© 2017 IJSRSET | Volume 3 | Issue 8 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

A Survey of Dynamic Power Saving Strategies in Real Systems

Sai Kiran Talamudupula
Senior Bios Engineer, Intel, Chandler, Arizona, USA

ABSTRACT

Energy efficiency and energy-proportional computing have become major constraints in the design of modern exascale platforms. Dynamic Voltage and Frequency Scaling (DVFS) is one of the most commonly used and effective techniques to dynamically reduce power consumption based on workload characteristics. The focus of this paper is to survey several energy saving strategies designed for improving power efficiency of CPU and DRAM systems. This paper also presents a characterization of the strategies based on their salient features, to help the research community in gaining insights into the similarities and differences between the them. The aim of the paper is to equip researchers with knowledge of the state of the art energy saving strategies and serve as a quick reference to engineers while they are devising novel energy saving strategies.

Keywords: DynamicVoltage and Frequency Scaling (DVFS), Power Efficiency, Energy Saving, Survey, Review.

I. INTRODUCTION

The last few decades have witnessed a tremendous rise in the design of scalable applications for various scientific domains. The sheer computational requirements of these applications have forced system engineers to develop ever more power and performance-efficient architectures. For the current topmost petascale computing platforms in the world, it is typical to consume power on the order of several megawatts, which at current prices may cost on the order of several million dollars annually. For operational sustenance of the exascale machines, the power consumption growth rate must slow down and deliver more calculations per unit of power. To address this challenge, power and energy optimizations have been proposed in modern computing platforms at all levels: application, system software, and hardware.

There are several architectural and technological trends that have resulted in increased power consumption of modern computing systems. In modern processors, the number of cores on a single chip has been increasing [4, 40] and consequently to serve the requests of many core architectures, the requirements for memory system have been increasing as well. Apart from the increase in number of cores and size of main memory, the clock rate on these devices has also been increasing rapidly and is only been constrained by the feature size, pipeline subcomputation length and power consumption since dynamic power consumption in a CMOS is dependent upon the operating frequency and voltage.

It is well established that the CPU and the memory subsystem are the major power consumers in a computer system. For example, the CPU consumes about 50% of the total power as was investigated in [10], considering both static and dynamic power consumption. Memory power consumption is a significant component in computer server power profile, which is comparable to or may even surpass processor power consumption for memory-intensive workloads. An early study [39] has reported that, on an IBM p670 server, the average memory power consumption is 1223 watts, compared to the average processor power consumption of 840 watts. A keynote speech from industry in 2011 [40] predicts that memory may consume about 70% in small scale servers in the future, excluding the power consumed by power supply and cooling. The current generation of Intel processors provides different P-states for dynamic voltage and frequency scaling (DVFS) [41,42] and T-states for introducing processor idle cycles (throttling).

In [36], Mei et. al review the past research works focusing on GPU DVFS and conduct experiments on GPUs, confirming that GPU DVFS provides lot of potential for energy savings. [37] discusses categories of power saving techniques such as DVFS, thread motion, parallelism etc. and proposes a new technique based the research work that was reviewed. Zhuravlev et. Al [38] survey scheduling techniques which provide energy efficient computation.

Different from all these works, this paper focuses on dynamic power reduction strategies making use of DVFS/Throttling in modern processors. Although the techniques which improve the performance of an application can consequently reduce the energy consumption as well but only consider the strategies which were devised to reduce the power consumption specifically. Moreover, all the techniques surveyed in this paper have been implemented and tested on real hardware platforms instead of simulation/emulation.

The contributions of this work include 1) an overview of the basic power consumption terminologies commonly used in research literature, 2) Studying the capabilities for reducing dynamic power consumption in modern processors and 3) provide a categorization for dynamic power saving strategies with respect to their design characteristics.

The rest of the paper is organized as follows. Section II provides an overview of types and sources of power dissipation along with the mechanisms to reduce dynamic power in modern processors to apply DVFS/Throttling. Section III describes the salient features of dynamic power saving strategies for their classification, showing the similarity and differences between them. Section IV provides a detailed discussion of various power saving strategies. Finally, Section V provides the concluding remarks

II. POWER DISSIPATION

In this section, we briefly discuss the types and sources of power consumption in modern processors, along with the mechanism to reduce dynamic power consumption, to aid in the discussion of dynamic power saving techniques in the next sections.

