© 2018 IJSRSET | Volume 4 | Issue 1 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Energy Saving Strategy Based on Profiling

Milan, Kanak Khanna
The NorthCap University, Gurugram, Haryana, India

ABSTRACT

Constraints imposed by power consumption and the related costs are one of the key roadblocks to the design and development of next generation exascale systems. To mitigate these issues, strategies that reduce the power consumption of the processor are the need of the hour. Techniques such as Dynamic Voltage and Frequency Scaling (DVFS) exist which reduce the power consumption of a processor at runtime but they should be used in such a manner so that their overhead does not hamper application performance. In this paper, we propose an energy saving strategy which operates on timeslice basis to apply DVFS under a user defined performance constraint. Results show energy savings up to 7% when NAS benchmarks are tested on a laptop platform

Keywords: DVFS, Power Management, Frequency Scaling, Energy Saving

I. INTRODUCTION

The ever increasing costs and constraints on power consumption are limiting the leap to the next generation exascale systems [34] as the power limit determined for these systems as per DoE guidelines is 20 MW. Therefore, there is an urgent need to limit the power consumption of modern computing systems to mitigate these issues.

To minimize the impact of the increasing power consumption on the computing performance, novel schemes/strategies need to be devised to reduce the power consumption of components within a computing system mainly processors without decreasing their performance by much. This means that a performance loss which can be tolerated for application execution can be pre-decided and then performance of the processor can be adjusted so to minimize power consumption. The current generation of Intel processors provides various P-states for dynamic voltage and frequency scaling (DVFS) [35][36] and T-states for introducing processor idle cycles (Throttling). For example, the Intel Haswell" micro-architecture provides fifteen P-states while the DRAM provides four frequencies. The delay of switching from

one state to another depends on the relative ordering of the current and target states, as discussed, e.g., in [42]. The user may write a specific value to model-specific registers (MSRs) to change the P- or T-states of the processor. Intel micro-architecture starting from Sandy Bridge onwards estimates power and energy consumption of the CPU and memory through the built-in MSRs.

This paper proposes a power saving strategy which operates in a fixed timeslice fashion to reduce the energy footprint in modern processors during parallel application execution. The strategy makes use of DVFS in each timeslice to reduce processor frequency based on the performance constraint and workload. Results depict that up to 7% of energy can be saved.

The rest of the paper is organized as follows. Section 2 provides the background DVFS in Intel processors.

IJSRSET1841171 | Received : 20 January 2018 | Accepted : 05 February 2018 | January-February-2018 [(4) 1 : 1040-1046]

Section 3 discusses the performance model that we use for the strategy. Section 4 provides the proposed energy saving strategy. Section 5 discusses the related works whereas Section 6 concludes the paper.

II. Dynamic Voltage and Frequency Scaling

There are mainly two sources of power dissipation in digital CMOS circuits which are namely: Static Power [26] and Dynamic Power. Static Power is the power consumption which is consumed by a digital circuit when it is not in operation and it is mainly caused by leakage currents within the circuit. The other type of power consumption known as dynamic power consumption is mainly caused by the switching activity in the circuit when it is in operations. It is mainly dependent upon the operating frequency and voltage of the processor.

The dynamic voltage and frequency scaling (DVFS) mechanism reduces the operating frequency and voltage of the processor on-the-fly during application execution, thereby reducing the dynamic power consumption. DVFS is applied by writing a specific value to the IA32_PERF _CTL [1][2] model specific register (MSR) in Intel processors and it is an architectural register which means it is present in different generations of Intel processors with the same address.

III. Applying DVFS

The execution time ((t_f)) of an application at a processor frequency f can be divided into two non-overlapping portions: On-chip time (t_{on}) and off-chip time (t_{off}) [41].

(t_f)=(t_{on})(f_{max})/f+ (t_{off}) …(1)

Where f_{max} is the maximum available processor frequency. The on-chip time scales linearly with the processor frequency accordingly whereas the off-chip time is not affected by the processor frequency.

This is due to the fact that off-chip time is predominantly memory accesses which runs in the order of many microseconds during which, the processor is simply waiting for the data and not doing much useful work. Therefore, this stall time can be exploited by simply reducing the processor frequency so that the power consumption of the processor can be reduced. This idea forms the basis of this work.

Basically, by monitoring the memory intensity, more specifically the memory accesses per instruction, we can assess the performance loss that would be caused on the application of DVFS on application performance. Therefore, if appropriate frequencies are chosen through DVFS, the performance loss can be minimized and substantial energy savings can be achieved. In the next section, we provide the details for the runtime strategy.

