Evaluating Energy Savings for Checkpoint/Restart in Exascale

Bryan Mills, Ryan E. Grant,

Kurt B. Ferreira and Rolf Riesen

E2SC Workshop – November 18, 2013

Requisite Agenda Slide

- Checkpointing
  - Why is power important here?
- Experimental Setup
- Power Profiles
  - Checkpoint writes
  - Whole Application Execution
- Conclusions
- Future Work

Major Challenges at Exascale

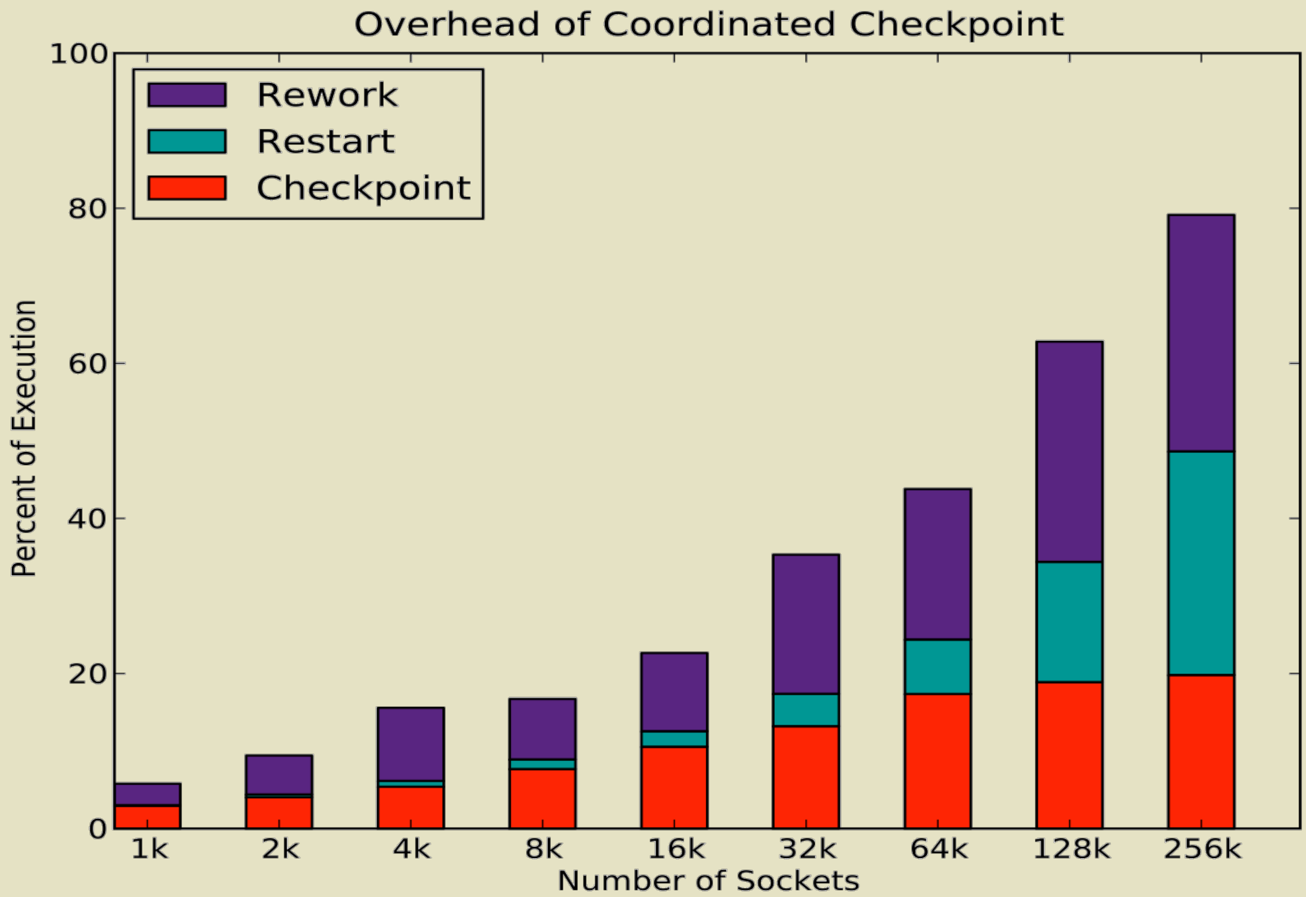
Making the transition to exascale poses numerous unavoidable scientific and technological challenges

- **Harnessing the Potential of Massive Parallelism**
  - Effective use of unprecedented levels of concurrency requires new conceptual and programming paradigms
- **Reducing Power Requirements**
  - Based on current technology, scaling today's systems to an Exaflop level would consume ~500 Megawatts of power
- **Resilience to Failure**
  - An immediate consequence of exascale computing is that the frequency of errors will increase

Checkpointing

- Periodically pause execution and write state to stable storage
- In event of failure restore from saved state
- Two main methods:
  - <u>Coordinated</u>
    - *Everyone rollback*
  - <u>Uncoordinated</u>
    - *Failed nodes rollback*

Time Spent in Checkpoint Operations



Research Question

Can we conserve energy during checkpoint operations?

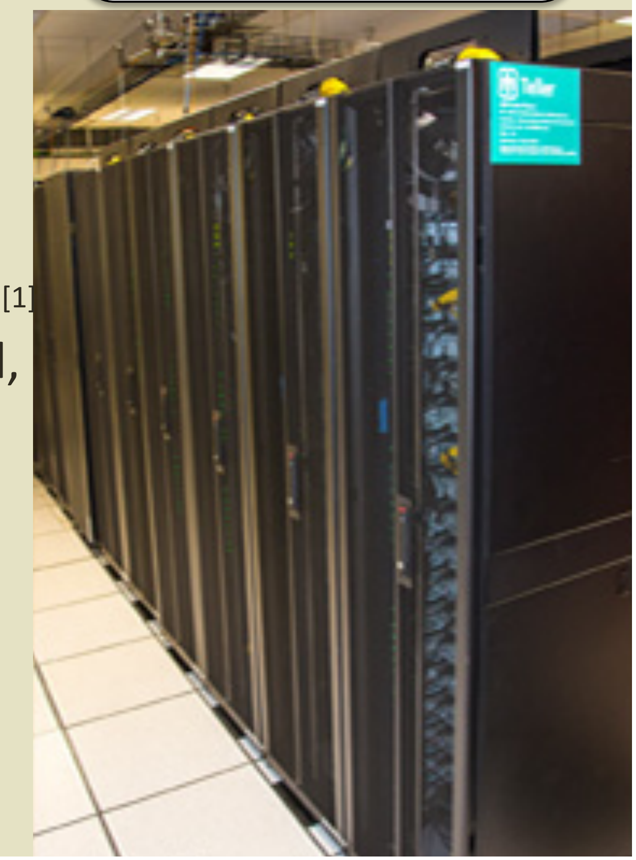
- Checkpoint write is an IO intensive operation, resulting in low CPU usage
- Can we reduce power by reducing the CPU speed without effecting the checkpoint?
  – Use Dynamic Voltage Frequency Scaling (DVFS)

