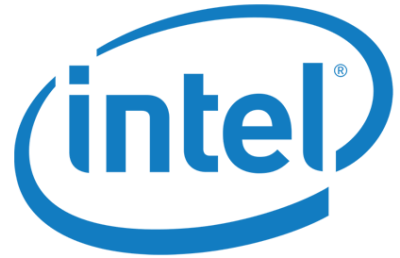
# Boosting Power Efficiency of HPC Applications with GEOPM




Jonathan Eastep [jonathan.m.eastep@intel.com]
Principal Engineer and PhD
29 August 2018

# Outline

- Challenges and Approach to Solving Them
- GEOPM Architecture, Use-Cases, and Deployments
- GEOPM Experimental Evaluation
- GEOPM Work in Progress and Future Work
- Takeaways and Call to Action

# Challenges Motivating Power R&D

- Original motivator for GEOPM was improving power efficiency for Exascale systems, but scope has grown to include current systems
- Exascale: US DOE set a target of 1 ExaFLOPs within ~45 MW by 2021
- With only traditional scaling techniques, facing 2-3x efficiency gap
  - Manufacturing process technology advances
  - Integration of HW components
  - Architectural advances
- We anticipate that no single silver bullet solution can close this gap

# Implications of Power/Energy Challenges

- Advances needed in multiple dimensions: architecture, power delivery, software, and power management
- Focus of this work: rethinking technologies for power management
  - Historically, power management was largely the responsibility of the HW/FW
  - Historical techniques waste significant power for a given level of performance
    - Node-local and lacking in application awareness (oblivious to impact of performance variation across nodes on overall performance in BSP applications, oblivious to phases)
  - Move toward a solution including SW layers of power management
  - SW provides global application-awareness and leverages existent (or enhanced) HW controls to guide HW to better decisions

# GEOPM Solution


- GEOPM = Global Extensible Open Power Manager
  - New software runtime for power management and optimization of HPC jobs
  - Adds scalable, application-aware layer to system power management
  - Community collaborative open source project, started and supported by Intel
  - Project page: https://geopm.github.io/
- Analyzes the application for patterns then coordinates optimizations to HW or SW control knob settings across compute nodes in a job to exploit those patterns
  - Feedback-guided optimization leveraging lightweight profiling of the application
  - Example knobs: node power budgets, processor core frequencies
  - Example patterns: load imbalance across nodes, distinct computational phases within a node
- Promises to increase performance or efficiency by 5-30% on current systems
  - Mileage varies depending on workload and what controls/monitors are available in the HW
  - See ISC'17 paper by Eastep et al. for experimental data (see later slides for summary)

# GEOPM: An Open Platform for Research

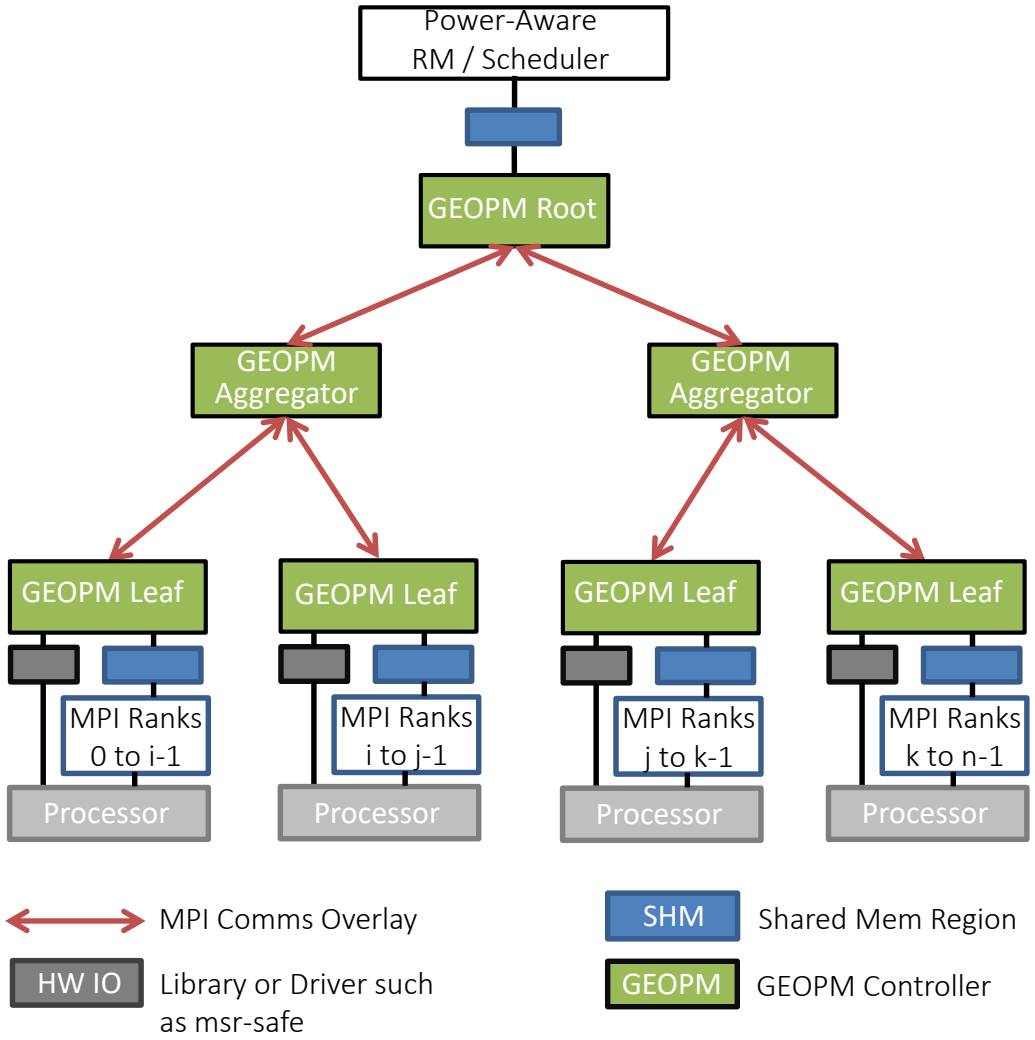
- Another goal is providing a highly extensible, open platform suitable for community research on SW power/energy optimization
  - Goal: accelerate innovation by aligning community on common SW framework for this type of research
  - Truly open: non-sticky BSD license and simple porting via plugin architecture
  - Extend GEOPM to explore new optimization strategies via ‘Agent’ plugins
  - Extend GEOPM to target new control knobs or HW platforms via ‘IOGroup’ plugins

