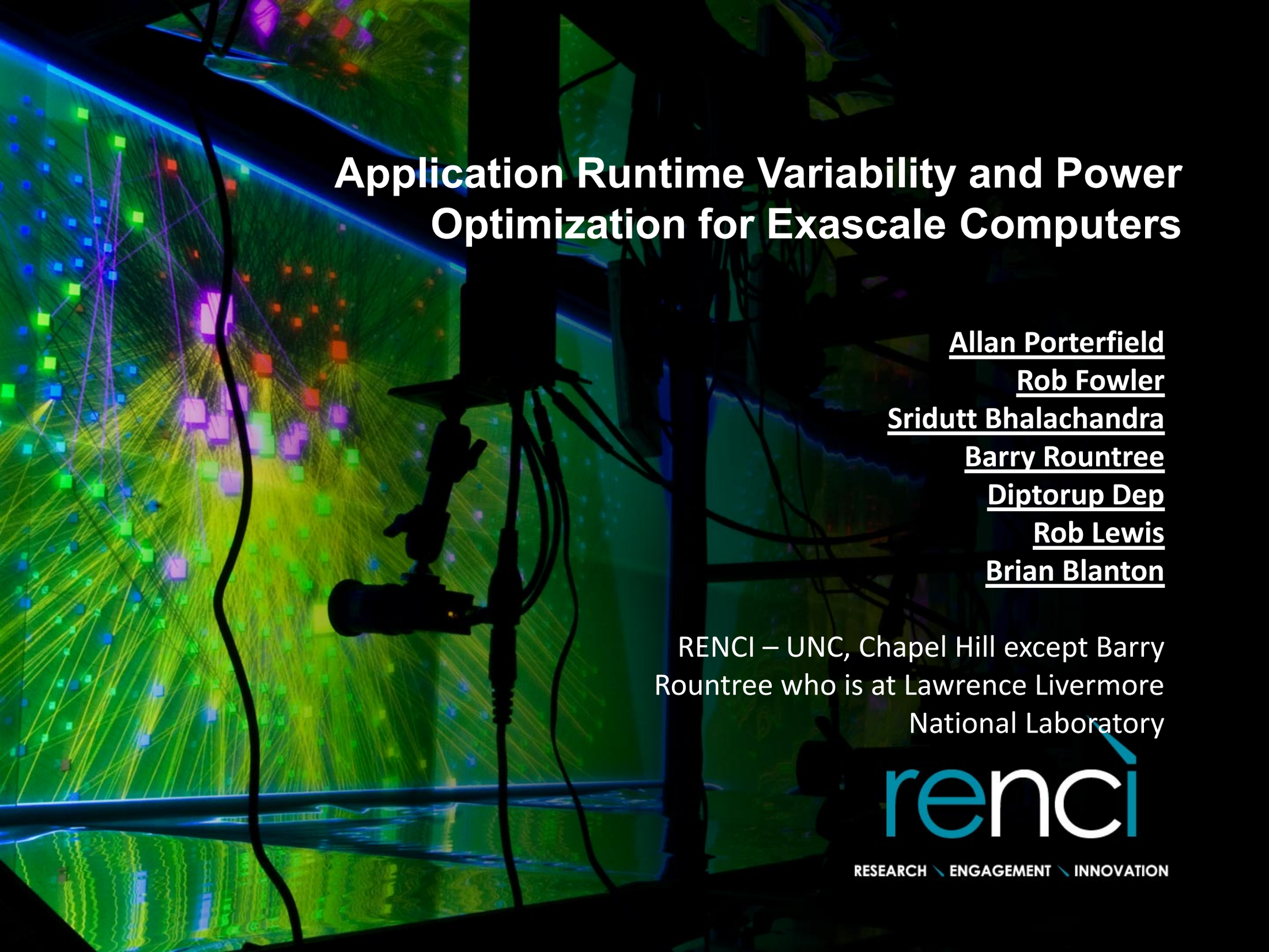


Application Runtime Variability and Power Optimization for Exascale Computers

Allan Porterfield
Rob Fowler
Sridutt Bhalachandra
Barry Rountree
Diptorup Dep
Rob Lewis
Brian Blanton

RENCI – UNC, Chapel Hill except Barry Rountree who is at Lawrence Livermore National Laboratory

renci

RESEARCH \ ENGAGEMENT \ INNOVATION

Energy Management

- Almost all techniques have traded performance for energy savings
  - Minimize Energy (E) and/or limit power
    - Effective for mobile devices – increase battery life
  - Optimize for Energy-Delay (ED) or Energy-Delay^2 (ED^2)
    - Takes performance into account

- Idle Supercomputers are expensive.
  - Depreciating capital expense of the Supercomputer hardware and facility
  - Opportunity cost of problems being delayed or not being solved

renci

Energy Management

- Almost all techniques have traded performance for energy savings
  - Minimize Energy (E) and/or limit power
    - Effective for mobile devices – increase battery life
  - Optimize for Energy-Delay (ED) or Energy-Delay^2 (ED^2)
    - Takes performance into account

- Idle Supercomputers are expensive.
  - Depreciating capital expense of the Supercomputer hardware and facility
  - Opportunity cost of problems being delayed or not being solved