Experimental Setup

- HPC Cluster at Sandia National Labs
  - 104 node cluster
  - AMD Llano Fusion APU
    - 4 core x86 + 400 core Radeon HF 6550D
    - 6 Power Gears 1.4Ghz – 3.8Ghz
  - 500Gb SSD in each node
  - Component level power measurement [1]
    - CPU, Memory, Network, SSD, Motherboard, etc.
  - Two networks
    - 1Gb Ethernet
    - Infiniband - Qlogic QDR InfiniBand HCA

Hello My Name is…

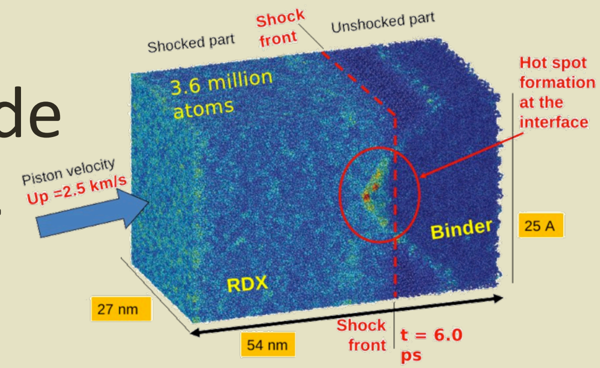
Teller



[1] J. H. L. III, P. Pokorny, and D. DeBonis. Powerinsight - a commodity power measurement capability. In *The Third International Workshop on Power Measurement and Profiling in conjunction with IEEE IGCC 2013,* Arlington Va, 2013.

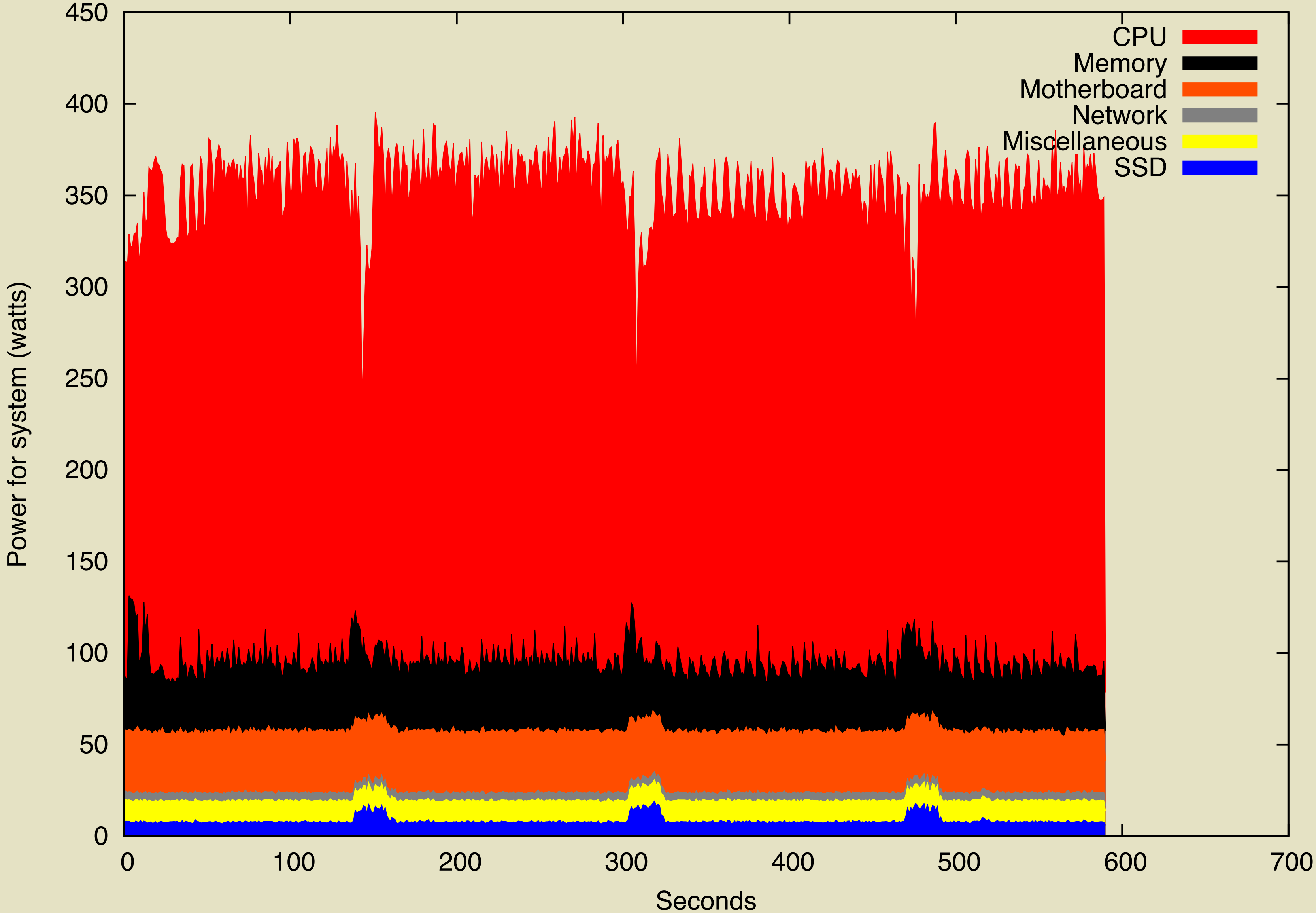
Software Stack

- Real applications running MPI
  - LAMMPS - molecular dynamics code
  - HPCCG - conjugate gradient solver



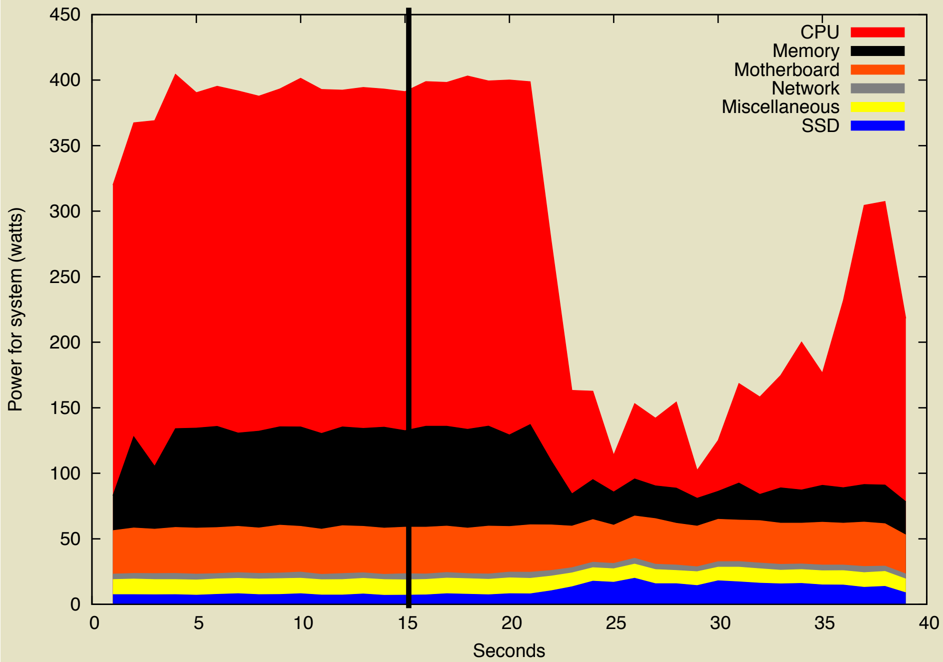
- OpenMPI 1.3.4 with BLCR
  - Berkley Lab Checkpoint/Restart
    - Kernel level module for checkpoint/restart processes
    - Coordinated - “stops” communication and checkpoints each process individually

