GEOPM Progress Updates
(Global Extensible Open Power Manager)
https://geopm.github.io/geopm

Jonathan Eastep [jonathan.m.eastep@intel.com]
Principal Engineer and PhD
14 November 2017

SC17 BoF: PowerAPI, GEOPM, and Redfish
Synergies Between Power API, GEOPM, and Redfish

- Power API is a specification for power monitoring and control interfaces
  - Proposes common interfaces for interoperability between power mgmt implementations

- Redfish is a specification for data center management
  - Provides convenient RESTful interface for power monitoring, control, and broader data center management functions

- GEOPM is a runtime for power management
  - Implements monitoring and control, and importantly: optimizes job power/performance
  - Would sit under Power API / Redfish to implement relevant power controls and monitors

- Ongoing collaboration between GEOPM, Power API, and Redfish
  - Redfish and Power API working toward compatibility
  - Power API and GEOPM have compatibility in their app-facing interfaces (mostly)

- Would love to see community Power API / Redfish implementations using GEOPM
- Runtime for in-band power management and optimization
  - On-the-fly monitoring of HW counters & application profiling
  - Feedback-guided optimization of HW control knob settings

- Open source software (flexible BSD three clause license)

- Extensible through plugin architecture
  - Add new energy optimization strategies
  - Add support for new architectures beyond x86 (truly open)

- Designed for holistic optimization
  - Job-wide global optimization of HW control knob settings
  - Application-awareness for max speedup or energy savings

- Scalable via distributed tree-hierarchical design, algorithms

Project url: [http://geopm.github.io/geopm](http://geopm.github.io/geopm)
Contact: jonathan.m.eastep@intel.com
GEOPM Use Cases

- **Turn-key (requires no app annotation):**
  - **Automatic online job profiling**
    - Node-level: trace samples of processor counters and correlate HW activity to each OpenMP parallel region
    - Job-level: aggregate the energy counters across all job compute nodes to monitor overall job power or energy
  - **Automatic offline or online optimization**
    - Will talk more about this today
  - **Offline visualization of profile data**
    - Python scripts leveraging pandas for data analysis
    - Helpful for debugging new plugins or understanding how they optimize energy or runtime
    - Plot trace of plugin decisions and data they’re based on

- **Advanced (requires using GEOPM profiling API for app annotation):**
  - **Automatic online rebalancing of power & perf among nodes**
    - Purpose: accelerate critical path nodes in MPI bulk-synchronous applications
    - Refer to ISC’17 paper on GEOPM by Eastep et al. for more info
  - **Note:** work in progress to make the annotation automatic / turn-key too
<table>
<thead>
<tr>
<th>Institution</th>
<th>Principal Investigator</th>
<th>Project Name</th>
<th>Project Scope</th>
<th>Contribution Type</th>
<th>Time Span</th>
<th>Quality Level</th>
<th>Funded?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argonne</td>
<td>Kalyan Kumaran Vitali Morozov</td>
<td>CORAL</td>
<td>1. GEOPM 1.0 product development</td>
<td>Sponsor</td>
<td>Q2’15 – Q4’17</td>
<td>Product</td>
<td>Yes</td>
</tr>
<tr>
<td>IBM STFC - Hartree</td>
<td>Vadim Elisseev Milos Puzovic Neil Morgan</td>
<td>1. GEOPM port to Power8 + NVLink 2. Integration of GEOPM with EAS</td>
<td>Contributor</td>
<td>Q4’16 – TBD</td>
<td>Research</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>LLNL</td>
<td>Barry Rountree Aniruddha Marathe</td>
<td>CRADA</td>
<td>1. Integration of GEOPM and Conductor runtime tech 2. Studies to motivate GEOPM/HW codesign</td>
<td>Contributor</td>
<td>Q3’13 – TBD</td>
<td>Research</td>
<td>Yes</td>
</tr>
<tr>
<td>LLNL Argonne U. of Arizona</td>
<td>Tapasya Patki Pete Beckman Dave Lowenthal</td>
<td>ECP PS ECP Argo-GRM</td>
<td>1. Exascale power stack leveraging GEOPM 2. Integration of GEOPM + Caliper framework 3. Integration of GEOPM with EAS 4. Port of GEOPM to non-x86 architecture</td>
<td>Contributor</td>
<td>Q1’17 – Q4’19</td>
<td>Near-Product</td>
<td>Yes</td>
</tr>
<tr>
<td>LRZ</td>
<td>Dieter Kranzmüller Herbert Huber Torsten Wilde</td>
<td>Power API</td>
<td>1. Energy optimization plugin for GEOPM 1.0 2. Power ramp limiting plugin for GEOPM 1.x</td>
<td>Contributor</td>
<td>Q3’17 – Q4’20</td>
<td>Near-Product</td>
<td>Yes</td>
</tr>
<tr>
<td>Sandia</td>
<td>James Laros Ryan Grant</td>
<td>Power API</td>
<td>1. GEOPM and Power API xface compatibility 2. Power API community WG kickoff at Intel</td>
<td>User</td>
<td>Q4’14 - TBD</td>
<td>Industry Standard</td>
<td>Yes</td>
</tr>
</tbody>
</table>

