GEO Progress Updates
(Global Energy Optimization)

Project Lead: Jonathan Eastep, PhD & Principal Engineer
jonathan.m.eastep@intel.com

May 12, 2016
Recap of GEO Project Scope and Goals

- GEO is a runtime for energy optimization in HPC systems
  - Application-level: launches with and runs with the application
  - Global: coordinates DVFS / power alloc decisions across nodes
  - Open source: BSD 3-clause license
  - Scalable: tree-hierarchical control and telemetry aggregation
  - Extensible: plug-ins for extensions + out-of-the-box functionality

- Goals:
  - Report per-job (or per-application-phase) energy/perf profile info
  - Provide out-of-the-box functionality to unlock substantially more performance in power-limited systems
  - Provide open platform for research community to accelerate innovation in HPC system energy management
Recap of Implementation Status (5 Dec 2015)

Reported initial public release of GEO on github

- **Package Name:** geopm (stands for GEO power management)
- **Release goals:**
  - Define GEO interfaces and publish user docs for community review
  - Nail down modular OO-design in C++11 (w/ C external interfaces)
  - Include solid autotools build system & gtest/gcov test infrastructure
  - Include support for basic static power management functionality
    - Example: Uniform Frequency Static mode
- **Non-Goals:**
  - Code / feature-completeness
    - No dynamic power management yet (runtime was still under construction)
    - No support for extensibility via plug-ins yet
Status Update on Implementation (Current)

Completed a significant new geopm release

- Release goals:
  - Achieve functional correctness of runtime for dynamic power mgmt
  - Provide plug-in frameworks for extending GEO in two dimensions:
    - Add new energy management strategies
    - Add support for new target hardware platforms
  - Provide an out-of-the-box plug-in for a key US DOE use-case:
    - Goal: maximize application performance within a job power bound
    - Approach: dynamically reallocate power to speed up nodes on critical path
  - Provide developer documentation and additional user documentation

- Non-Goals:
  - Production quality test coverage (much testing included, more needed)
  - Benchmark and regression test infrastructure (work in progress)
  - Tuned-up power balancer plug-in (results not yet optimized)
Status Update on Collaborations: Argonne

- **Goal:** develop GEO for deployment on Aurora in 2018
  - Note: earlier intercepts probable on other Phi or Xeon systems

- **Scope:**
  - Work with Argonne/Cray to integrate GEO into Aurora software stack
  - Nail down key use-cases for GEO & user incentives for running it
  - Explore power-aware scheduler functions in Cobalt Job Mgmt Suite

- **Status:**
  - [COMPLETE] Define GEO design and integration architecture
  - [NEXT STEP] Bring up test cluster at Argonne for integration work
  - [NEXT STEP] Demo GEO running on KNL cluster (proxy for Aurora)
Goal: work toward deploying GEO on LLNL production systems

Scope:
- Develop high-performance safe userspace interfaces to power/perf monitors and controls (build on msr-safe)
- Study /enhance GEO scalability on LLNL catalyst test cluster
- Explore integrating Conductor energy mgmt technology into GEO

Status:
- [COMPLETE] msr-safe enhancements for performance
- [NEXT STEP] Work with LLNL and Cray and attempt to get msr-safe adopted in OpenHPC and SLES/RHEL Linux distros
- [NEXT STEP] Begin GEO scaling work
- [NEXT STEP] Begin exploring Conductor integration
Status Update on Collaborations: Sandia

- **Goal:** work toward compatibility between Sandia Power API and GEO APIs and explore integration feasibility

- **Scope:**
  - Exchange feedback to influence future API versions, simplify wrapping
  - Explore feasibility of having GEO provide some of the control and monitoring functionality specified in Sandia API

- **Status:**
  - [COMPLETE] GEO team to modify application API for simpler wrapping
  - [COMPLETE] GEO team to suggest changes to Sandia application API for compatibility
  - [NEXT STEP] Sandia working to incorporate feedback on application API into a future version of the spec
  - [NEXT STEP] Exchange feedback on design of interfaces between Workload Managers and Job-Level Energy Managers like GEO
## Project Information

<table>
<thead>
<tr>
<th>Type</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Email contact</td>
<td><a href="mailto:jonathan.m.eastep@intel.com">jonathan.m.eastep@intel.com</a></td>
</tr>
<tr>
<td>Project page url</td>
<td>geopm.github.io/geopm</td>
</tr>
<tr>
<td>Project repo url</td>
<td>github.com/geopm</td>
</tr>
<tr>
<td>Release notes url</td>
<td>github.com/geopm/geopm/releases/tag/v0.1.0</td>
</tr>
<tr>
<td>EE HPC WG Webinar on GEO hyperlinks</td>
<td>slides and audio</td>
</tr>
</tbody>
</table>


GEO Team Acknowledgements

GEO Core Team (Intel)
- Fede Ardanaz
- Chris Cantalupo
- Jonathan Eastep
- Stephanie Labasan
- Kelly Livingston
- Steve Sylvester
- Reza Zamani
- ... and hiring!

Collaborators (Intel)
- Tryggve Fossum
- Al Gara
- Richard Greco
- David Lombard
- Ram Nagappan
- Mike Patterson
Backup Slides
GEO Capabilities

- Comprehend and mitigate dynamic load imbalance by globally coordinating frequency and power allocations across nodes
- Leverage application-awareness and learning to recognize patterns in application (phases), then exploit patterns to optimize decisions
- React to phase changes at aggressive time scales (low milliseconds) and rapidly redistribute limited power to performance-critical resources
- Tackle the scale challenges prior techniques have swept under the rug to enable holistic joint optimization of power policy across the job
Recap of GEO Integration Architecture

- **User Interface** (Work w/ job queues)
- **Admin Interface** (Work w/ WLMs)

**Workload Manager**
- Power-Aware Sched (Work w/ Intel to implement)

**Workload Mngr Interface** (Work w/ WLMs)

**Application Power Mngr** = GEO

- PCU RAPL and Perf Counter Interfaces (Work w/ Intel to enhance)

**Owner**
- 3rd parties
- Intel GEO team
- Intel PM Arch team

Initially required; later optional if not tuning SW knobs
GEO Hierarchical Architecture

GEO manages job to a power budget and globally coordinates frequency & power allocation decisions

Scaling challenge is addressed via tree-hierarchical design & hierarchical policy
- Each agent owns sub-problem: decide how to divide/balance power among children
- Power/perf telemetry is scalably aggregated so network traffic is minimal
- Tuning is globally optimized despite distributed tuning: achieved through Hierarchical-POMDP learning techniques

GEO tree runs in 1 reserved core per CN
- Leaf & non-leaf agents run in these cores
- Enables fast reaction times, deep analysis
- Overhead is negligible in manycore chips
- Designing for minimal memory footprint

CN ≡ Compute Node
(in compute node racks)
Zoom-In on Leaf Agent

Power budgeting inside the processor:
- Spatio-Temporal Energy Scheduling (phase-adaptively allocate power among RAPL power domains)
Auto-Tuner Prototype Results Summary

Speedup from Auto-Tuner at ISO Power

- **miniFE**: 1.18x
- **FFT**: 1.31x
- **IS**: 1.17x
- **NEKbone**: 1.22x

Speedup derives from two factors: correcting load imbalance across nodes and node-local spatio-temporal energy scheduling optimizations exploiting phases.

Bars represent average results over a range of assumptions about how much power the job is allocated and how much load imbalance is present.

Experimental setup carefully emulates large-cluster load imbalance on a small cluster.

Results collected while running on Xeon hardware (not simulation).