A. Static and Dynamic Power

There are three sources of power dissipation in digital CMOS circuits [23] which are summarized as

$$P = P_{switching} + P_{short-circuit} + P_{static}$$
$$= \alpha C_L V_{dd}^2 + I_{sc}V_{dd} + I_{leakage}V_{dd}.$$

Pswitching refers to the dynamic component of the power, where CL stands for load capacitance, f is the clock frequency and α is the activity factor. Assuming voltage swing is equal to the supply voltage Vdd, the short circuit power Pshort−circuit is due to the direct-path short circuit current Isc which flows when both PMOS and NMOS transistors are simultaneously active. The static power consumption Pstatic is due to the leakage current Ileakage which arises due to various factors such as sub-threshold leakage, reverse biased pn-junction etc.

It can be observed from Eq. (1), that only the dynamic power consumption can be modified at application runtime through software since it is dependent upon the frequency of the processor whereas the other two components in general are fixed during the manufacturing process. Therefore, in this paper we review power saving strategies which attempt to reduce only the dynamic power consumption.

B. DVFS and Throttling

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 as per Eq. (1). DVFS can be applied by writing a specific value to the IA32_PERF _CTL model specific register (MSR).

CPU throttling can be viewed as an equivalent to dynamic frequency scaling as it inserts a given number of idle cycles in the CPU execution obtaining a particular operating frequency without changing the operating voltage of the cores. Hence, dynamic frequency scaling is less effective than DVFS in terms of reducing dynamic power consumption, but when used with conjunction with DVFS, it has been shown to reduce the dynamic power consumption significantly [28].

III. CHARACTERISTICS OF THE STATE OF THE ART ENERGY SAVING STRATEGIES

In particular, the existing DVFS based techniques can be characterized through the following features:

a) Application/Library: If a technique makes modification to the application or library code to apply DVFS.

b) Transparent: If a technique applies DVFS transparently without making any modification to the application or library code.

c) Processor/DRAM: If a technique applies DVFS to either one of processor/DRAM or both.

d)Workload Prediction Mechanism: If a technique uses *history* or *trace* based workload predictors. The *history* predictors employs window of past *n* samples to predict the future workload whereas the *trace* predictors use the traces of past execution to predict the future behavior [15].

e) Performance Loss Availability: If a technique provides the user with fine grained control of degradation in performance of the application resulting from DVFS usage. In [47], authors discussed that how always selecting a higher performance loss constraint doesn't always translate into energy savings.

f) DVFS/Throttling: If a technique makes use of DVFS or throttling or both to save power.

IV. OVERVIEW OF THE DYNAMIC POWER SAVING STRATEGIES

Table I categorizes the various dynamic power saving strategies as per their salient characteristics.

We begin by reviewing the linux frequency scaling governors [43, 44] provided by the cpufreq infrastructure. cpufreq is implemented on top of frequency scaling drivers such as acpi-cpufreq, intel pstate etc. The governors are basically an in-built energy saving strategy and only one of them can be active at time which are:

Performance: Statically sets the processor to its maximum frequency.
Powersave: Statically sets the processor to its minimum frequency.
Ondemand: Sets the processor frequency to maximum or minimum if processor is working or idle, respectively.
Conservative: Similar to *Ondemand*, except it increases/decreases frequency in a stepwise manner.

The *Userspace* frequency scaling governor is not listed above since it allows userspace programs to set the frequency. Since, all the above-mentioned governors have their own frequency scaling rules, the *Userspace* governor is chosen to deploy user developed energy saving strategies.

CPU Miser [8] selects a performance loss for the underlying application chosen by the user and divides the execution of an application into intervals of a particular duration. Next, it predicts the execution characteristics, such as memory stalls, of the upcoming interval based on similar recent intervals using the *history* predictor. CPU Miser primarily considers memory accesses to choose a suitable frequency for a given time slice, and has been shown to attain significant energy gains [8]. It can also either overestimate or underestimate the number of stalls in an application, which can lead to an inaccurate estimation of the frequency. Also, it does not consider the instantaneous power consumption of the unit under test when choosing a suitable frequency.

Adagio [21] discusses critical path analysis to identify the tasks to be slowed down as it does not provide the user with an option to choose a performance loss. The workload prediction mechanism used in Adagio is similar to the history predictor with some feedback added, such that the future behavior of a communication call is predicted based on its last invocation. Lim *et al.* [19] propose a scheme that applies DVFS during communication phases in MPI applications. Dynamic determination of communication phases and selection a suit- able processor frequency to minimize energy consumption are the key features of this scheme. This strategy does not apply DVFS in computation portions of an application and, therefore, may not save a significant amount of energy for an application that has a relatively low communication activity.