* = collaborator will be sharing their GEOPM usages and experiences at SC17: BoF on Power API, GEOPM, and Redfish
<table>
<thead>
<tr>
<th>Institution</th>
<th>Principal Investigator</th>
<th>Project Name</th>
<th>Project Scope</th>
<th>Contribution Type</th>
<th>Time Span</th>
<th>Quality Level</th>
<th>Funded?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argonne</td>
<td>Kalyan Kumaran Vitali Morozov Kevin Harms</td>
<td>1. GEOPM &gt;1.0 feature development 2. GEOPM enablement for system power capping + EAS 3. Studies to motivate GEOPM / hardware codesign</td>
<td>Sponsor</td>
<td>Q1’18 – Q4’21</td>
<td>Product</td>
<td>WIP</td>
<td></td>
</tr>
<tr>
<td>CINECA</td>
<td>Carlo Cavazzoni</td>
<td>1. System level runtime for power capping and power ramp limiting leveraging GEOPM</td>
<td>Contributor</td>
<td>Q2’18 – Q1’21</td>
<td>Near-Product</td>
<td>WIP</td>
<td></td>
</tr>
<tr>
<td>IT4I</td>
<td>Lubomir Riha</td>
<td>1. GEOPM ports to OpenPOWER and ARM 2. Extensions to GEOPM application profiler 3. Integration of GEOPM with EAS</td>
<td>Contributor</td>
<td>Q2’18 – Q1’21</td>
<td>Near-Product</td>
<td>WIP</td>
<td></td>
</tr>
<tr>
<td>E4</td>
<td>Fabrizio Magugliani</td>
<td>1. GEOPM port to OpenPOWER</td>
<td>Contributor</td>
<td>Q2’18 – Q1’21</td>
<td>Near-Product</td>
<td>WIP</td>
<td></td>
</tr>
<tr>
<td>PNNL</td>
<td>Leon Song</td>
<td>1. GEOPM extensions to tune new HW control knob settings 2. GEOPM extensions for coordinated tuning of SW params and HW control knob settings</td>
<td>Contributor</td>
<td>Q1’19 – Q4’20</td>
<td>Research</td>
<td>WIP</td>
<td></td>
</tr>
</tbody>
</table>

† = letter of intent or equivalent in-hand (non-binding)
GEOPM Release Schedule

Commitment:
- Alpha Q2’17
- Beta Q2’18
- v1.0 Q4’18

Stretch Goal:
- Alpha Q2’17
- Beta Q1’18
- v1.0 Q2’18

Announcement: OpenHPC application has been submitted. Under consideration.

ISC’18
TOSS 3.x

SC’18
GEOPM Core Team Acknowledgements

Lead Architect:
• Jonathan Eastep, Principal Engineer

Hardware Team:
• Processor Firmware
  • Revathy Rajasree
• Hardware Architecture and Design
  • Fede Ardanaz
  • Fuat Keceli
  • Kelly Livingston
  • Lowren Lawson

Software Team:
• GEOPM Development
  • Chris Cantalupo
  • Diana Guttman
  • Brad Geltz
  • Brandon Baker
• Research
  • Sid Jana
  • Asma Al-Rawi
  • Matthias Maiterth
Backup Slides
What Problems Does GEOPM Address?

1. **At-scale load imbalance due to manufacturing variation in power-capped systems.** This problem is deemed one of the key Exascale-era power challenges. Developing GEOPM and techniques to address this problem over the past 6 years made me a Principal Engineer at Intel.

2. **Gap in community energy management research tools.** There was previously no platform for energy management research that was open, scalable, robust, flexible, portable (truly open), and backed by serious engineering resources. Now the community is using GEOPM, porting to non-x86 architectures, integrating their optimization techniques into it, and integrating it with other software components.

3. **Gap in industry server power management roadmaps and technical directions.** Power management was previously done node-locally. Techniques were oblivious to application-level information such as bottlenecks on remote nodes that could limit overall performance and were unable to forecast what computation was going to happen in the future and optimize power-performance policy accordingly. GEOPM adds a critical layer of global optimization across nodes, application and application phase awareness, and forecasting capabilities. See ISC’17 paper for demo of benefits (up to 32% speedup).
Experimental Setup: 3 Investigations

1. Opportunity Analysis
   • Use proxy app (model application) to determine envelope of energy-to-solution and time-to-solution impact we’ll see over the landscape of BSP applications
   • Measure energy-to-solution decrease and time-to-solution tradeoff relative to running at sticker on the JLSE cluster at Argonne
   • Compare two different use-cases for the offline technique we developed:
     • ‘Offline automatic application best-fit:’ all phases run at common frequency (best-fit across all)
     • ‘Offline automatic per-phase best fit:’ each phase runs at the best frequency for it

