Power-aware and Temperature Restrain Modeling for Maximizing Performance and Reliability

Laxmikant V. Kale, Akhil Langer[‡], and Osman Sarood
Department of Computer Science
University of Illinois at Urbana-Champaign
[‡]alanger@illinois.edu

ABSTRACT

Ability to constrain power consumption in the recent hardware architectures is a powerful capability that can be leveraged for efficient utilization of available power. We propose to develop power-aware performance models that can predict job performance given a resource configuration, that is, the CPU/memory power cap, the number of nodes, etc. In addition to performance optimization under a fixed power budget, our proposed model also alleviates the difference in thermal profiles amongst different processors to achieve a balance in the overall temperature distribution of the data center. Reduced temperature of operation improves the reliability of the system in addition to saving cooling energy of the data center, while minimizing the overall execution time of the jobs. The power-aware performance model can be used to determine the optimal resource configurations for a job or for a set of jobs, with the aim of efficient utilization of power.

1. POWER-AWARE PERFORMANCE MODELING

Power requirements of a data center are computed using the Thermal Design Power (TDP) of its subsystems. However, TDP limit is hardly reached in normal operation for any individual processor. Nonetheless, TDP amount of power has to be allocated to the subsystems, in order to avoid circuit trips on the rare occasions when the power draw reaches TDP. Clearly, this is excessive and wasteful allocation of power. Recent microprocessor architectures such as, Intel SandyBridge [1], IBM Power6 [2], IBM Power7 [3], AMD Bulldozer [4], allow constraining the CPU and memory power consumption to below their TDP limit. This feature can be used to constrain the power consumption of nodes, and using the saved power to add more nodes to the data center. This is also called as overprovisioning [5, 6, 7].

Applications do not yield proportionate improvements in performance as the power allocated to the CPU and/or memory is increased. For a given power budget, it might be beneficial to run an application on larger number of nodes with each node capped at a power level below its TDP than running the fewer nodes each allocated TDP amount of power [5, 6]. In addition to cores/memory, caches constitute a significant portion of the node power consumption. However, the benefits of using different levels of caches on application performance may not be proportional to their power consumption. Ability to dynamically enable/disable caches at various levels through software, is also being supported by the hardware architectures [8, 9]. Similar to the power savings by capping CPU/memory power, judicially turning the caches on/off can save power, which can be used to add more nodes. Different applications respond differently to changes in CPU/memory power and/or availability of caches. In order to allocate the resources (i.e. CPU/memory power caps, number of nodes, etc.) to jobs or to a set of jobs, the Power-aware Strong Scaling (PASS) performance model of the jobs is required [7]. This is where modeling can be used to predict the applications performance for any given resource configuration.

Application performance modeling using DVFS has been extensively studied [10, 11, 12, 13]. Because of the difference in the CPU/memory characteristics of an application, different applications running under the same CPU power cap can have different CPU frequencies. Based on the on-chip activity of the application, user-specified CPU power cap is ensured by using a mixture of DVFS and CPU throttling [1, 14]. Power consumption of the chip can be modeled as a function of the leakage power of the chip, cache and memory access rates of the application, and the fixed idle/ base power of the chip [15, 16]. Both leakage power, and cache, memory access rates can be modeled as functions of chip frequency. Chip frequency, in turn, can be used to model the execution time of the application [10, 11, 12, 17]. These models can be combined together with the strong-scaling model (e.g. Downey's strong scaling model [18]) to get a holistic model that can predict an application's performance for any resource configuration.

2. MODELING CHIP TEMPERATURES

The temperature of the data center is maintained such that the cooling is sufficient to cool-down the processors with hot-spots, whose temperature can be up to 30°C more than the processor with lowest temperature [19]. This is done due to fear of increased node failures at higher temperatures because Mean Time Between Failure (MTBF) of a processor is directly proportional to the exponential of its temperature [20, 21, 22]. It has also been reported that for every 10°C increase in temperature, fault rate doubles [20, 23, 24, 25]. Therefore, besides reducing the cooling energy of the data center, restraining the temperature of the processors also increases the MTBF of the system. Temperature control is achieved by reducing the frequency of the chip (using Dynamic Voltage and Frequency Scaling, DVFS), when the temperature increases beyond a threshold, and by increasing the frequency when the temperature decreases below a certain limit [26, 27]. Since, the optimum checkpoint frequency

