A Brief Survey of Current Power Limiting Strategies

Milan

The Northcap University, Gurugram, Haryana.

Abstract— *Constraints imposed by power consumption and the related costs are one of they key roadblocks to the design and development of next generation exascale systems. To mitigate these issues, strategies that constrain the power consumption of compute devices such as processors, MICs and GPUs systems to remain within the physical power limits have been proposed. While there is a plethora when it comes to review work dealing with techniques such as Dynamic Voltage and Frequency Scaling (DVFS), there is a dearth of work reviewing power constrained computing. In this paper, we review several strategies which deal explicitly with directly limiting the power consumption of the compute devices while minimizing its affect on the performance of the executing applications. We believe such a review would help the researchers to compare their work to the contemporary strategies. Also, it can serve as a compendium to quickly introduce researchers with the current state of power limiting research.*

1. Introduction

The ever increasing costs and constraints on power delivery are limiting the leap to the next generation exascale systems as the power limit determined for these systems as per DoE guidelines is 20 MW. Therefore, exascale systems will be power bounded so that the set power limits can be respected which means that all the components within a compute device may not work at their maximum performance.

To minimize the impact of the power limiting on perform mance, strategies need to be devised to optimally allocate the given power budget among the devices. applications execution time. By optimal allocation, we mean that a component gets a power allocation based on its overall utilization. For example, limiting the power consumption of CPU during a memory intensive application is optimal since the CPU utilization would be considerably lower at that time. Determining the extent to which different types of computations are sensitive to reduced power caps on CPU and DRAM subsystems is, therefore, the prerequisite in the development of optimal power capping strategies.

It is well-established that CPU and the memory sub-system are the major power consuming components in a modern computing system [1]. The current generation of Intel processors employs different P-states for dynamic volt- age and frequency scaling (DVFS) and clock modulation for introducing processor idle cycles (throttling). The delay

Of applying DVFS/ Throttling depends relative ordering of the current and target freqencies [21].

Various approaches exist to intelligently employ DVFS in modern computing systems to apply frequency scaling. The two manners in which frequency scaling strategies is applied is 1) through a fixed size timeslice With workload classification through performance counters [11], [13], [14], [15], [26], [28]; and 2) the other that applies frequency scal- ing to message passing etc. based communication intervals. [8], [18], [25], [27], [29], [32]. While DVFS has been quite widely used to reduce the power consumption, it doesnt exactly provide the information regarding the instantaneous power consumption of the processor. Therefore, power lim- iting comes into picture so that power consumption of the processor can be directly controlled.

This paper provides an overview of some of the most common power limiting strategies. While there have been many works in the past which surveyed reduction of power through DVFS [20], [37], there has been a dearth of works when in comes to power limiting. This paper attempts to fill that gap and would serve as a quick reference for a person relatively new to the subject.

The rest of the paper is organized as follows. Section 2 provides the background for power limiting in Intel processors. Section 3 surveys the existing power limiting strategies. Section 4 provides the conclusions for the paper.

2. Power Capping in Intel Processors

RAPL provides a set of counters and model specific registers (MSR) providing energy and power consumption information along with power limiting capability. RAPL uses a software based power model [2] which calculates energy and power consumption of different power domains. Intel processors starting from Sandy bridge provide up to four power domains Fig. 1:.

- PKG: The complete processor chip.
- PP0: Compute cores in the PKG.
- PP1: Uncore.
- DRAM: Main Memory.

Intel has separated its processor families into two classes namely *client* and *server*. Both classes support package (PKG) and Power Plane 0 (PP0) domains, while the server adds a separate DRAM domain and the client adds a second power plane (PP1). The power consumption of both PKG and DRAM are managed by the model specific registers MSR_PKG_RAPL_POWER_LIMIT and MSR_DRAM POWER_LIMIT, respectively.