Authors in [29] proposed a novel DVFS application strategy, which used both DVFS and Throttling to maximizes energy savings by selecting appropriate values for DVFS and throttling based on the predicted communication phases considering both the CPU offload (provided by the network protocol, such as Infiniband) during communication and the architectural stalls during computation. [25, 32], discuss the potential pitfalls of the performance-loss based approaches and propose a strategy that depends on the instantaneous power consumption of the computing platform using regression analysis to choose the best processor frequency which would minimize the energy consumption.

[35] implements a runtime communication library named PASCoL making use of Aggregate Remote Memory Copy Interface (ARMCI) which uses both DVFS and polling/blocking mechanisms to achieve energy saving in communication calls. A quad-state runtime system targeting point-to-point calls in MPI was proposed in [27, 33] which applied DVFS in those calls using a trace based workload prediction mechanism. [28, 24] and [16] both focus on collective communication calls such as MPI_AlltoAll which are extremely communication intensive and employ both DVFS and Throttling to achieve impressive energy savings up to 20% for select applications in NAS [3] and CPMD [1].

Timeslice profiling through performance counters and application of DVFS by determining the memory intensiveness of a workload is done in [11]. [12] improves upon [11] by employing advanced modeling of workload and taking into account the out-of-order nature of modern processor pipelines. Authors in [7] devise a strategy which divides the application into various phases, assign it a frequency with and then when this phase is further encountered, it is assigned a frequency which was calculated through experiments and heuristics when that particular phase was first seen.

By determining the memory intensiveness of different algorithmic steps in NwChem [34] in [30], authors in [31] propose a frequency scaling scheme code in application for both processor and memory In [26], a runtime system is proposed which addresses both processor and memory frequency scaling based on a detailed performance model minimizing energy consumption of an application. An important point to note here is that DRAM frequency scaling is currently not supported through software and it was emulated through changes in memory frequency through BIOS in [26].

Table I : Summary of Several Dynamic-Power Saving Strategies as to the Six Salient Characteristics

Name	App./Library	Transparent	Proc./DRAM	Prediction	Perf. Loss	DVFS/Throttling
[8]	No	Yes	Processor	History	Yes	DVFS
[21]	No	Yes	Processor	History	No	DVFS
[29]	No	Yes	Processor	Trace	Yes	DVFS
[19]	No	Yes	Processor	History	No	DVFS
[25]	No	Yes	Processor	History	Yes	DVFS
[35]	Yes	No	Processor	History	No	DVFS
[14]	No	Yes	Processor	Trace	Yes	DVFS
[28]	Yes	No	Processor	None	No	Both
[12]	No	Yes	Processor	History	Yes	DVFS
[11]	No	Yes	Processor	History	Yes	DVFS
[7]	No	Yes	Processor	Trace	No	DVFS
[16]	Yes	No	Processor	None	No	DVFS
[33, 27]	No	Yes	Processor	Trace	Yes	DVFS
[30, 31]	Yes	No	Both	None	No	DVFS
[18]	No	Yes	Processor	History	Yes	DVFS
[26]	No	Yes	Both	History	Yes	DVFS
[32]	No	Yes	Processor	History	No	DVFS
[4]	No	Yes	Processor	None	No	Both
[45]	No	yes	Processor	History	No	DVFS
[46]	No	Yes	Processor	History	No	DVFS

Authors in [18] propose a performance model based on power limiting provided through the running average power limit (RAPL) [5] feature present in Intel processors, which under a performance loss minimizes energy consumption in Intel Xeon Phis [2]. An adaptive core-specific runtime (ACR) is proposed in [4] which makes use of both DVFS in its per core variant and throttling to achieve energy savings in parallel applications. Optimal power allocations using RAPL power limiting features to different components of a compute node along with workload behavior with respect to those applications was studied in [9]. Green GPU strategy [20] for heterogeneous architectures, distributes workload between GPU-CPU so that both the components can finish around the same time and also applies DVFS to GPU cores based on their utilization. [17] compares the energy efficiency of ARM32, ARM 64 and an x86 processor for a GAMESS [22] execution for solution to energy which determined that ARM32 machine was the most energy efficient followed by its 64-bit variant and the x86 processor. Lately, oversubscription [13, 6] has been shown to reduce power consumption and increase performance of an application by executing multiple threads/processes on a single core of a processor.