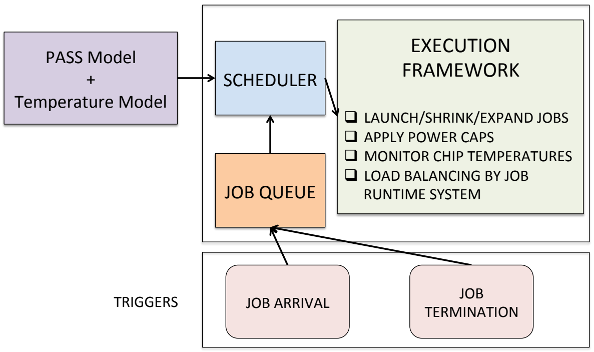

Figure 1: High level overview of the resource manager

for fault tolerance is computed based on the MTBF of the system, increase in MTBF due to temperature control, reduces the checkpoint frequency. This reduces the overhead of checkpoint/restart in the overall execution time of the job. As can be understood from the context, temperature control brings a trade-off between the checkpoint/restart overhead and job program time which increases due to control of frequency. The optimal temperature, where the overall execution time of the job is the least, varies from job to job [19].

In the contemporary research, focus has been on reducing the energy consumption of the applications which is achieved by using DVFS. However, the focus is shifting towards efficient utilization of available power as power is becoming a limiting factor. We propose modeling maximum temperature of a processor as a function of the the power cap of the CPU and the cooling temperature of the data center. Identical chips exhibit significant variation in their temperatures even when running under identical settings. This is attributed to chip-to-chip fabrication precision during manufacturing. This effect will be even more pronounced as new revolutionary chip technologies will be developed to reach exascale. For example, the recent 2014 DoE report on top ten exascale research challenges ([28]) shows that with the Near Threshold Voltage (NTV) operation, the variability in circuit speeds increases dramatically to 50%. This implies that processor temperatures will have to be individually modeled for each processor. However, the cost of temperature modeling is a one time cost and hence negligible as compared to the overall operations of the data center.

3. USING THE MODELS FOR IMPROVED PERFORMANCE AND RELIABILITY

Figure 1 shows the overall block-diagram of an online resource manager that makes resource allocation decisions while taking into account the power-aware strong-scaling performance of the applications and the temperature response of the processors under a given power cap, and machine room temperature settings. Power-aware performance of the jobs can be used to determine optimal allocation of resources to the set of jobs being scheduled by the data center. Use of online integer linear program optimization for optimal allocation of power to jobs being submitted to a data center has been shown in [7]. In order to improve the reliability of the system, additional temperature constraints are added to the linear program to restrain processor temperature from going beyond a threshold.

We also plan to address the important challenge of handling speed variability across identical nodes. Even in current architectures this variability becomes pronounced when the processors are power capped or temperature restrained. For example, Rountree, et. al. [1] show variation of 8% in performance across 64 processors when the CPU is power capped at 50W (where the TDP is 85W). We have observed further increase in variation with CPU power caps below 50W. In our earlier work [19], we have shown variation in temperatures across processors can be up to 30°C. Hence, temperature restrain through DVFS leads to different processor speeds. These variations cause load imbalance and hence synchronization issues in HPC applications. We propose the use of over-decomposition and subsequent dynamic load balancing through object migration to achieve load balance. Overdecompositon and object migration also allows for dynamic restriction or extraction of jobs to a different number of processors, during its execution. This gives an additional degree of freedom to the online resource manager to remake optimal resource allocation decisions with the current set of jobs, by changing the configuration of running jobs as new jobs arrive and/or running jobs terminate.