# Outline

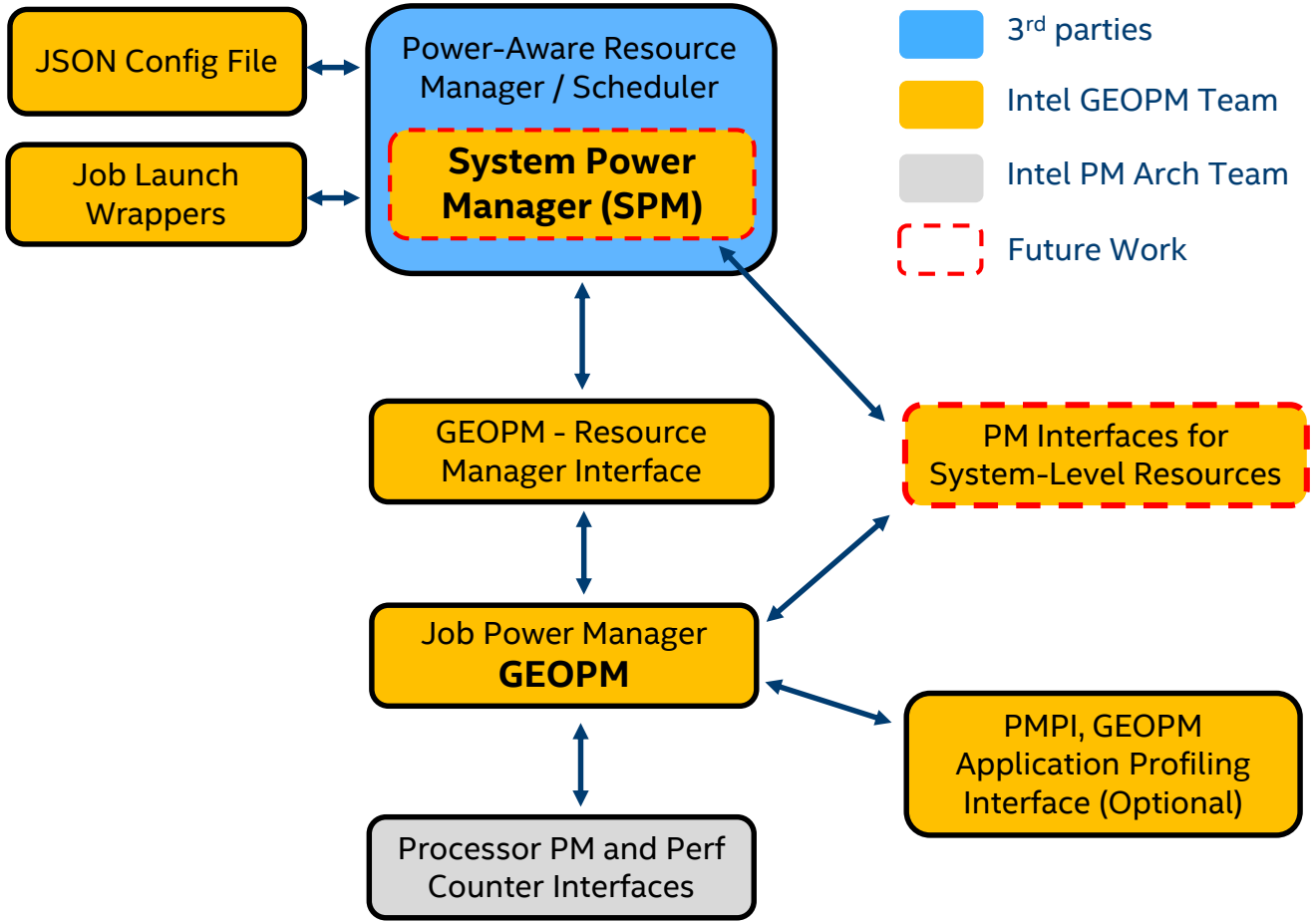
- Challenges and Approach to Solving Them
- GEOPM Architecture, Use-Cases, and Deployments
- GEOPM Experimental Evaluation
- GEOPM Work in Progress and Future Work
- Takeaways and Call to Action

# GEOPM: Hierarchical Design & Comms

- GEOPM = job-level runtime that coordinates tuning across all compute nodes in job
- Scalability achieved via tree-hierarchical design and decomposition
  - Tree hierarchy of controllers
  - Each controller coordinates with parent and children via recursive control and feedback
  - Controller code is extensible via 'Agent' plugins
- Implementation info
  - Access to HW controls achieved via drivers/libs
  - Application profile data collected via PMPI and optional programmer API over shared memory
  - Controllers run in job compute nodes; preferred mode runs them in a core reserved for the OS
  - Controller tree comms currently use in-band MPI (this is easy to modify if desired)



