

Reducing Power Dissipation in Multi-Core Processors using Effective Core Switching

Vijayalakshmi Saravanan¹
WINCORE Lab, Ryerson University
Toronto, Canada
Email: vsaravan {at} rnet.ryerson.ca

Aniket Shivam² & Sudeep Chauhan³
CSE, National Institute of Technology,
Srinagar, Uttarakhand, India

Abstract—The recent development of microprocessors has raise up the demand for high-performance and fast processing computing systems capable of performing multiple tasks. Multi-core processors are being accepted to achieve higher performance but maintaining sustainable power consumption is still an issue. Hence, the need for developing alternatives for modern CMPS is highly indispensable. In this paper, therefore, we have proposed a method of effective core switching based on the processes workload, so as to maintain the level of performance, with significant reduction in the power dissipation of a multi-core processor. Theoretical results are provided, showing that our proposed approach can be efficient in terms of power consumption, based on various power-performance metrics.

Keywords- Multi-Core, power dissipation, core switching.

I. INTRODUCTION

The multi-core architecture being used now-a-days, from our mobiles, laptops to high-end servers are a viable solution to achieve high performance, but under the constraints associated with power bounds. Maintaining the power consumption of processors at an acceptable level is still a challenge. As an example, the size of the transistor is defined to be around 22 nm. As the size decreases beyond 30 nm, the dynamic voltage frequency scaling will be rendered less useful. Smaller the size, lesser will be the absolute maximum operating voltage although the lower limit voltage will be constant at 2.3 times the threshold voltage. Hence, one needs to develop better alternatives for managing the energy consumption. The technique proposed for switching cores on/off effectively can be used as a potential successor to the DVFS technique.

IT organizations aim for higher performance while sustaining acceptable power consumption and heat. The ongoing progress in processor designs has enabled servers to continue delivering increased performance, which in turn helps fuel the powerful applications that are crucial for rapid business growth. However, increased performance incurs a corresponding increase in processor's power consumption and heat is a consequence of power use. Huge power dissipation

by data center servers increases the demand for cooling equipment.

The overall model that we are going to implement depends upon turning on-off the multiple cores effectively based on the workloads of the processes to be executed and deciding the best configuration for a window of processes. Every core of a chip multiprocessor is an independent unit of processing. To take full advantage of chip multiprocessors, the system must use the execution characteristics of each application to predict its future processing needs and then schedule it to the number of cores that matches those needs.

II. BACKGROUND

The two adopted industrial practices for power reduction are gating based and voltage/frequency based. Both the techniques are applicable on a single core and are rely on program behavior for power reduction. But these techniques suffer from some drawbacks. For instance, in clock gating technique, the gating circuitry has power and area overheads like dissipation of power for inactive blocks. Additionally, considerable amount of power is spent on moving data over unused, gated off portions of the chip for long distances. Large unused portions of the chip dissipate leakage power too.

The method of Dynamic voltage and frequency scaling (DVFS) aims to reduce power consumption by adapting to changing workloads' voltage and frequency. Voltage/frequency scaling-based techniques suffer from similar limitations. But these results are subject to issues involving inability of the slow, off chip voltage regulators in adjusting to different voltage levels in small intervals. But a significant improvement has been observed in processor energy dissipation owing to the ability to switch between cores and power down the unused cores causing leakage.

The energy minimization of the homogeneous chip multiprocessor is based on the factor that the model minimizes the power dissipation, given a throughput constraint. Adjusting the number of active cores helps in optimizing the performance without increasing the power consumption. The energy dissipation of cores in a chip multiprocessor in a time interval can be defined as the summation of the energy consumption of each of the cores. In our work, we considered that unused cores should be completely powered down, rather

than left idle. Thus, unused cores suffer no static leakage or dynamic switching power. This does, however, introduce latency for powering a new core up.

By our estimation, a processor core can be switched on in approximately one thousand cycles of the 2.1GHz clock. The time required by power buses to charge and stabilize determines this time for switching-on. Also, it has been concluded that switching overhead involved in switching core at operating system scheduling advantages has least impact on performance.

The following metrics have been considered to measure the power and performance of the architectures: Instructions per second (IPS), instructions per cycle (IPC), IPS per Watt, $(IPS)^2$ per Watt, and $(IPS)^3$ per Watt.

