

Simulation of Asymmetric Access Time Cache in the Multi-Core Architecture

Julio C. S. Anjos, Felipe L. Severino
Cláudio R. Geyer, Philippe O. A. Navaux
UFRGS - Universidade Federal do Rio Grande do Sul
Instituto de Informática
Rio Grande do Sul, RS, Brasil
{jcsanjos,flseverino,geyer,navaux}@inf.ufrgs.br

August 21, 2009

Agenda

- 1 Motivation
- 2 Related Works
- 3 Goals
- 4 Methodology
- 5 Results
- 6 Conclusions
- 7 Bibliography

Motivation

- ✓ The demand for greater processing capacity propitiated the use of two or more CPUs simultaneously on a single chip;
- ✓ The development of a multi-core technology creates new questions about other devices, such as memory. Taking the use of one or more intermediate levels of memory, to minimize the length of time to access the main memory;
- ✓ This improvement is still insufficient for Tera-scale processing of applications.

Related Works

- ✓ Several studies have been made by manufacturers Intel and AMD;
- ✓ Several studies have been made by researchers
 - ① H. Aboudja and J. Simonson - Real-Time Systems Performance Improvement with Multi-Level Cache Memory [1];
 - ② M. A. Z. Alves *et al.* - Investigation of Shared L2 Cache on Many-Core Processors [2];
 - ③ H. C. Freitas *et al.* - NOC Architecture Design For Multi-cluster Chips [3];
 - ④ F. N. Sibai - On the performance benefits of sharing and privatizing second and third-level cache memories in homogeneous multi-core architectures [5] and
 - ⑤ Jie Tao *et al.* - Evaluating the Cache Architecture of Multicore Processors [6]
- ✓ To identify an ideal model that enables the use of various cores with a meaningful speed-up increase, lower consumption of energy and low heat generation by watts.

Goals

- ✓ Our goal is to make the simulation the use of cache memory with asymmetric latencies
- ✓ Using the Simics simulator
- ✓ Compare the asymmetric access time to the L2 cache with the symmetric access time
- ✓ To confront some of the obtained results with the experiment obtained by Jie Tao *et al.*

Methodology

- ✓ The simulations it will be applied NAS *Parallel Benchmark*, to be able to have the same simulation environment of Jie Tao *et al.*
- ✓ For this experiment only algorithms BT and CG are used, compiled with OpenMP;
- ✓ Simics is a simulator created by Virtutech;
- ✓ Caches Configurations

Cache	Size	Associative	Cycles	Update
L1	32 KB	4-way	4	write-through
L2	256 KB	8-way	10	write-back
L3	8MB	16-way	35	write-back