IV. Energy Saving Runtime Strategy

MAPI Range	Chosen Frequency
0-0.004 <=	2.4. GHz
>0.004-0.01 <=	2.2 GHz
>0.01-0.04<=	1.6 GHz
>0.04	1.2 GHz

Table I: MAPI ranges and the respective processor frequencies selected for the strategy.

For the strategy, we operate it on a timeslice basis such that a frequency is selected for the next timeslice in the previous timeslice based on the memory access per-instruction metric (MAPI) which is measured through the Intel processor performance counters.

Table I shows the range of MAPI values for a timeslice and the appropriate value of frequency that will be selected in case the actual MAPI value falls in that range. These values were obtained through extensive profiling on our hardware platform on which we executed several single and multi-threaded applications to notice the change in performance with varying processor frequencies. The performance degradation was noted down with the corresponding frequency and the MAPI value. A pattern was noticed for the range of MAPI values such that when the actual MAPI value was in that range, the performance loss for a particular frequency peaked at a particular value. This methodology was used to choose suitable frequencies for MAPI ranges as shown in Table I.

So, the strategy works as follows. The application can be divided into equally sized “k” timeslices. At the end of each timeslice, we measure the MAPI during that timeslice using the performance counters and record them in a register. To predict the MAPI for the next timeslice, we make use of a history-based predictor which averages “n” previous values of MAPI to predict its next value.

This predicted value is used in Table I to set the frequency accordingly for the next timeslice. In this manner, the strategy runs for the “k” timeslices, setting the frequency for each timeslice and reducing power and energy consumption. The algorithm I provides underlying pseudocode algorithm for the runtime energy-saving strategy.

Algorithm I

```
Parameters:
R => MAPI Register.
$M_{avg}$ => Average value of MAPI for past “x” timeslices.
1.    Initialize MAPI Register R.
2.    After “x” timeslices have passed, calculate $M_{avg}$.
3.    Based on Table I, select the appropriate frequency based on the value of $M_{avg}$.
4.    Go to Step 2 till “k” timeslices are finished.
```

V. Experimental Results

We conducted experiments on a Desktop platform with Intel Core 2 Quad 6600, quad-core processor which has processor frequency ranging from 2.4-1.2 GHz. NAS NPB parallel benchmarks [4] were used for evaluation of the strategy. The power and energy consumption of the platform were measured by using the P3 4400 Kill-a-watt power meter and “*userspace*” governor was used throughout the experiments so that frequency decided by the *userspace* could be used.

Table 1: NAS NPB Benchmark Information

CG (Conjugate Gradient)	Estimates the smallest eigenvalue of a large sparse symmetric positive-definite matrix using the inverse iteration with the conjugate gradient method as a subroutine for solving systems of linear equations.
FT (Fast Fourier Transform)	Solves a three-dimensional partial differential equation.
MG (Multigrid)	Approximates the solution to three dimensional discrete poisson equation using the multigrid method.
SP (Scalar Pentadiagonal)	Solves a nonlinear system of partial differential equations.

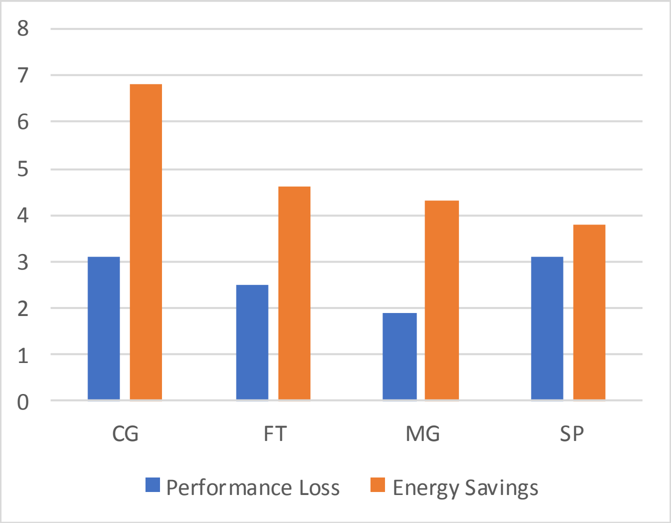

Figure 1. Performance Loss and Energy Savings for NAS NPB Benchmarks.