NOT IMPLEMENTED on HPC

renci

Exascale Limitations

- Supercomputers are using more power
  - Increased parallelism ➔ Increased processors ➔ Increased power

- Power Limited – 20 MW
  - Enough for 15,000 – 24,000 average houses
  - Difficulty in supplying the power and cooling for the Data Center
  - Operational expense (multiple millions of $ per year)
    - Energy Cost > Supercomputer Cost

renci

Potential Exascale Direction

- Overprovisioning Processors (or under provisioning for power)
  - More processors than can be supported at peak power and use processor power limits to control overall system power demand
  - Hardware power limits on each socket can be used to control overall power usage
  - Power is now resource that the OS/runtime must schedule
    - Simple – divide evenly
    - Advanced – uneven distribution based on utility function

renci

Application Power Limits

- Applications use different amounts of power
  - Memory-heavy, ALU-heavy, Network-heavy or balanced
    - Depends on the system architecture and implementation
  - Staying under the power limit using different numbers of processors
- Application power varies between phases
- Application performance need not be linear with power.
  - A memory-bound applications may see minor impact from reducing ALU performance
  - A compute-bound may see a major impact from the ALU but little if the DIMMS are slowed

renci

Performance with Power Limits?

- Systems become slightly performance heterogeneous between nodes
  - Die-to-Die Manufacturing Variation
    - Transistors use more/less power to switch
    - Leakage current varies
  - Cooling Variation
    - Better cooled processor can run faster
    - Run cooler => less leakage current

renci

Performance with Power Limits?

- Systems become slightly performance heterogeneous between nodes
  - Die-to-Die Manufacturing Variation
    - Transistors use more/less power to switch
    - Leakage current varies
  - Cooling Variation
    - Better cooled processor can run faster
    - Run cooler => less leakage current
  - **EACH CORE RUNS DIFFERENT FREQUENCY**

renci

Performance with Power Limits?

- Systems become slightly performance heterogeneous between nodes
  - Die-to-Die Manufacturing Variation
    - Transistors use more/less power to switch
    - Leakage current varies
  - Cooling Variation
    - Better cooled processor can run faster
    - Run cooler => less leakage current
  - **EACH CORE RUNS DIFFERENT FREQUENCY**
  - **How are HPC applications going to perform?**

