Proceedings of The National Conference
On Undergraduate Research (NCUR) 2016
The University of North Carolina at Asheville
Asheville, North Carolina
April 7 – 9, 2016

Implementing Power Optimization Methodologies to Improve Energy Efficiency of Multi-Core Processors

Mohamed Azard Rilvan
Computer Science Department
Southern Connecticut State University
New Haven, CT 06515USA

Shervin Hajiamini
Electrical Engineering and Computer Science Department
[1]Washington State University
Pullman, WA 99163 USA

Lauren Hiland
Computer Science Department
Western Oregon University
Monmouth, OR 97361 USA

Faculty Advisor: [1]Dr. Behrooz Shirazi

Abstract

Chip multi-core processors consume a large portion of system's total power. A core contains periods of internal computation and communication during an execution of an application. In the periods of communication, core requests data from the memory during which it remains in an idle state until it receives the requested data. The waiting time of the cores provide an opportunity to save energy by reducing the voltage and frequency at the cost of performance degradation. Dynamic voltage and frequency scaling (DVFS) is a technique that predicts voltage and frequency based on the computation and communication characteristics of a running application on the multi-core system. We integrate a history-based DVFS methodology with Gem 5, a full system cycle accurate simulator, to investigate the maximum energy efficiency gained by this methodology for various computation and communication intensive applications. We measure the intensity of CPU-bound and memory-bound portions of the applications periodically during the runtime. The CPU-bound portions are the busy utilizations of the processor cores, and the memory bound portions are the utilization of the memory, that is the amount of the data requests the memory receives from the CPUs (cores). We aim to compute the prediction accuracy, which is the tracking distance between the voltage and frequency (predicted by the methodology) and the actual CPU-bound/memory-bound intensity of a program in each time period. We compare the proposed history-based DVFS methodology with a simple history based DVFS methodology. The latter adjusts the voltage and frequency solely based on the core's busy utilization for the past few time periods, while our proposed DVFS methodology adjusts voltage and frequency based on the core's busy utilization as well as the memory utilization for the current and past few time periods. We evaluate these methodologies in terms of energy savings (energy reduction) and execution time penalty (performance degradation) compared to the non-DVFS setup.

Keywords: Dynamic voltage and frequency (DVFS), history based DVFS, CPU and memory bound operations

1. Introduction

Power consumption of microprocessors has led to a major concern regarding the battery life and the temperature of processors[1, 2]. The increase of the temperature in a processor leads to cooling cost increases, impacting the operational costs. High power consumption would also affect the environment concerning the increase in the production of the electricity.

Dynamic voltage and frequency scaling (DVFS) is a power management technique, and it adjusts the voltage and frequency of a processor based on the intensity of an executed workload[2]. By implementing DVFS methodologies for multi-core processor architecture during idle cycles, voltage and frequency can be lowered to reduce the power dissipation with minimal impact on performance degradation.

History based DVFS methodologies improve the energy efficiency of a multi-core system by predicting the future behavior of the workload and adjusting the voltage and frequency for each core during runtime of the workload (benchmark). In this paper we compare and evaluate two history based DVFS methodologies for various benchmarks to adjust voltage and frequency based on the amount of busy and idle portions of the core in a time period. The improved history based DVFS that we propose, considers core's and network's history of busy utilization in order to predict per-core voltage and frequency levels during the execution time of a benchmark. We compare the improved history based DVFS with a simple history based DVFS; the latter adjusts the voltage and frequency solely based on core's history of the busy utilization. The paper is organized as follows. Section 2 describes related works that were previously studied. In section 3 we explain the two history based DVFS methodologies. In section 4 we describe the experimental setup. In section 5 we analyze our experimental results. We conclude our results in section 6.