Four NAS NPB benchmarks namely CG, FT, MG and SP were chosen for the evaluation of the strategy which are explained in Table 1.

Figure 1 shows the resultant performance loss and energy savings for the four chosen NAS NPB benchmarks. It can be observed from Figure 1 that for all the four benchmarks, the energy saving strategy does not result in a performance loss greater than 3%. Also, the average performance loss across the four benchmarks was determined to be ~2.4%.

In terms of energy savings, the largest amount of energy savings were obtained for the CG benchmark. This is due to the fact that CG benchmark was determined to be the most memory intensive out of the bunch with the MAPI figure staying above the 0.01 value. Therefore, it executes at frequencies of 2.2 and 1.6 GHz for different intervals. The other three benchmarks were not as memory intensive as CG, and depicted variable MAPI behavior through different timeslices, therefore, relatively lesser energy savings were obtained for them. Overall, the average energy savings for the four NAS benchmarks were ~5%.

VI. Related Work

There are different strategies for reducing energy consumption in modern computing systems through DVFS. The first uses a fixed size timeslice based profiling methods with workload classification through performance counters [8], [10], [11], [14], [17], [19], [23],[24],[25],[30],[33],[37],[38],[39],[40],[41]. The other type identify communication phases which could be present in message passing etc. based communication intervals, to apply frequency scaling [7], [13], [15], [18], [20], [22], [25], [26], [27],[28],[29],[31],[32],[33]. While DVFS has been quite widely used to reduce the power consumption, it doesn't exactly provide the information regarding the instantaneous power consumption of the processor. Therefore, power limiting comes into picture so that power consumption of the processor can be directly controlled. Intel Running Average Power Limit (RAPL) [6] provides power clamping and energy metering capabilities starting from Intel SandyBridge processor generation. Runtime system "conductor" does power budgeting based on available power to different compute nodes and the communication slack available. Power limiting research focusing on capabilities of processor and DRAM along with a profiling based power budgeting strategy was proposed in [44]. Authors in [45] study and propose predictive models which facilitate forecasting values of performance parameters and the appropriate power limits for different components within a compute node to maximize performance.

VII. Conclusions

The desire for achieving exascale performance has pushed the modern computing systems to operate at their maximum operating frequency and bandwidths. Consequently, the energy consumption and failure rates are also reaching prohibitive levels for these computing systems. To mitigate this issue, an energy saving strategy is proposed in this work which makes use of the DVFS capability present in the Intel processors to reduce the processor power and energy consumption. The strategy makes use of offline profiling to determine the frequency levels for different memory intensities and uses this information to scale the frequency for applications accordingly. Results on an Intel quad-core machine with four chosen NAS NPB benchmarks depict that energy savings of up to 7% can be achieved by using our strategy. Future work will explore a runtime performance model which would disable use of offline profiling and would enable the use to determine the frequency levels for an application at runtime.

VIII. REFERENCES

[1]. A. Intel Power Governor. https://software.intel.com/en-us/articles/intel-power-governor.

[2]. Intel Software Developer's Manual. https://software.intel.com/en-us/articles/intel-sdm

[3]. N. Almoosa, W. Song, Y. Wardi and S. Yalamanchili. A Power Capping Controller for

Multicore Processors. In 2012 American Control Conference (ACC), pages 4709-4714. June 2012.

[4]. D.H. Bailey et. Al. The NAS Parallel Benchmarks Summary and Preliminary Results. In Proceedings of the 1991 ACM/IEEE conference on Supercomputing, pages 158-165, 1991.

[5]. S. Bhalachandra, A. Porterfield, S. L. Olivier, and J. F. Prins. An Adaptive Core-specific Runtime for Energy Efficiency. In 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pages 947-956, May 2017.

[6]. H. David, E. Gorbatov, U.R. Hanebutte, R. Khannal, and C. Le RAPL: Memory Power Estimation and Capping. In Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design, ISLPED'10, pages 189-194, New York, NY, USA, 2010. ACM.

[7]. V.W. Freeh and D.K. Lowenthal. Using Multiple Energy Gears in MPI Programs on a Powerscalable Cluster. In Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming, pages 164-173, 2005.

[8]. R. Ge, X. Feng, W. Feng, and K.W. Cameron. CPU MISER: A Performance Directed, Run-time System for Power-aware Clusters. In Parallel Processing, 2007. ICPP 2007. International Conference on, page 18, Sep. 2007.

