Power Management for Exascale*

Kamil Iskra, Kazutomo Yoshii, Rinku Gupta, Pete Beckman

Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue, Argonne, IL 60439, USA
{iskra,kazutomo,rgupta,beckman}@mcs.anl.gov

1 Background

In order to meet the exascale goals, today's top HPC systems will need to scale by two orders of magnitude, at the same time increasing their power consumption by only an integer factor. Power consumption of both individual nodes and the overall system is thus a critical issue to address [2].

Most performance studies of large-scale HPC systems and their workloads have focused primarily on flops, bandwidth, and latency. Few concrete studies exist that focus on quantifying power and energy consumption at the hardware and software levels. Until recently, system vendors have had little incentive to expose extensive system and component-level power interfaces to users. Consequently, the power-management methodology is lacking, and the underlying capabilities in today's computer systems are limited or missing.

Exascale systems, consisting of hundreds of thousands of nodes drawing tens of megawatts of electrical power, mandate a need for new, systemwide methodologies and procedures for power monitoring, management, and scheduling.

2 Power Monitoring

On current large-scale HPC systems, we are forced to rely on built-in environmental-monitoring capabilities designed primarily to help identify insufficient cooling and power distribution, not to monitor energy usage driven by application workloads. Our evaluation of IBM Blue Gene systems in that respect has shown [18] limited temporal and spatial resolution of the data available through these channels, further complicated by the latency with which the raw power data becomes available.

To perform meaningful, informed power management, we will need convenient access to accurate power consumption information. Key capabilities of such interface include the following:

Low overhead access: ideally, the overhead should be of the same order of magnitude as obtaining the current system time.

Instantaneous data: the returned data should correspond to the current power draw; if only older data is available, it needs to be accurately timestamped.

Local monitoring: separate power monitoring information should be available for every compute node; rack-scale bulk power data is insufficient.

High granularity data: raw power data should be divided between node components; at least the CPU, memory, and interconnect should be measured individually.

Energy vs power measurement: while for system management purposes power draw is probably the metric of choice, developers wishing to optimize their code for optimal power usage may be more interested in overall energy consumption during the execution of a section of code. Energy consumption can be estimated by repeatedly sampling the instantaneous power draw, which should be done by a kernel thread running on a different core from the application, or, to reduce jitter, performed in hardware.

Obviously, the characteristics of such software interface depend on the capabilities of the measuring hardware; these two should be codesigned.

3 Power Management

Our experiments on current HPC systems indicate [18] that very little power management is taking place. The systems are tuned for energy efficiency under sustained high load (performance per watt), not for power saving during idle periods. In particular, the power draw of the interconnect on Blue Gene/Q appears to be independent of load, and that of the CPU varies only by some 20%; the only component showing significant changes in power draw under different loads is DRAM—by a factor of 2 or more.

For meaningful node power management at exascale, we expect that the OS and runtime will need capabilities such as the following:

- adjust the CPU speed,

*This work was supported by the Office of Advanced Scientific Computer Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

- put individual cores in low-power (idle) state with minimum latency,
- adjust the memory bus speed (if there is a memory bus at exascale),
- put individual memory banks in low-power state, and
- put the network interface in low-power state.

The corresponding interfaces are conceptually simple; however, major changes to the hardware designs will be required in order to enable this level of power management.

4 Power Scheduling

Electrical power is predicted to be a greatly oversubscribed resource at exascale and thus to become first-class scheduling constraint. We expect an application job to receive a power budget from the global system scheduler when the job is first being started. The job, through the runtime system, should then be able to dynamically allocate portions of that budget to different HPC system components (enclaves) on an as-needed basis. Namely, operations on the critical path should be supplied with adequate power budget for maximum efficiency, at the expense of less-critical operations. For example, when a job initializes, a large portion of the budget could be allocated to the storage system to read the input data into memory as quickly as possible; but once the data has been read, power should be reallocated to the compute and interconnect fabric until the next checkpoint or job termination time.

For such execution model to be feasible, a multilevel power scheduling infrastructure must be provided, from global system level, through enclaves, to individual nodes, comprising at each level scheduling, monitoring, and enforcement subcomponents, with request and feedback propagation between the levels and with interfaces to the runtime system at multiple levels. The runtime system and the application should be involved in the decision-making process; for example, if the power budget available to the compute fabric needs to be reduced, user code could provide input on whether it is preferable to slightly slow all the CPUs or to idle some of the CPUs (perhaps even release the nodes to the global scheduler) while keeping the remaining ones running at full speed.

5 Related Work

Power consumption has increasingly been recognized as a limiting factor in large data centers and supercomputer facilities [3–5, 15, 16]. Research in power-aware scheduling has been vast and diverse [8, 9, 11–14, 17].

Garcia et al. [7] developed an instruction-level energy consumption model for many-core architectures and demonstrated its accuracy by experimenting on an IBM Cyclops-64 chip. Feng et al. [6] developed a power/energy profiling framework for HPC cluster systems, measuring power consumption by tapping digital multimeters into DC lines. Alam et al. [1] compared various performance aspects of IBM Blue Gene/P and Cray XT4, including performance per watt and power consumption of HPC applications. Hennecke et al. [10] presented an overview of the power measurement capabilities of Blue Gene/P.

6 Summary

Challenges addressed: Power, specifically global control of power management.

