Variability: A Performance Nightmare

Allan Porterfield
RENCI, UNC-Chapel Hill

renci
RESEARCH \ ENGAGEMENT \ INNOVATION

My Interests

Compiler Optimization – improve application performance
Adaptive Runtime Scheduling – reduce application energy demands

Variability make both of these HARD (impossible?).

Chips are different (center of the wafer vs edge)
Temperature changes both time and energy required
BIOS settings change time and energy required
Random latencies from other system load (network, file …)
Butterfly effect on adaptive algorithms

renci

Autotuning Problems

Run #	nofuse	nofuseX16	nofuseX32	nofuseX64
1	5.333394	5.147411	5.086881	5.304340

Autuning results from Jacobi-2D from PolyBench
Dell M620 with 2 Intel E5-2680 @ 2.7GHz
best 4 results from over a thousand compiler configurations

Looks like tiling improves cache utilization until a cache size is exceeded at which point it falls almost 6%

renci

Autotuning Problems

Run #	nofuse	nofuseX16	nofuseX32	nofuseX64
1	5.333394	5.147411	5.086881	5.304340
2	5.144302	5.077856	5.125735	5.138848
3	5.369150	5.345376	5.203415	5.314528
4	5.037441	5.115155	5.179577	5.340713
5	5.408004	5.333346	5.156341	5.083266
6	5.242719	5.217077	5.373121	5.022142
7	5.055487	5.156706	5.199737	5.084126
Average	5.22721	5.19899	5.18926	5.18399

But on multiple runs a different story appears
– Original answer sub-optimal

renci

Better Autotuning?

Make one pass of over all of the options (may be thousands)

Identify the fastest options (within 10%(?) of the absolute fastest)
(hopefully single digits versions)

Rerun subset 10+ times to find best average

renci

Scheduling problem

ADCIRC – storm surge simulation
- synchronous SPMD with global synchronization each time step
Test on 6 Nodes Intel E5-2450 @ 2.1 GHz
10 runs on each region – same input used for all tests
Test ran approximately 1 hour – about 20% of production run

Nodes	Minimum	Maximum	Average	Range	Slowdown
0-5	3419	3898	3523 (3481)	14(3.8)%	n/a
6-11	3495	3594	3553	2.8%	2.1%
12-17	3614	3743	3677	3.5%	5.6%
18-23	3434	3597	3500	4.7%	0.5%
24-29	3489	3573	3529	2.4%	1.3%

Average and Slowdown ignore one very slow test on region 0-5

renci

Better SPMD Scheduling?