renci

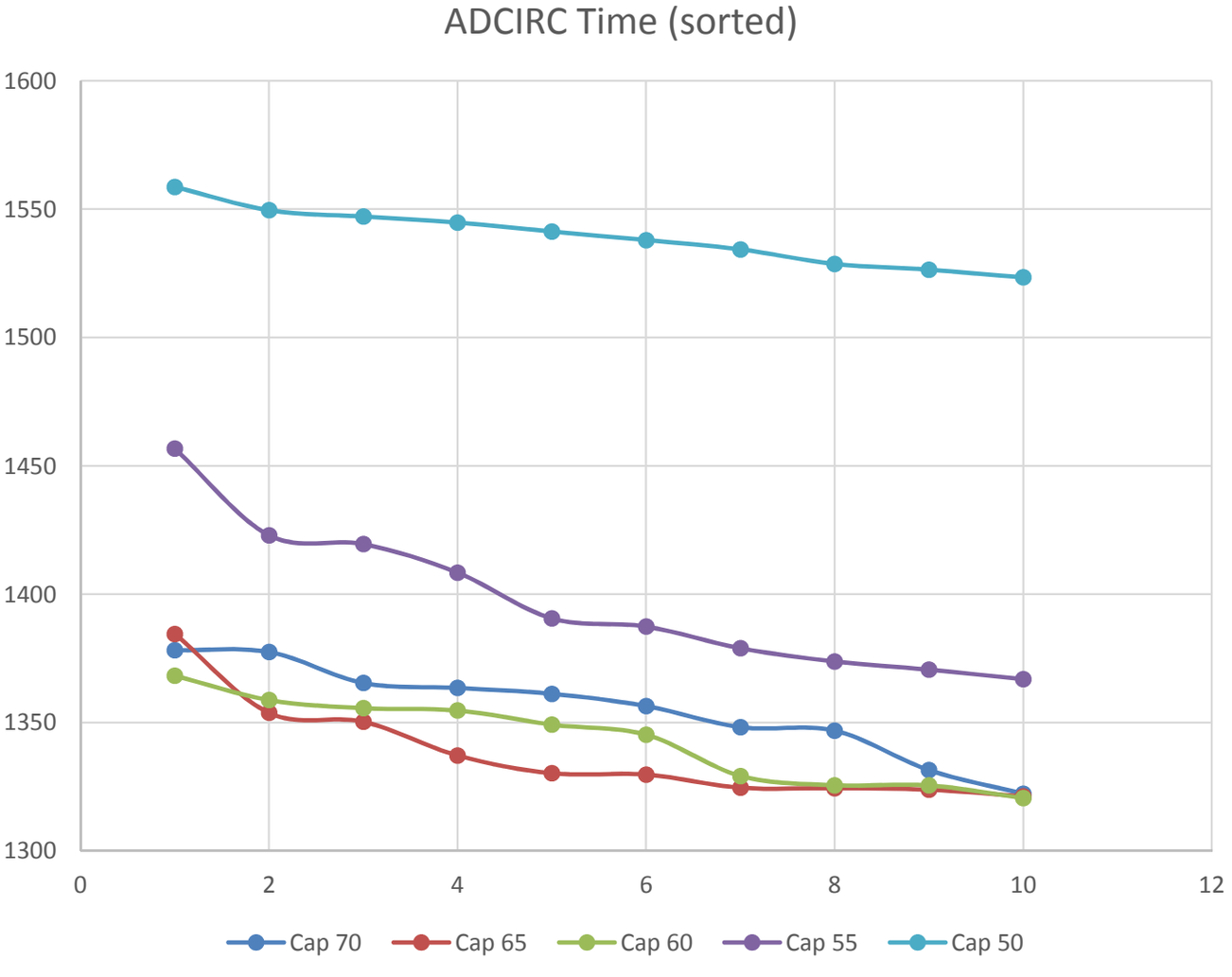
Problems and Execution Environment

- ADCIRC – Storm surge modeling code
- WRF – Standard weather modeling code
- CHROMA – LQCD - t_leapfrog test case

- Dell bladecenter
  - 16 M420 blades
  - 2 Intel Xeon E5-2450 (SandyBridge) @2.1GHz
    - Air flows over one socket to reach the second socket
  - Infiniband interconnect
  - Hyperthreading enabled but only 1 thread per core was used

renci

ADCIRC Variability (16 node execution)