Component Level Power Monitoring

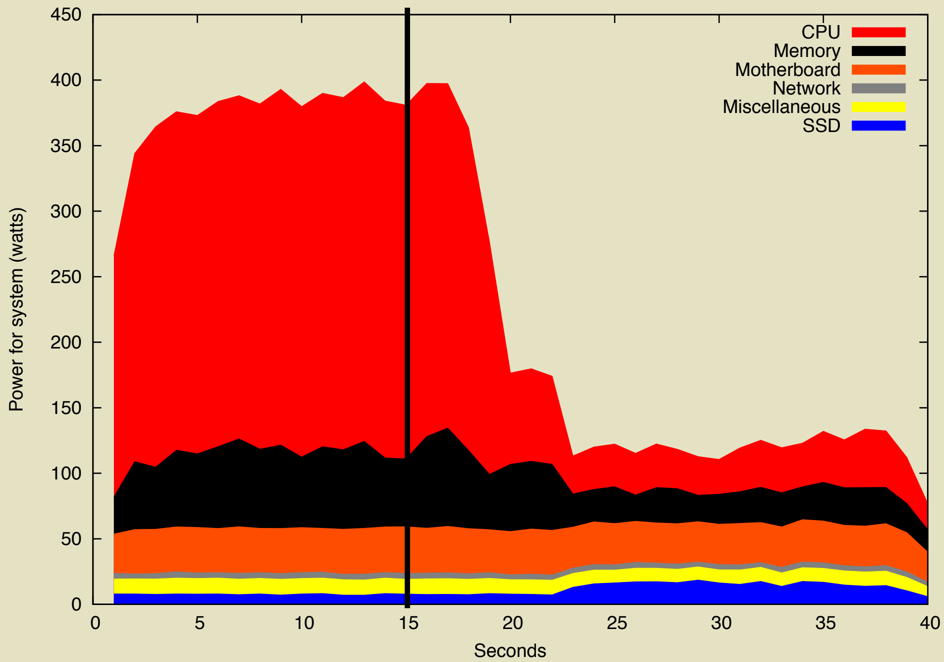


Local Checkpoint

- Write checkpoint to local SSD only
  - 4 nodes running HPCCG



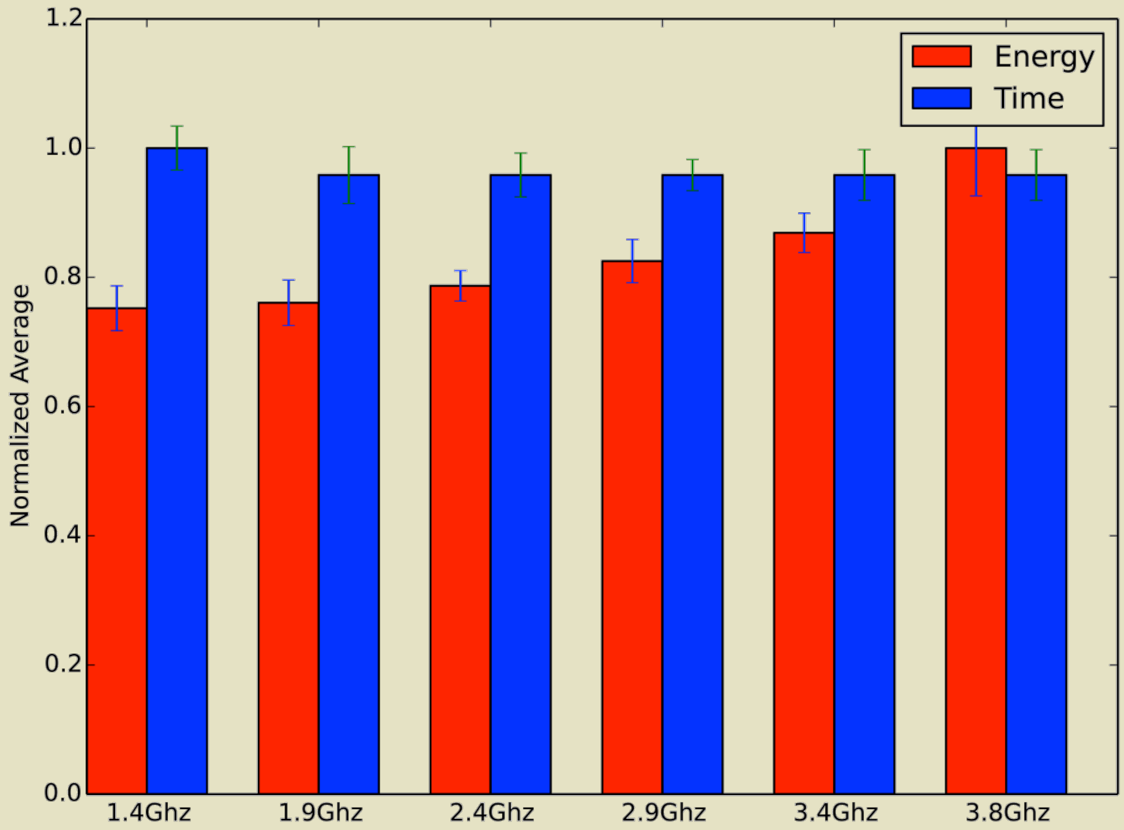
High Power 3.8 Ghz



Low Power 1.4 Ghz

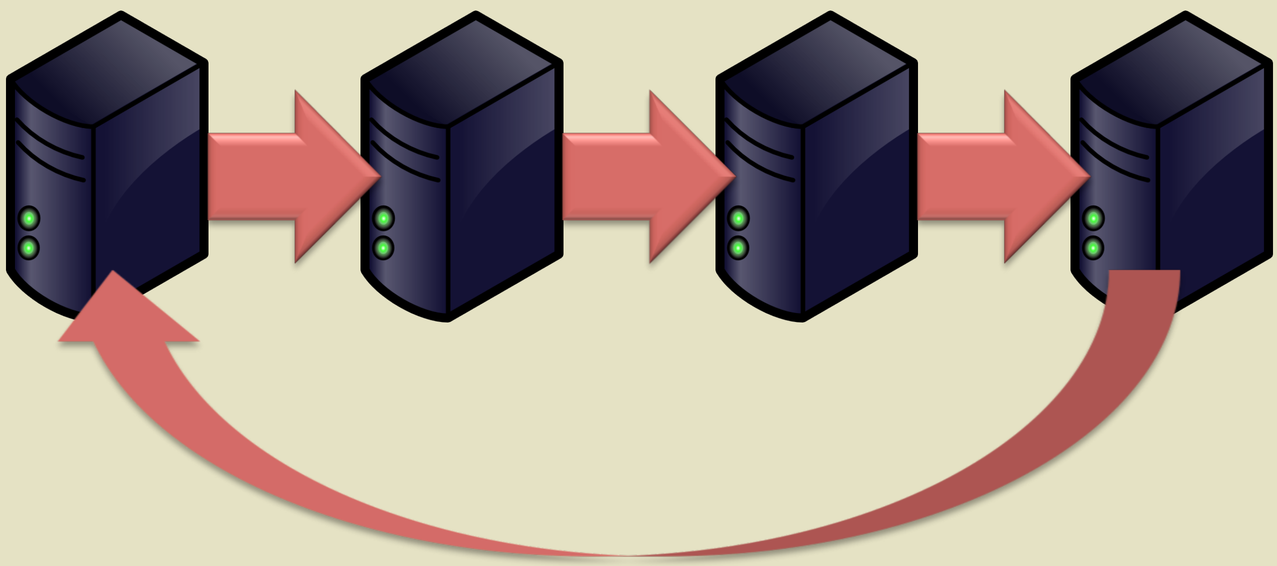
Power vs Energy

- Write checkpoint to local SSD only
  - Average over 10 runs each



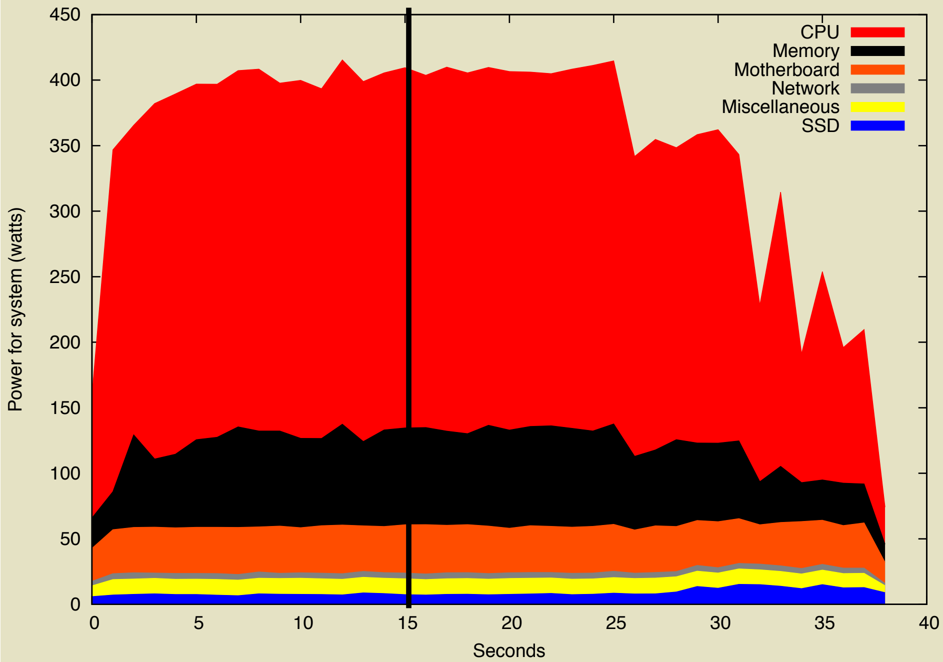
Remote Checkpoints

- Checkpoints not useful on a dead node
- Write checkpoint to remote system over NFS
  - IP over Infiniband
  - RDMA over Infiniband