# GEOPM Interfaces and HPC Stack Integration



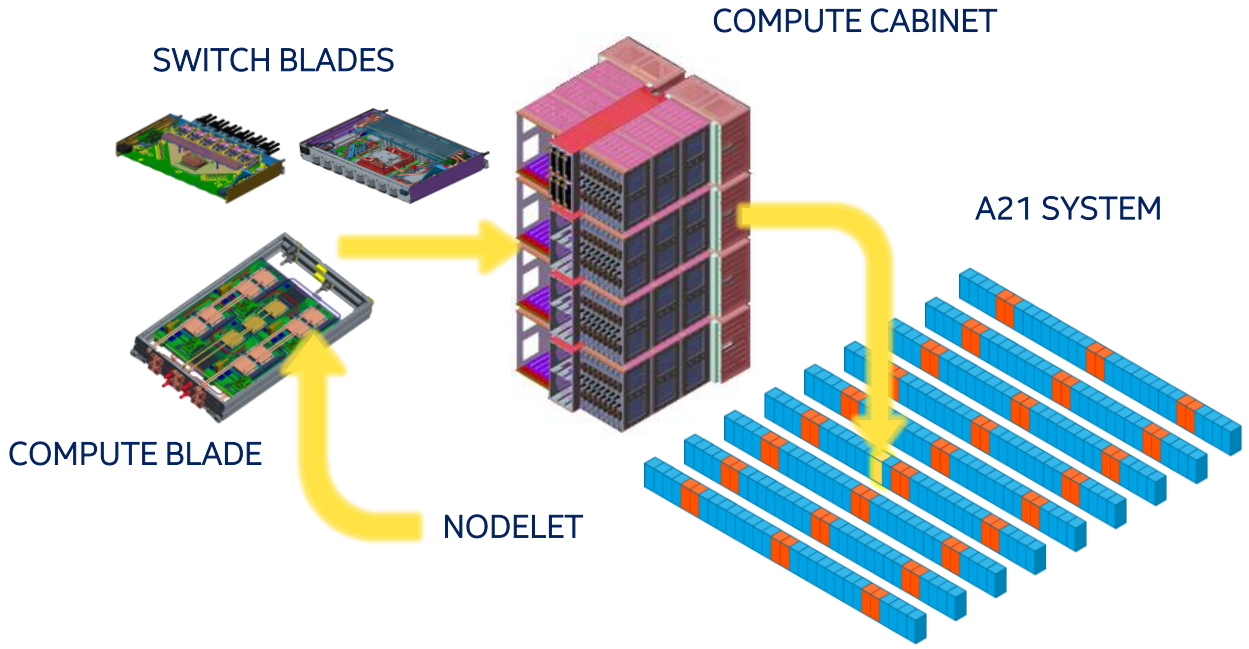

- Long-term: GEOPM sits underneath SPM layer of power-aware resource manager
  - SPM and GEOPM work together for full-system mgmt, talk via RM interface or JSON config file
  - Based on site policy, SPM decides power budgets (or other constraints) for all jobs
  - Based on site policy, SPM decides what 'Agent' optimization plugin GEOPM will use for a given job (e.g. maximize performance or energy efficiency)
  - One instance of GEOPM runs with each job and enforces the power budget (or other constraints) while carrying out the SPM-desired optimization
  - Exploring models where GEOPM runs w/ all jobs
- Near-term, GEOPM is standalone and opt-in
  - Power-aware RMs are still under development
  - Job-level management only: SPM not present to coordinate GEOPM configurations across jobs
  - User requests GEOPM and selects 'Agent' plugin when queuing jobs. We provide wrappers around popular job queue tools (e.g. aprun and srun) which intercept the GEOPM-specific options
  - User configures GEOPM 'Agent' w/ JSON config file

# GEOPM: Highlighting Simpler Use-Cases

| Agent | Use-Case | How To |
|---|---|---|
| General Users | Optimize workload energy or performance | Use GEOPM's built-in optimization capabilities |
| Developers | Tune up application or library code | Use GEOPM's reporting capabilities to characterize runtime and energy of your app or its phases |
| Admins, Researchers | Monitor job or system statistics or trace GEOPM's settings of HW control knobs | Use GEOPM's reporting and tracing capabilities |
| Researchers, Vendors, Integrators | Tailor optimization strategies to a specific HPC Center or applications | Extend GEOPM optimization strategies by adding 'Agent' plugins |
| Researchers, Vendors, Integrators | Port GEOPM to support new vendor HW platforms or new HW controls + monitors | Port GEOPM by adding 'IOGroup' plugins |
| Vendors, Integrators | Codesign HW, SW, and FW | Codesign GEOPM plugins + HW/FW features through open or internal efforts. GEOPM is a codesign vehicle |
| HPC Center Leadership | Prepare for future where power will be constrained | Provide GEOPM to scientists as a platform to research power/energy optimization strategies |
| HPC Center Leadership | Optimize system energy efficiency or throughput under power caps | Leverage GEOPM + Resource Manager / Scheduler integration for system-level optimization [coming soon] |
| Everyone | Help shape the GEOPM v1.0 product | Follow GEOPM documentation to install; participate in Beta testing and provide feedback |

# GEOPM Tech Adopted in **A21** for Exascale

- GEOPM is a pillar of Intel's solution for reaching Exascale targets
- Intel working on a contract with Argonne to build first US Exascale system called 'A21'
- A21 system will achieve >1 Peak ExaFLOPs in 2021
- GEOPM is expected to be enhanced and deployed on A21



# GEOPM Benefits Current Systems Too

- While funded to address Exascale challenges, GEOPM is being deployed on current systems too and will benefit them too
- Expected to be in production on **Theta** system at Argonne this year
  - Knights Landing system
  - Provides vehicle for production / at-scale testing of GEOPM software
  - Enables users to perform research leveraging GEOPM
- Expected to be in production on **SuperMUC-NG** system at LRZ, early ‘19
  - Aiming for a Top 10 ranking when the system comes online this year
  - Sky Lake based system
  - Strong emphasis on energy efficiency due to European electricity pricing

# GEOPM Coming to More Systems via OpenHPC


+


Accepted into OpenHPC, expected to intercept SC'18 OpenHPC release

Impact: adds an advanced power management runtime to OpenHPC and further expands the GEOPM community

# Outline

- Challenges and Approach to Solving Them
- GEOPM Architecture, Use-Cases, and Deployments
- GEOPM Experimental Evaluation
- GEOPM Work in Progress and Future Work
- Takeaways and Call to Action

# Intro to GEOPM Power Balancer Plug-In

- Power Balancer is a built-in optimization plugin provided with GEOPM SW package
  - Demonstrates the benefits of application-aware optimization techniques
- What it does:
  - Speeds up iterative bulk-synchronous MPI jobs in power-capped systems
  - Identifies critical path nodes which determine time-to-solution
  - Reallocates more power to critical path nodes to accelerate them
  - Improves time-to-solution by proactively avoiding wait time at the synchronization point
- Feedback and control mechanisms:
  - Programmer annotates application's outer loop containing a global synchronization operation via a lightweight profiling API we provide with GEOPM (i.e. 'epoch' hints)
  - Power balancer plugin measures loop time on each node and compares times across nodes to identify critical path nodes and how much performance/power correction is needed
  - Power balancer reconfigures each node's HW RAPL limits to set a new power allocation

