



Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by **Battelle** Since 1965

Modeling the Performance and Energy Impact of Dynamic Power Steering

KEVIN J. BARKER, DARREN J. KERBYSON
PACIFIC NORTHWEST NATIONAL LABORATORY

Modeling and Simulation

August 12, 2015. Seattle, WA

▶ Trends in systems

- Restrictive power budgets – possible that not all architectural components may be active at full capability simultaneously
- Fine-grained power measurement and allocation – codes can closely monitor and modify power consumption characteristics
- Default mode of execution may be “throttled down” leaving performance on the table

▶ Trends in applications

- Adaptivity and asynchrony
- Input-dependent execution cannot be optimized in advance
- Evolving computation leads naturally to dynamic load imbalance

Can these trends be exploited to improve performance without negatively impacting power consumption?



Power saving mechanisms tend to be local

- ▶ Most energy saving mechanisms rely on exploiting *slack*
 - Down-clocking under utilized resources
 - DVFS is available mechanism

- ▶ Some use predictive models to determine forthcoming slack and duration periods
 - Energy Templates: use of per-core micro-models

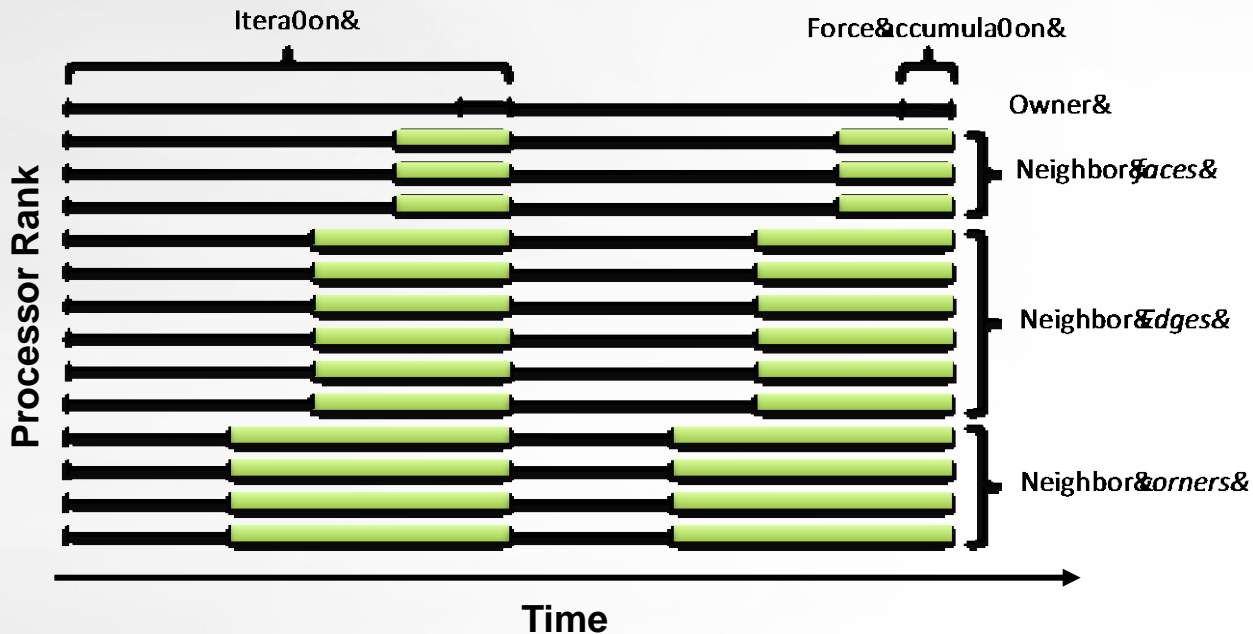
- ▶ In power constrained systems a more global view is needed:
 - Which parts of the system to down-clock to satisfy the power-cap
 - At a socket level (overcome dark-silicon),
 - At a rack level (power distribution)
 - At a system level (machine room constraints)



Energy Templates

- ▶ Expression of complex activities
- ▶ Use per core model to determine when savings possible
- ▶ Run-time uses DVFS and/or to idle-core
- ▶ Template interfaces between application and hardware

Example: ARGOS MD code, parallelized over cell-cell interaction pairs results in input-dependent load imbalance



“Energy Templates: Exploiting Application Information to Save Energy”,
Kerbyson, Vishnu, Barker, IEEE CLUSTER, 2011.