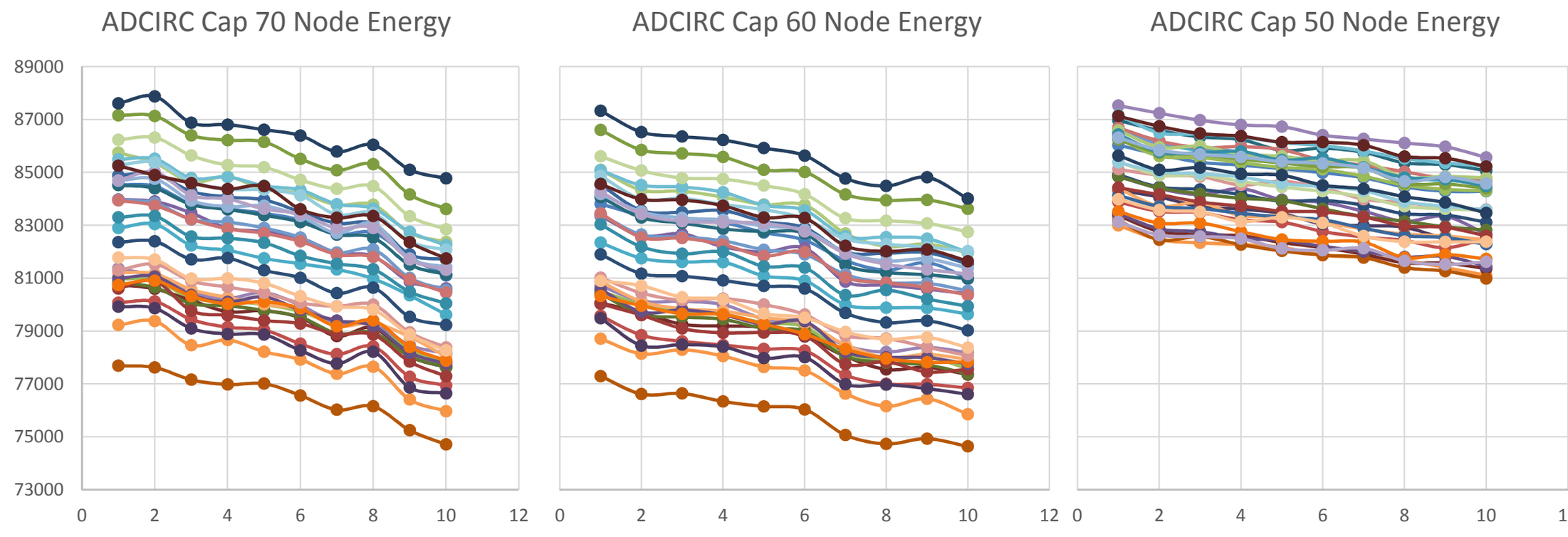


70W Limit - 5% variance

60W Limit - Average +2% - variance unchanged

50W Limit - Average +15% - variance 2%

ADCIRC Node Energy



At 70W ~13% Energy Variation

cores different power for same length of time

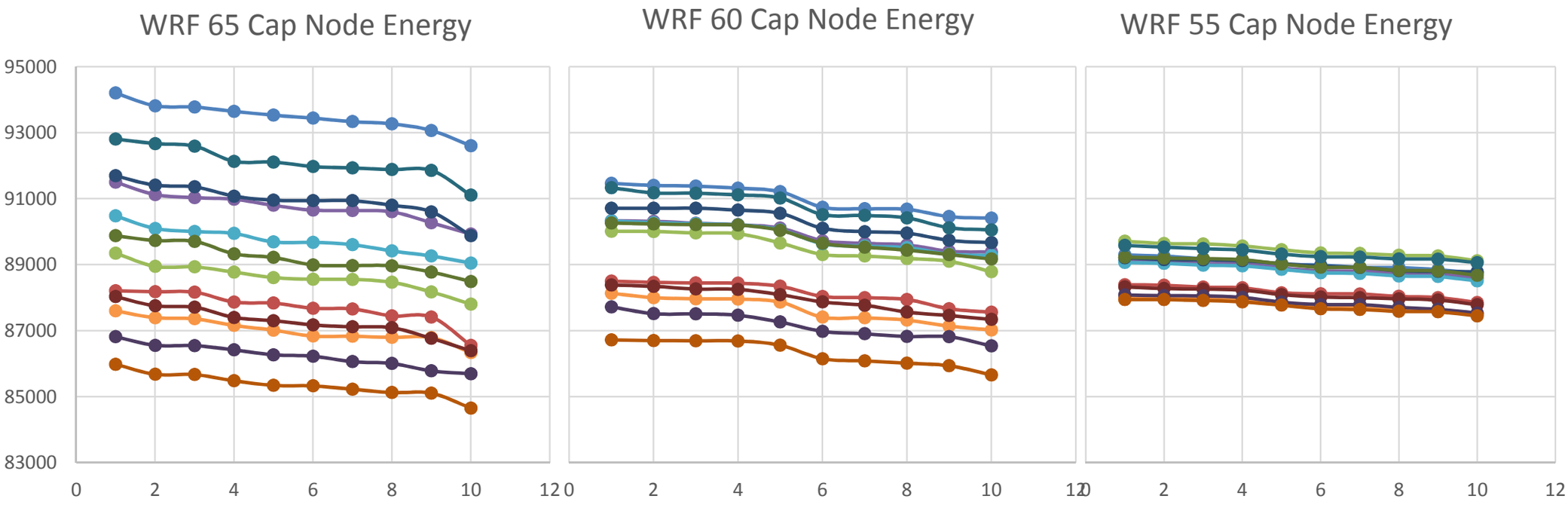
Can see difference in cooling

At 50W ~4% variance – energy usage more consistent

cores using same power for same length of time

renci

WRF Node Energy (6 Node execution)



Energy about the same at all power limits but more evenly distributed with lower power limits

With power limits clear separation between hot and cold sockets

renci

Core Based Energy Management

- All the cores now heterogeneous
  - Effective clock rates differ
  - Clock rates change over time
- Duty Cycle Modulation (Clock Skipping) allows core specific clock frequency control
  - Mechanism to locally control each cores clock rate
  - Haswell extend DVFS to allow this level control
    - Also allows control of Uncore (memory controller)

