



Energy Efficiency Evaluation of Distributed Systems

James Phung¹(✉), Young Choon Lee², and Albert Y. Zomaya¹

¹ School of Computer Science, The University of Sydney, Sydney, Australia
{james.phung,albert.zomaya}@sydney.edu.au

² Department of Computing, Macquarie University, Sydney, Australia
young.lee@mq.edu.au

Abstract. Rapid growth in Big Data and Cloud technologies has fueled rising energy demands in large server systems such as data centers, leading to a need for effective power management. In this paper, we investigate the energy consumption characteristics of data-intensive distributed applications in terms of the CPU and memory subsystem. To this end, we develop *PowerSave* as a lightweight software framework that enables dynamic reconfiguration of power limits. PowerSave uses Running Average Power Limit (RAPL) to impose power limits. Our evaluation study, conducted on three different real systems, demonstrates that for workloads typical of servers used in data centers, higher power caps correlate with higher overall CPU energy use.

Keywords: Big Data · Cloud · Energy monitoring · Power management · RAPL · Virtualization

1 Introduction

The rise of Cloud technologies and Big Data has led to rapid growth in storage and computational power requirements, and consequently, energy demands. An example of this phenomenon is the popularity of virtual computing services provided by Cloud service providers such as Amazon Web Services. These services range from database services to bare metal processing power. Likewise, scientific and engineering applications such as the Montage astronomy application [6] impose enormous processing and data demands, requiring that large amounts of real-time data (often on a terabyte scale) be collected and processed in a short period of time.

In the interests of economic and environmental sustainability, it is desirable to design large-scale server systems with the goal of optimizing energy use. To this end, various energy-aware scheduling algorithms [4, 9], power models [3, 7, 11] and techniques to reduce power consumption of data centers [2, 10, 12] have been developed.

James Phung's work was supported by an Australian Government Research Training Program (RTP) Scholarship.

© Springer Nature Switzerland AG 2019

J. M. F. Rodrigues et al. (Eds.): ICCS 2019, LNCS 11540, pp. 756–763, 2019.

https://doi.org/10.1007/978-3-030-22750-0_75

In this paper, we investigate the impact of CPU power limiting on CPU energy consumption and performance for a Macbook Pro laptop and two different types of Amazon EC2 virtualized instances. We use the RAPL feature of modern Intel CPUs to implement power limiting and monitor real-time energy use.

In order to streamline the process of setting power limits, we have developed PowerSave, a power management framework that provides a user-friendly interface for setting or locking various RAPL power limits. Also, we use an RAPL energy tracker that has been developed in-house in order to monitor the RAPL counters in real-time.

Specific contributions of this paper are as follows:

1. We develop PowerSave and an RAPL energy tracker to streamline the energy monitoring procedures; and
2. We evaluate the impact of CPU power limiting on CPU energy consumption and performance on a laptop, a virtual machine and a cluster of bare metal instances using a variety of big data benchmarks (HiBench benchmarks [5]).

We use HiBench benchmarks to investigate the impacts of CPU power limiting. We use various combinations of CPU power limits, turbo setting and benchmarks.

Our evaluation shows that there is a correlation between CPU power limits and CPU energy efficiency when CPU utilization is high. Energy efficiency rapidly decreases if the CPU consumes power beyond the thermal design power for sustained periods.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the basic functionality of our power management framework *PowerSave*. Section 4 describes the implementation details of *PowerSave*. Section 5 describes our experimental setup, evaluation methodology and results for our experiments. We draw our conclusions in Sect. 6.

2 Related Work

The RAPL feature of Intel CPUs has been applied by various researchers in a range of energy efficiency and monitoring applications. In [13], the problem of achieving proportionality in energy usage relative to workload is explored. The author proposes “a methodology to efficiently cap the power consumption of applications while meeting strict performance targets”. To confirm the validity of the power-capping scheme, the author runs the SPECweb benchmark for various workloads. During the benchmark, a Watts Up power meter is used to measure the energy usage of the entire system while subsystem energy use is measured using the RAPL interface’s energy meters.