4. EFFORT

Work requires developing performance models and its empirical validation in a data center that supports power capping and temperature control of the room. Charm++ [29, 30] runtime system provides support for writing custom load balancers. Existing load balancers will have to be adapted to take into account the heterogeneity of nodes under the proposed settings. Charm++ features can be easily realized for legacy MPI codes by using the Adaptive MPI framework (AMPI) provided by Charm++. AMPI [31, 32] is built on top of Charm++ framework and uses light-weight user levels threads instead of processes. This allows us to virtualize several MPI ranks on a single physical core, which brings the benefits of over-decompisition. The ranks can then be migrated to realize benefits such as load balancing and fault tolerance. Some effort will be required towards development of a tool for automated conversion of MPI programs to AMPI.

Empirical validation on a relatively small-scale data center will be followed by large scale projections for exascale through simulation. We estimate that an effort with 1 FTE and 2 graduate research assistants, over a period of two years will be needed to carry out the proposed research program.

5. REFERENCES

[1] Barry Rountree, Dong H Ahn, Bronis R de Supinski, David K Lowenthal, and Martin Schulz. Beyond DVFS: A First Look at Performance Under a Hardware-enforced Power Bound. In *IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, 2012.

[2] Brad Behle, Nick Bofferding, Martha Broyles, Curtis Eide, Michael Floyd, Chris Francois, Andrew Geissler, Michael Hollinger, Hye-Young McCreary, Cale Rath, et al. IBM Energyscale for POWER6 Processor-based Systems. *IBM White Paper*, 2009.

[3] Martha Broyles, Chris Francois, Andrew Geissler, Michael Hollinger, Todd Rosedahl, Guillermo Silva, Jeff Van Heuklon, and Brian Veale. IBM Energyscale for POWER7 Processor-based Systems. *white paper, IBM*, 2010.

[4] Advanced Micro Devices. BIOS and Kernel Developer's guide (BKDG) for AMD Family 15h Models 00h-0fh Processors. January 2012.

[5] Tapasya Patki, David K Lowenthal, Barry Rountree, Martin Schulz, and Bronis R de Supinski. Exploring Hardware Overprovisioning in Power-constrained, High Performance Computing. In *Proceedings of the 27th international ACM conference on International conference on supercomputing*, pages 173–182. ACM, 2013.

[6] Osman Sarood, Akhil Langer, Laxmikant Kalé, Barry Rountree, and Bronis de Supinski. Optimizing Power Allocation to CPU and Memory Subsystems in Overprovisioned HPC Systems. In *Cluster Computing (CLUSTER), 2013 IEEE International Conference on*, pages 1–8. IEEE, 2013.

[7] Osman Sarood, Akhil Langer, Abhishek Gupta, and Laxmikant V Kale. Maximizing Throughput of a Data Center Under a Strict Power Budget. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '14, New York, NY, USA, 2014. ACM.

[8] Intel®64 and IA-32 Architectures Software Developer's Manual. `http://science.energy.gov/~/media/ascr/ascac/pdf/meetings/20140210/Top10reportFEB14.pdf`.

[9] David H Albonesi. Selective Cache Ways: On-demand Cache Resource Allocation. In *Microarchitecture, 1999. MICRO-32. Proceedings. 32nd Annual International Symposium on*, pages 248–259. IEEE, 1999.

[10] Chung-Hsing Hsu and Wu-Chun Feng. Effective Dynamic Voltage Scaling through CPU-Boundedness Detection. In *Proceedings of the 4th International Conference on Power-Aware Computer Systems*, PACS'04, 2005.

[11] Kihwan Choi, Ramakrishna Soma, and Massoud Pedram. Fine-grained Dynamic Voltage and Frequency Scaling for Precise Energy and Performance Tradeoff based on the Ratio of Off-chip Access to On-chip Computation Times. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 24(1):18–28, 2005.

[12] Kihwan Choi, Ramakrishna Soma, and Massoud Pedram. Dynamic Voltage and Frequency Scaling based on Workload Decomposition. In *Proceedings of the 2004 international symposium on Low power electronics and design*, pages 174–179. ACM, 2004.

[13] K. Seth, A. Anantaraman, F. Mueller, and E. Rotenberg. Fast: frequency-aware static timing analysis. In *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*, pages 40–51, Dec 2003.

[14] Juan M Cebrian, Juan L Aragón, José M García, Pavlos Petoumenos, and Stefanos Kaxiras. Efficient Microarchitecture Policies for Accurately Adapting to Power Constraints. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–12. IEEE, 2009.