# Background: Manufacturing Variation

- In iterative bulk-synchronous programs, nodes must reach global synch point before next iteration can begin
- If some nodes arrive late, other nodes must wait: power is wasted and potential performance is lost
- Even though workloads may be work-balanced, some nodes can fall behind due to performance differences across nodes
- Where do these performance differences come from?
  - Manufacturing variation. Power capping exposes different frequencies on different processors at a given power cap
  - This is true even if processors are taken from same product SKU

# Experimental Setup

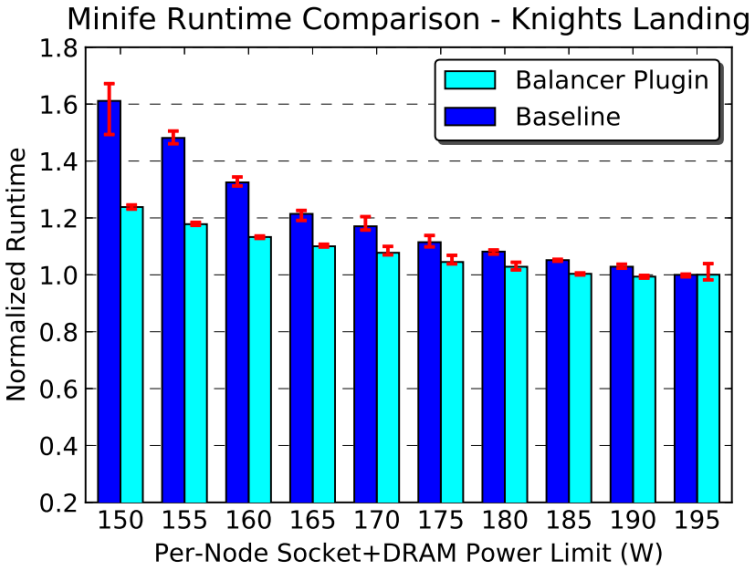
- Evaluations of power balancing plug-in performed on a Knights Landing cluster
- Included 5 key **CORAL procurement benchmarks plus ExMatEx CoMD and NAS FT**

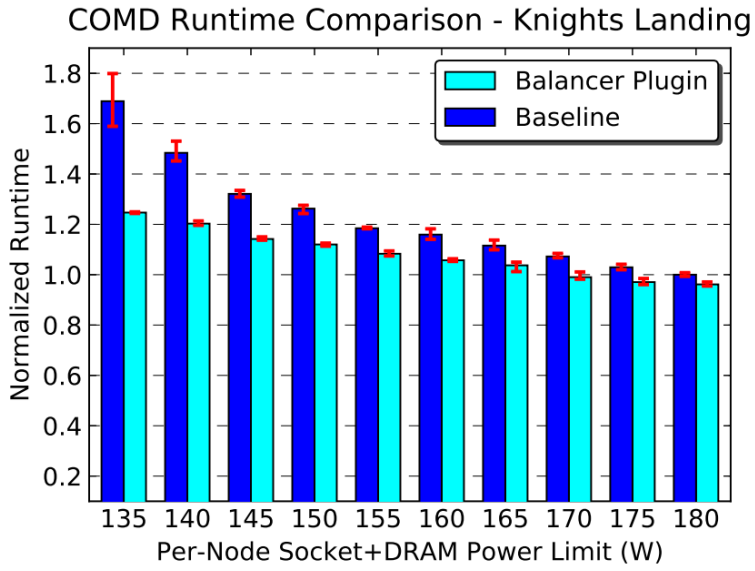
| Knights Landing Xeon Phi Cluster | |
|---|---|
| Hardware | Software |
| 12x KNL-F nodes B0 Beta **(all same SKU)** | CentOS 7, 'performance' frequency governor |
| 64 cores per node, 4-way hyperthreaded | Intel toolchain for C/C++/Fortran, MVAPICH2 MPI |
| 16GB MCDRAM, 256GB DRAM per node | Workloads instrumented with GEOPM profiling APIs |
| Integrated OmniPath HFI, OmniPath Fabric | Workloads run on optimal core and hyperthread count |
| Turbo enabled, 1.3GHz sticker frequency | Found that workloads run best if kept off of Linux CPU0 |
| 230W TDP processors | GEOPM and app affinitized to non-overlapping core |
| | 5 runs averaged per data point, error bars included |

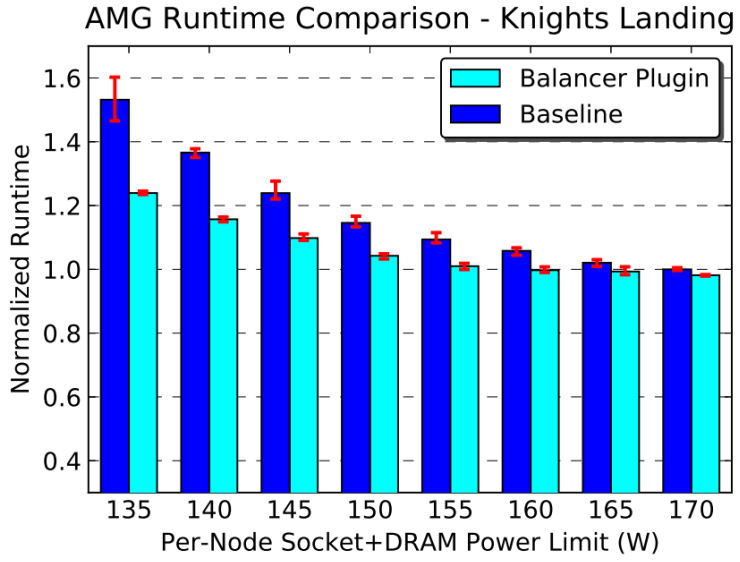
These results from ISC paper by Eastep et al. published in June 2017