In [8], Khan et al. model the system-level power consumption of two different computers using RAPL counters: one contains an Intel Haswell CPU and the other contains an Intel Skylake CPU. They establish that there is a strong correlation between system power usage and CPU power usage. Also, they investigate important issues such as performance overhead, sampling rate, and potential register overflows. Furthermore, they conduct their investigations on some non-bare metal Amazon AWS EC2 instances.

In [2], the researchers develop a new MapReduce workload manager to address the problem of processing batch MapReduce workloads efficiently in situations where some MapReduce jobs have to be performed on demand with little tolerance for latency such as an ad-hoc database query. To validate the workload manager's performance, the researchers performed simulations involving Facebook MapReduce traces. In addition, a live experiment was performed on Amazon EC2.

Energy-aware resource allocation is also studied in [1]. Unlike [2], the authors of [1], consider the problem of allocating virtual machines to servers and virtual machine migration in order to minimize energy use. New algorithms are designed and tested by simulation.

3 PowerSave

PowerSave¹ is a lightweight software framework that allows the user to specify power limits for the CPU package, PP0, PP1 and DRAM using a text interface. Two CPU package power limits can be set: a short-term power limit and long-term power limit. Each of these limits is defined in terms of maximum average power consumption over a time window; both the average power value and time window are set by the user. It may be used in physical systems that feature an RAPL enabled CPU; also, it may be used in bare metal and virtualized instances if RAPL support is available. Thus, PowerSave enables rapid testing of the impact that different CPU power limits have on energy consumption and application performance in distributed systems.

4 Implementation

In this section, we describe the implementation details of our PowerSave framework and RAPL energy tracker.

4.1 PowerSave

Our PowerSave framework consists of the RAPLCap library and a C program PowerSet that provides a user-friendly interface. The framework can operate in a variety of UNIX environments.

The C program uses the RAPLCap library to read and write to RAPL registers that control the power limiting behaviors of the CPU and memory subsystem. When the user runs the C code by invoking it on the UNIX command line, it checks to see if the CPU supports RAPL. If RAPL support is not found, the program aborts with an error message; otherwise, the user is presented with the

¹ <https://github.com/SydneyDSysGuy/PowerSave/blob/master/README.md>.

current RAPL power limits for the CPU package, PP0, PP1 and DRAM. Also, the user is notified as to whether the power limits are locked or not. If the power limits are unlocked, the user is prompted to enter new power limits if they wish to change them; also, the user is offered the choice of locking the power limits (an operation that cannot be reversed without restarting the system). Whenever the user requests a power limit change, the appropriate RAPL register is updated immediately.

4.2 RAPL Energy Tracker

Our RAPL energy tracker is a C program that can be run in a variety of UNIX environments. The CPU's RAPL counters are accessed via an interface provided by the UNIX *msr* (model specific registers) driver that provides raw access to RAPL counters. RAPL counters are sampled at regular intervals with a nominal sampling rate of one sample per second. RAPL energy readings are exported to a specified text file in real-time. The user is notified of the energy usage in real-time via the console.

5 Experiments

In this section, we describe our experimental setup, evaluation of our hypothesis, and results.

5.1 Experimental Setup

For our experiments, we use three different systems: a 2017 model Macbook Pro containing an Intel Mobile Core i5 Kaby Lake 2.3 GHz CPU (hereafter called “Macbook Pro”), an Amazon AWS EC2 i3.metal instance (hereafter called “AWS i3.metal”) and a cluster of 11 Amazon AWS EC2 m4.large instances (hereafter called “AWS m4.large”). For the Macbook Pro machine, we run our experiments directly on the machine; we do not use any virtualization technique. For the Amazon AWS instances, we run them on an *on-demand* basis; we do not use reserved spot instances or dedicated server reservations.

We use the RAPL energy tracker as described in Subject. 4.2 to capture RAPL energy values in real-time. A sampling rate of one measurement per second is used. The readings are saved to comma separated values (CSV) files for later offline analysis and evaluation.

The PowerSet utility is used to set the CPU RAPL power caps to the required values. The *cpufreq-set* utility is used to adjust the CPU frequency to the appropriate value and toggle the turbo boost feature as required. Idle time is allowed between experiments to permit the CPU (and system) to cool to its normal idle temperature.