2. Related Work

Several previous studies investigated history based DVFS methods to adjust voltage and frequency of the processors and their interconnecting network. Rahimi et al. proposed a history based dynamic voltage scaling scheme (DVS) for NoC architecture by estimating future network loads based on past and current link utilizations[3]. According to the predicted link utilization level, link voltage is adjusted to higher/lower. Rahimi et al. reported 17% energy delay saving from this proposed technique compared to a system with a fixed voltage at 1.0 vdd. Ababei et al. proposed that a history based DVFS is applied at the router level[5] by considering link utilization and buffer utilization to measure the network congestion. Link utilization and buffer utilization were predicted based on the weighted average of previous and current corresponding utilizations. Depending on the congestion of neighboring routers, decisions are made to predict buffer utilization and link utilization, and the DVFS algorithm determines whether to increase or decrease the frequency of a router to handle the load. Shang et al. incorporated similar history based DVFS approach with links to minimize power consumption of interconnection networks[6]. The results show 4.6-fold increase in power savings on average, with a small impact on performance. Liang et al. proposed an adaptive controller mechanism for the on-chip communication network and compared with a fixed high frequency network[4]. The proposed technique considers past load (number of data packets) and frequency in the network, and adjusts the voltage and frequency of the entire network to operate below a saturation point, which is defined as the maximum network load when packet delay is stabilized in transmission. Based on the comparison of the saturation point and the predicted network load, the frequency is increased or decreased. The adaptive controller mechanism was found to be more power efficient compared to a fixed high frequency network. Isci et al. and Choi et al. considered history intervals to capture the behavior of the computational workload to adjust the voltage and frequency of the processor[7, 8]. In [7] a history table is used to track the patterns of previously observed phases to deduce the behavior of the future phase during the runtime. The predicted behavior from the history table is used to translate to one of the 6 DVFS setting. The chosen setting is then used to adjust the voltage and frequency of the processor. Energy delay product (EDP) is used to evaluate the performance of the DVFS approach. The experiment results show that by using the proposed history table, the EDP was improved. A frame based history is proposed in [8], predicts the computational workload for the incoming frame to adjust the voltage and frequency of the processor, so that an efficient amount of computing power could be provided to decode the frame of a MPEG decoder. Computational workload is predicted by considering average workload of a window size (past 6 frames) and a weighting factor. Based on the predicted computational workload, decoding time is predicted for the next frame and voltage values are mapped to the selected frequency value. A prediction based shutting down method proposed in [10], predicts the idle cycles of the CPU by considering accumulative average of previous idle cycles. The predicted duration value of the idle cycles is compared with a threshold value to determine

whether the CPU must keep running or enter the sleep state to save power. In [11], utilization of the core is predicted by using a table based predicting structure. The table contains a pattern history length of two entries which is used to predict the level of busyness in the next phase. According to the predicted core utilization, one of five available voltage and frequency combinations is selected.

3. History Based Voltage and Frequency Scaling Techniques

In this section we describe two history based DVFS methodologies which are used to adjust voltage and frequency according to the amount of busy and idle cycle in previous time periods (windows). These techniques are referred to as history based DVFS and improved history based DVFS. History based DVFS technique predicts core utilization for the future window based on the core utilization in the past few windows. The improved history based DVFS technique predicts the core utilization for the future window based on the core utilization and network utilization of the current and past windows.

3.1. History Based DVFS

In the simple history based DVFS technique we compute the core busy utilization for the next window in order to predict voltage and frequency in that window. The predicted core utilization is characterized by,

$$U_{predicted}{}^{core} = \frac{1}{num\_win} \sum_{i=1}^{num\_win} U_i{}^{core} \quad (1)$$

num_win is the number of past windows and $U_i{}^{core}$ is the utilization of the core in the i[th] window. Based on a mapping table, voltage and frequency is adjusted similar to [9], where the core utilization falls in a category within the table, and the corresponding voltage and frequency pair is assigned to that core.

3.2. improved History Based DVFS

Improved history based DVFS predicts voltage and frequency of the core, based on predicted utilizations of the core and memory bound network traffic in the current as well as the previous windows. Past utilizations for the core or network,

$$U_{past}{}^{core/netowrk} = \frac{1}{num\_win} \sum_{i=1}^{num\_win} U_i^{core/network} \quad (2)$$

where $U_{past}{}^{core/network}$ denotes the average of the past busy utilization for the core or network. *num_win* is the number of windows and $U_i^{core/network}$ is the processor/network utilization measure for the i[th] window. Core utilization is based on the core's busy cycles and network utilization is based on the amount of load and store requests in the core's local queue.

Predicted core/network utilization is the weighted average of the utilization in the current window and average of the utilizations in the past few windows,

$$U_{prediced}{}^{core/network} = W * U_{current}{}^{core/network} + (1 - W) * U_{past}{}^{core/network} \quad (3)$$

where $U_{predicted}{}^{core/network}$ denotes predicted processor/network utilization. *W* is a weight given to the current utilization of the processor/network and it is proportional to the core/network utilization in the current window. Voltage and frequency pair for the next window is derived using the weighted average of the predicted core utilization and the predicted network utilization, according to,

$$core^{V/Flevel} \propto T * U_{predicted}{}^{core} + (1 - T) * (1 - U_{predicted}{}^{network}) \quad (4)$$

$core^{V/F\ level}$ is the predicted core's voltage and frequency level. $U_{predicted}{}^{core}$ and $U_{predicted}{}^{network}$ are the predicted utilization for the core and network, respectively. T is a constant which is the overall ratio of the computation to communication characteristics of the benchmark. The term $1 - U_{predicted}{}^{network}$ helps to capture the computation workload through the pending load/store requests that ultimately impacts the core's busy utilization.