Dynamic power steering

Concept: Route power to those resources that are over-loaded and away from under-loaded resources to compensate

- ▶ Optimizes power consumption in two ways:
 - Leaves data in place – minimizes power lost to data migration
 - Routing available power to where the work is – *Power Balancing*

- ▶ Targeting workloads
 - In which static calculation of ideal power distribution is not possible (e.g., data-dependent execution, variation over time)
 - In which performance is impacted by changes to node or core p -state (i.e., by allocated more power, performance may be improved)

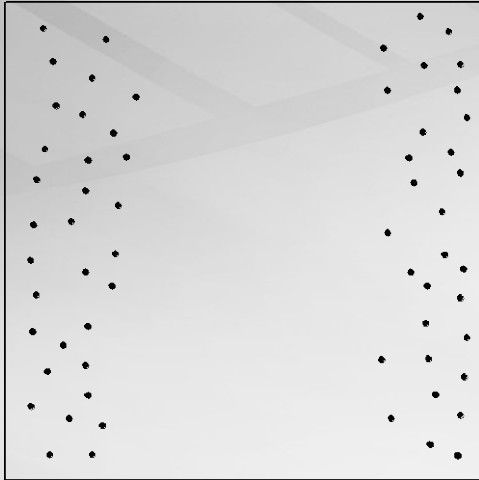
Key challenge: understanding how application characteristics impact effectiveness of dynamic power steering strategy

Example: Charged Particles within electric field

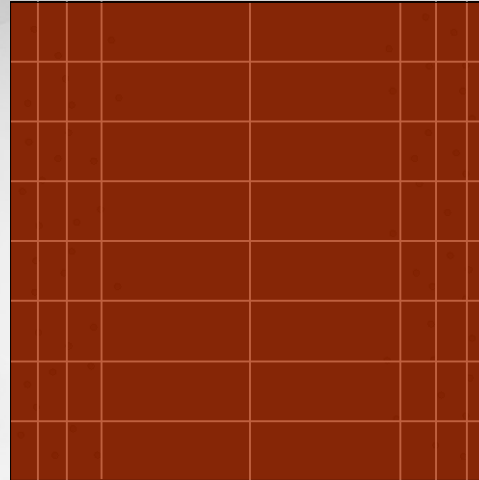


Pacific Northwest
NATIONAL LABORATORY

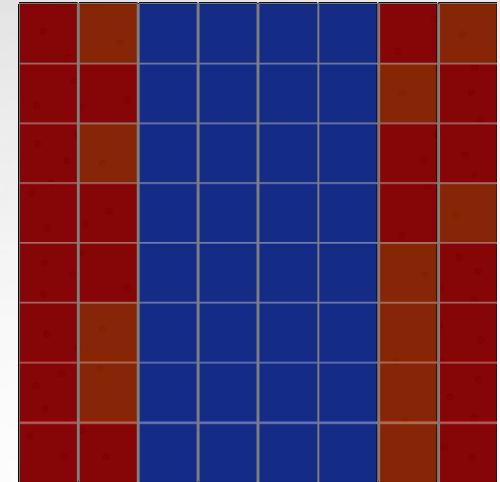
Proudly Operated by **Battelle** Since 1965



Example: Non-uniform distribution of particles (work)



Load-balancing of particles:
Each sub-grid contains ~equal particles (work) & uniform power distribution



Dynamic Power Steering:
Particles left in-place (no data movement), power allocation is optimized

- ▶ Temporally varying load imbalance due to charged particle movement

Traditional approach: Load-balance particles over processors

Power-Steering: particles left in place & power-balance over processors



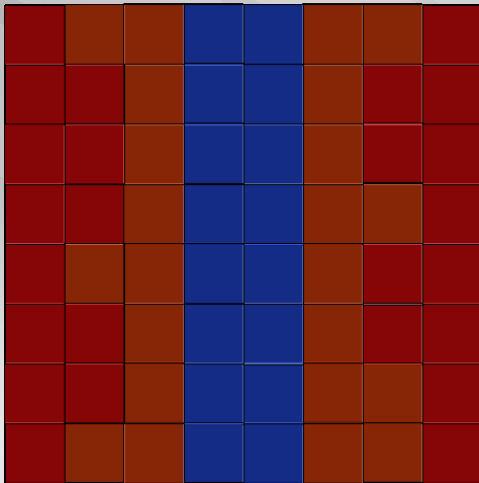
Focus on exploring the possibilities of Dynamic Power Steering