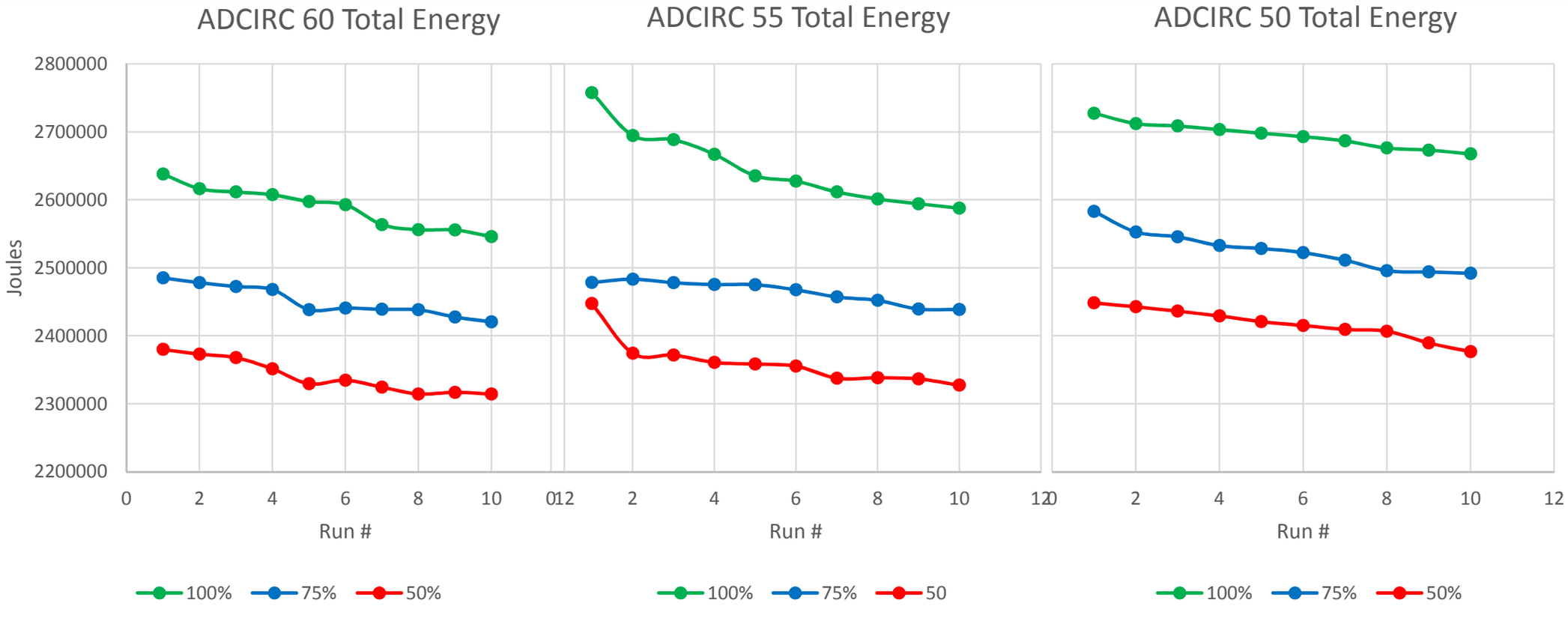
renci

Simple Energy Management

- Threads that arrive at MPI collectives early can be slowed
  - Intercept MPI calls using the MPI profiling interface
    - Pick clock rate for this core until next collective
    - Clock rate based on % time spent waiting
  - All decisions local – no communication overheads
  - Save energy
  - Overall execution time unaffected

