Dynamic Adaptability in Support of Extreme Scale

Jayaraman Suresh Babu (M.S. Candidate), Seetharami Seelam (Ph.D. Candidate), Nidia Pedragon (B.S. Candidate), Alejandro Castaneda (B.S. Candidate), Rodrigo Romero, Ph.D., and Patricia J. Teller, Ph.D.
Department of Computer Science, University of Texas at El Paso

Goals
Generalized resource management with Fixed parameters = Current Performance
Customized resource management with Dynamically Adaptable parameters = Enhanced Performance

Challenges
- What to Adapt?
- When to Adapt?
- How to Adapt?
- How to measure effects of adaptation?

Deliverables
- Mechanisms to dynamically sense, analyze, and adjust common performance metrics based on fluctuating workload situations and overall system conditions
- A general-purpose methodology for dynamic adaptation of commodity operating systems

Methodology
Identify possible adaptation targets
Characterize application resource usage patterns
Offline

Example Adaptations
- Process scheduling parameters, algorithms (e.g., dynamic adaptation of scheduling policy of processes)
- I/O scheduling parameters, algorithms (e.g., dynamic adaptation of scheduler algorithms – prototype in process)
- Memory management algorithms (e.g., dynamic adaptation of page management thresholds)

Implementation Platform
Dynamic Adaptation / Self-tuning will be demonstrated via Linux prototypes and experiments for HPC environments

Examples of Adaptation
- Different I/O scheduler algorithms already implemented in Linux 2.6 kernel
- Default scheduler is “AS” since it gives best performance in most cases
- But experimentation reveals cases where each scheduler shines – No Silver Bullet in schedulers
- Hence appropriate dynamic scheduler selection necessary to ensure good performance