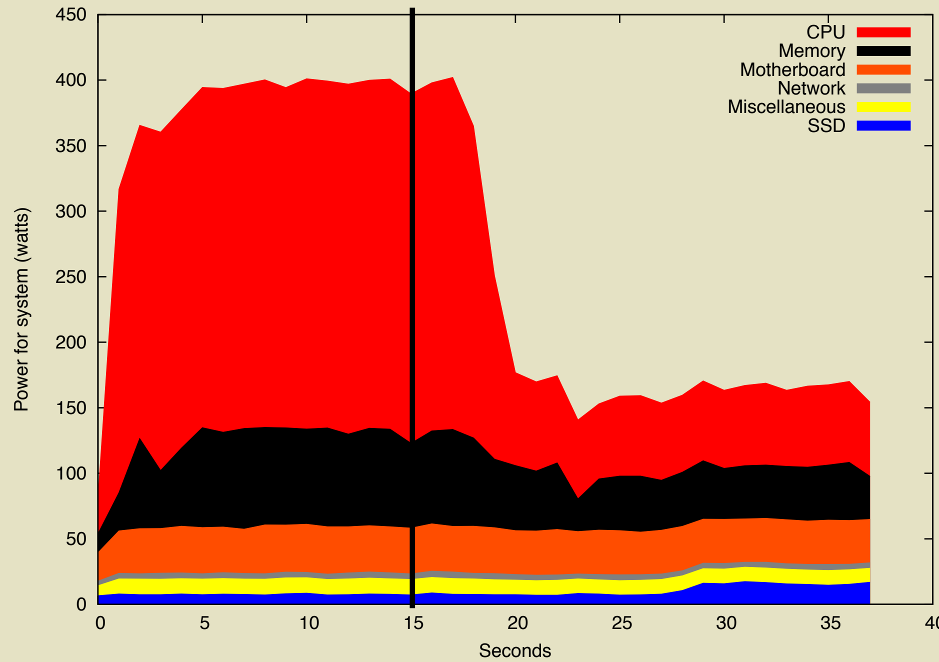


IP Over Infiniband

- Write checkpoint to remote SSD using IP
  - 4 nodes running HPCCG



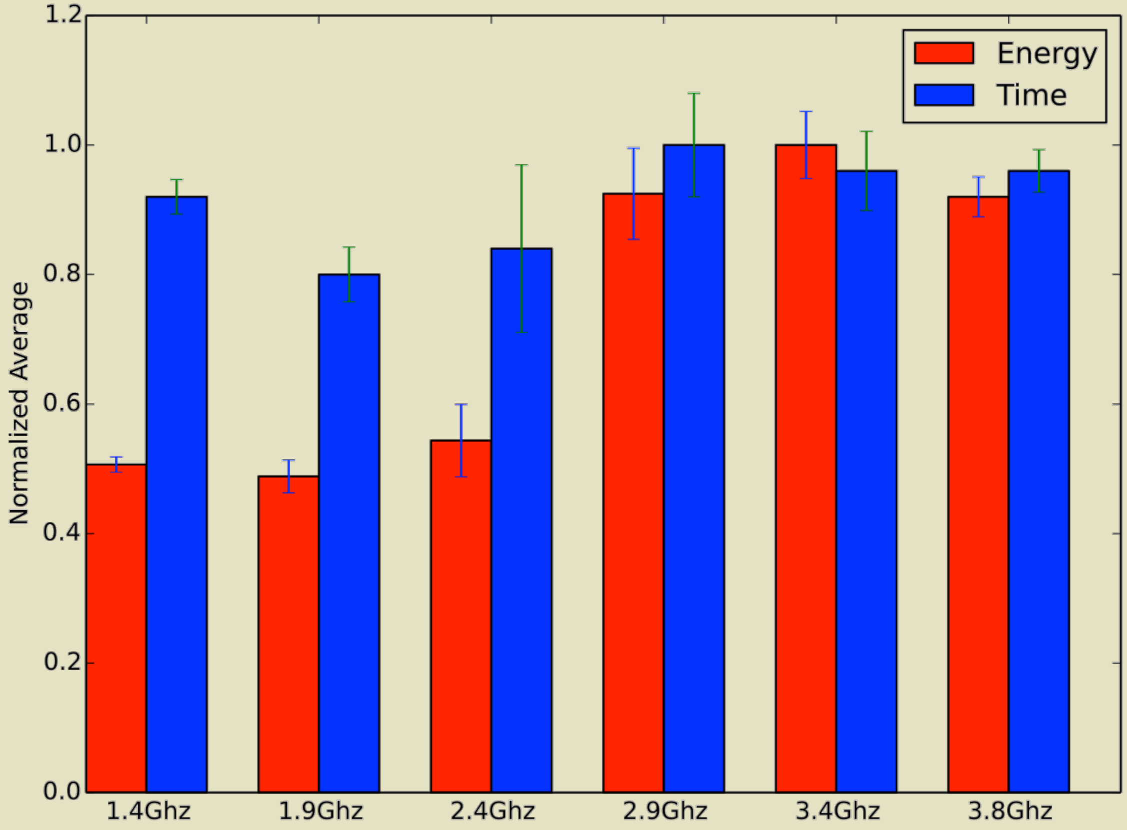
High Power 3.8 Ghz



Low Power 1.4 Ghz

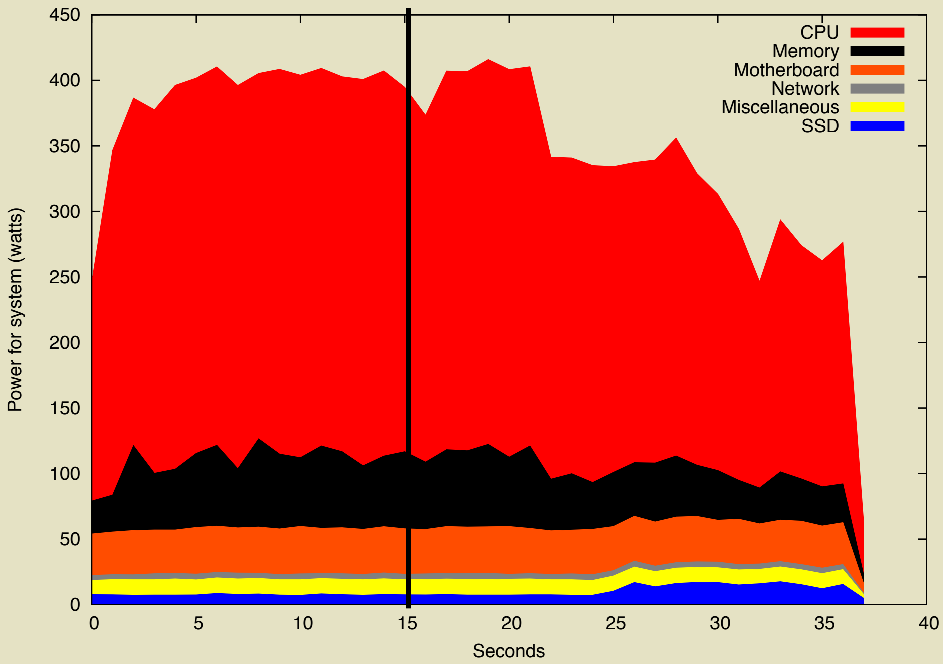
Power vs Energy

- Write checkpoint to remote SSD using IP
  - Average over 10 runs

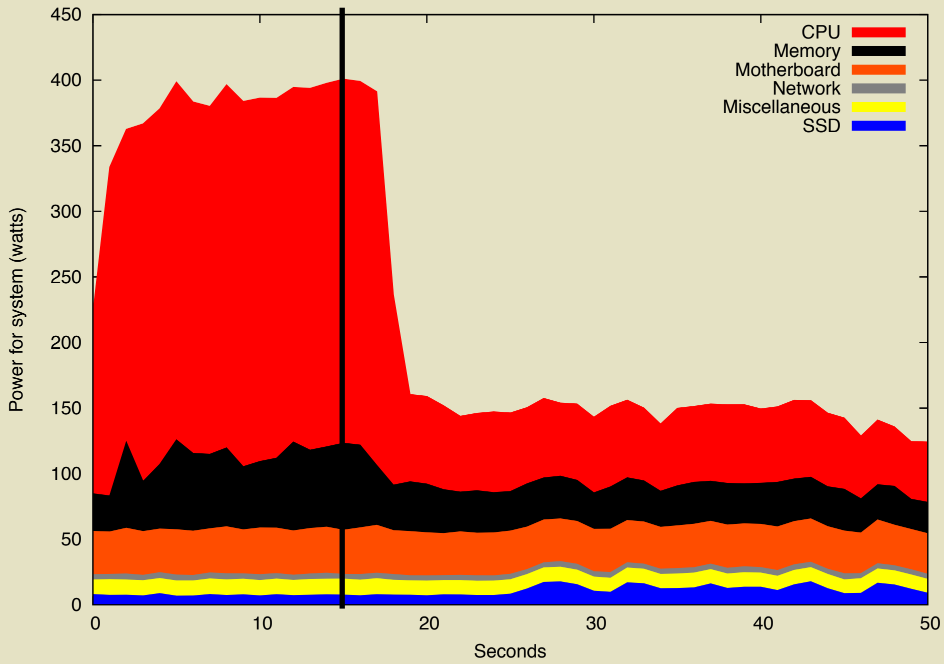


RDMA Over Infiniband

- Write checkpoint to remote SSD using RDMA
  - 4 nodes running HPCCG



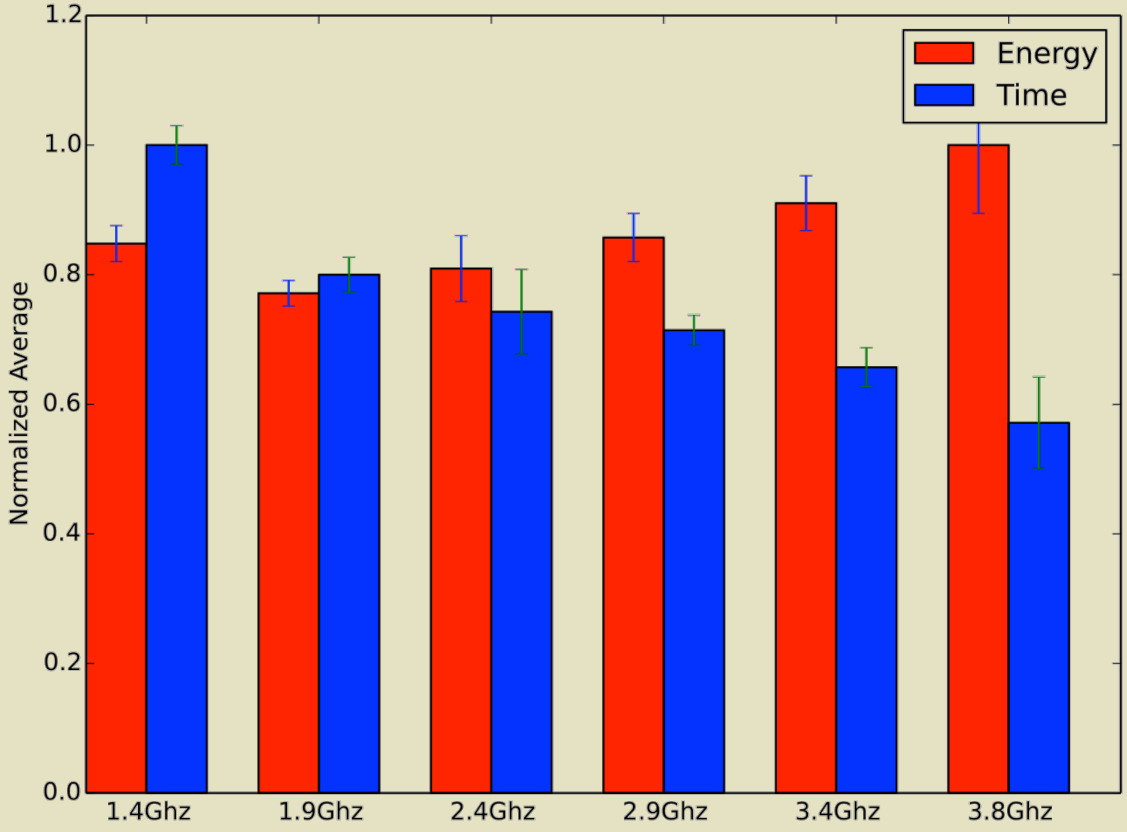