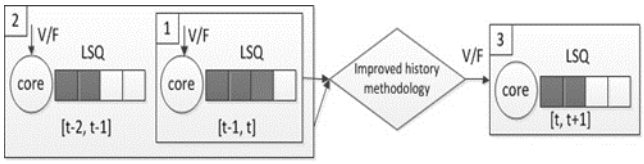

Figure 1. Improved history based DVFS uses the current window (1) and the past two windows (2) as a history to predict the core's voltage and frequency in the next window (3).

As shown in figure 1, the prediction for the next window, [t, t+1], is based on the core utilization and the number of the pending loads/stores in the queue (LSQ) for the current window, [t-1, t], as well as the history windows, [t-2, t].

4. Experimental Setup

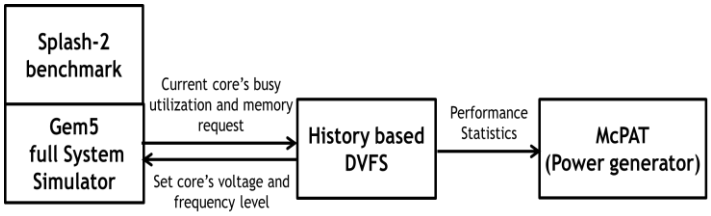

Figure 2. Experimental setup to obtain performance statistics.

We implement our DVFS algorithms in Gem5[10]. Gem5 is a full system simulator designed to capture the runtime data for the processor and network. In our simulations, we use a system with 64 alpha cores running in a Linux based operating system. We have used three SPLASH-2 benchmarks, RADIX, FFT, and WATER; that have communication and computation characteristics[11]. Processor level statistics obtained from Gem5 are incorporated in to McPAT (Multi-core Power, Area, and Timing) to obtain processor level power consumption data. As shown in figure 2 the computation and memory reference patterns of cores running the benchmarks optimizes the selection of voltage and frequency levels predicted by the history based algorithm. After performing the DVFS, the performance output of the runtime of the benchmark is fed to McPAT for estimating the power saving. To track the predicted V/F levels with the actual busy utilization of the cores we discretize the utilization into six categories (Figure 4)[12]. If a core's utilization falls in a range for a given window, that core runs with the corresponding V/F level pair (computed from equation 4) for the next window.

Benchmarks are run on Gem5 for each experimental window size and these simulations are repeated for up to and above hundred thousand window sizes. Voltage and frequency is adjusted based on the predicted utilizations of the core and network traffic of each window as described in the section 3.

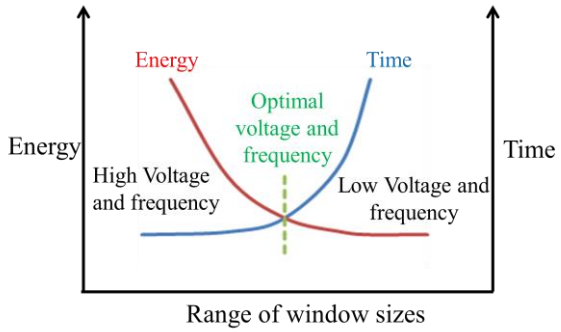

Figure 3. Selecting the optimum window size which gives a good balance between energy and time.

Voltage (V), Frequency (GHz)	CPU utilization%
0.5,1.25	< 50
0.6,1.5	>= 50 & < 60
0.7,1.75	>=60 & < 70
0.8,2.0	>= 70 & < 80
0.9,2.25	>=80 & < 90
1.0,2.5	>=90 & <= 100

Figure 4. V/F levels and the corresponding core's utilization.

We evaluate the performance of the history based DVFS and improved history based DVFS techniques by considering energy delay product (EDP). EDP weighs energy savings and time equally. Thus, a decrease in energy is "canceled" by a proportional increase in time. To reduce the EDP, the relative energy decrease must be greater than the relative time increase[13]. Among the experimented number of window sizes we select a window size range within which a minimum EDP is gained. A minimum EDP represents an optimum voltage and frequency. For certain number of windows the history gives minimum execution time and maximum energy and vice versa. An optimal windows size corresponds to a voltage and frequency pair that strikes a good balance between the execution time and energy consumption (Figure 3).

We compare the minimum EDP which was achieved within the range of window sizes of history based DVFS technique with the minimum EDP which was achieved within the range of window sizes of improved history based DVFS technique.