- IPC: The number of instructions executed per cycle.
- IPS: The number of instructions executed per second. This is also known as peak instruction throughput.
- IPS per Watt: This is generally referred to as the energy metric (denoted E). Energy/Instruction or its inverse IPS/W can be used as power-performance metric.
- $(IPS)^2$ per Watt: Generally, the energy-delay product is inversely proportional to IPS^2/W
- $(IPS)^3$ per Watt: The energy-delay-square product is inversely proportional to IPS^3/W .

It can be a fair assumption that a low power consuming processor may give poor performance. An optimum level of power-performance can be obtained by taking delay into the account. Power efficiency metrics like Energy-Delay Product (EDP) and Energy-Delay² Product (ED^2P) will be used for simulation of architectures with independent power supply voltage alongside IPS/W, IPS^2/W and IPS^3/W metrics.

The choice of metric depends on the processor class. For example, processors that use frequency scaling or capacitance scaling can use E metric. EDP or ED^2P metric can be used to compare power-performance efficiencies on processors which use voltage scaling as the primary method. High end machines like servers primarily use ED^2P metric.

III. OUR APPROACH

We have implemented a model in which the workload of the incoming processes will be the deciding factor for the number of cores that we want active for processing rather than all cores to be active at all time, so as to save the power consumption. All processes need not to be executed at the full potential of the processor. Suppose, for a 8-core processor, some processes of low intensity or low instruction count may require least number of cores, i.e. 2 cores for its execution and hence switching off the other 6 cores will not hamper the performance but will consume less power than the power consumed by all 8 cores, if they happen to be active for this low intensity process. This phenomenon is the basis of our approach for switching cores on/off.

The other crucial part of the approach is to use a ‘window’ of processes of an optimal size to decide which configuration of active cores will be best suited for them. This strategy is used to save the performance loss in terms of the wastage in clock cycles that are used during switching of cores which in turn delays the instruction execution. Therefore to prevent performance loss due to frequent switching, we choose the configuration for a set of processes i.e. a window, instead of switching cores on/off for every process.

For our results, we have taken the window of 3 processes, which seems to be optimal as neither it is too large that choosing a configuration for them becomes ineffective because in a large window, number of high intensity processes will increase which will force us to use the full core configuration for every window and hence reducing the importance of the motive behind this strategy nor it is too small that we have to change the configuration of cores for every next process.

We have taken an 8-core processor for simulation and results analysis as it is the common multi-core processor that is being used from the personal laptops to the servers used in the data centers.

A. Functionality of Our Proposed System

The algorithm that we have used to decide the number of cores needed for a window depends on the majority type of processes in a window like if there are two medium intensity process and one high intensity process, we will switch on 4 cores only, for all three processes. The full description of the algorithm is defined below.

For window size of 3 processes on 8 core processor:
(**L** - Low intensity process (around 2 million instructions), **M** - Moderate intensity process (around 4 million instructions), **H** - High intensity process (around 8 million instructions).

We have simulated and then analyzed results to deduct that low intensity process need 2 cores, moderate intensity process need 4 cores, and high intensity process need 8 cores.

1. LLM (order may shuffle as in MLL or LML) - 2 cores
2. MML - use 4 cores

3. MMH - use 4 cores
4. LLH - use 4 cores
5. LMH - use 4 cores
6. LHH or MHH - use 8 cores
7. LLL - 2 cores, MMM- 4 cores, HHH- 8 cores

TABLE I. Simulation analysis for all three process types on various no. of cores

	No. of cores	Instruction Count	Simulation time(in cycles)	Power Dissipation	IPC	Simulation Time	IPS/W	IPS ² /W	IPS ³ /W
Low intensity process	2	1,918,454	1,129,294	359	1.7	10	534.39	102.52	19,667.98
Low intensity process	4	1,694,445	453,325	366	3.74	9	514.40	96.85	18,233.66
Low intensity process	8	1,632,853	226,336	378	7.21	11	392.70	58.29	8,653.09
Medium intensity process	2	3,833,336	2,250,866	359	1.7	19	561.99	113.38	22,875.75
Medium intensity process	4	3,396,974	905,988	366	3.75	19	488.49	87.34	15,614.74
Medium intensity process	8	3,282,942	453,325	378	7.24	20	434.25	71.28	11,700.59
High intensity process	4	7,294,899	1,890,253	366	3.86	39	511.06	95.59	17,880.64
High intensity process	8	7,593,404	1,026,780	378	7.4	45	446.41	75.33	12,711.01

Based on table I, we can see the instruction count is the basis for defining the process' intensity. The simulation data is collected for these process types on various configurations of cores. The data is then analyzed to calculate Instructions per second (IPC) and various power-performance metrics (IPS/W, IPS²/W, IPS³/W) to measure performance vs power for each case possible as per the proposed method of deciding no. of active cores.