5.2 Energy Efficiency Evaluation

To verify our hypothesis that there is an optimum RAPL power limit for minimizing energy use, we use a series of real-world MapReduce workloads. These workloads are representative of real-world data analytics jobs. They are drawn from the HiBench benchmark suite [5]. They are Aggregate, Join, PageRank, Scan, Sort, TeraSort and WordCount.

We run our experiments for a range of RAPL CPU package power limits in the case of the Macbook Pro and AWS i3.metal instance, or for various numbers of Hadoop DataNodes in the case of the AWS m4.large cluster. We run them without turbo boost when a power limit has been set; otherwise, we run them with turbo boost. The workload is allowed to run to completion.

5.3 Results

Figures 1, 2 and 3 show the average RAPL counter readings for CPU package and DRAM for various workloads, RAPL CPU package power limit and turbo settings running on the Macbook Pro, AWS i3.metal instance and AWS m4.large virtual cluster. For the Macbook Pro, average RAPL counter readings for PP0 are also shown.

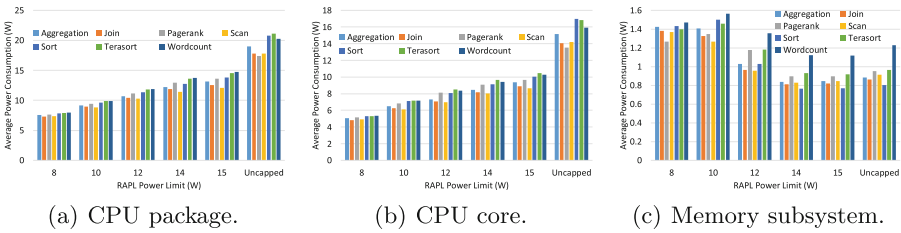


Fig. 1. Average power use for CPU package, CPU core and memory subsystem for various workloads on Macbook Pro.

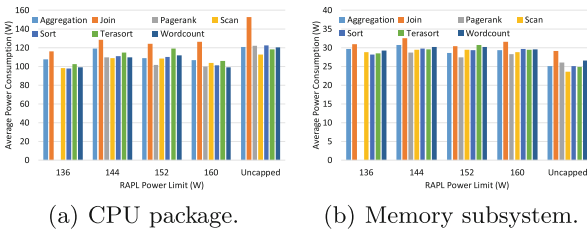


Fig. 2. Average power use for CPU package and memory subsystem for various workloads on AWS i3.metal instance.

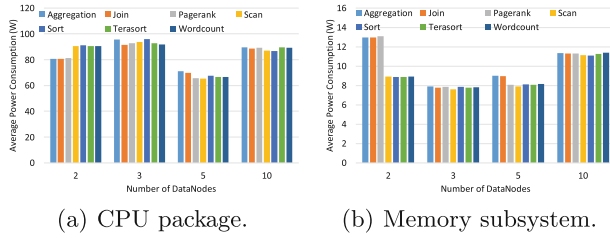


Fig. 3. Average power use for CPU package and memory subsystem for various workloads on AWS m4.large virtual cluster.

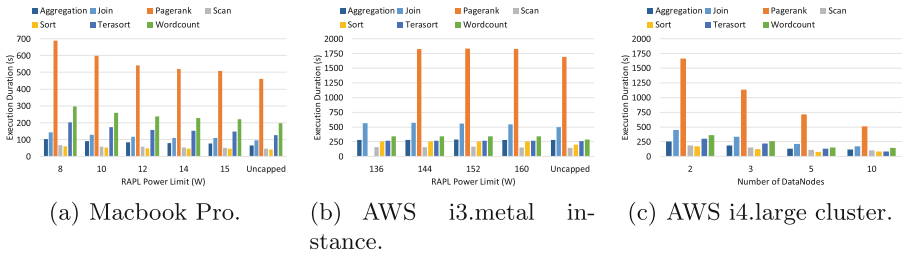


Fig. 4. Total execution durations for various workloads on various systems.

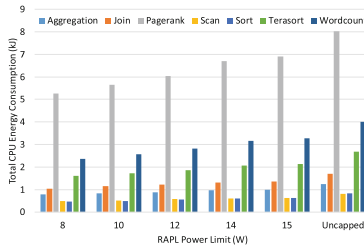


Fig. 5. Total CPU Energy Consumption for various workloads on Macbook Pro.