High Power 3.8 Ghz



Low Power 1.4 Ghz

Power vs Energy

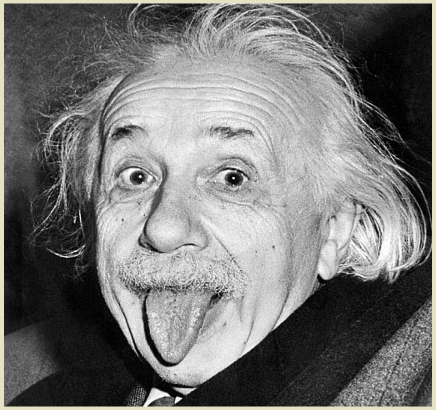
- Write checkpoint to remote SSD using RDMA
  - Average over 10 runs



What does this show us?

- Previous research suggested that one should *always* reduce CPU frequency during IO operations [1,2]

"In theory, theory and practice are the same. In practice, they are not." - Einstein

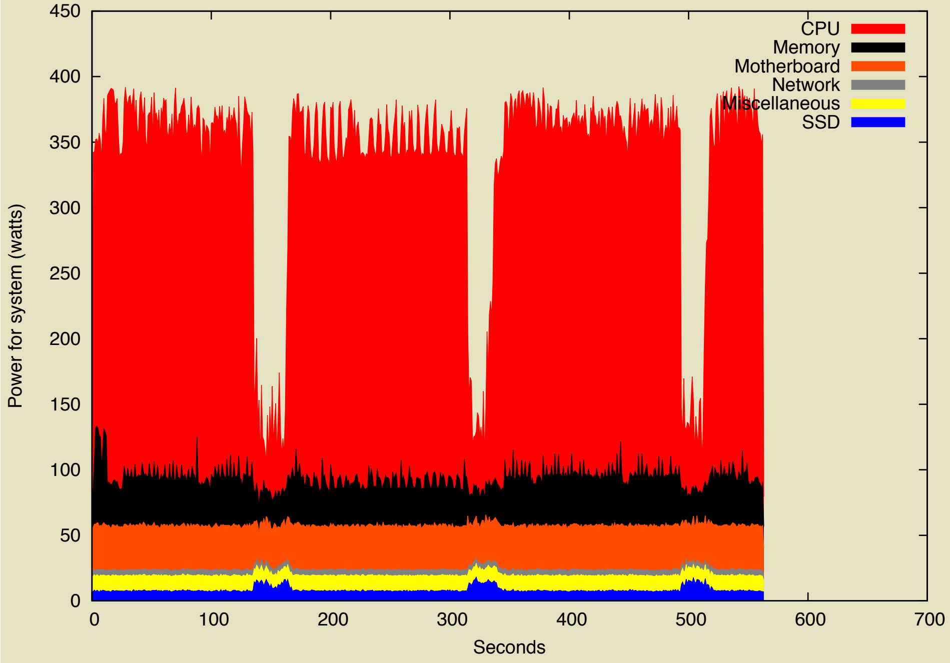


- Depends on IO subsystem, especially if network IO is involved as it would be for checkpoints