Fig. 1 describes the Instructions per Second for various numbers of cores and types of processors.

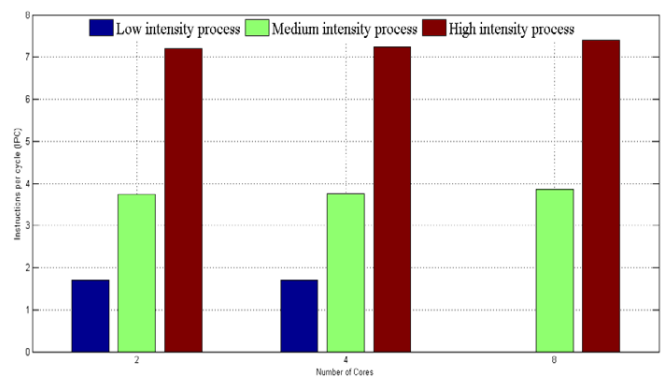


Fig. 1: IPC vs Number of Cores for various intensity processes

IV. SIMULATION ANALYSIS

Our approach needed to be analyzed in a way, so as to give the results that are expected to come when the processor using our proposed technique is implemented.

We also need to compare the results for power and performance with the processor using the techniques that are already implemented in current multi-core processors. Therefore, we have taken set of 21 processes, covering each of the cases that our algorithm covers, and chose the number of cores required for each window of process and also calculated the power-performance metrics for the whole set of processes. We will then compare our results with the power-performance statistics that a current 8-core processor will give, which is being used to it full ability i.e., all 8 cores being active at all the time for the same set of 21 processes.

The tables II, III, IV will cover each aspect of analysis for our test case mentioned above. It shows the instruction count of each process which then decides its intensity. Then our algorithm as discussed before is applied to them and the result for the configuration of core appropriate for them is mentioned. The metrics like IPC, IPS and all e-metrics are calculated with reference to table I and hence compared to generate our final result.

Table II. Proposed solution

	Instruction count(millions)	Process Intensity	Active cores
Process Window 1	2.2	Medium	2
	1.1	Low	2
	1.2	Low	2
Process Window 2	2.3	Medium	4
	1.7	Low	4
	3.3	Medium	4
Process Window 3	3.5	Medium	4
	2.9	Medium	4
	7.7	High	4
Process Window 4	1.3	Low	4
	1.5	Low	4
	5.6	High	4
Process Window 5	1.2	Low	4
	2.6	Medium	4
	6.6	High	4
Process Window 6	6.7	High	8
	7.2	High	8
	3.3	Medium	8
Process Window 7	4.2	High	8
	5.1	High	8
	7.8	High	8

Table II particularly shows the execution of our proposed approach for using effective core switching as per the processes workloads.

Table III defines average value for each metrics like IPC, IPS, IPS/W, IPS^2/W and IPS^3/W while using our approach.

Table IV defines average value for each metrics like IPC, IPS, IPS/W, IPS^2/W and IPS^3/W when using the current techniques used in 8-core processors.