# Results with a KNL Xeon Phi System

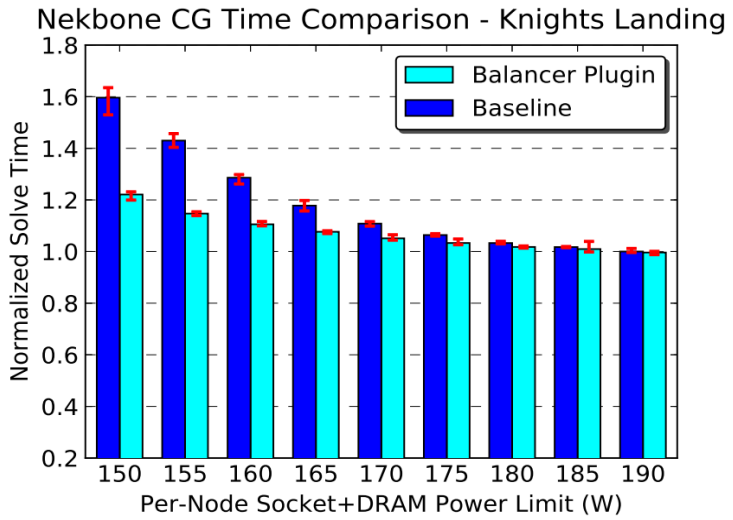
- Comparing time-to-solution when capping job power with GEOPM power balancer plug-in vs. static uniform power division (baseline), sweeping over job power caps
- Focusing on job power caps in the region of interest: low-end of caps avoids inefficient clock throttling and high-end is the unconstrained power consumption of the workload
- Main result: **up to 30% improvement** in time-to-solution at low caps (miniFE, CoMD, AMG), with **up to 9-23% for the rest.** Improvement generally increases as power is more constrained

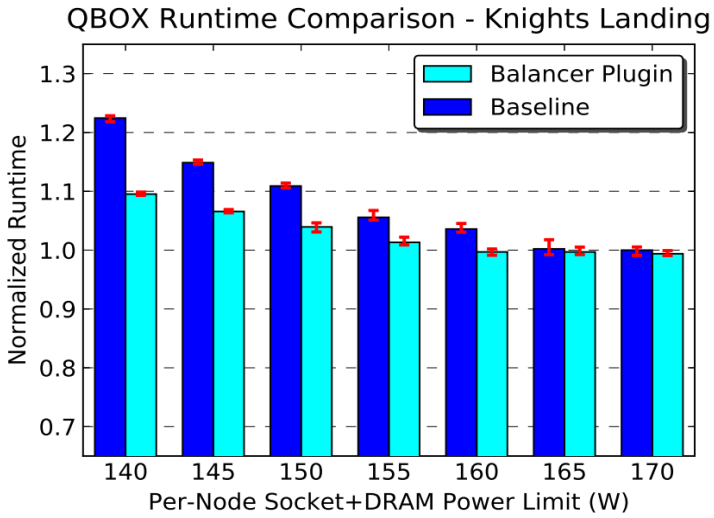


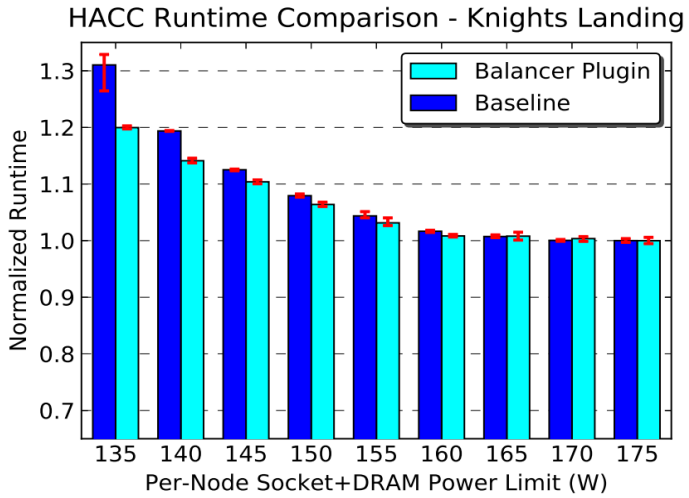



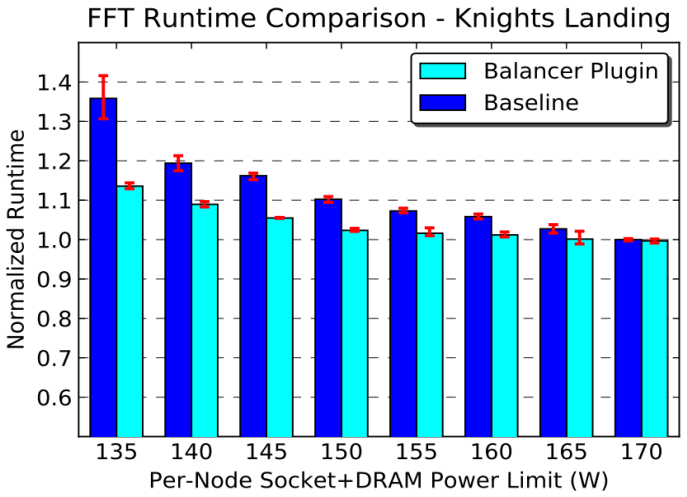



# Results with a KNL Xeon Phi System (2)



# Intro to GEOPM Energy Efficient Plugin

- We provide another built-in plugin w/ GEOPM that optimizes energy efficiency
- Collaborating with LRZ, Cineca, UniBo to develop this plugin
- Optimize processor core frequency to save energy with minimal TtS impact
  - i.e. reduce frequency for apps or phases that are memory-limited or MPI-communication-limited

| Energy-to-Solution and Time-to-Solution Comparison on Quartz BDX System at LLNL | | | | |
|---|---|---|---|---|
| | Offline Automatic ***Application*** Best-Fit | | Offline Automatic ***Per-Phase*** Best-Fit | |
| Workload | EtS Decrease vs Sticker | TtS Increase vs Sticker | EtS Decrease vs Sticker | TtS Increase vs Sticker |
| NAS FT | 9.5% | 6.8% | 15.8% | 4.8% |
| CORAL miniFE | 8.5% | 5.8% | Collecting data soon | Collecting data soon |
| CORAL Nekbone | 7.9% | 2.4% | Collecting data soon | Collecting data soon |
| LRZ Gadget | 17.2% | 7.1% | Collecting data soon | Collecting data soon |