[45] proposes the energy aware race-to-halt (EARtH) runtime strategy which uses a theoretical model evaluating effect of processor frequency on platform power and determine its effect on application using the workload scalability (SCA) parameter. The EARtH algorithm calibrates itself at system production by measuring power consumption at various processor frequencies and employs it in the theoretical model to determine the optimal frequency which minimizes energy consumption.

FoREST DVFS controller [19], based on the past studies which depicted static power as a function of dynamic power [58, 38], derived through modeling that power ratios at different frequencies for an application are independent of the nature of the application. Next, by computing these power ratios in an online manner, an online algorithm is proposed which depicted significant energy gains. Moreover, it also provides an option for providing a user defined performance loss.

V. CONCLUSION

Recent advances in chip design and the desire to extract maximum performance from the modern computing systems has greatly increased their power consumption. To alleviate this issue, several techniques have been proposed in the past to reduce dynamic power consumption of the processor and memory by modifying their frequency and operating voltage. In this work, we reviewed many dynamic power saving strategies which make use of DVFS and throttling to reduce the dynamic power consumption. We also discussed a categorization of the reviewed strategies as per their salient characteristics to inform the reader regarding their similarities and differences. We hope that this survey work will inform researchers, processor architects and software engineers to regarding the prominent strategies employed to reduce dynamic power consumption and their underlying behavior. Moreover, it will also serve as a quick reference for the designers of future dynamic power saving strategies to compare their work with the past researches.

VI. REFERENCES

[1]. www.cpmd.org.

[2]. A. Sodani. Knights Landing (KNL): 2nd generation Intel Xeon phi Processor. In HotChips 2015.

[3]. 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.

[4]. 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.

[5]. 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.

[6]. Ellie L. Fought, Vaibhav Sundriyal, Masha Sosonkina, and Theresa L. Windus. Saving time and energy with oversubscription and semi-direct Moller-Plesset second order perturbation

methods. Journal of Computational Chemistry, 38(11):830-841, 2017.

[7]. V.W. Freeh and D.K. Lowenthal. Using Multiple Energy Gears in MPI Programs on a Power-scalable 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 Run-time 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]. C. Iancu, S. Hofmeyr, F. Blagojevi, and Y. Zheng. Oversubscription on Multicore Processors. In 2010 IEEE International Symposium on Parallel Distributed Processing (IPDPS), pages 1-11, April 2010.

[14]. 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.

[15]. 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.

[16]. 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.

[17]. Kristopher Keipert, Gaurav Mitra, Vaibhav Sunriyal, 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.

[18]. 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.

[19]. 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.

[20]. 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.

[21]. 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.

[22]. M. W. Schmidt, K.K. Baldridge, J.A. Boatz, S.T. Elbert, M.S. Gordon, J.H. Jensen, S. Koseki, N. Matsunaga, K.A. Nguyen, S. Su, T.L. Windus, M. Dupuis, and Jr. J.A. Montgomery. General atomic and Molecular Electronic Structure System. J. Comput. Chem., 14:1347-1363, Nov. 1993.

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

[24]. 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. Springer-Verlag.

[25]. 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.

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

[27]. 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.

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

[29]. 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.

[30]. 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.

[31]. 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.

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

[33]. 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.

[34]. M. Valiev, E.J. Bylaska, N. Govind, K. Kowalski, T.P. Straatsma, H.J.J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T.L. Windus, and W.A. de Jong. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. Computer Physics Communications, 181(9):1477 - 1489, 2010.

[35]. 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.

[36]. Xinxin Mei, Qiang Wang, and Xiaowen Chu. A Survey and Measurement Study of GPU DVFS on Energy Conservation. Digital Communications and Networks, 3(2):89 -100,2017.

[37]. Khaled M. Attia, Mostafa A. El-Hosseini, and Hesham A. Ali. Dynamic power management techniques in multi-core architectures: A survey study. Ain Shams Engineering Journal, 8(3):445-456, 2017.

[38]. S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto. Survey of energy cognizant cheduling techniques. IEEE Transactions on Parallel and Distributed Systems, 24(7):1447-1464. July 2013.

[39]. 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.

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

[41]. 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.

[42]. 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.

[43]. Linux Kernel Governors. https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt. Online].

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

[45]. 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.

[46]. 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.

[47]. 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.