Latency asymmetric cache memory model

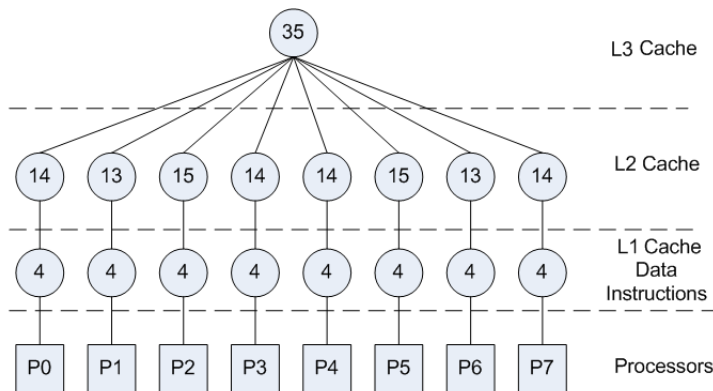


Figure: Latency asymmetric cache memory model

Latency cache memory symmetric model

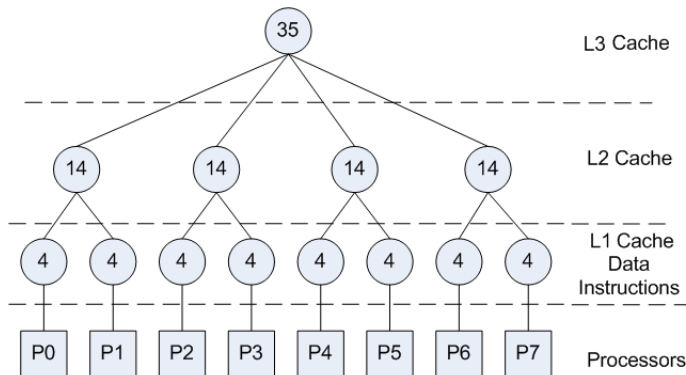


Figure: Latency cache memory symmetric model

CPU Cycles

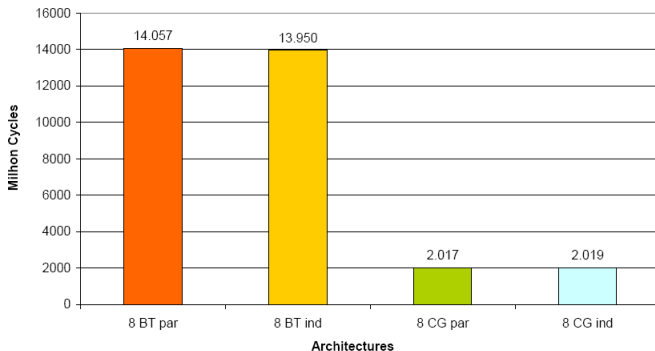


Figure: Execution cycles - standard deviation $\sigma < 0.00001\%$

CPU Steps

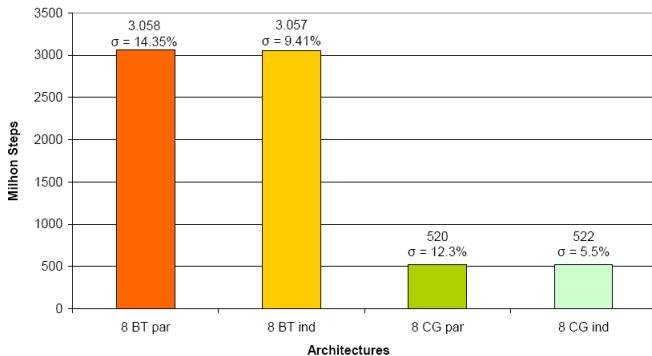


Figure: Execution step (instructions) numbers

L2 Read Miss (%)

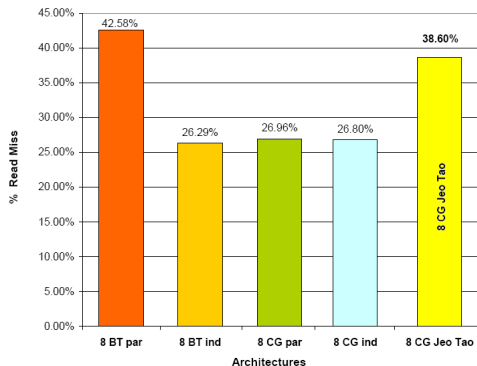


Figure: Read miss on L2 cache memory - BT standard deviation was $\sigma < 2.74\%$ and CG was $\sigma < 0.25\%$

L2 Write Miss (%)

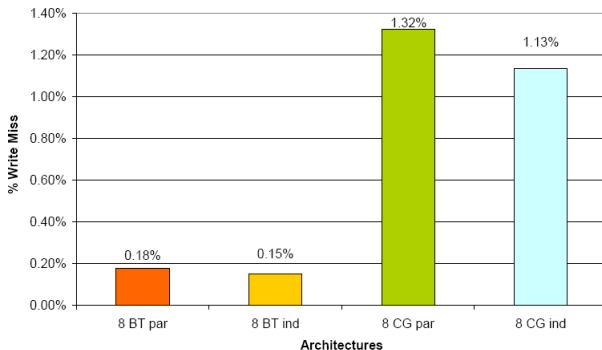


Figure: Write miss on L2 cache memory - standard deviation of $0.01\% < \sigma < 0.02\%$

L3 Read Hits (%)