[9]. R. Ge, X. Feng, Y. He, and P. Zou. The Case for Cross-component Power Coordination on Power Bounded Systems. In 2016 45th International Conference on Parallel Processing (ICPP), pages 516-525, Aug 2016.

[10]. R. Ge, X. Feng, S. Song, H.C. Chang, D. Li, and K.W. Cameron. PowerPack: Energy Profiling and Analysis of High-performance systems and Applications. Parallel and Distributed Systems, IEEE Transactions on, 21:658-671, 2010.

[11]. C.H. Hsu and W. Feng. A Power-aware Runtime System for High-Performance Computing. In Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference, page 1, Nov. 2005.

[12]. S. Huang and W. Feng. Energy-efficient Cluster Computing via Accurate Workload Characterization. In Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on, pages 68- 75, May 2009.

[13]. N. Ioannou, M. Kauschke, M. Gries, and M. Cin- tra. Phase-based Application-driven Hierarchical Power Management on the Single-chip Cloud Computer. In Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on, pages 131-142, Oct. 2011.

[14]. C. Isci and M. Martonosi. Runtime Power Monitoring in High-end Processors: Methodology and empirical data. In Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture, MICRO 36, pages 93-, Washington, DC, USA, 2003. IEEE Computer Society.

[15]. K. Kandalla, E.P. Mancini, S. Sur, and D.K. Panda. Designing Power-aware Collective Communication Algorithms for InfiniBand Clusters. In Parallel Processing (ICPP), 2010 39th International Conference on, pages 218-227, 2010.

[16]. Kristopher Keipert, Gaurav Mitra, Vaibhav Sundriyal, Sarom S. Leang, Masha Sosonkina, Alistair P. Rendell, and Mark S. Gordon. Energy Efficient Computational Chemistry: Comparison of x86 and ARM Systems. Journal of Chemical Theory and Computation, 11(11):5055-5061, 2015. PMID: 26574303.

[17]. Gary Lawson, Vaibhav Sundriyal, Masha Sosonkina, and Yuzhong Shen. Runtime Power Limiting of Parallel applications on Intel Xeon Phi Processors. In Proceedings of the 4th International Workshop on Energy Efficient

Supercomputing, E2SC ’16, pages 39- 45, Piscataway, NJ, USA, 2016. IEEE Press.

[18]. M.Y. Lim, V.W. Freeh, and D.K. Lowenthal. Adaptive, Transparent Frequency and Voltage scaling of Communication Phases in MPI Programs. In Proceedings of the 2006 ACM/IEEE conference on Supercomputing, 2006.

[19]. K. Ma, X. Li, W. Chen, C. Zhang, and X. Wang. GreenGPU: A Holistic Approach to Energy Efficiency in GPU-CPU Heterogeneous Architectures. In 2012 41st International Conference on Parallel Processing, pages 48-57, Sept 2012.

[20]. B. Rountree, D.K. Lowenthal, B.R. de Supinski, M. Schulz, V.W. Freeh, and T. Bletsch. Adagio: Making DVS Practical for Complex HPC Applications. In Proceedings of the 23rd international conference on Supercomputing, ICS’09, pages 460-469, New York, NY, USA, 2009. ACM.

[21]. Jayanth Srinivasan. An Overview of Static Power Dissipation. Technical report.

[22]. V. Sundriyal and M. Sosonkina. Per-call Energy Saving Strategies in All-to-all Communications In Proceedings of the 18th European MPI Users’ Group conference on Recent advances in the message passing interface, EuroMPI’11, pages 188-197, Berlin, Heidelberg, 2011. SpringerVerlag.

[23]. V. Sundriyal and M. Sosonkina. Initial Investigation of a Scheme to use Instantaneous CPU Power Consumption for Energy Savings format. In Proceedings of the 1st International Workshop on Energy Efficient Supercomputing, E2SC ’13, pages 1:1-1:6, New York, NY, USA, 2013. ACM.

[24]. V. Sundriyal and M. Sosonkina. Joint Frequency Scaling of Processor and DRAM. The Journal of Supercomputing, 72(4):1549-1569, 2016.

[25]. V. Sundriyal, M. Sosonkina, and A. Gaenko. Runtime Procedure for Energy Savings in Applications with Point-to-point Communications. In Computer Architecture and High-Performance Computing (SBAC-PAD), 2012 IEEE 24th International Symposium on, pages 155-162, 2012.