- ▶ Need for Emulation: no power constrained system was available for our study
 - Mimic a power cap on a current system which is lower than the normal operating power.
 - Allow for core p -state to vary up or down using Heuristic
 - Overall power is constrained to be that of initial operating point
 - Improve performance along critical path
- ▶ Test-bed platform:
 - 36 nodes dual-socket, 8-core AMD Interlagos processors
 - Power measurement @ 0.3Hz sampling rate (Outlet based)

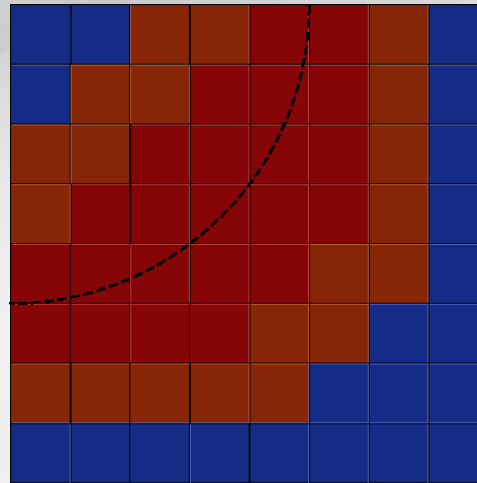
Frequency (GHz)	Core Active Power (W)
2.1	21.1
1.7	18.0
1.4	15.6

$$P_{constraint} = CP_{base} \geq \sum_{i=1}^{N_{P-states}} C_i P_i$$

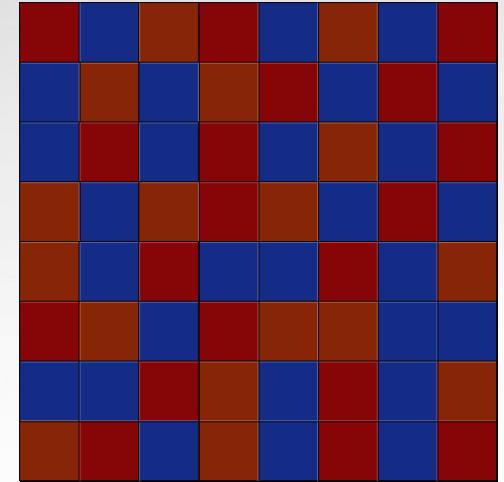
Three Synthetic workloads



Charged Field



Wavefront



Random

- ▶ **Charged Field**: particle positions vary over time due to application of electric field
- ▶ **Wavefront**: quadrant of circular wavefront propagates from corner of global grid
- ▶ **Random**: control case – work load levels are assigned randomly
- ▶ Variation in Computational Intensity & Load-imbalance

Power assignment heuristic

Start

1. PWR_{max} = maximum globally available power
2. $p\text{-state}_{max}$ = highest performance $p\text{-state}$
3. $N_{work_max} = \max(N_{work_i}) \forall i \in \{P_i\}$
4. $t_{work_max} = N_{work_max} \times t_{work}(p\text{-state}_{max})$
5. $\forall i \in \{P_i \mid P_i \neq P_{work_max}\}$ find slowest $p\text{-state}$ such that $t_{work_i} < t_{work_max}$
6. $PWR_i = t_{work_i}(p\text{-state}_i)$
7. $PWR_{global} = \text{SUM}(PWR(p\text{-state}_i))$
8. If $PWR_{global} > PWR_{max}$ then reduce $p\text{-state}_{max}$ and go to step 3
9. Assign $p\text{-state}$ calculated to each processor-core

