Towards Modeling Parallelism and Energy Performance of Multicore Systems

Bogdan Marius Tudor, Yong Meng Teo
[bogdan],[teoym]@comp.nus.edu.sg
Department of Computer Science
National University of Singapore

Motivation
Multicore systems are mainstream

Objective and Approach

Objective
Develop an approach for users and developers to analyze the performance of a shared-memory program for different programming models and languages across multicore platforms.

Scope of the Models
- Programming Models: OpenMP, Pthreads, Mobile Apps, CUDA
- Analytical Models: Power Monitor, Hardware Events Counters
- Measurement Analysis
- Multicore Platforms: Intel / AMD, ARM, NVIDIA

Approach
- Shared-memory Program Executions
- Baseline Program Executions
- Parallelism Performance $A(m,n)$ - inherent parallelism with $m$ threads
- Energy Usage $E(n, f)$ - energy on $n$ cores with clock freq. $f$
- $\lambda$, $\mu$, $\rho$ - service of OS run-queue
- $\omega$, $\mu_n$, $\lambda$ - cycle accounting

Measurement Analysis

Experimental setup
- Systems: 8 cores - Intel UMA (2 * Xeon E5320), 24 cores - Intel NUMA (2 * Xeon X5690), 48 cores - AMD NUMA (4 * Opteron 6172)

Observations of memory contention
- O1. Memory controllers reduce contention
- O2. Memory contention increases with problem size
- O3. Caching is effective for small problem size

Validations & Applications
- Prediction: Optimal #cores
- Target architecture: X86 / x64 Intel & AMD

Summary
- Contributions: Parallelism performance model for UMA and NUMA systems across different languages and architectures
- Energy model linking parallelism and memory performance with energy usage
- Analysis of memory contention showing that memory accesses are not always bursty
- Performance optimizations in memory-bound applications