TtS = time-to-solution
EtS = energy-to-solution

# Outline

- Challenges and Approach to Solving Them
- GEOPM Architecture, Use-Cases, and Deployments
- GEOPM Experimental Evaluation
- GEOPM Work in Progress and Future Work
- Takeaways and Call to Action

# Work in Progress and Future Work

- Prepping for general availability of GEOPM on Iota/Theta
  - Scheduling GEOPM tutorial for ALCF, timing TBD
- Prepping for inclusion of GEOPM in SC'18 OpenHPC release
- Prepping for v1.0 release of GEOPM for end of 2018
- Researching and developing GEOPM extensions with collaborators
  - Phase-adaptive optimization of processor core frequency for energy efficiency
  - Port supporting systems based on OpenPOWER + NVLink microarchitecture
  - Extensions for tuning parameters in SW (e.g. thread concurrency + MPI/OpenMP runtime params)
- Integrating GEOPM with other components of the HPC power stack:
  - Scalable global (job-level) monitoring
  - System-level power capping frameworks
  - Power-aware scheduling frameworks
  - System power forecasting frameworks

# Longer-Term Vision: HW/FW Codesign

- GEOPM project is not just a software project. It also drives codesign of the features in Intel hardware for power-performance monitoring and control
- Goals are to significantly advance the state-of-the-art in HPC power management technology and to ensure GEOPM runs best on Intel
- Research areas:
  - Processor: improvements to granularity, reaction time, and interfaces for existing features
  - Processor: hooks for GEOPM to guide allocation of Turbo headroom among cores
  - Memory: hooks for GEOPM to hint to mem controller when it's best to enter low-power states
  - Memory: hooks for GEOPM to hint to prefetchers how aggressively they should prefetch

# Outline

- Challenges and Approach to Solving Them
- GEOPM Architecture, Use-Cases, and Deployments
- GEOPM Experimental Evaluation
- GEOPM Work in Progress and Future Work
- Takeaways and Call to Action

# GEOPM: Take-Aways and Call-to-Action

- Substantial performance and energy benefits can be obtained with application-aware power management algorithms
- You can explore your own algorithms via new GEOPM plugins
- You can port GEOPM to your preferred architecture via plugins
- GEOPM coming soon to Iota/Theta and other production systems
- Try GEOPM. Need feedback to polish code for v1.0 release EO'18
- Call to Action: join the growing GEOPM user or developer community!

# Contact and Further Information

- Points of contact
  - Jonathan Eastep jonathan.m.eastep@intel.com
  - Slack message board here
- Tutorial on GEOPM
  - Coming soon to ALCF community. Now scheduling it
  - GEOPM will be made available to ALCF community on Iota/Theta at this time
- ISC'17 paper introducing GEOPM
  - Paper here by Eastep et al.
- GEOPM project page and info
  - http://geopm.github.io
  - More tutorials for offline instruction on GEOPM
  - Source code and user/developer documentation
  - Contribution instructions and github collaboration tools

# Team Acknowledgements


Principal Investigator:

- Jonathan Eastep, Principal Engineer and PhD

Team (in alphabetical order):

- Asma Al-Rawi
- Brandon Baker
- Brad Geltz
- Sid Jana
- Lowren Lawson
- Matthias Maiterth (PhD Intern)
- Revathy Rajasree
- Fede Ardanaz (HW/FW Lead)
- Chris Cantalupo (SW Dev Lead)
- Diana Guttman
- Fuat Keceli
- Kelly Livingston
- Ali Mohammad

# Questions?

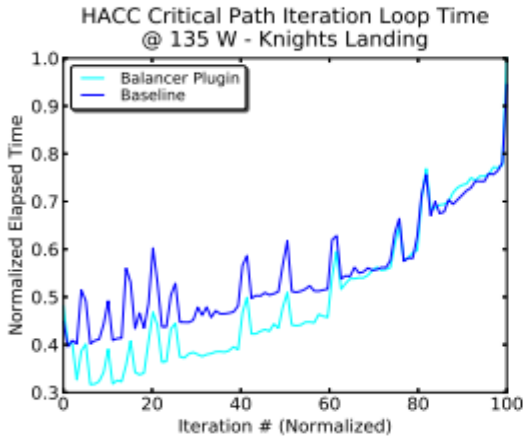
# Backup

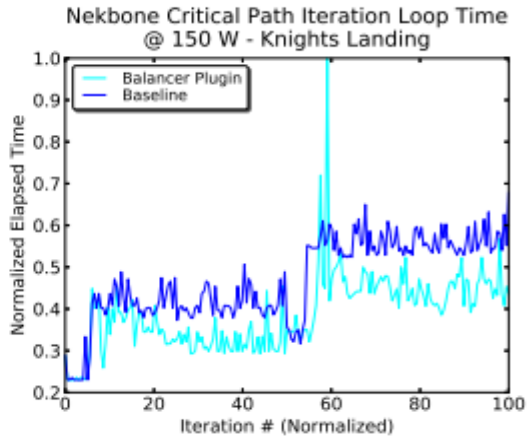
# GEOPM Speedup Analysis

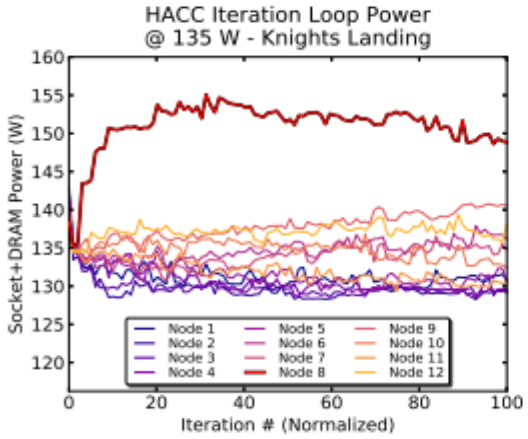
## (using included GEOPM Trace and Python Visualization Tools)