- There might still be a benefit
  - Next experiment looks at entire application execution

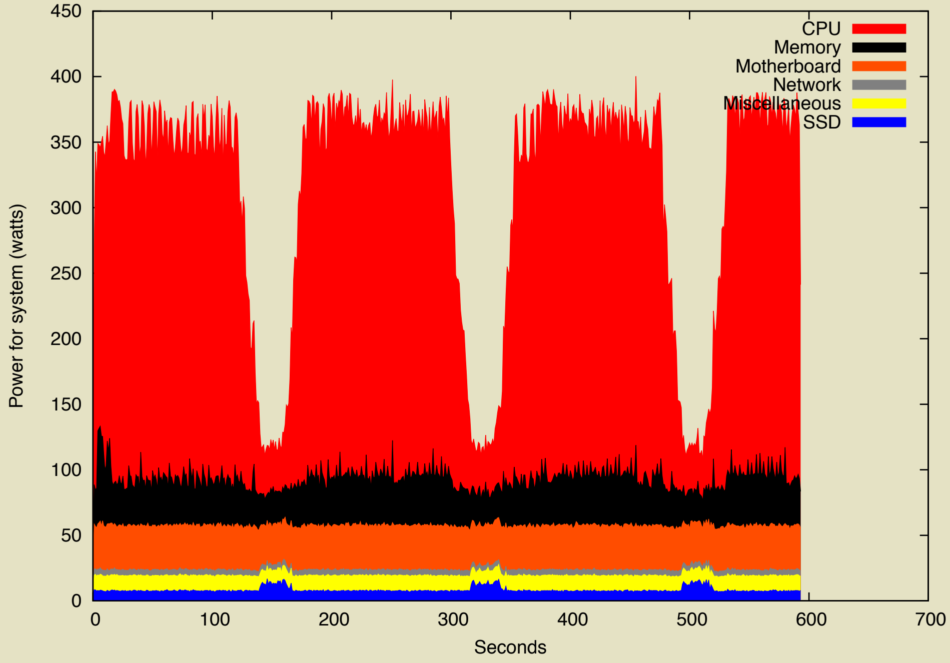
[1] M. Diouri, et.al. Energy considerations in checkpointing and fault tolerance protocols. In *Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on,* pages 1–6. IEEE, 2012.
[2] T. Saito, et.al. Energy-aware I/O optimization for checkpoint and restart on a NAND flash memory system. In *Proceedings of the 3rd Workshop on Fault-tolerance for HPC at extreme scale,* pages 41–48. ACM, 2013.