Rank nodes within a cluster according to their “slowest” processor

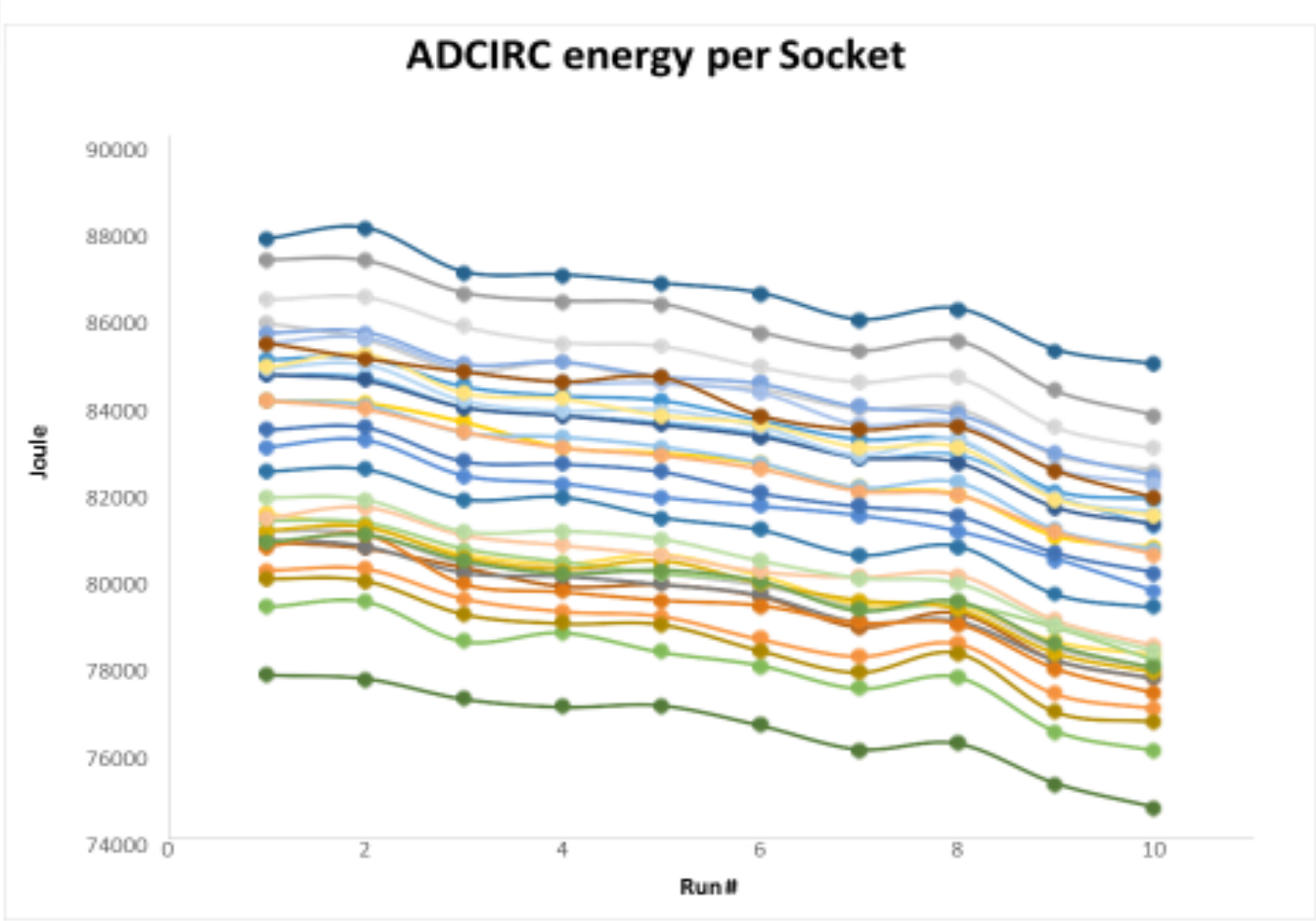
- for Intel processors – probably the amount of time a node spends in TurboBoost

Try to schedule jobs on nodes on groups of nodes.

- ignores network demand issues which can outweigh SPMD synchronization delays

renci

Energy Usage Problem



ADCIRC – storm surge simulation
16 Nodes of Intel E5-2450 @ 2.1 GHz
10 identical executions – sorted by time
>10% between chips

renci

Better Energy Usage?