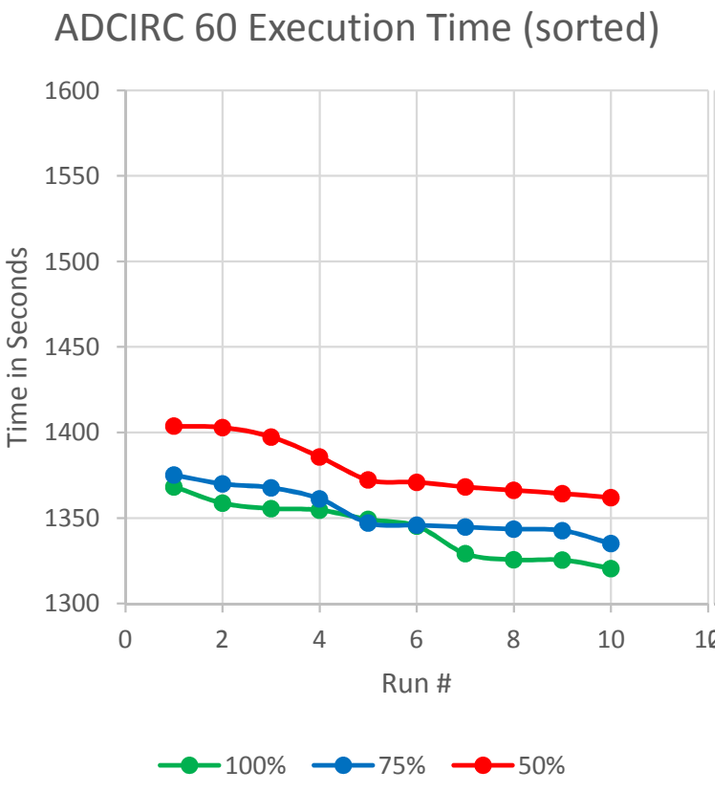
renci

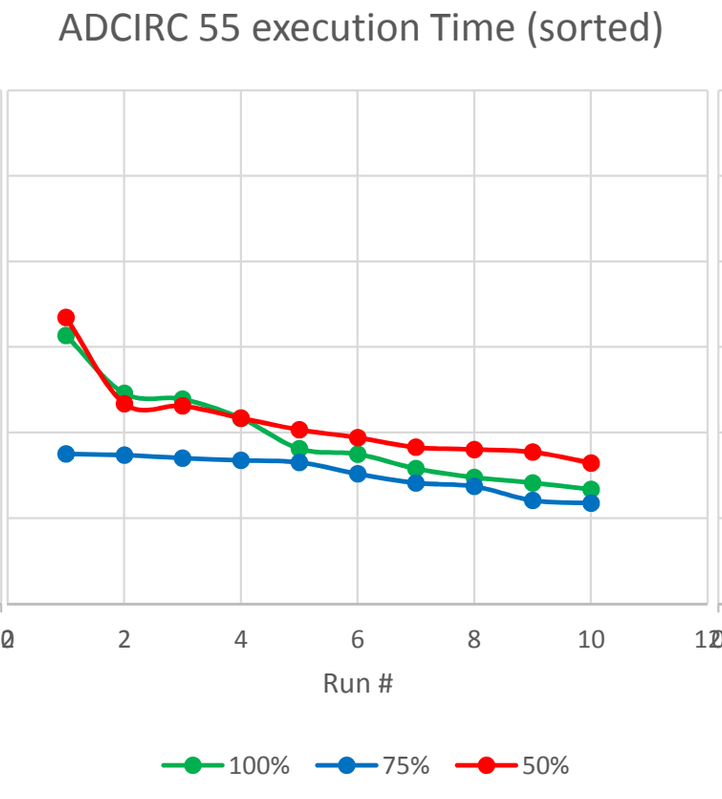
ADCIRC With Policy

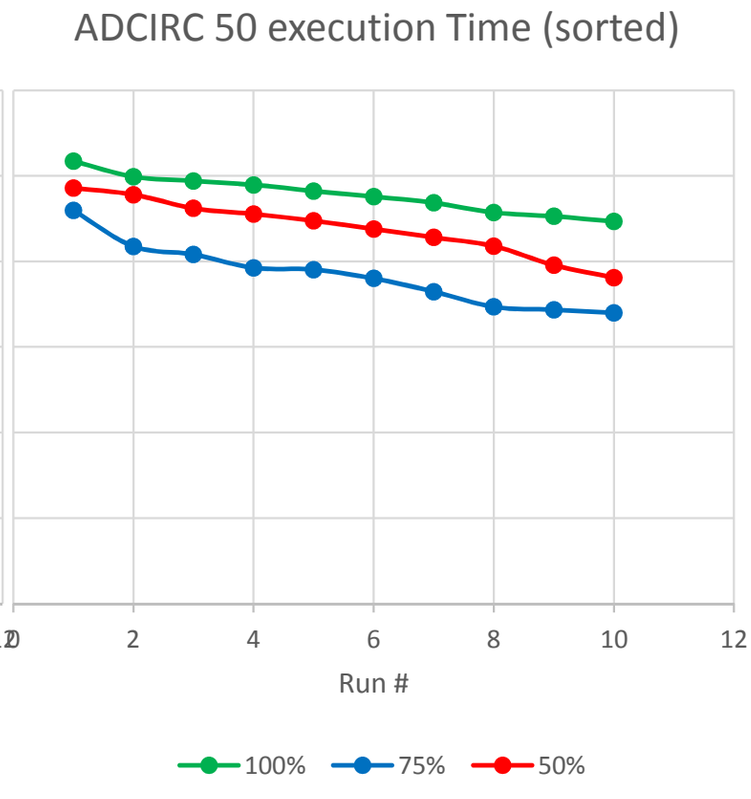


At 50W Energy Management reduces energy ~8%

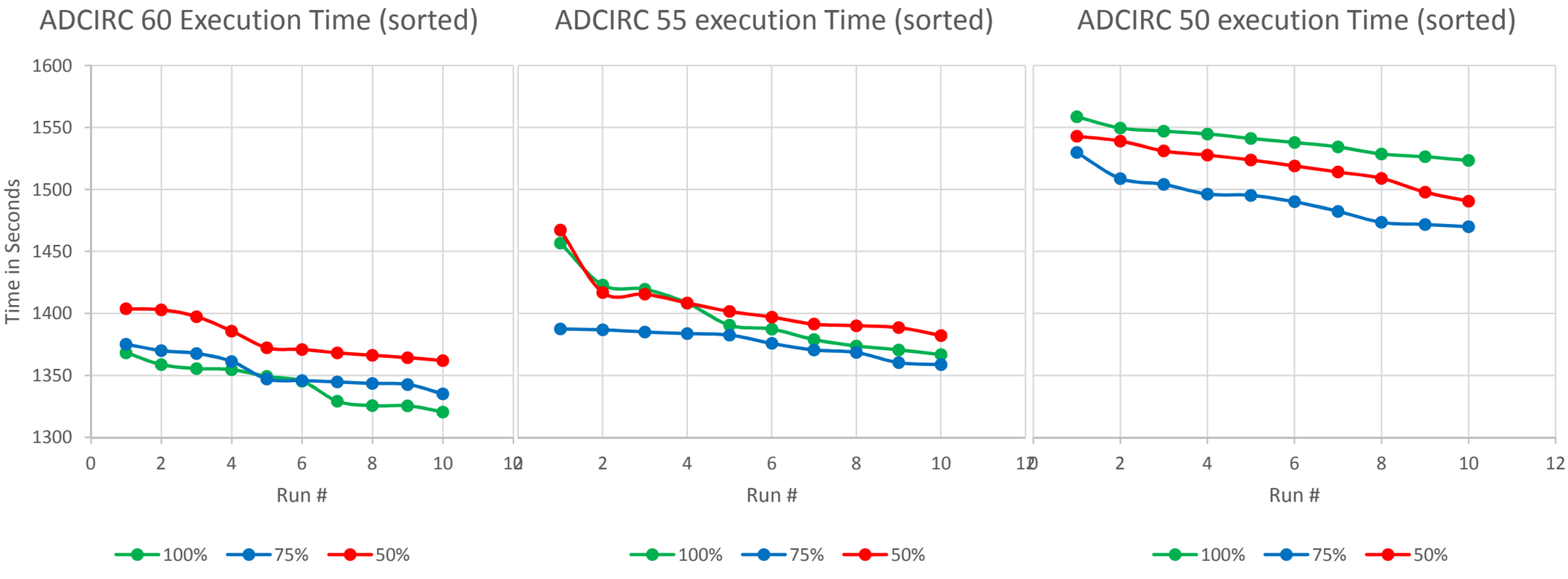
ADCIRC With Policy





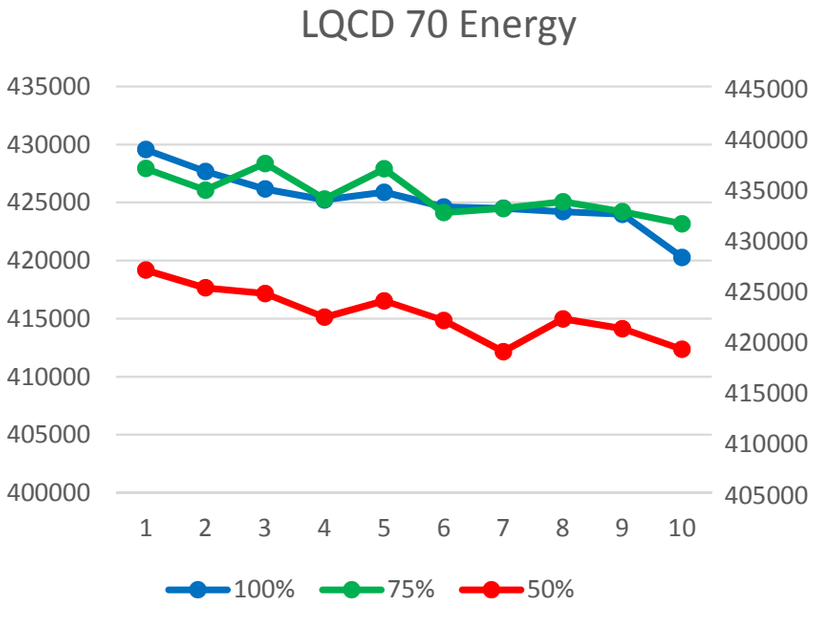


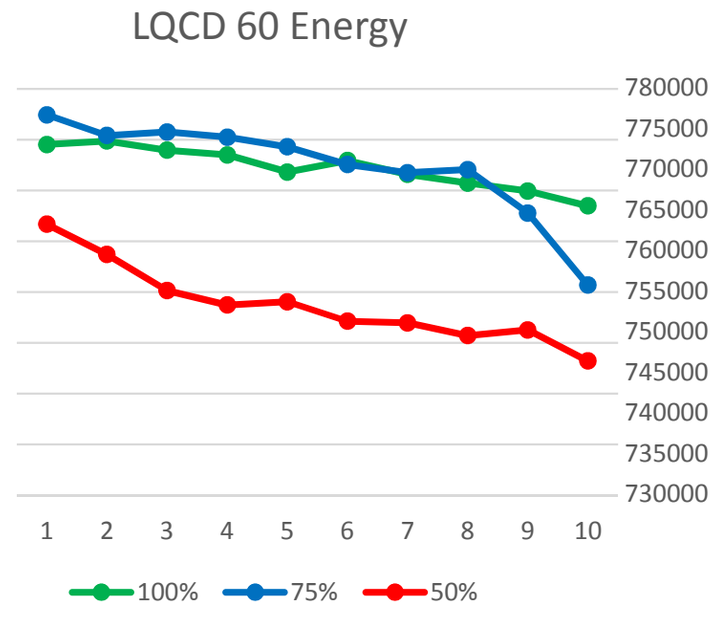
ADCIRC With Policy

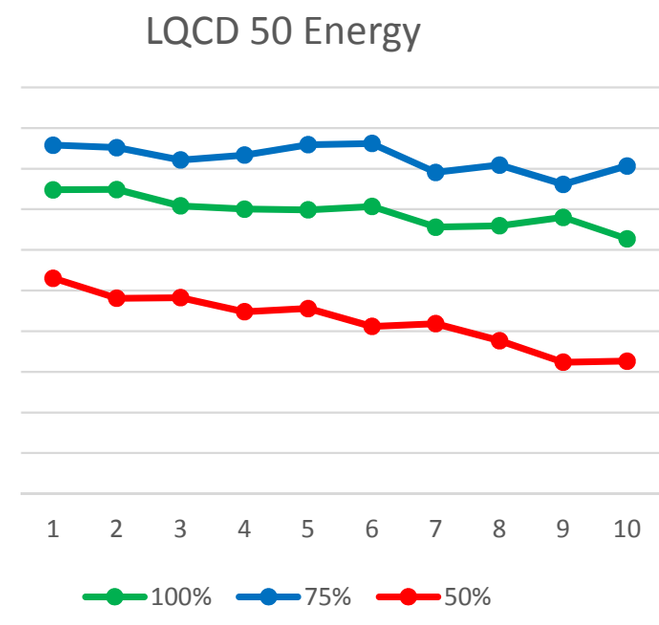


At 50W Energy Management REDUCES time ~5%

LQCD (t_leapfrog) with Policy



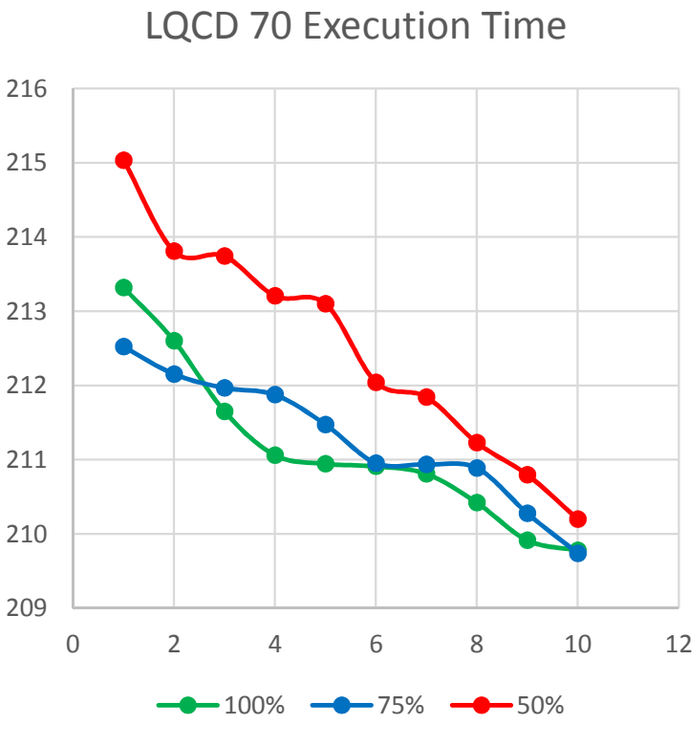


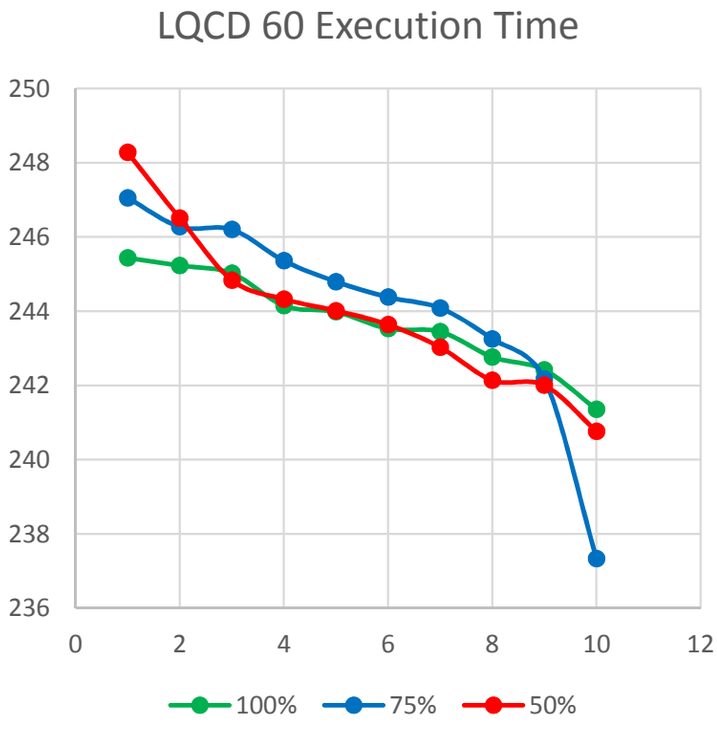