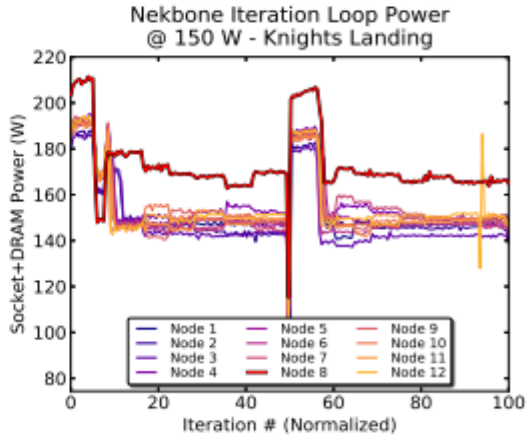
Take-away points:

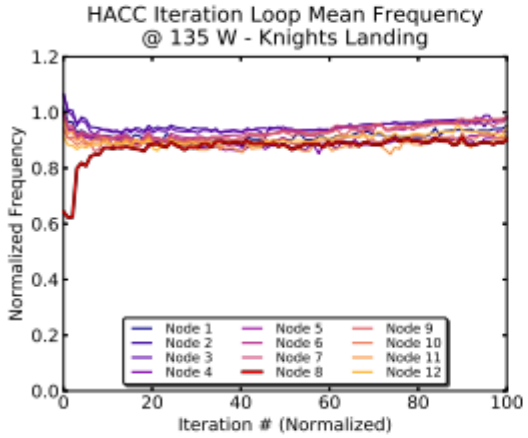
- Results demonstrate robustness of power balancing algorithm against time-varying amounts of work in the outer loop and sharp shifts in computational-intensity (top graphs)
- Node 8, with lowest power efficiency in our KNL cluster, is allocated more power (middle graphs)
- Power balancing algorithm improves critical path loop time by finding the power allocation that roughly equalizes the frequencies of all nodes (bottom graphs)

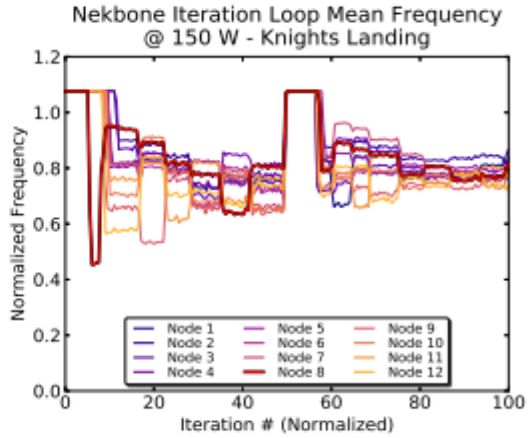


# GEOPM Beta Community Partners

| | Institution | Principal Investigator | Project Name | Project Scope | Contribution Type | Time Span | Quality Level | Funded? |
|---|---|---|---|---|---|---|---|---|
| | Argonne | Ti Leggett<br>Paul Rich<br>Kalyan Kumaran | CORAL -> A21 | 1. GEOPM 1.0 product development<br>2. GEOPM >1.0 feature development<br>3. GEOPM enablement for system power capping + EAS in Cobalt | Sponsor | Q2'15 – Q4'21 | Product | Yes |
| * | IBM<br>STFC<br>LLNL | Vadim Elisseev<br>Tapasya Patki<br>Aniruddha Marathe | | 1. GEOPM port to Power8 + NVLink<br>2. Integrate GEOPM with energy aware scheduler research | Contributor | Q4'16 – TBD | Near-Product | Yes |
| * | LLNL<br>Argonne<br>U. Arizona<br>U. of Tokyo | Tapasya Patki<br>Aniruddha Marathe<br>Pete Beckman<br>Dave Lowenthal | ECP PS ECP Argo-GRM | 1. Exascale power stack leveraging GEOPM<br>2. Integrate GEOPM + Caliper framework<br>3. Integrate GEOPM w/ SLURM power capping and power-aware scheduling extensions<br>4. Port of GEOPM to non-x86 architectures | Contributor | Q1'17 – Q4'19<br>SLURM PoC in '18 | Near-Product | Yes |
| | LRZ | Herbert Huber<br>Et al. | Super MUC-NG | 1. Enhance GEOPM monitoring features<br>2. Energy optimization plugin for GEOPM 1.0 | Contributor | Q3'17 – Q4'20 | Product | Yes |
| | Sandia | James Laros<br>Ryan Grant | Power API | 1. GEOPM and Power API xface compatibility<br>2. Power API community WG kickoff at Intel | User | Q4'14 - TBD | Industry Standard | Yes |
| * | UniBo<br>CINECA | Andrea Bartolini<br>Carlo Cavazzoni | | 1. Enhance GEOPM monitoring features<br>2. Energy optimization plugin for GEOPM<br>3. Integrate GEOPM + EXAMON<br>4. Integrate GEOPM w/ SLURM extensions | Contributor | Q2'18 – Q4'19 | Near-Product | Yes |

* = collaborator shared their GEOPM usages and experiences at ISC'18: in the GEOPM Tutorial