Table III. Metrics value according to our approach

Instruction count(millions)	Process Intensity	Active cores	Power Dissipation(watts)	IPC	IPS	IPS/W	IPS^2/W	IPS^3/W
2.2	Medium	2	359	1.7	201755	561.99	113.38	22875.75
1.1	Low	2	359	1.7	191845	534.39	102.52	19667.98
1.2	Low	2	359	1.7	191845	534.39	102.52	19667.98
2.3	Medium	4	366	3.75	178788	488.49	87.34	15614.74
1.7	Low	4	366	3.74	188272	514.4	96.85	18233.66
3.3	Medium	4	366	3.75	178788	488.49	87.34	15614.74
3.5	Medium	4	366	3.75	178788	488.49	87.34	15614.74
2.9	Medium	4	366	3.75	178788	488.49	87.34	15614.74
7.7	Low	4	366	3.86	187049	511.06	95.59	17880.64
1.3	Low	4	366	3.74	188272	514.4	96.85	18233.66
1.5	Low	4	366	3.74	188272	514.4	96.85	18233.66
5.6	High	4	366	3.86	187049	511.06	95.59	17880.64
1.2	Low	4	366	3.74	188272	514.4	96.85	18233.66
2.6	Medium	4	366	3.75	178788	488.49	87.34	15614.74
6.6	High	4	366	3.86	168742	446.41	75.33	12711.01
6.7	High	8	378	7.4	168742	446.41	75.33	12711.01
7.2	High	8	378	7.4	168742	446.41	75.33	12711.01
3.3	Medium	8	378	7.24	164147	434.25	71.28	11700.59
4.2	High	8	378	7.4	168742	446.41	75.33	12711.01
5.1	High	8	378	7.4	168742	446.41	75.33	12711.01
7.8	High	8	378	7.4	168742	446.41	75.33	12711.01
			Average Metrics value:	4.51	180150.95	488.84	88.43	16045.14

Table IV. Metrics value for present techniques

Instruction count(millions)	Process Intensity	Active cores	Power Dissipation(watts)	IPC	IPS	IPS/W	IPS2/W	IPS3/W
2.2	Medium	8	378	7.24	164147	434.25	71.28	11700.59
1.1	Low	8	378	7.21	148441	392.7	58.29	8653.09
1.2	Low	8	378	7.21	148441	392.7	58.29	8653.09
2.3	Medium	8	378	7.24	164147	434.25	71.28	11700.59
1.7	Low	8	378	7.21	148441	392.7	58.29	8653.09
3.3	Medium	8	378	7.24	164147	434.25	71.28	11700.59
3.5	Medium	8	378	7.24	164147	434.25	71.28	11700.59
2.9	Medium	8	378	7.24	164147	434.25	71.28	11700.59
7.7	Low	8	378	7.4	168742	446.41	75.33	12711.01
1.3	Low	8	378	7.21	148441	392.7	58.29	8653.09
1.5	Low	8	378	7.21	148441	392.7	58.29	8653.09
5.6	High	8	378	7.4	168742	446.41	75.33	12711.01
1.2	Low	8	378	7.21	148441	392.7	58.29	8653.09
2.6	Medium	8	378	7.24	164147	434.25	71.28	11700.59
6.6	High	8	378	7.4	168742	446.41	75.33	12711.01
6.7	High	8	378	7.4	168742	446.41	75.33	12711.01
7.2	High	8	378	7.4	168742	446.41	75.33	12711.01
3.3	Medium	8	378	7.24	164147	434.25	71.28	11700.59
4.2	High	8	378	7.4	168742	446.41	75.33	12711.01
5.1	High	8	378	7.4	168742	446.41	75.33	12711.01
7.8	High	8	378	7.4	168742	446.41	75.33	12711.01
Average Metrics value:				7.29	161410	427.011	69.111	11214.8

V. RESULT ANALYSIS

A. Performance-only Metrics

1) IPC (Instructions per cycle)

The comparison in performance-only metrics i.e., IPC for both our method and present method is drawn in Fig. 2. We can see that because of lowering the number of active cores from 8 cores to 4 and 2 cores for some set of processes, IPC is reduced by 38%.

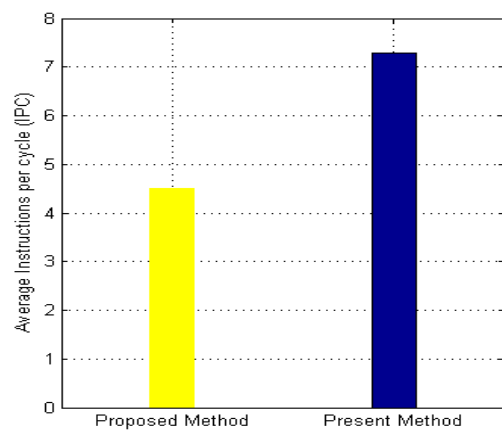


Fig. 2 IPC Comparison

2) *IPS (Instructions per second)*

The comparison in performance-only metrics i.e., IPS is drawn in graphs Fig.3. We can see that IPS is improved by 11.6% in our approach. The improvement is due to the fact that a process may be assigned more number of cores than are required for execution, hence, resulting in idle cores. Even though multi-threading occurs, but threading can occur only to some extent depending upon each program. Therefore, it does not increase level of parallelism beyond a limit and in turn does not reduce the total simulation time.