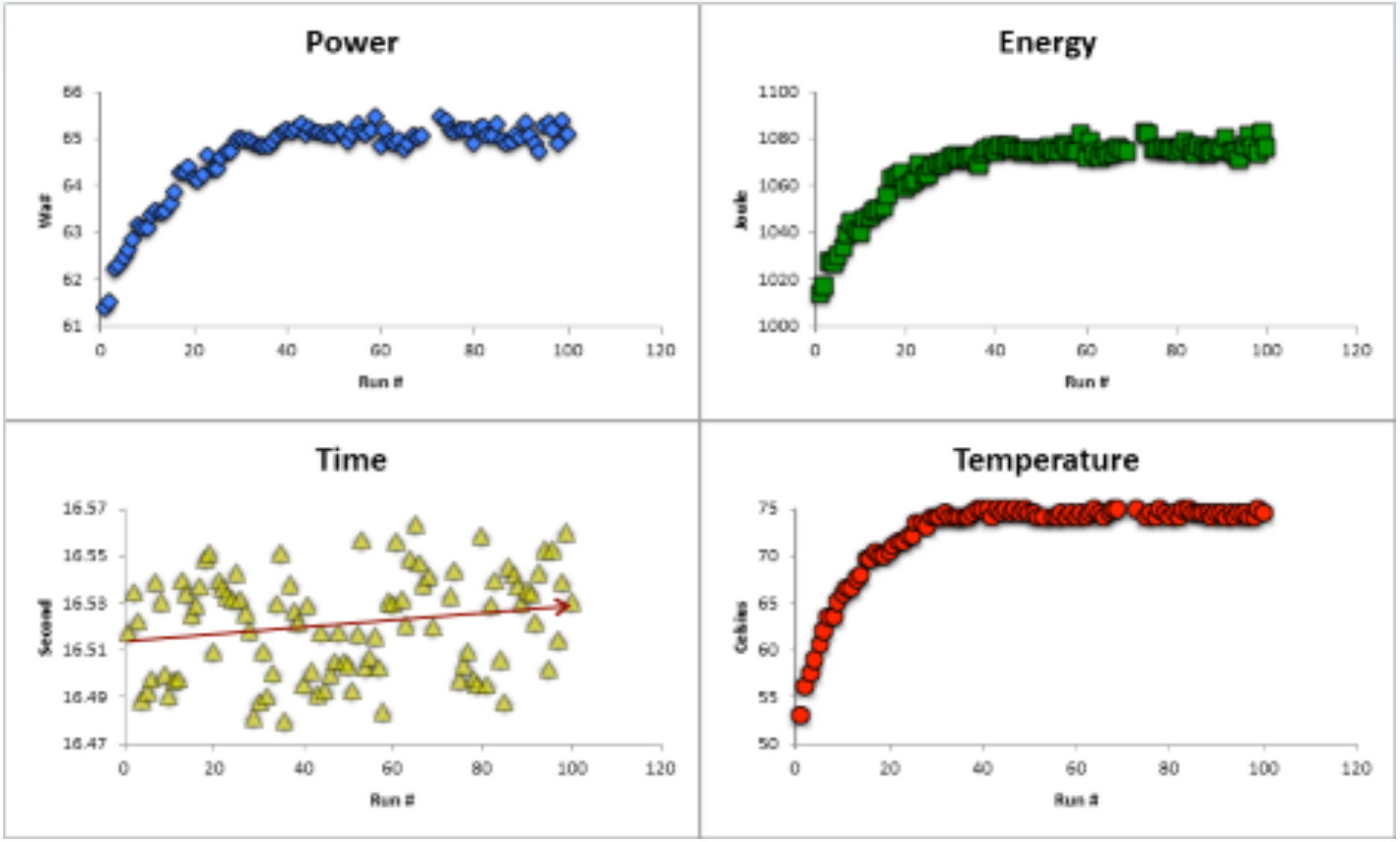
Even socket energy demands – by reducing use by ‘expensive’ sockets

