Classifying-based Partial-Global Scheduling

A scheduling framework with CPU and thread selector co-design for multi-thread workloads on asymmetric multicore processors

Teng Yu, Mingcan Zhu, John Thomson, Hugh Leather
{ty33,j.thomson}@st-andrews.ac.uk
{mingcan.zhu,hleather}@inf.ed.ac.uk

MOTIVATION

- Global Scheduling
- Frequent migrations
- Greedy sub-optimal decisions
- Partition-based Scheduling
- NO uniform best framework

INSIGHTS

1. Predicating the performance gain for a thread on different types of cores: Core Sensitive/Speedup Factor
2. Maintaining fairness and handle with thread migration overheads: Fairness Aware Factor
3. Accelerating the most necessary/block threads: Bottleneck Identification Factor

RESEARCH IDEAS

- Off-line training model
- Fairness, Speedup, Bottleneck
- Runtime classifier
- Hierarchical round-robin CPU selector
- Non-greedy decisions
- Load balancing
- Execution of multiple multi-thread workloads

EXPERIMENTAL SETUP

- Workloads: PARSEC3.0, SPLASH-2, SPEC2006
- Proposed Hardware: ARM big.LITTLE architectures
- Simulator: Gem5

PRIMARILY RESULTS