15-740/18-740 Computer Architecture

Course Project Proposal

Athula Balachandran
Lavanya Subramanian

October 21, 2009

1 Introduction

Dynamic Voltage Frequency Scaling (DVFS) is a traditional technique that is used to exploit run time variability and conserve power with minimum performance degradation. DVFS is typically employed at the OS scheduler intervals. However, recent research shows that applications' variability behaviour is more fine-grained and cannot be exploited effectively, by performing DVFS at OS scheduler intervals. Employing DVFS at fine-grained intervals imposes a huge delay overhead for the regulator voltage level transitions and is practically impossible, with off-chip regulators. In this light, [1] proposes a scheme, where the different cores are assigned different voltage/performance levels and can be used based on the applications' performance requirements. The authors call this mechanism Thread Motion. Our project looks into the challenges/bottlenecks in applying this to a generic chip multiprocessor.

2 Project Description

In [1], the authors employ an architecture similar to the Sun ROCK processor. This architecture groups processors into clusters and they share an L1 cache. Migrations that are performed within a cluster do not suffer the impact of missing L1 cache data. However, in most Chip Multiprocessor Systems, each processor has a private L1 cache. So, we aim at exploring the effectiveness of the "Thread motion" scheme in this scenario. Specifically, we would like to quantify the performance degradation, that would result from the L1 misses, when migration is performed. We observe that the concept of intra and inter clusters does not apply in this scenario.

Migration to a far-off core would also result in increased L2 access latency. We plan to also fine tune the migration algorithm/strategy to minimize this.

- 75% Goal : Preliminary Evaluation with Thread Motion Manager implemented
- 100% Goal : Thorough Performance evaluation with/without migration with private L1 architecture
- 125% Goal : Algorithm fine tuning and evaluations for L2 access latency minimization

3 Related Work

[1] looks at migration, at fine grained intervals as described above.Previous work does do migration either at OS intervals, [2] for process variation-aware application mapping combined with DVFS and [3] during thermal hotspots/emergencies. Apart from [1], there isn't any work to our knowledge, that looks at migration at finer-grained intervals than the OS scheduling interval.

4 Resources

- BLESS simulator for simulating the Chip Multi Processor
- SPEC 2006 benchmarks for evaluation. (We plan to pre-characterize SPEC 2006 benchmarks on Wattch/Simplescalar for incorporation in BLESS.)
- Our own PCs and laptops for carrying out the work

5 Schedule

- Week 1 : Understanding the BLESS CMP simulator. Identifying changes to BLESS to build a Thread Motion Manager
- Week 2,3 : Implementing and testing the Thread Motion Manager in BLESS
- Week 4 : Preliminary Evaluation to compare performance with and without migration
- Week 5 : Detailed Throughput evaluations to study and quantify performance degradation with migration
- Week 6 : Fine tune algorithms to minimize L2 access latency

We both will be working on the design and implemention of the thread motion manager in the simulator and the evaluation process. Once we are done with the design, we may suitably modularize and divide the implementation work between the two of us.

6 Milestone

Preliminary Evaluation with Thread Motion Manager implemented

References

[1] K. K. Rangan, G.-Y. Wei, and D. Brooks, "Thread motion: fine-grained power management for multi-core systems," in *ISCA '09: Proceedings of the 36th annual international symposium on Computer architecture*. New York, NY, USA: ACM, 2009, pp. 302–313.

[2] R. Teodorescu and J. Torrellas, "Variation-aware application scheduling and power management for chip multiprocessors," in *ISCA '08: Proceedings of the 35th International Symposium on Computer Architecture*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 363–374.

[3] A. K. Coskun, R. Strong, D. M. Tullsen, and T. Simunic Rosing, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," in *SIGMETRICS '09: Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM, 2009, pp. 169–180.