Schedule less work

- idle one or more cores
- use DVFS/DCM to reduce clock rate of cores

renci

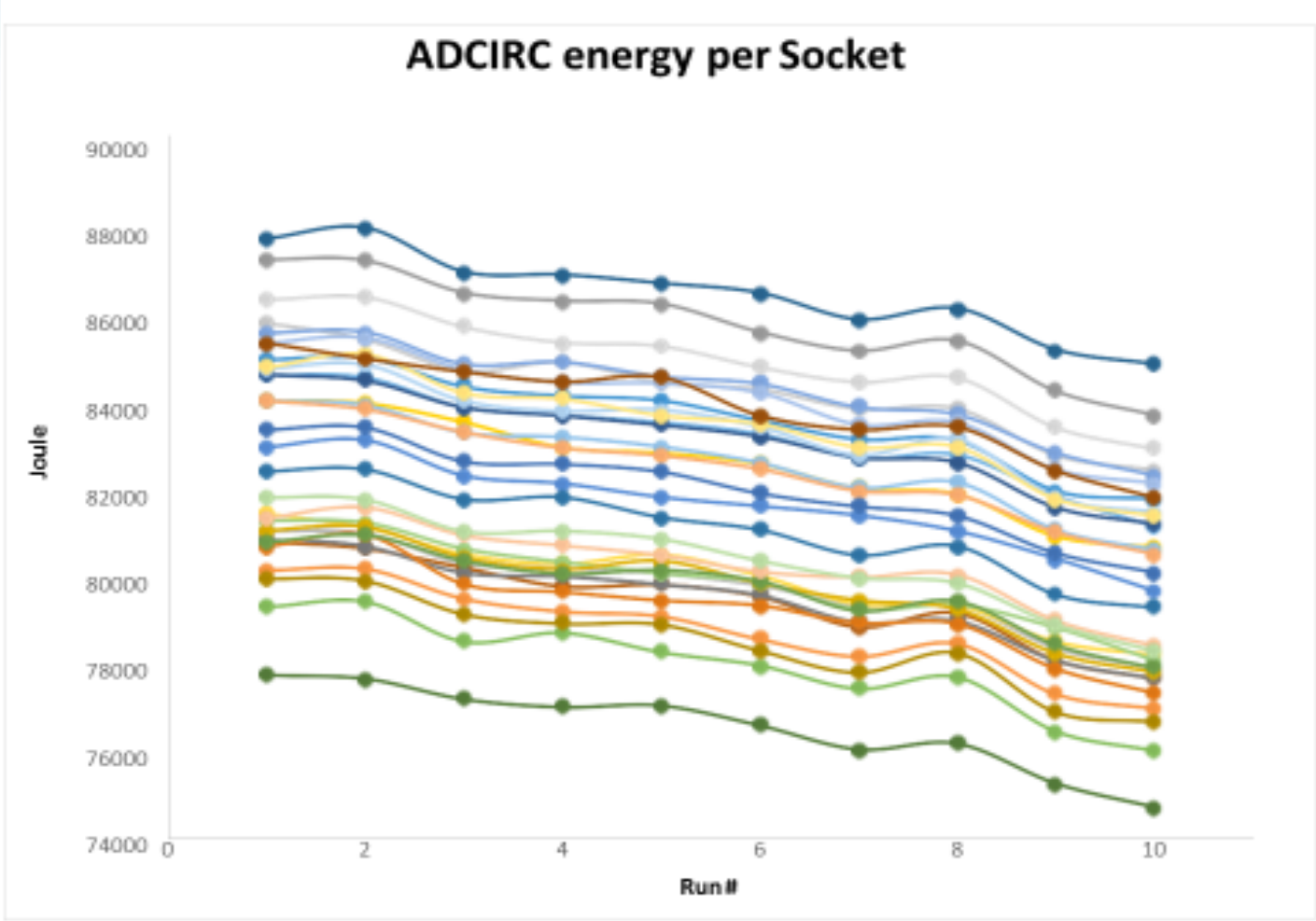
Problem with Understanding Results



100 executions (in order) of HPCCG
2 socket Intel E5-2650 @2.3GHz

Which run used?

Trusting Energy Results



System design evident
-- Socket 0 blue-brown lines
-- Socket 1 yellow-green lines
Air blows over Socket 1 then Socket 0, which runs measurable hotter

renci

Trusting Results

100 runs of test Castro AMR execution on 2 different days
input- inputs.2d.cyl_in_cartcoords
4 Nodes Intel E5-2450 @ 2.1 GHz
Same nodes, executable and inputs used both days

Date	Minimum	Maximum	Average
4/18/2016	18.56	34.51	25.11
4/19/2016	16.51	19.13	17.71

Differences
- Slurm Queue 4/18 ~4000 run as root
  - basically every core in the other nodes of the bladecenter being used executing a Genetics workflow
- Slurm Queue 4/19 ~680 run as user
  - only half of nodes in bladecenter busy executing ADCIRC

During debugging could repeat once but vanished during efforts to understand the cause

renci

Easier to trust results?

Papers need to describe execution environment

- Temperature
- Other workload on system
- …

Graphs that show the variance

- Candlestick, Whisker …

renci