2. Benchmark offline energy optimization technique
   • Target FT, miniFE, and Nekbone workloads
   • Same as above but targets less synthetic workloads and performs experiments on LLNL Quartz system

3. Benchmark online energy optimization technique
   • Target the proxy app and perform experiments on JLSE cluster at Argonne
   • Compare the online and offline techniques we developed:
     • ‘Offline automatic per-phase best-fit:’ scripts identify best frequency via offline characterization
     • ‘Online automatic per-phase best fit:’ GEOPM plugin performs characterization/tuning online
Big energy savings are possible with frequency optimization in GEOPM vs running workloads at sticker: up to 16.5% energy savings at 0.3% increase in time-to-solution

With per-phase optimization, energy savings increase with increase in % time in memory-limited phase

Per-phase optimization simultaneously offers better energy-to-solution AND time-to-solution versus optimizing frequency across the blended characteristics of all application phases
## Results: Offline App vs Per-Phase Best-Fit

### Energy-to-Solution and Time-to-Solution Comparison on Quartz

<table>
<thead>
<tr>
<th>Workload</th>
<th>Offline Automatic <strong>Application</strong> Best-Fit</th>
<th>Offline Automatic <strong>Per-Phase</strong> Best-Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>EtS</td>
<td>EtS Decrease vs Sticker</td>
<td>EtS Decrease vs Sticker</td>
</tr>
<tr>
<td>TtS</td>
<td>TtS Increase vs Sticker</td>
<td>TtS Increase vs Sticker</td>
</tr>
<tr>
<td>FT</td>
<td>9.5%</td>
<td>6.8%</td>
</tr>
<tr>
<td>miniFE</td>
<td>8.5%</td>
<td>5.8%</td>
</tr>
<tr>
<td>Nekbone</td>
<td>7.9%</td>
<td>2.4%</td>
</tr>
</tbody>
</table>

Results starting to confirm that GEOPM provides benefits for a number of workloads beyond our proxy app.

More data on the way, but data starting to suggest per-phase frequency optimization simultaneously offers better energy-to-solution AND time-to-solution vs optimizing frequency across blended characteristics of whole app.
Results: Online vs Offline Technique

Remember, offline approach is brittle. The goal: same (or better) results via more robust online approach. We think much of the EtS and TtS gap can be closed via addressing frequency latency & doing longer runs. Fine-tuning needed, but already seeing promising decreases in energy-to-solution with online approach.

Explanation of EtS and TtS gaps:

- Runs were shorter than real apps -> noticeable “learning” overhead
- Reduced # samples in learning period to reduce overhead -> more noise-related control errors
- Observed latency between frequency change requests and enactment (10s of milliseconds) -> not running at desired frequency immediately, confusing algorithm
See GEOPM ISC’17 paper by Eastep et al. for details of experimental setup and further analysis

Paper demonstrates power balancing plugin: it leverages annotation of application’s outer synchronization loop to detect critical path nodes and then reallocates power among nodes in order to equalize their time to complete a loop iteration

Compared overall time-to-solution when capping job power on 12-node KNL cluster with power balancer plug-in vs. static uniform power division (baseline); swept over a range of different job power caps

Region of interest in job power caps: low-end of job power caps was selected to avoid inefficient clock throttling and the high-end of the job power caps equals the unconstrained power consumption of the workload

Main result: up to 30% improvement in time-to-solution at low end of caps (miniFE, CoMD, AMG), with up to 9-23% for the rest. Improvement generally increases as power is more constrained
Results: Four Additional Workloads

- Nekbone CG Time Comparison - Knights Landing
- QBOX Runtime Comparison - Knights Landing
- HACC Runtime Comparison - Knights Landing
- FFT Runtime Comparison - Knights Landing
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)
Research on GEOPM/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
  - Network: hooks for GEOPM to estimate power, manage tradeoffs between power and bandwidth in HFI and switches, and hint to HFI when it’s best to enter low-power states
GEOPM New Business Opportunities

- GEOPM software package is open source, provides a rich feature set free of charge
- Intent is for Intel’s future work on the software to be open source as well
- 3rd parties are able to make proprietary extensions of GEOPM (BSD 3-clause license)
  - Enables integrators like Dell/Cray/HPE to develop commercial for-profit plugins (i.e. add power management secret sauce to differentiate your systems vs the competition)
  - GEOPM team can help integrators with this in a consulting capacity
- Intel can explore developing custom processor firmware enhancements for customers
  - Enables processor power management firmware and GEOPM plugins to be co-optimized for individual customer needs
  - Enables management of hardware control knob settings which are not (yet) publicly available
  - Providing GEOPM NRE funding in a system contract is a good way to establish such an engagement

Inquire with Jonathan Eastep for more information: jonathan.m.eastep@intel.com