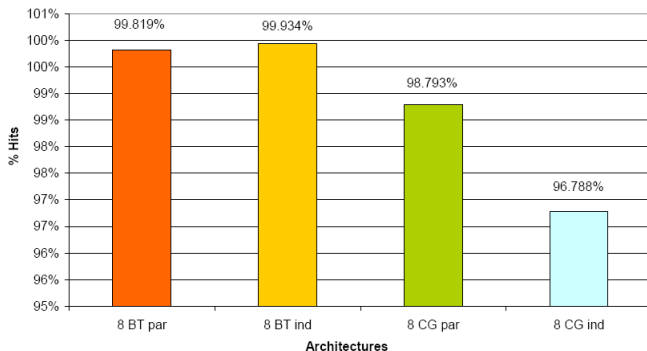


Figure: L3 Cache read/write for BT and CG benchmarks - standard deviation of $\sigma < 0,01\%$

L3 Write Hits (%)

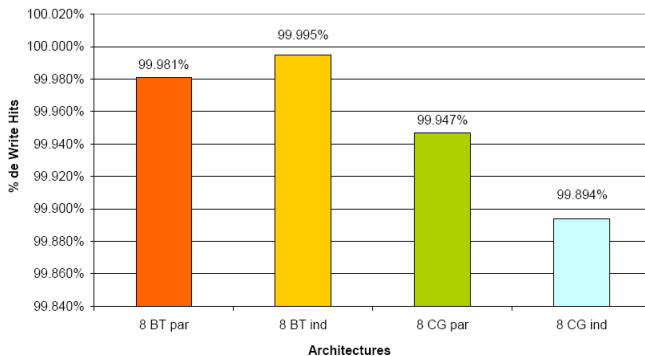


Figure: L3 Cache write hits - standard deviation of $\sigma < 0,01\%$

Conclusions

- ✓ On Figure Read miss on L2 cache memory it is suggested that the asymmetry time use, might allow a more efficient use of resources of hardware than the symmetric approach, however it will still need more tests with other types of applications.
- ✓ The fact of obtaining a inferior number of read miss to the one obtained by Jie Tao *et al.* [6] experiment of 38,6% for the CG application, compared with 26,96% for symmetric cache and 26,80% for shared cache with a lesser hardware, indicates that there are relevances in the results of performed experiment
- ✓ For future works, a simulation with 16 and 32 cores is still necessary for RubyGems simulator model [4] within Simics to be able to have the routing and also to execute other simulations models of the NAS, to observe if the behavior of the architecture will be the same with other applications formats.

Bibliography



H. Aboudja and J. Simonson.

Real-time systems performance improvement with multi-level cache memory.

CCECE '06. Canadian Conference on Electrical and Computer Engineering, pages 78–81, May 2006.



M. A. Z. Alves, H. C. Freitas, and P. O. A. Navaux.

Investigation of shared L2 cache on many-core processors.

In *ARCS '09 - 22th International Conference on Architecture of Computing Systems 2009*, volume 1, pages 21–30. VDE Verlag, 2009.



H. C. Freitas, P. O. A. Navaux, and T. G. S. Santos.

Noc architecture design for multi-cluster chips.

IEEE International Conference on Field Programmable Logic and Applications, pages 53–58, September 2008.



RubyForge.

Rubygems manuals.

<http://docs.rubygems.org/>.



F. N. Sibai.

On the performance benefits of sharing and privatizing second and third-level cache memories in homogeneous multi-core architectures.

Microprocess. Microsyst., 32(7):405–412, 2008.



J. Tao, M. Kunze, and W. Karl.

Simulation of Asymmetric Access Time Cache in the Multi-Core Architecture

Julio C. S. Anjos, Felipe L. Severino
Cláudio R. Geyer, Philippe O. A. Navaux
UFRGS - Universidade Federal do Rio Grande do Sul
Instituto de Informática
Rio Grande do Sul, RS, Brasil
{jcsanjos,flseverino,geyer,navaux}@inf.ufrgs.br

August 21, 2009