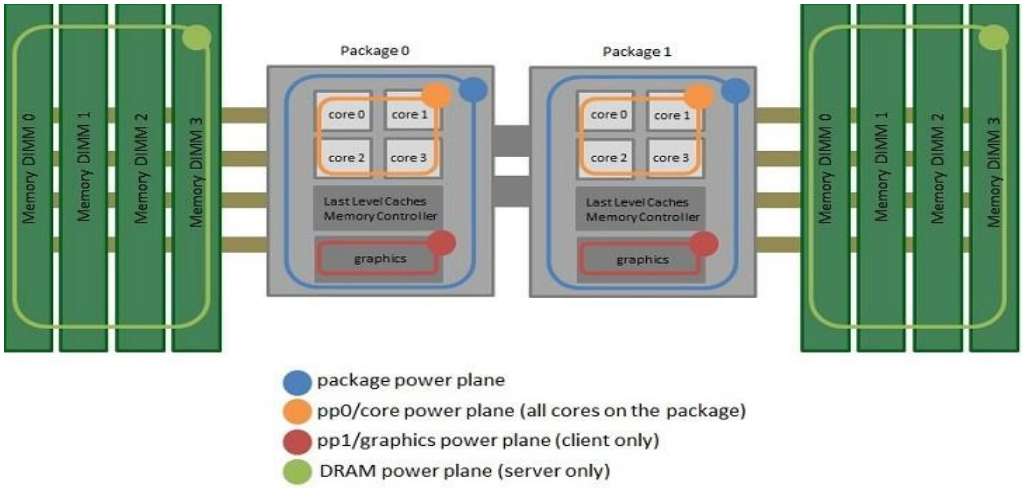

Figure 1. RAPL Power Domains [1].

3. Review of Power Limiting Strategies

Authors in [17] implement a feedback controller on an IBM server. The feedback controller is implemented in such a way that the system stays within a fixed power constraint while selecting the highest performance state periodically. Closed-loop controller is used to provide better application performance under power constraint instead of open-loop policies as it readily reacts to workload changes and selects the desired performance state to meet the power budget. As a result, 82% increase in application performance is observed. Authors in [22] propose 3 techniques for peak power management that firstly prevent instantaneous power from exceeding the peak power budget and secondly work on processors with a large number of cores on the same die. The 3 approaches speed up the decision process for peak power management by limiting or pruning the search space of global power management states or by selecting a set of cores and scaling down their voltages.

[7] presents a power budget guided job scheduling policy implemented in High performance Computing (HPC) centers. DVFS is used to significantly improve performance as at low frequencies, more jobs can be executed simultaneously and long wait queues can be avoided. The policy assigns frequency to each job at the scheduling time depend- ing on instantaneous power and thresholds. As a result, two times better performance is observed compared to power budgeting without DVFS.

[3] discusses an online controller for tracking power-budgets in multicore processors using dynamic voltage frequency scaling. The controller adjusts its gains in response to changes in the workload to ensure effective regulation and fast settling time. Each core in the multicore processor is equipped with the controller, assigned a power budget and operates independently in tracking its power budget.

Authors in [34] propose a method to partition the total power budget between the computing servers and cooling units in such a way that the cooling power meets the heat removal requirements for the computing power. This is done using an optimal power budgeting technique which identifies the optimal power cap for the servers such that the total server power meets the computing budget and the total throughput is maximized. Also, a throughput predictor is proposed where measurements from performance counters are used to estimate the changes in throughput as a function of the servers power cap.

[30] discusses a runtime power limiting strategy for quantum chemistry software GAMESS [23]. The proposed strategy applies a power limit based on the underlying behavior of data servers and compute processes such that a higher allocation is made to compute processes because they are actually doing all the compute based work.

[16] presents an efficient power capping technique by coordinating DVFS and task mapping in a CPU-GPU heterogeneous system. The frequency scaling can incur load imbalances and power violations between the CPUs and GPUs and so to avoid this, empirical models are demonstrated to precisely predict the performance and the maximum power consumption with the given settings of the CPU frequency, GPU frequency and task mapping.

Authors in [31] propose an energy saving mechanism based on DVS that takes into account energy consumption restrictions imposed by the system during the execution of tasks. The mechanism maximizes performance without violating energy consumption restrictions and achieves great energy savings by lowering processors frequency in addition to executing tasks under a low power budget than the one imposed by the system.

[24] introduces PPEP, an online Performance, Power and Energy prediction framework that estimates PPE at a particular voltage frequency state and predicts PPE at all other states by using execution statistics gathered on real processors. PPEP periodically reads hardware performance counters from the CPU cores, allowing it to quickly adjust to program phase changes.

[35] evaluates the performance of Intels Running Average Power Limit (RAPL) interface in the 4 main SASO properties: stability, accuracy, settling time and maximum overshoot in addition to its efficiency. RAPL performs well on the first four standard metrics but is inefficient at very low power limits and for some applications even at higher limits.

[4] proposes two techniques to enforce power capping constraints on a real supercomputer (Eurora supercomputer)

1.) A priority rules based algorithm and 2.) A novel hybrid approach which combines a CP and a heuristic technique. Power capping is achieved by acting on the number of jobs entering the system. The two approaches are compared by using average queue times as an evaluation metric and it is found that the quality of solution varies with the levels of power capping considered.