Maturity: The underlying power optimization techniques, such as dynamic voltage and frequency scaling or power gating, are well studied and understood. To utilize these techniques properly, however, we need a coordinated, multilevel power-scheduling infrastructure running on top, which is a new research area with unknown risks.

Uniqueness: While power-limited computing is the order of the day in mobile consumer devices such as cell phones, the efforts there focus on optimizing the *idle* power draw, which is the state those devices are predominantly in. HPC systems are instead expected to be under load for much of the time, so the set of challenges is different. Because of their scale, exascale systems will be the ones most affected by limited available power, so we expect that they will be the first to need a working solution.

Novelty: We stress the different objectives of the related work presented earlier compared with what we have outlined. The bulk of the prior research has focused on *slowing* system components in order to optimize performance per watt and reduce the overall energy consumption, thus saving on running costs—a noble goal in itself. Large-scale HPC systems, however, represent significant financial investments and need to be utilized as highly as possible before the inevitable progress makes them obsolete and too expensive to keep running. Thus, we are not interested in making the system as a whole slower for minor energy savings; rather, we want to redirect the limited available power budget to those components that can benefit most, with the goal of *accelerating* the computations in order to maximize the system throughput.

Applicability: We expect power management solutions developed for exascale systems to trickle down to smaller-scale HPC systems and possibly also to other commercial high-load systems. Traditional data-center servers, however, which are idle much of the time, need a different set of solutions.

Effort: Given the multilevel solution required, a larger team would be preferable, comprising OS, runtime, and machine scheduling experts: possibly five people working for three to five years.

References

[1] S. Alam, R. Barrett, M. Bast, M. R. Fahey, J. Kuehn, C. McCurdy, J. Rogers, P. Roth, R. Sankaran, J. S. Vetter, P. Worley, and W. Yu. Early evaluation of IBM BlueGene/P. In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, SC '08. IEEE Press, 2008.

[2] DOE. Report from the Architectures and Technology for Extreme Scale Computing Workshop, 2009.

[3] W. Feng and K. Cameron. The Green500 list: Encouraging sustainable supercomputing. *Computer*, 40:50–55, Dec. 2007.

[4] W. Feng, X. Feng, and R. Ge. Green supercomputing comes of age. *IT Professional*, 10:17–23, Jan. 2008.

[5] W. Feng and T. Scogland. The Green500 list: Year one. *Parallel and Distributed Processing Symposium*, 2009.

[6] X. Feng, R. Ge, and K. W. Cameron. Power and energy profiling of scientific applications on distributed systems. In *Proceedings of the International Parallel and Distributed Processing Symposium*. IEEE Computer Society, 2005.

[7] E. Garcia, D. Orozco, and G. Gao. Energy efficient tiling on a many-core architecture. In *Proceedings of 4th Workshop on Programmability Issues for Heterogeneous Multicores (MULTIPROG-2011); 6th International Conference on High-Performance and Embedded Architectures and Compilers (HiPEAC)*, pages 53–66, Heraklion, Greece, January 2011.

[8] F. Harada, T. Ushio, and Y. Nakamoto. Power-aware resource allocation with fair QoS guarantee. In *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, RTCSA '06, pages 287–293. IEEE Computer Society, 2006.

[9] T. Heath, B. Diniz, E. V. Carrera, W. Meira, Jr., and R. Bianchini. Energy conservation in heterogeneous server clusters. In *Proceedings of the 10th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '05, pages 186–195. ACM, 2005.

[10] M. Hennecke, W. Frings, W. Homberg, A. Zitz, M. Knobloch, and H. Böttiger. Measuring power consumption on IBM Blue Gene/P. *Computer Science – Research and Development*.

[11] C. Hsu and W. Feng. A feasibility analysis of power awareness in commodity-based high-performance clusters. In *Proceedings of the IEEE International Conference on Cluster Computing*, Sept. 2005.

[12] C. Hsu and W. Feng. A power-aware run-time system for high-performance computing. In *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*. IEEE Computer Society, 2005.

[13] J. Liu, D. Poff, and B. Abali. Evaluating high performance communication: A power perspective. In *Proceedings of the 23rd International Conference on Supercomputing*, ICS '09, pages 326–337. ACM, 2009.

[14] T. M. Lynar, R. D. Herbert, S. Chivers, and W. J. Chivers. Resource allocation to conserve energy in distributed computing. *Int. J. Grid Util. Comput.*, 2(1):1–10, May 2011.

[15] E. Pakbaznia, M. Ghasemazar, and M. Pedram. Temperature-aware dynamic resource provisioning in a power-optimized datacenter. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '10, pages 124–129, 3001 Leuven, Belgium, 2010. European Design and Automation Association.

[16] N. Rasmussen. Calculating total cooling requirements for data centers. *American Power Conversion, white paper*, 2007.

[17] L. Wang, G. von Laszewski, J. Dayal, and F. Wang. Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with DVFS. In *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, CCGRID '10, pages 368–377. IEEE Computer Society, 2010.

[18] K. Yoshii, K. Iskra, R. Gupta, P. Beckman, V. Vishwanath, C. Yu, and S. Coghlan. Evaluating power-monitoring capabilitieson IBM Blue Gene/P and Blue Gene/Q. In *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER '12)*, Beijing, China, 2012. (to appear).