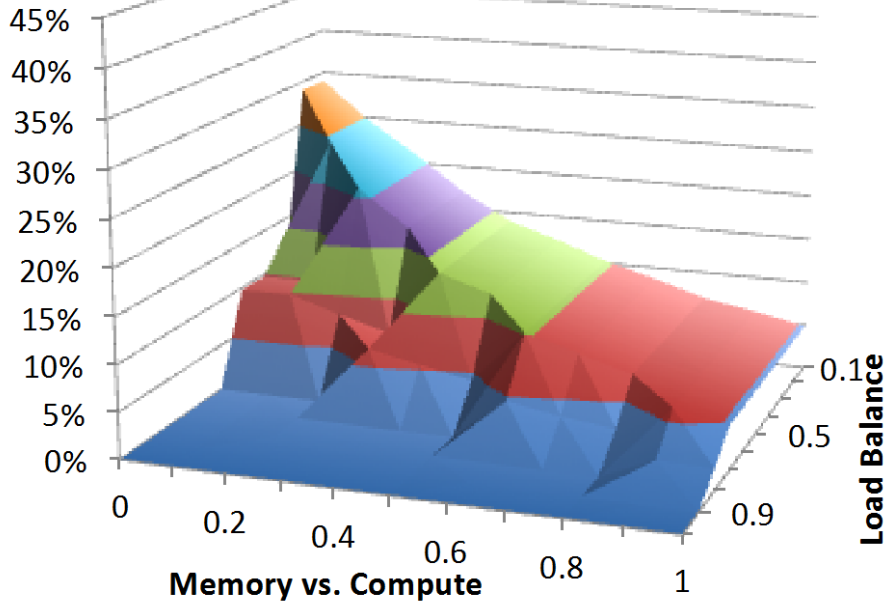
End

Assign highest-performing $p\text{-state}$ to cores with heaviest load, and then assign the lowest $p\text{-state}$ to all others such that there is no increase in execution time

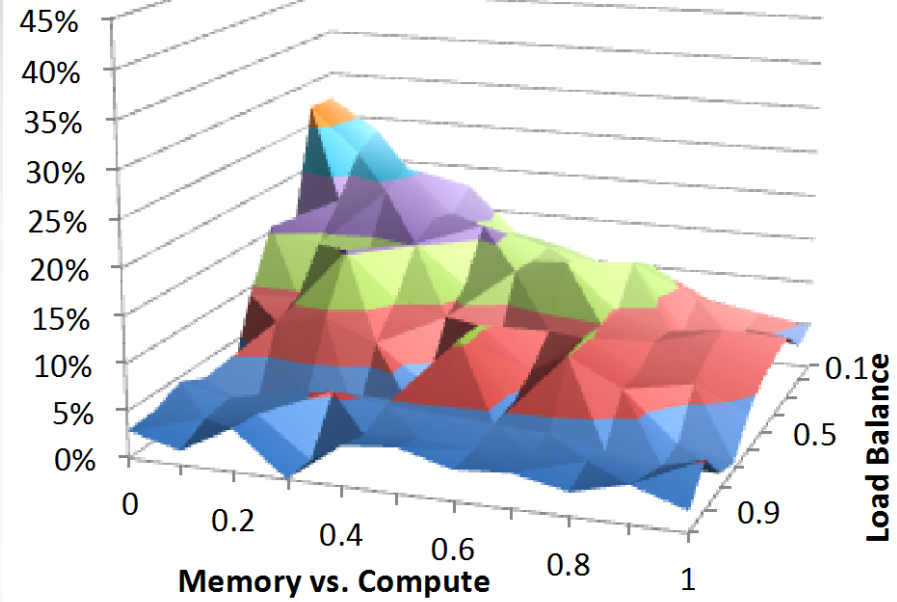
“On the Feasibility of Dynamic Power Steering”, Barker, Kerbyson & Anger, Energy Efficient Supercomputing (E2SC), SC’14, 2014.

Results: Charged Field workload (runtime)