[15] Robert Graybill and Rami Melhem. *Power aware computing*. Kluwer Academic Publishers, 2002.

[16] Xi Chen, Chi Xu, and R.P. Dick. Memory Access Aware on-line Voltage Control for Performance and Energy Optimization. In *Computer-Aided Design (ICCAD), 2010 IEEE/ACM International Conference on*, pages 365–372, 2010.

[17] Kiran Seth, Aravindh Anantaraman, Frank Mueller, and Eric Rotenberg. Fast: Frequency-aware Static Timing Analysis. *ACM Transactions on Embedded Computing Systems (TECS)*, 5(1):200–224, 2006.

[18] Allen B Downey. A Parallel Workload Model and Its Implications for Processor Allocation. *Cluster Computing*, 1(1):133–145, 1998.

[19] Osman Sarood, Esteban Meneses, and Laxmikant V. Kale. A 'cool' way of improving the reliability of hpc machines. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 58:1–58:12, New York, NY, USA, 2013. ACM.

[20] Ericsson. Reliability Aspects on Power Supplies. *Technical ReportDesign Note 002, Ericsson Microelectronics, April 2000.*

[21] J. Srinivasan, S.V. Adve, P. Bose, and J.A. Rivers. The impact of technology scaling on lifetime reliability. In *Dependable Systems and Networks, 2004 International Conference on*, pages 177–186, 2004.

[22] J. Archuleta Chung H. Hsu, W. Feng. Towards Efficient Supercomputing: A Quest for the Right Metric.

[23] Chung hsing Hsu, Wu chun Feng, and Jeremy S. Archuleta. Towards efficient supercomputing: A quest for the right metric. In *In Proceedings of the HighPerformance Power-Aware Computing Workshop*, 2005.

[24] Wu-chun Feng. Making a case for efficient supercomputing. volume 1, pages 54–64, New York, NY, USA, October 2003. ACM.

[25] Wu-chun Feng. The Importance of Being Low Power in High-Performance Computing. *Cyberinfrastructure Technology Watch Quarterly (CTWatch Quarterly)*, 1(3), August 2005.

[26] Osman Sarood and Laxmikant V. Kalé. A 'cool' load balancer for parallel applications. In *Proceedings of the 2011 ACM/IEEE conference on Supercomputing*, Seattle, WA, November 2011.

[27] Osman Sarood, Phil Miller, Ehsan Totoni, and L. V. Kale. 'Cool' Load Balancing for High Performance Computing Data Centers. In *IEEE Transactions on Computer - SI (Energy Efficient Computing)*, September 2012.

[28] The Department of Energy Report on Top Ten Exascale Research Challenges. `http://science.energy.gov/~/media/ascr/ascac/pdf/meetings/20140210/Top10reportFEB14.pdf`.

[29] Bilge Acun, Abhishek Gupta, Nikhil Jain, Akhil

Langer, Harshitha Menon, Eric Mikida, Xiang Ni, Michael Robson, Yanhua Sun, Ehsan Totoni, Lukasz Wesolowski, and Laxmikant Kale. Parallel Programming with Migratable Objects: Charm++ in Practice. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '14, New York, NY, USA, 2014. ACM.

[30] Laxmikant Kale, Anshu Arya, Nikhil Jain, Akhil Langer, Jonathan Lifflander, Harshitha Menon, Xiang Ni, Yanhua Sun, Ehsan Totoni, Ramprasad Venkataraman, and Lukasz Wesolowski. Migratable Objects + Active Messages + Adaptive Runtime = Productivity + Performance A Submission to 2012 HPC Class II Challenge. Technical Report 12-47, Parallel Programming Laboratory, November 2012.

[31] Orion Lawlor, Milind Bhandarkar, and Laxmikant V. Kalé. Adaptive mpi. Technical Report 02-05, Parallel Programming Laboratory, Department of Computer Science, University of Illinois at Urbana-Champaign, 2002.

[32] Chao Huang, Gengbin Zheng, Sameer Kumar, and Laxmikant V. Kalé. Performance Evaluation of Adaptive MPI. In *Proceedings of ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming 2006*, March 2006.