5. Results

In this section we analyze our experimental results for RADIX, WATER, and FFT benchmarks. The results help us to identify the effect of the window size ranges (in percentage) of history DVFS and improved history based DVFS techniques which impact the energy efficiency of the multi-core system. Table 1 compares ranges of window size within which a minimum EDP is gained for history based DVFS and improved history based DVFS techniques. These percentages of range of window sizes indicate that history based DVFS methodology achieves a minimum EDP within a larger window size while improved history based DVFS technique achieves a minimum EDP within a shorter window size.

Table 1. Range of percentage of window size within which minimum EDP is gained.

Methodology	FFT	WATER	RADIX
History	68-100	66-100	70-100
Improved history	7-10	17-25	20-28

Table 2. Comparison of energy efficiency of history based DVFS methodologies for small window size across the benchmarks.

Benchmark	Methodology	Power	Time	Energy	EDP
RADIX	History	37.05	0.0457	1.6932	0.0774
	Improved history	30.73	0.0451	1.3858	0.0625
FFT	History	45.93	0.0224	1.0266	0.0229
	Improved history	41.83	0.0215	0.9010	0.0194
WATER	History	61.22	0.0316	1.9345	0.0611
	Improved history	57.68	0.0316	1.8229	0.0576

Table 3. Comparison of energy efficiency of history based DVFS methodologies for large window size across the benchmarks.

Benchmark	Methodology	Power	Time	Energy	EDP
RADIX	History	29.88	0.0455	1.3591	0.0618
	Improved history	31.69	0.0464	1.4694	0.0681
FFT	History	40.12	0.0218	0.8745	0.0191
	Improved history	39.94	0.0219	0.8767	0.0192
WATER	History	56.71	0.0317	1.7950	0.0568
	Improved history	58.65	0.0317	1.8580	0.0589

Tables 2 and 3 compare benchmarks' performance results for a small and large window sizes, respectively. The energy efficiency of a benchmark for each methodology correspond to an optimum window size for that methodology.

From table 2 we can observe that within a small window size, improved history based DVFS technique gains a lower EDP and from table 3 we can observe that within a large window size, history based DVFS technique gains a lower EDP. We can also observe from tables 2 and 3 that the percentage of EDP improvement by improved history based DVFS methodology is higher for the small window size compared to the percentage of EDP improvement by history based DVFS methodology for the large window size.

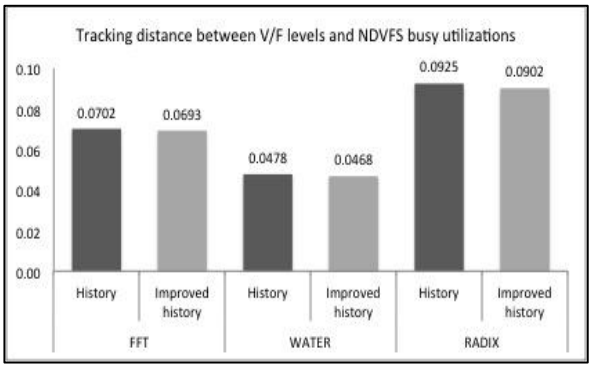

Figure 5. Tracking distance between voltage and frequency (V/F level) and non DVFS (NDVFS) busy utilization for small window size.

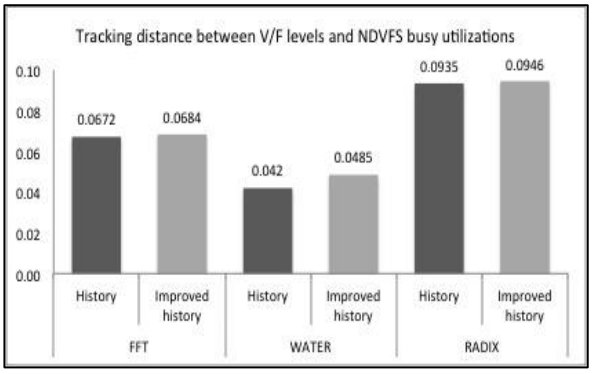

Figure 6. Tracking distance between voltage and frequency (V/F level) and non DVFS (NDVFS) busy utilization for large window size.

Lower EDP indicates a more accurate voltage and frequency prediction. To know the quality of the predicted voltage and frequency, we measure the tracking distance of the predicted voltage and frequency with the non DVFS core's busy utilization. We use this as a justification on the better EDP gained by one of the aforementioned history based DVFS methodologies over the other. Figures 5 and 6 displays the tracking distance between each methodology's voltage and frequency level (V/F level) and non-DVFS busy utilization (normalized) for the small window size and the large window size, respectively.