Runtime Improvement

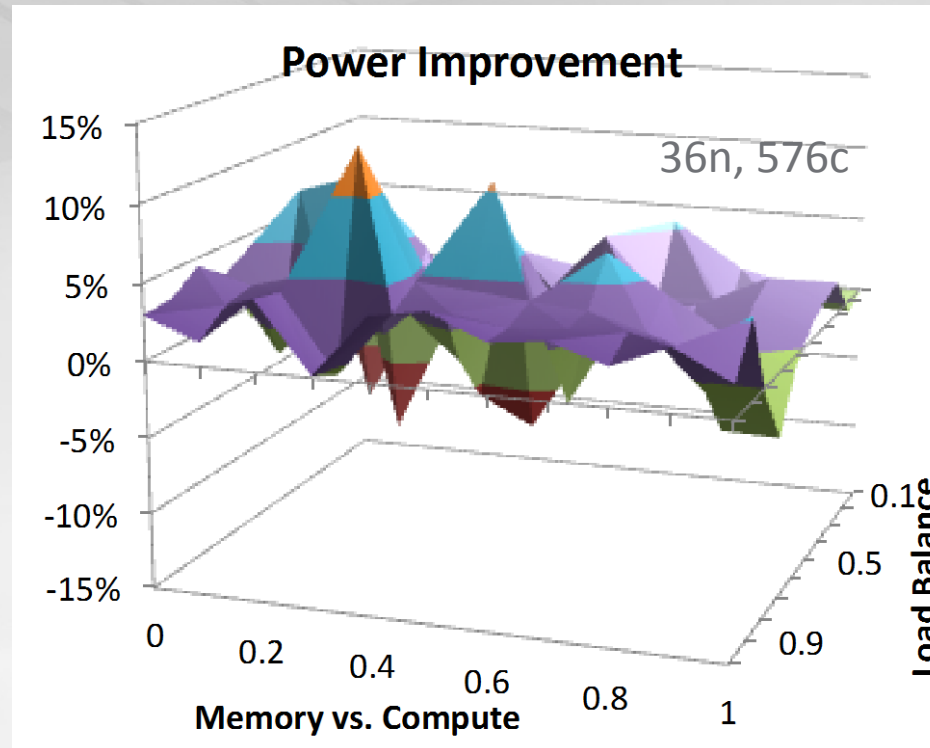


Energy Improvement



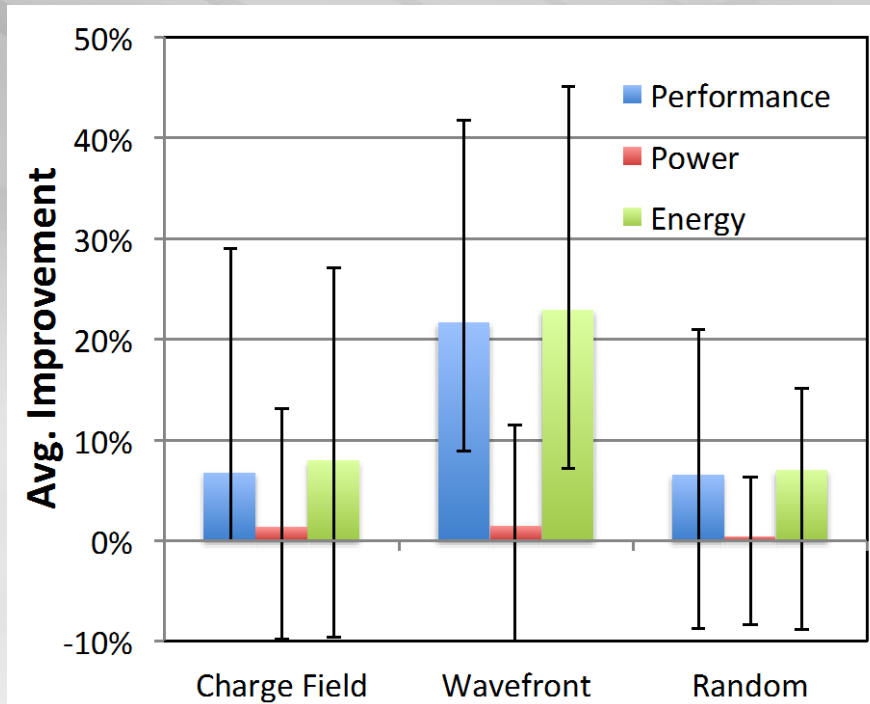
- ▶ Run-time improves as critical path has more power applied
 - Greater impact when compute than memory bound
 - Greater impact as load-imbalance increases (balance decreases)
- ▶ Up to 27% energy savings obtained compared with operating point

Results: Charged Field workload (power)



- ▶ Aim to keep power at the power cap
 - Due to quantization we mostly see a reduction in power use from the operating point

Summary of results



Average (and min/max) performance, power, and energy consumption results for all three workloads over the range in compute intensity, load-imbalance, and time-step.

- ▶ Performance is improved in all cases.
- ▶ Slight improvements in power consumption,
- ▶ Results in slightly greater improvements in overall energy efficiency.
- ▶ Wavefront exhibits greater improvements as a degree of load imbalance persists in all cases.

Conclusions

- ▶ Exploration of Power steering has shown that power-balancing could replace load-balancing in a power constrained system
 - Provide more power to processor-cores with more work
 - Leave work in place (no load-balancing)

- ▶ Impact of dynamic power steering will increase with system scale

- ▶ Work in progress to use modeling to explore full potential of power-steering
 - Socket-level, rack-level and system-levels
 - CESAR co-design center exploring applications under development
 - Possible impact on workflows with wide-area distribution

Acknowledgements

- ▶ Advanced Scientific Computing Research (ASCR) of the US Dept. of Energy
 - Beyond the Standard Model (BSM)
 - Performance Health Monitoring (PHM)
 - DOE CESAR Exascale Co-design Center

- ▶ Advanced Computer Systems Research Program (ACS)