consistent with expectations
  - Show overprovisioning can improve performance
- Turbo frequency depends on active core count
- All nodes engage in Turbo mode similarly
- Application power profiles
  - No application uses all allocated power
  - Some applications are more memory intensive than others, which implies higher DRAM power percentage

Vocabulary for multiple node results

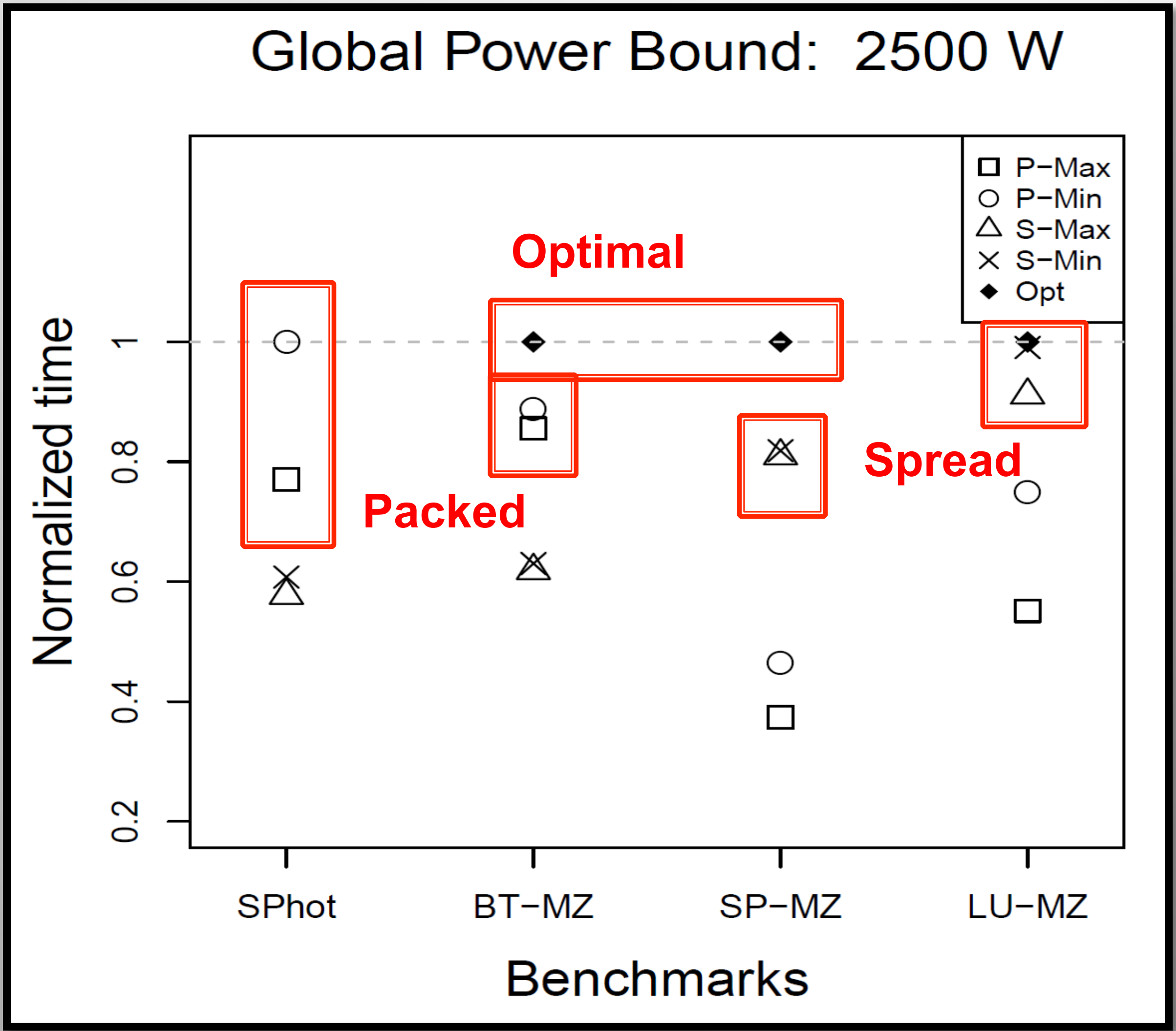
- Configuration
  - Number of nodes: n
  - Number of cores per node: c
  - PKG power cap per socket: p
  - Denote configuration as (n x c, p)
- Canonical configurations
  - `Packed`: All cores on a node before adding node
  - `Spread`: 4 cores on a node, spread evenly over nodes
  - `Max`: 115W PKG power
  - `Min`: 51W PKG power

Overprovisioning yields speed ups between 50% and 73%



- Compare `packed-max` to `optimal` under a power bound

Different configurations suit different applications



- Some applications prefer `packed` over `spread`
- Significant performance difference between `packed` and `spread`, `max` and `min`
- Best configuration depends on power bound and is not necessarily a canonical configuration

SP-MZ shows how optimal configuration varies with the global power bound

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

Fewer total cores at lower power can perform better: SP-MZ 192 vs. 176 cores

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

Overprovisioning supercomputers is a promising power bound technique

- Power is the limiting factor for exascale
  - 20MW is system power bound target
  - Worst-case provisioning unnecessarily limits system size
  - Overprovisioning will require new infrastructure
    - System-wide measurement and control
    - Resource manger innovations to support and to exploit
- We observe application performance improvements of more than 50% under a power bound with overprovisioning

Lawrence Livermore National Laboratory