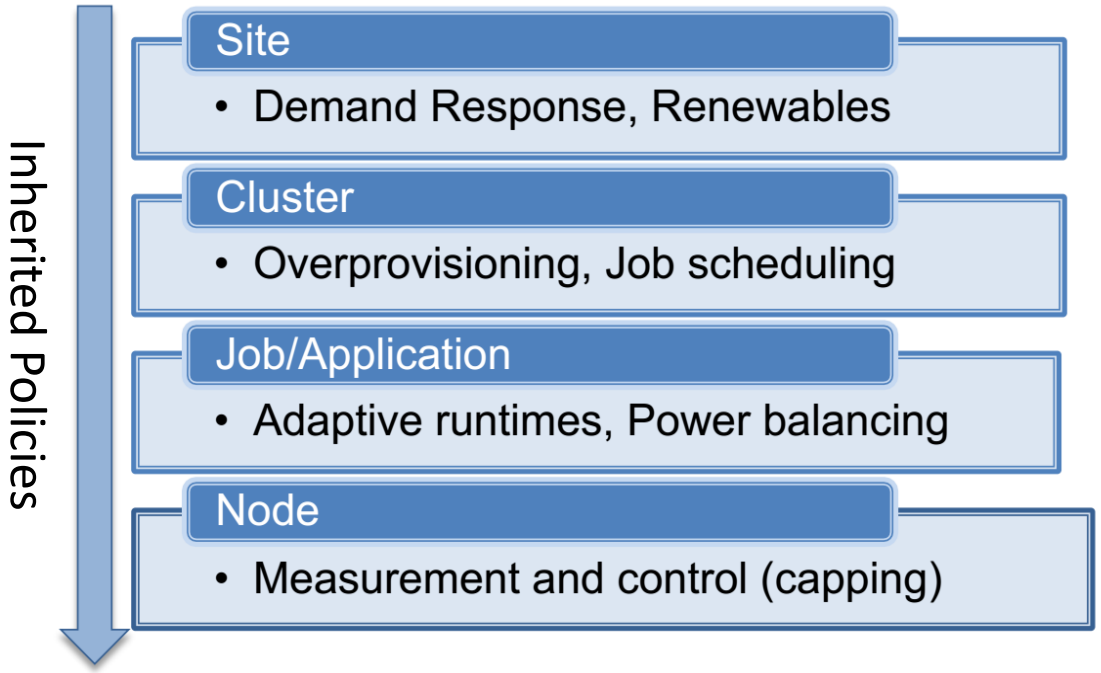
# GEOPM Profiling Interface

| Function | Description |
|---|---|
| geopm_prof_epoch() | Synchronization loop iteration beacon |
| geopm_prof_region() | Get region ID from name |
| geopm_prof_enter() | Mark region entry |
| geopm_prof_exit() | Mark region exit |
| geopm_prof_progress() | Report region progress |

- Functions enable programmer to mark up application with hints for GEOPM
  - Epoch: hint about a global synchronization loop where all MPI ranks will wait if 1 rank falls behind
  - Region: hint about computational phases between synch events for phase-adaptive energy mgmt
  - Progress: give application-level performance signal that can be used for intra-phase energy mgmt
- Interface is simple to use and has a low-overhead implementation
  - Nonetheless, research is under way to automatically extract this info with no programmer effort

# PowerStack Solution

- New production SW power mgmt stack to address Exascale challenges
- Design goals: holistic (site-wide), dynamic, flexible, application-aware




- Collaboration: my team, Argonne, LLNL, U. of Arizona, U. of Tokyo
- DOE ECP project + Argonne A21 NRE project (Intel's source of funding)
- Hierarchical design leveraging HPC community work:
  - Cluster: PowSched/RMAP, SLURM power capping extensions
  - Job/App: Conductor, READEX, **GEOPM**
  - Node: msr-safe, libmsr, Caliper, PowerAPI

GEOPM provides the foundations for PowerStack's dynamic adaptivity, flexibility, application-awareness, scalability, and product quality

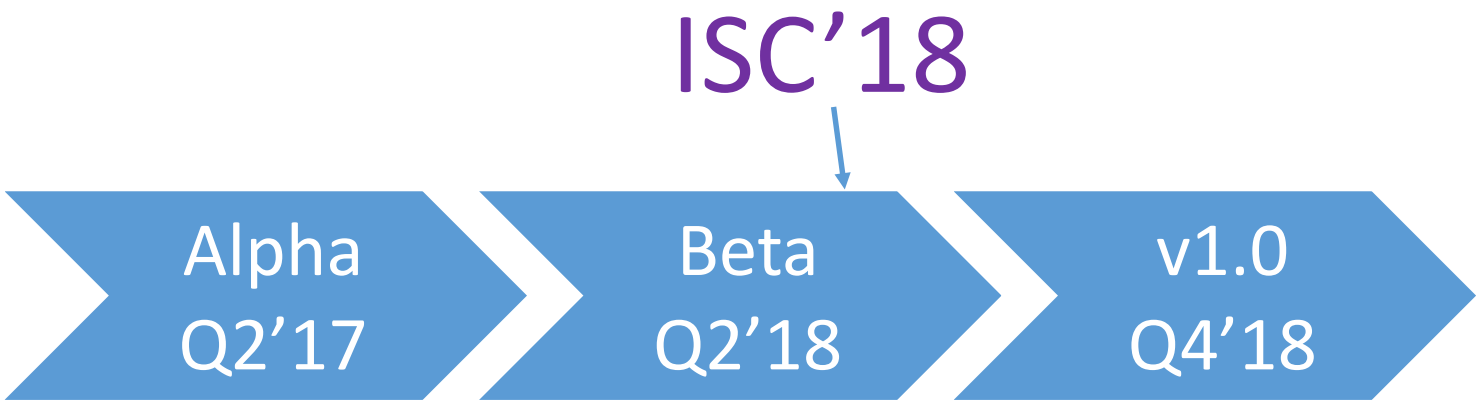
# GEOPM: Role Within PowerStack

- GEOPM = Global Extensible Open Power Manager
- Provides the job-level runtime in PowerStack
  - Global control system
    - Globally coordinates control knob settings across compute nodes in a job which is key to achieving ***application-awareness*** (esp: addressing load imbalance and variation)
    - Achieves ***scalability*** via distributed tree-hierarchical design, recursive control/feedback
  - Improves job performance or efficiency via feedback-guided optimization
    - ***Dynamically*** tunes HW control settings like node power caps & processor core frequencies
    - ***Application-aware*** tuning decisions via on-the-fly application profiling and analysis
    - 10-30% perf upside under power caps -or- 10-20% energy savings with ~5% perf impact (no cap)
  - Extensible framework via plugin architecture
    - Plugins provide ***flexibility*** to address broad range of Exascale power/energy challenges
    - Plugins can range from improving time-to-solution to improving efficiency to custom
    - Plugins allow easy porting to new hardware architectures (not limited to Intel x86)
  - ***Production-grade*** open source software that is backed by Intel

# GEOPM Beta is Available Now

## Coming soon: General Availability on Iota/Theta

Launch timeline:




GEOPM Beta is out now at geopm.github.io, released ahead of ISC'18

GEOPM v1.0 trending to be released before end of year