From figure 5 it can be observed that for the small window size, improved history based DVFS methodology contains a lower V/F tracking distance compared to history based DVFS methodology, while in figure 6 (for the large window size), history based DVFS methodology contains a lower V/F tracking distance compared to improved history based DVFS methodology. The more the difference between EDPs of history algorithms, the more the tracking distance is between voltage and frequency predictions of these methodologies and the non-DVFS core's busy utilization.

6. Conclusion

We proposed an improved history based DVFS technique, which by considering the network utilization, and the core utilization, predicts future utilization not only based on the past few windows, but also the current window. We have compared improved history based DVFS technique with history based DVFS technique which adjusts voltage and frequency based on the core's utilization of only past few windows.

Based on the experimental results we find that for small window size, improved history based DVFS technique gains EDP saving by 19%, 15%, and 5.7% for RADIX, FFT, and WATER respectively, while for a large window size, the history based DVFS technique gains EDP saving by 9%, 0.5%, and 3.6% for RADIX, FFT, and WATER respectively. The percentage of EDP improvement for a small window size of improved history based DVFS technique is more than the percentage of EDP improvement for a large window size of history based DVFS technique.

7. Acknowledgements

This work was supported by the National Science Foundation's REU program under grant number: CNS 1359461.

8. References

1. D.Suleiman, M. Ibrahim, and I. Hamarash, "Dynamic voltage frequency scaling (DVFS) for microprocessors power and energy reduction," 4th International Conference on Electrical and Electronics Engineering, December 2005.

2. Kong, Joonho, Jinhang Choi, Lynn Choi, and Sung Woo Chung. "Low-cost application-aware DVFS for multi-core architecture." In Convergence and Hybrid Information Technology, 2008. ICCIT'08. Third International Conference on, vol. 2, pp. 106-111. IEEE, 2008.

3. Rahimi, Abbas, Mostafa E. Salehi, Mohammad Fattah, and Siamak Mohammadi. "History-based dynamic voltage scaling with few number of voltage modes for GALS NoC." In Future Information Technology (FutureTech), 2010 5th International Conference on, pp. 1-6. IEEE, 2010.

4. Liang, Guang, and Axel Jantsch. "Adaptive power management for the on-chip communication network." In Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on, pp. 649-656. IEEE, 2006.

5. Ababei, Cristinel, and Nicholas Mastronarde. "Benefits and costs of prediction based DVFS for NoCs at router level." In System-on-Chip Conference (SOCC), 2014 27th IEEE International, pp. 255-260. IEEE, 2014.

6. Shang, Li, Li-Shiuan Peh, and Niraj K. Jha. "Dynamic voltage scaling with links for power optimization of interconnection networks." In High-Performance Computer Architecture, 2003. HPCA-9 2003. Proceedings. The Ninth International Symposium on, pp. 91-102. IEEE, 2003.

7. Isci, Canturk, Gilberto Contreras, and Margaret Martonosi. "Live, runtime phase monitoring and prediction on real systems with application to dynamic power management." In Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 359-370. IEEE Computer Society, 2006.

8. Choi, Kihwan, Karthik Dantu, Wei-Chung Cheng, and Massoud Pedram. "Frame-based dynamic voltage and frequency scaling for a MPEG decoder." In Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design, pp. 732-737. ACM, 2002.

9. Murray, Jacob, Teng Lu, Paul Wettin, Partha Pratim Pande, and Behrooz Shirazi. "Dual-Level DVFS-Enabled Millimeter-Wave Wireless NoC Architectures." ACM Journal on Emerging Technologies in Computing Systems (JETC) 10, no. 4 (2014): 27.

10. Hwang, Chi-Hong, and Allen C-H. Wu. "A predictive system shutdown method for energy saving of event driven computation." ACM Transactions on Design Automation of Electronic Systems (TODAES) 5, no. 2 (2000): 226-241.

11. Bircher, William Lloyd, and Lizy John. "Predictive power management for multi-core processors." In Computer Architecture, pp. 243-255. Springer Berlin Heidelberg, 2010.

12. Hajiamini, Shervin, Behrooz Shirazi, and Mohamed Azard Rilvan. "Enhancing EDP of multicore processors through DVFS." In Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International, pp. 1-6. IEEE, 2015.

13. Freeh, Vincent W., David K. Lowenthal, Feng Pan, Nandini Kappiah, Rob Springer, Barry L. Rountree, and Mark E. Femal. "Analyzing the energy-time trade-off in high-performance computing applications." Parallel and Distributed Systems, IEEE Transactions on 18, no. 6 (2007): 835-848.