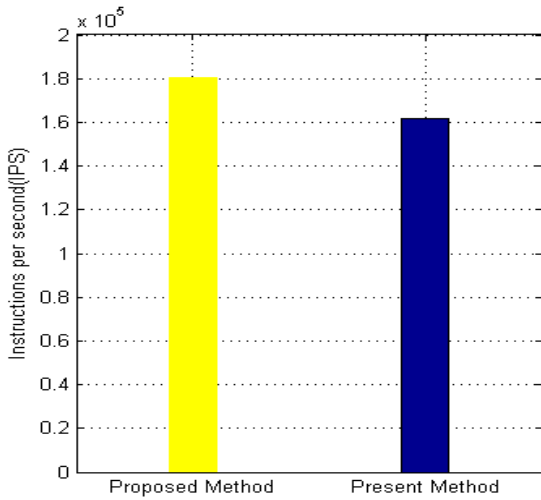


Fig.3 IPS comparison

B. *Power-Performance Metrics*

1) *IPS/W (E-Metrics)*

The comparison in power-performance metrics i.e., IPS/W is drawn in Fig.4. We can see that there is an improvement of 14.3% in this metrics for our approach. This proves that performance will improve across processors using frequency scaling or capacitance scaling technique, ex. clock gating or adaptive micro-architectural techniques.

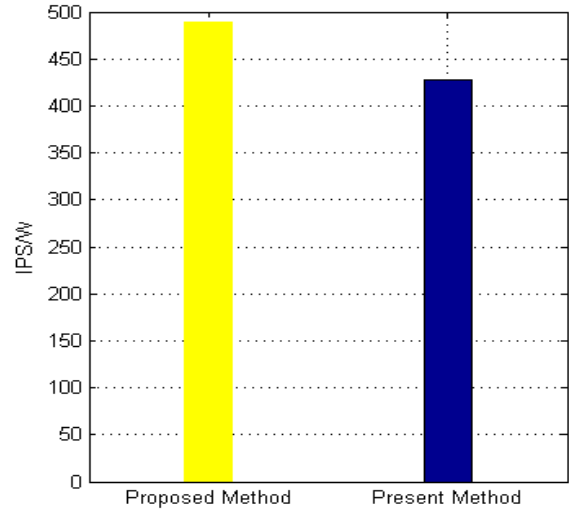


Fig.4. E-Metrics comparison

2) *IPS²/W (EDP Metrics)*

The comparison in power-performance metrics i.e., IPS²/W is drawn in Fig. 5. We can see that there is an improvement of 27.9% in this metrics for our approach. Hence, the performance will improve across processors employing voltage scaling technique.

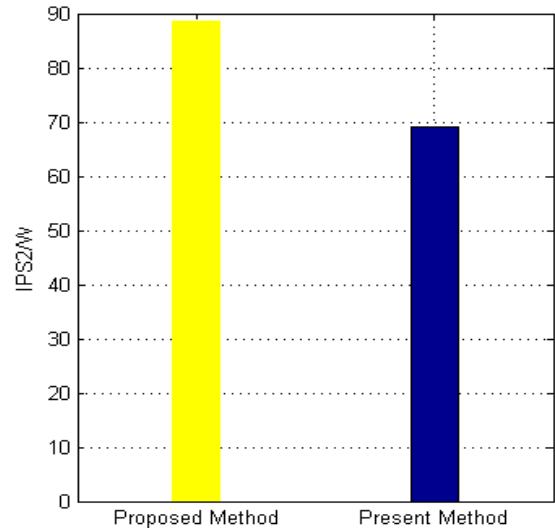


Fig.5. EDP comparison

3) IPS^3/W (ED²P Metrics)

The comparison in power-performance metrics i.e., IPS^3/W is drawn in Fig. 6. We can see that there is an improvement of 43.1% in this metrics for our approach. This shows that our approach will perform better across high-end machines such as servers.