Authors in [36] propose *PUPiL*, a hardware/software power capping system which combines hardwares fast reac- tion time with softwares flexibility. Performance of PUPiL is evaluated and tested against a pure software approach and Intels RAPL and it is observed that it achieves at least 18% greater mean performance. The work shows that capping cannot be left to hardware alone but requires the cooperation of both hardware and software.

[5] Proposes a novel server power control solution that can control the power consumption of a server to the desired budget. The solution shifts power between processor and main memory in a coordinated manner by dynamically adjusting the voltage/frequency of the processor and placing memory ranks into different power states, based on the power demands indicated by the memory queue level to achieve optimized system performance. The solution also features a control algorithm designed to achieve control accuracy and system stability.

Authors in [9] firstly investigate how power allocation affects server frequency in a single server using DVS, DVFS, DVS + DVFS for various workloads because optimal power allocation can vary for different scenarios. The power to frequency relationship is found to be linear for CPU bound processes and cubic for memory bound processes. Using a queueing theoretic model which takes into account the power frequency relationship, the mean response time for a server farm as a function of many factors is determined.

[10] introduces IdleCap, a power capping technique that provides a higher time- averaged processor frequency for a given power budget. It works by repeatedly alternating be- tween the extreme states maintaining a fixed average power budget and reducing the mean response time significantly as compared to other capping techniques.

[33] Uses DVFS jointly with rate adaptation for utilization control. A two-layered CPU utilization control architecture is presented in which the primary loop uses frequency scaling to control the CPU utilization of each processor while the secondary loop controls the utilization of all the processors at the cluster level by adopting rate adaptation. The results show that this control solution outperforms other controllers that rely solely on rate adaptation.

Authors in [19] propose a power limiting framework conductor which dynamically distributes available power to different compute nodes and cores based on the available slack to improve performance along with

upscaling and downscaling of processor frequency to decrease execution time.

In [6], authors present a power capping framework, Star-Cap, that incorporates software based models of power consumption, rather than physical measurements to enforce system level power budgets. This removes the need for physical measurement infrastructure to implement power capping. Star-Cap allows the power caps for individual machines to adapt to the demands of the workload and the results show that a better response time can be achieved with minimal overhead.

In [12] authors explore the power allocation budgeting problem for the PKG and DRAM domain for maximizing performance and subsequently, an optimal power allocation strategy is proposed. Same authors then present power lim- iting framework at cluster level *CLIP* [38], which divides application types into three kinds for application character- ization and performance modeling to divide a given power budget to different nodes in a cluster and their components such that execution time is minimized.

4. Conclusions

The desire for achieving exascale performance has pushed the modern computing systems to operate at their maximum operating frequency and bandwidths. Conse- quently, their power consumption has also increased drasti- cally, subsequently increasing their power and energy con- sumption. For mitigation of this problem, several strategies making use of DVFS/Throttling have been proposed. As the formulation of the common issue shifts from reducing power consumption to limiting it and then maximizing performance, power limiting through RAPL has become quite relevant in modern computing systems. In this work, we have reviewed several power limiting strategies which make use of mostly Intel RAPL and in some cases DVFS to limit the power consumption of a computing system and subsequently maximize performance under that envelope. We hope that this review work will server as a reference to future researchers who want to have a quick overview of the area of power limiting.

References

[1] Intel power governor.https://software.intel.com/en-us/articles/ intel-power-governor. Accessed: 2017-22-12.

[2] Intel Software Developer's Manual. In 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] Andrea Borghesi, Francesca Collina, Michele Lombardi, Michela Milano, and Luca Benini. Power Capping in High Performance Com- puting Systems, pages 524–540. Springer International Publishing, Cham, 2015.

[5] Ming Chen, Xiaorui Wang, and Xue Li. Coordinating processor and main memory for efficientserver power control. In Proceedings of the International Conference on Supercomputing, ICS '11, pages 130–140, New York, NY, USA, 2011. ACM.

[6]J. D. Davis, S. Rivoire, and M. Goldszmidt. Star-cap: Cluster power management using software-only models. In 2014 43rd International Conference on Parallel Processing Workshops, pages 114–120, Sept 2014.

[7]M. Etinski, J. Corbalan, J. Labarta, and M. Valero. Optimizing job performance under a given power constraint in hpc centers. In International Conference on Green Computing, pages 257–267, Aug 2010.

[8] 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.

[9] Anshul Gandhi, Mor Harchol-Balter, Rajarshi Das, and Charles Le- furgy. Optimal power allocation in server farms. In Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '09, pages 157–168, New York, NY, USA, 2009. ACM.