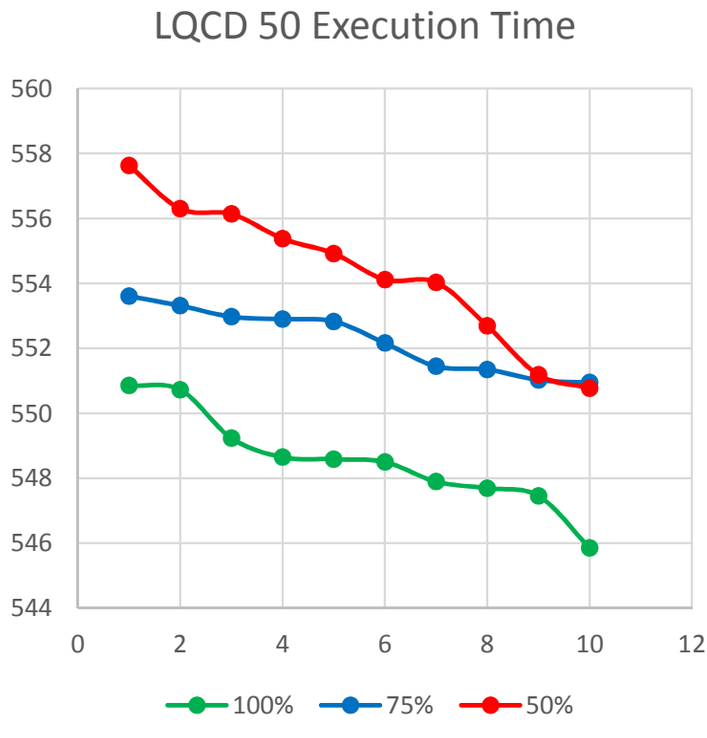
Does not fit into inner caches
Power Limit significant increase time and energy

With minimum speed set at 50% energy savings (~2%)

LQCD (t_leapfrog) with Policy







Does not fit into inner caches
Power Limit significant increase time and energy
Model has little effect at 70 and 60W
Model increases time and energy 1-2 % at 50W

WRF with Policy

- Near zero effect -- model never detected imbalance big enough to justify changing the clock frequency
  - Duty Cycle Modulation comes in 16 steps (6.25%)
  - Safety margin – we don't slow a clock until a thread arrives 12.5% early
  - WRF well balance and no opportunities to reduce the clock rate were detected

renci

Conclusion

- HPC Energy management may make sense for Exascale systems.
  - Can save Energy and more importantly Time
- Low overhead clock rate control can be used by the runtime to reduce energy of memory-bound or slightly imbalanced applications
  - With the introduction of core specific DVFS and Power-limited systems, previous work in Energy Management techniques should be re-evaluated

renci

Questions?