Entire Application Execution

- Write 3 checkpoints local SSD write
  - 4 nodes running LAMMPS



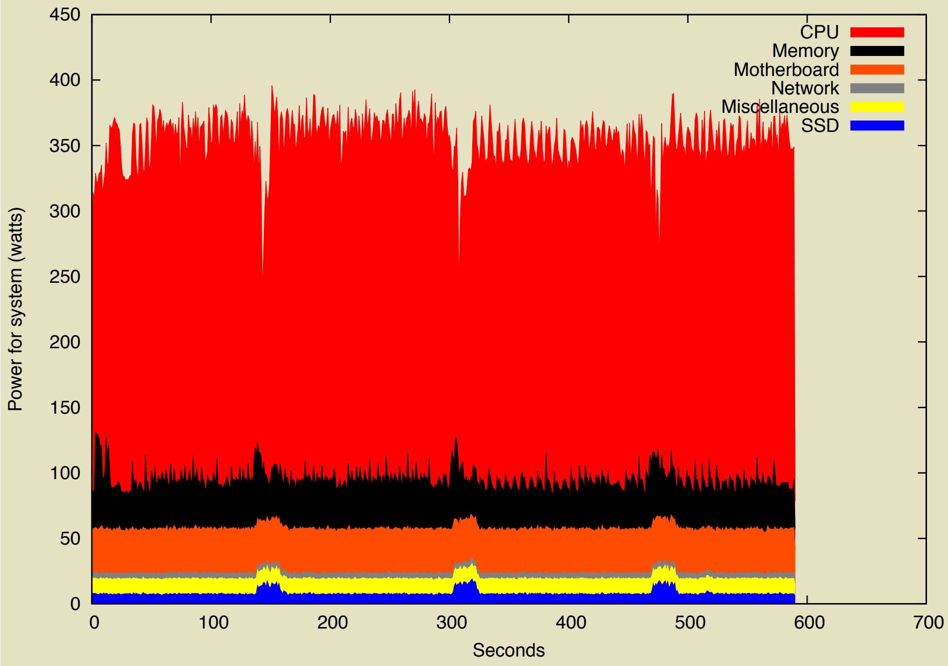
High Power 3.8 Ghz



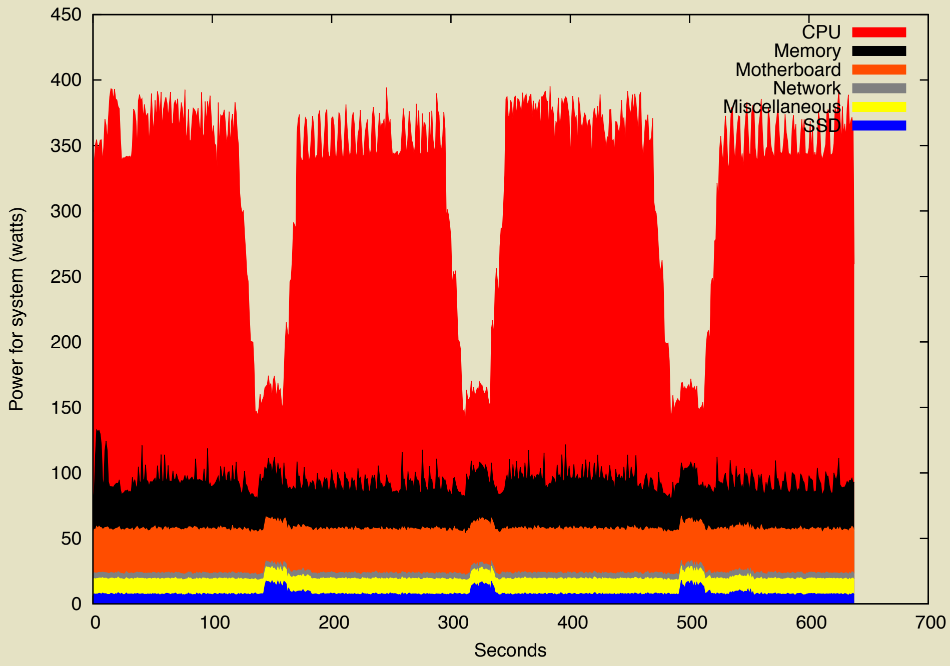
Low Power 1.4 Ghz

Entire Application Execution

- Write 3 checkpoints remote over IPoIB
  - 4 nodes running LAMMPS



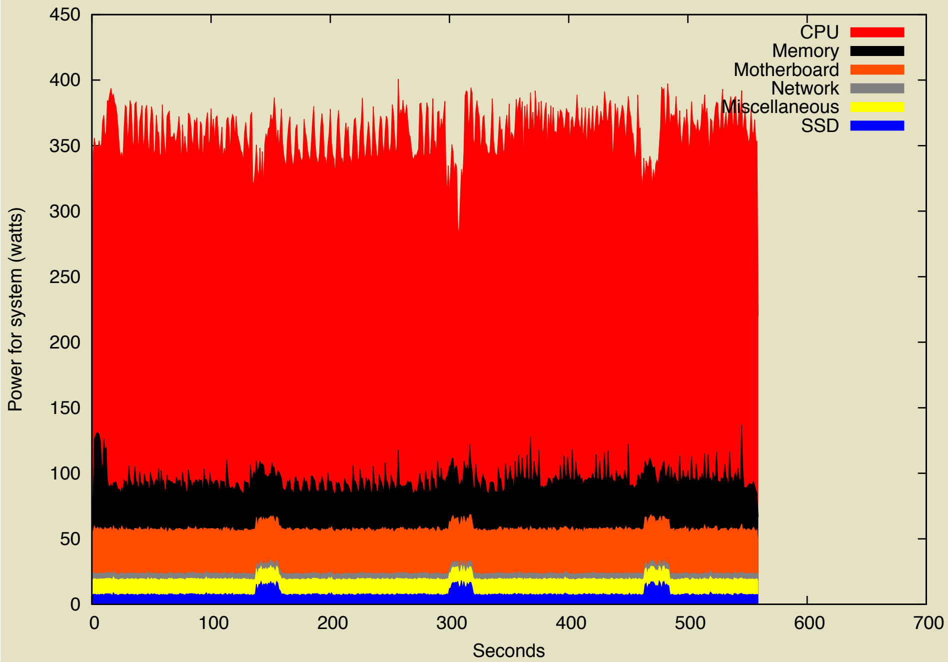
High Power 3.8 Ghz



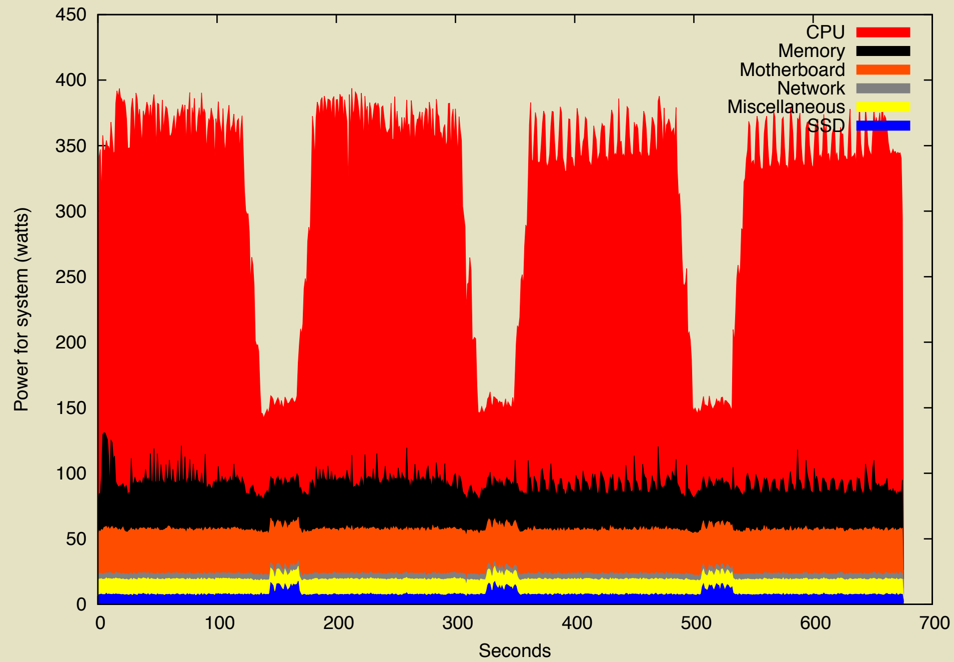
Low Power 1.4 Ghz

Entire Application Execution

- Write 3 checkpoints remote over RDMA
  - 4 nodes running LAMMPS



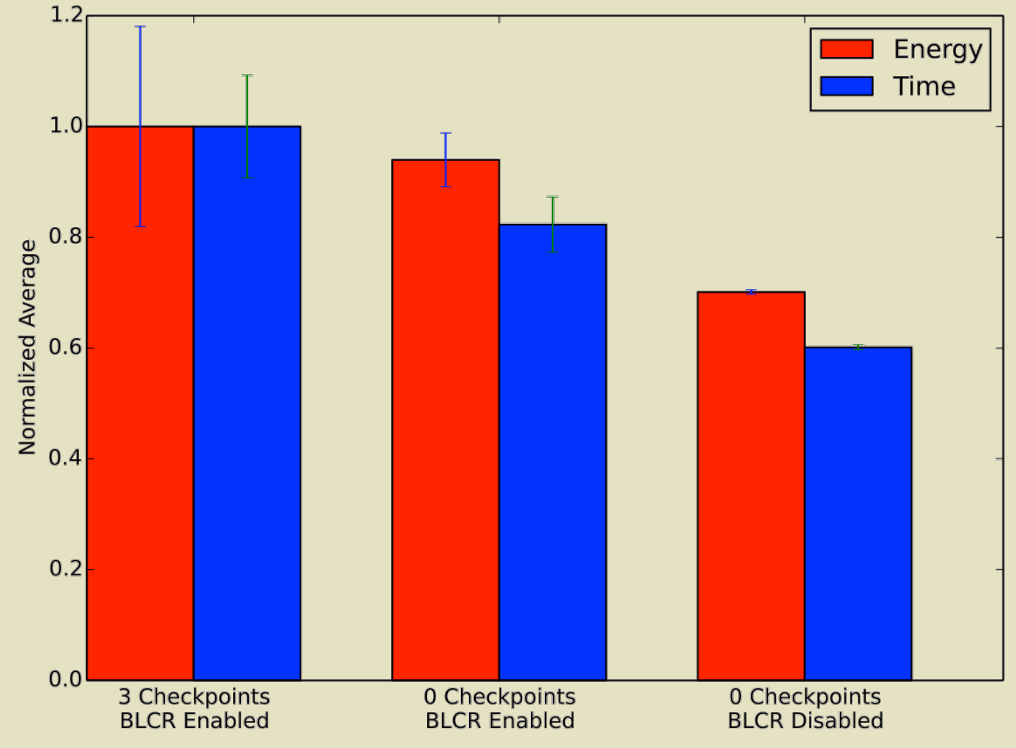
High Power 3.8 Ghz



Low Power 1.4 Ghz

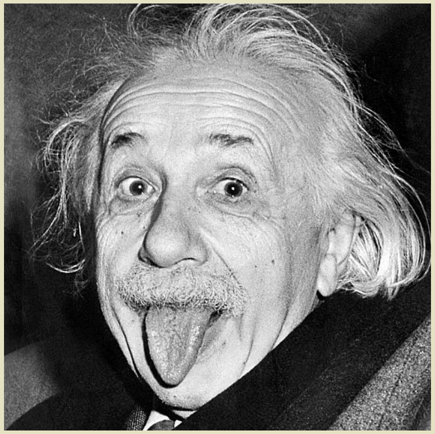
BLCR/OpenMPI Variability

- Unable to draw any conclusion from entire application experiments due to variation in time to solution
  - Simple experiment using 10 runs on 4 nodes local checkpoints



Conclusions

- Can save 50-60% of energy during checkpoint write translating to 5-15% of overall application energy savings in exascale systems
- IO operations are sometimes CPU intensive
  - Especially with the Qlogic Infiniband
- BLCR in OpenMPI is problematic
  - Control thread causes lots of variance
- Staged Checkpoints?



Future Work

- Measure power in restart operation
- Fix BLCR control thread
  - Underway but difficult (might just dedicate core)
- Test at larger scale
- Look at fully offloaded Infiniband cards
  - Initial results look very promising
- Parallel Filesystem instead of NFS
- Staged Checkpoints

Staged Checkpoints

- Multi-tiered checkpoint write
  - First write to local SSD, copy to network, etc.
  - Continue working after local SSD write
- Our work implies this might not be beneficial
  - If the network copy consumes CPU cycles then application performance will suffer
