Reliability-Aware Scheduling on Heterogeneous Multicore Processors

Ajeya Naithani
Ghent University, Belgium

Stijn Eyerman
Intel, Belgium

Lieven Eeckhout
Ghent University, Belgium

HPCA 2017
Motivation

• Wide use of Heterogeneous Chip Multi-Processors (HCMPs) in mobile SoCs
  - e.g., ARM’s big.LITTLE, Nvidia’s Tegra
Goal

• Increasing robustness of HCMPs against soft errors

• Reliability-aware scheduling of HCMPs based on:
  - Core type
  - Workload
Terminology

• **ACE bit**: Architecturally Correct Execution bit

• **ABC**: ACE Bit Count

• **AVF**: Architectural Vulnerability Factor

• **IFR**: Intrinsic Fault Rate

• **SER**: Soft Error Rate
Reliability-Aware Scheduling

• Requires a metric for system-wide reliability

\[ \text{System } SE = \sum_{i=1}^{n} \text{SER}_i \]

Applications may slow down!
Reliability-Aware Scheduling

• Reliability Metric: System-level Soft Error Rate (SSER)

\[ SSER = \sum_{i=1}^{n} wSER_i \]

• Weighted SER:

\[ wSER = \frac{ABC}{T} \times \frac{T}{T_{ref}} \times IFR = \frac{ABC}{T_{ref}} \times IFR \]
Reliability-Aware Scheduling

Sampling Data per Application:
1. ABC on Different Cores
2. Performance on Different Cores
Hardware Overhead

• Profiled structures:
  - ROB
  - Issue queue
  - Load/store queue
  - Physical output registers
  - Functional units

• Total hardware overhead:
  - 904 bytes per big core
  - 67 bytes per little core

• Approximation: Only profile ROB → HW overhead: 296 bytes per big core
Evaluation

• Three schedulers:
  - Random
  - Reliability-optimized
  - Performance-optimized

• Benchmark suite
  - SPEC CPU2006
Application Characteristics

![Diagram showing AVF (%) for different applications across Low, Medium, and High categories.](Fig. 1 in the paper)
Results – Symmetric HCMP (2B2S)

32% improvement over random scheduling
25.4% improvement over performance-optimized scheduling

Same as random scheduling
6.3% degradation over performance-optimized scheduling

[Fig. 6 in the paper]
Results – Asymmetric HCMP

[Fig. 8 in the paper]
Results – Workload Categories

(a) Reliability (SSER)

SSER normalized to random

HHHH  HHMM  HHLL  MMMM  MMLL  LLLL

random  performance-optimized  reliability-optimized
Results – Approximate Profiling

[Modified version of Fig. 10 in the paper]
Conclusion

• Applications and cores → Different vulnerability characteristics

• Reliability-aware scheduling:
  - System reliability 25.4% ↑
  - Performance 6.3% ↓

• The proposed scheduler is robust across:
  - Core configurations
  - Workload types
Discussion Points

• HCMPs are mainly used in mobile SoCs, while the evaluated benchmarks target server/desktop processors. Is the sampling method actually applicable to mobile applications, which are heavily I/O based?

• The scheduling is implemented in software. Is it efficient in a battery-constrained platform?

• What if the number of applications is more than the number of cores?