[10]Anshul et. al Gandhi. Power capping via forced idleness. In Proceedings of Workshop on Energy-Efficient Design, 2009.

[11] 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.

[12] R. Ge, X. Feng, Y. He, and P. Zou. The case for cross-component power coordination on power bounded systems. In 2016 45th Inter- national Conference on

Parallel Processing (ICPP), pages 516–525, Aug 2016.

[13] R. Ge, X. Feng, S. Song, H.C. Chang, D. Li, and K.W. Cameron. PowerPack: Energy profiling and analysis of high-performance sys- tems and applications. Parallel and Distributed Systems, IEEE Transactions on, 21:658–671, 2010.

[14] 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.

[15] 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.

[16] T. Komoda, S. Hayashi, T. Nakada, S. Miwa, and H. Nakamura. Power capping of cpu-gpu heterogeneous systems through coordi- nating dvfs and task mapping. In 2013 IEEE 31st International Conference on Computer Design (ICCD), pages 349–356, Oct 2013.

[17] Charles Lefurgy, Xiaorui Wang, and Malcolm Ware. Power capping: A prelude to power shifting. Cluster Computing, 11(2):183–195, June 2008.

[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] Aniruddha Marathe, Peter E. Bailey, David K. Lowenthal, Barry Rountree, Martin Schulz, and Bronis R. de Supinski. A Run-Time System for Power-Constrained HPC Applications, pages 394–408. Springer International Publishing, Cham, 2015.

[20] Xinxin Mei, Qiang Wang, and Xiaowen Chu. A survey and measure- ment study of gpu dvfs on energy conservation. Digital Communi- cations and Networks, 3(2):89 – 100, 2017.

[21] J. Park, D. Shin, N. Chang, and M. Pedram. Accurate modeling and calculation of delay and energy overheads of dynamic voltage scaling in modern high-performance microprocessors. In 2010 International Symposium on Low-Power Electronics and Design (ISLPED), pages 419–424, 2010.

[22] J. Sartori and R. Kumar. Three scalable approaches to improving many-core throughput for a given peak power budget. In 2009 International Conference on High Performance Computing (HiPC), pages 89–98, Dec 2009.

[23] 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.

[24] B. Su, J. Gu, L. Shen, W. Huang, J. L. Greathouse, and Z. Wang. Ppep: Online performance, power, and energy prediction framework and dvfs space exploration. In 2014 47th Annual IEEE/ACM Interna- tional Symposium on Microarchitecture, pages 445–457, Dec 2014.

[25] 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.

[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, F. Liu, and M.W Schmidt. Dynamic frequency scaling and energy saving in quantum chemistry applica- tions. In Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum, IPDPSW '11, pages 837–845, Washington, DC, USA, 2011. IEEE Computer Society.

[29] V. Sundriyal, M. Sosonkina, and Z. Zhang. Achieving energy effi- ciency during collective communications. Concurrency and Compu- tation: Practice and Experience, 2012.

[30] Vaibhav Sundriyal, Masha Sosonkina, and Mark Gordon. Runtime power limiting in gamess on dual-socket nodes. In 2017 4th Annual Conference on Computational Science and Computational Intelli-gence, 2017.

[31] G. Terzopoulos and H. D. Karatza. Dynamic voltage scaling schedul- ing on power-aware clusters under power constraints. In 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications, pages 72–78, Oct 2013.

[32] A. Vishnu, S. Song, A. Marquez, K. Barker, D. Kerbyson, K. Cameron, and P. Balaji. Designing Energy Efficient Communi- cation Runtime Systems for Data Centric Programming Models. In Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Com- puting 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]X. Wang, X. Fu, X. Liu, and Z. Gu. Power-aware cpu utilization control for distributed real-time systems. In 2009 15th IEEE Real- Time and Embedded Technology and Applications Symposium, pages 233–242, April 2009.

[34] Xin Zhan and S. Reda. Techniques for energy-efficient power budgeting in data centers. In 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pages 1–7, May 2013.

[35] Huazhe Zhang and Henry Hoffmann. A quantitative evaluation of the rapl power control system.

[36] Huazhe Zhang and Henry Hoffmann. Maximizing performance under a power cap: A comparison of hardware, software, and hybrid techniques. SIGPLAN Not., 51(4):545–559, March 2016.

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

[38] P. Zou, T. Allen, C. H. D. IV, X. Feng, and R. Ge. Clip: Cluster-level intelligent power coordination for power-bounded systems. In 2017 IEEE International Conference on Cluster Computing (CLUSTER), pages 541–551, Sept 2017.