PShifter: Feedback-based Dynamic Power Shiftering within HPC Jobs for Performance
Neha Gholkar¹, Frank Mueller¹, Barry Rountree², Aniruddha Marathe²
¹ North Carolina State University  ² Lawrence Livermore National Laboratory

Motivation
- Exascale: 10X performance improvement under 2X power budget with respect to today’s technology
  - Discrete power management → assign power budgets to jobs
  - Achieve performance improvements for jobs while enforcing job power constraints

Problem
- Enforcing a job power budget
  - Uniform power distribution across sockets of a job (Naïve)
- Uniform Power
  - Manufacturing Variation → Hardware-induced imbalance
- Typical scientific simulations
  - Single Program Multiple Data (SPMD) codes
  - Non-uniform workload distribution → Input-induced imbalance
- Performance Imbalance
  - CPU cycles and energy wasted during wait time

Proposed Solution Approach
- Goals 1: Reduce wait times
  - To reduce wastage of CPU cycles and energy
- Goal 2: Speedup the critical path participants
  - To reduce the completion time → performance improvement
- Goal 3: Dynamic adaptation
  - Load driven management

We propose Power Shifting to achieve these objectives

Power Shifter (PShifter)
- Hierarchical runtime system
  - Multi-level feedback controller
- Agents
  - Job-level: Cluster Agent
  - Socket-level: Local agent

Local Agent
- Reduces power of its socket if it incurs wait time → slowdown
  - Reduces wait time and hence energy wastage
  - Uses Proportional-Integral (PI) feedback-controller to calculate power deductions
- Multiple waiting sockets that are not on the critical path operate at lower power
  - Job’s power budget is underutilized

Cluster Agent
- Calculates unused job power budget
  - Re-distributes it uniformly across all sockets
- Sockets on the critical path operate at higher power → speedup
  - Shortens the critical path → improving job performance

Conclusions
- PShifter achieves performance improvement while enforcing job-level power constraint. It reduces performance imbalance (hardware-induced/input-induced) by automatically and dynamically shifting power from waiting sockets to the sockets on the critical path.

For more details please attend our presentation
Session 4: Runtime Systems
10:30AM on June 14th