  - Implies that you want fully offloaded network operations
  - What about network bandwidth? Do we need a separate network for checkpoint writes?

Questions? Peanuts? Comments?

Bryan Mills

bmills@cs.pitt.edu

Exascale Computing

"One or more key attributes of the system achieve a 1,000 times the value of a corresponding attribute of a "Petascale" system" [1]

Three dimensions

- Functional performance
  - Flops per second
- Physical attributes
  - Shrink Petascale down to a desktop
- Application performance
  - Speed of science

[1] K. Bergman, et.al. Exascale computing study: Technology challenges in achieving exascale systems. 2008

Functional Performance

- 1,000 times more powerful than petascale
  - Tianhe is 30x smaller

Computer	Petaflops	Growth
Exascale	1000	
<u>Exascale GAP</u>		
Tianhe-2	33.86	30x
Titan	17.59	58x
Sequoia	16.32	62x
K Computer	10.51	100x
Mira	8.16	125x
JUQUEEN	4.14	250x

* Top 500 (http://www.top500.org/)

Energy Challenge

- DoE has set an energy target of 20 megawatt for exascale computing
  - Requires a minimum of 23x reduction in energy!

Computer	Energy (MW)	Growth	Projected (MW)
Exascale	-	-	20
	Exascale GAP		
Tianhe-2	17.80	30x	534.0
Titan	8.20	58x	475.6
Sequoia	7.89	62x	489.2
K Computer	12.65	100x	1265.0
Mira	3.95	125x	493.8
JUQUEEN	1.97	250x	492.5

Resilience Challenge

- Mean time between failure (MTBF) projected to be 5-20 years per node
  - At best we are looking at a node failure every 20 minutes if we simply scale todays technology

Computer	# Nodes	Growth	Projected	MTBF (20yr)
Tianhe-2	16,000	30x	480,000	21.9 minutes
Titan	18,688	58x	1,083,904	9.69 minutes
Sequoia	98,304	62x	6,094,848	1.72 minutes
K Computer	80,000	100x	8,000,000	1.32 minutes
Mira	49,152	125x	6,144,000	1.71 minutes
JUQUEEN	28,672	250x	7,168,000	1.46 minutes