[26]. V. Sundriyal, M. Sosonkina, and Z. Zhang. Achieving Energy Efficiency During Collective Communications. Concurrency and Computation: Practice and Experience, 25(15):2140-2156, 2013.

[27]. V. Sundriyal, M. Sosonkina, and Z. Zhang. Automatic Runtime Frequency Scaling System for Energy Savings in Parallel Applications. The Journal of Supercomputing, 68(2):777-797, 2014.

[28]. Vaibhav Sundriyal, Ellie Fought, Masha Sosonkina, and Theresa L. Windus. Power Profiling and Evaluating the Effect of Frequency Scaling on NWChem. In Proceedings of the 24th High Performance Computing Symposium, HPC ’16, pages 19:1-19:8, San Diego, CA, USA, 2016. Society for Computer Simulation International.

[29]. Vaibhav Sundriyal, Ellie Fought, Masha Sosonkina, and Theresa L. Windus. Evaluating effects of application based and automatic energy saving strategies on NWChem. In Proceedings of the 25th High Performance Computing Symposium, HPC ’17, pages 16:1-16:12, San Diego, CA, USA, 2017. Society for Computer Simulation International.

[30]. Vaibhav Sundriyal and Masha Sosonkina. Runtime power-aware energy-saving scheme for parallel applications. 2015.

[31]. Vaibhav Sundriyal, Masha Sosonkina, Alexander Gaenko, and Zhao Zhang. Energy saving strategies for parallel applications with point-to-point communication phases. Journal of Parallel and Distributed Computing, 73(8):1157 - 1169, 2013.

[32]. A. Vishnu, S. Song, A. Marquez, K. Barker, D. Kerbyson, K. Cameron, and P. Balaji. Designing Energy Efficient Communication Runtime Systems for Data Centric Programming Models. In Proceedings of the 2010 IEEE/ACM Int’l Conference on Green Computing and

Communications & Int'l Conference on Cyber, Physical and Social Computing, GREENCOM-CPSCOM '10, pages 229-236, Washington, DC, USA, 2010. IEEE Computer Society.

[33]. C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T.W. Keller. Energy management for commercial servers. Computer, 36(12):39- 48, December 2003.

[34]. S. Borkar. The exascale challenge, 2011. Keynote speech, the 12th International Conference on Parallel Architectures and Compilation Techniques.

[35]. R. Gonzalez, B. M. Gordon, and M. A. Horowitz. Supply and Threshold Voltage Scaling for Low Power CMOS. IEEE Journal of Solid-State Circuits, 32(8):1210-1216, Aug 1997.

[36]. G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott. Energy-Efficient Processor Design Using Multiple Clock Domains with Dynamic Voltage and Frequency Scaling. In Proceedings Eighth International Symposium on High Performance Computer Architecture, pages 29-40, Feb 2002.

[37]. Linux Kernel Governors. https://www.kernel.org/doc/Documentation/cpufreq/governors.txt. Online].

[38]. V Pallipadi and A Starikovskiy. The Ondemand Governor: Past, Present and Future. 2:223-238, 01 2006

[39]. R. Efraim, R. Ginosar, C. Weiser, and A. Mendelson. Energy Aware Race to Halt: A Down to Earth Approach for Platform Energy Management. IEEE Computer Architecture Letters, 13(1):25-28, Jan 2014.

[40]. J. P. Halimi, B. Pradelle, A. Guermouche, N. Triquenaux, A. Laurent, J. C. Beyler, and W. Jalby. Reactive DVFS Control for Multicore Processors. In 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pages 102-109, Aug 2013.

[41]. Vaibhav Sundriyal, Masha Sosonkina, Fang Liu, and Michael W. Schmidt. Dynamic Frequency Scaling and Energy Saving in Quantum Chemistry Applications. In IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum, pages 837-845, 2011.

[42]. J. Park et. al. Accurate Modeling and Calculation of Delay and Energy Overheads of Dynamic Voltage Scaling in Modern High-Performance Microprocessors. ISPLED 2010. Pages 419-424.

[43]. A. Marathe et. al. A Run-time System for Power Constrained HPC Applications. International Conference on High Performance Computing. ISC 2015.

[44]. R. Ge et. al. The Case for Cross-Component Power Coordination on Power Bounded Systems. International Conference on Parallel Processing. 2016.

[45]. A. Tiwari et. al. Predicting Optimal Power Allocation for CPU and DRAM Domains. IEEE International Parallel and Distributed Processing Symposium. 2015.