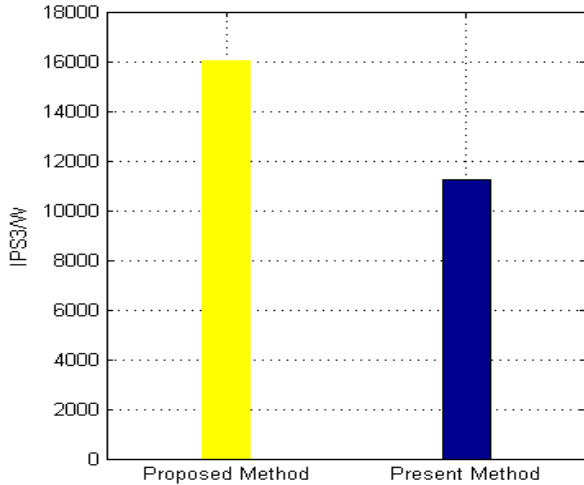


Fig.6. ED²P comparison

Table V: Percentage improvement in various metrics

	Avg. Metrics value for proposed approach	Avg. Metrics value for present techniques	% Improvement
IPC	4.5	7.3	-38.4
IPS	180151	161410	11.6
IPS/W	488	427	14.3
IPS ² /W	88.4	69.1	27.9
IPS ³ /W	16045.1	11214.8	43.1

VI. CONCLUSION

We have proposed an effective core switching approach in multi-core processors for reducing the power dissipation. Our results show that, if performance only metric is considered, our

approach is lacking behind when IPC is considered but is showing some improvement in terms of IPS. By applying our proposed method of effective core switching, we found that our method achieve better E-metrics, EDP and ED²P performance.

We believe that our proposed technique is a viable solution for designing multiprocessors whose transistors can be made of a sub 30nm process.

As future work, we intend to develop new optimizing algorithms for the implementation of core switching based on workload classification and assessment.

REFERENCES

- [1] S. Vijayalakshmi, A. Anpalagan, Isaac Woungang, D. P. Kothari, "Power Management in Multi-core Processors using Automatic Dynamic Pipeline Stage Unification", Proc. of the ACM Intl. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2013), Toronto, Canada, pp. 120-127, July 7-10, 2013.
- [2] S. Vijayalakshmi, A. Anpalagan, D. P. Kothari, Isaac Woungang, Mohammad S. Obaidat, "A Comparative Simulation Study on the Power-Performance of Multi-Core Architecture" (Accepted June 27, 2014), Journal of Supercomputing, Springer, (Impact Factor: 0.917). In Press.
- [3] Ghasemazar, M., Pakbaznia, E., and Pedram, M. 2010. Minimizing energy consumption of a chip multiprocessor through simultaneous core consolidation and DVFS. In Proceedings of the IEEE International Symposium on Circuits and Systems. 49--52.
- [4] E. Grochowski and M. Annavaram, "Energy per instruction trends in Intel microprocessors," Technology@Intel Magazine, pp. 1–8, Mar. 2006
- [5] Mohammad Abdul Qayum "Heterogeneous Chip Multiprocessors: A Survey", ECEN 6253: Advanced Topics in Computer architecture.
- [6] James Balfour , William Dally , David Black-Schaffer , Vishal Parikh , JongSoo Park, An Energy-Efficient Processor Architecture for Embedded Systems, IEEE Computer Architecture Letters, v.7 n.1, p.29-32, January 2008
- [7] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen. Single-ISA Heterogeneous Multi-core Architectures: The Potential for Processor Power Reduction. In International Symposium on Microarchitecture, Dec. 2003.
- [8] David Levinthal. Performance Analysis Guide for Intel® Core™ i7 Processor and Intel® Xeon™ 5500 processors.
- [9] J. Sharkey, "M-Sim: A Flexible, Multi-threaded Simulation Environment." Tech. Report CS-TR-05-DP1, Department of Computer Science, SUNY Binghamton, 2005. <http://www.cs.binghamton.edu/~jsharke/m-sim>.
- [10] W. Kim, M. Gupta, G.-Y. Wei, and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in International Symposium on High-Performance Computer Architecture, February 2008.