Figures 4(a) and (b) show the total execution durations for various workloads, RAPL CPU package power limit and turbo settings running on the Macbook Pro and AWS i3.metal instance.

Figure 4(c) shows the total execution durations for various workloads running on the AWS m4.large virtual cluster for various numbers of DataNodes.

Figure 5 shows the CPU energy consumption for various workloads, RAPL CPU package power limit and turbo settings running on the Macbook Pro.

6 Conclusion

We have developed PowerSave, a lightweight software framework that allows the user to set CPU power limits instantly using a simple user interface. It exploits the RAPL functionality of modern Intel CPUs in order to reduce CPU energy consumption in distributed systems. We use the framework to dynamically change power limits during our Big Data experiments on the Macbook Pro, an Amazon EC2 i3.metal instance and a cluster of Amazon EC2 m4.large instances.

We have shown that for heavy workloads, there exists an optimum power limit that minimizes energy use per unit of work. We have demonstrated that this is the case on both the laptop and Amazon EC2 instances.

References

1. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Gener. Comput. Syst.* **28**(5), 755–768 (2012). <https://doi.org/10.1016/j.future.2011.04.017>. <http://www.sciencedirect.com/science/article/pii/S0167739X11000689>. special section: energy efficiency in large-scale distributed systems
2. Chen, Y., Alspaugh, S., Borthakur, D., Katz, R.: Energy efficiency for large-scale mapreduce workloads with significant interactive analysis. In: *Proceedings of the 7th ACM European Conference on Computer Systems, EuroSys 2012*, pp. 43–56. ACM, New York (2012). <https://doi.org/10.1145/2168836.2168842>. <http://doi.acm.org/10.1145/2168836.2168842>
3. Dargie, W.: A stochastic model for estimating the power consumption of a processor. *IEEE Trans. Comput.* **64**(5), 1311–1322 (2015). <https://doi.org/10.1109/TC.2014.2315629>
4. Hoffmann, H.: JouleGuard: energy guarantees for approximate applications. In: *Proceedings of the 2015 Symposium on Operating Systems (SOSP 2015)*, ACM (2015)
5. Intel: HiBench Suite (2018). <https://github.com/intel-hadoop/HiBench>
6. Jacob, J.C., et al.: Montage; a grid portal and software toolkit for science grade astronomical image mosaicking. *Int. J. Comput. Sci. Eng.* **4**(2), 73–87 (2009)
7. Kansal, A., Zhao, F., Liu, J., Kothari, N., Bhattacharya, A.A.: Virtual machine power metering and provisioning. In: *Proceedings of the 1st Symposium on Cloud Computing*, ACM (2010)
8. Khan, K.N., Hirki, M., Niemi, T., Nurminen, J.K., Ou, Z.: RAPL in action: experiences in using RAPL for power measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.* **3**(2), 9–26 (2018). <https://doi.org/10.1145/3177754>. <http://doi.acm.org/10.1145/3177754>
9. Lee, Y.C., Zomaya, A.Y.: Energy conscious scheduling for distributed computing systems under different operating conditions. *IEEE Trans. Parallel Distrib. Syst.* **22**(8), 1374–1381 (2011). <https://doi.org/10.1109/TPDS.2010.208>
10. Lee, Y.C., Zomaya, A.Y.: Energy efficient utilization of resources in cloud computing systems. *J. Supercomputing* **60**(2), 268–280 (2012). <https://doi.org/10.1007/s11227-010-0421-3>. <http://dx.doi.org/10.1007/s11227-010-0421-3>

11. Möbius, C., Dargie, W., Schill, A.: Power consumption estimation models for processors, virtual machines, and servers. *IEEE Trans. Parallel Distrib. Syst.* **25**(6), 1600–1614 (2014)
12. Outin, E., Dartois, J.E., Barais, O., Pazat, J.L.: Seeking for the optimal energy modelisation accuracy to allow efficient datacenter optimizations. In: *Proceedings of the 16th IEEE/ACM Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, IEEE (2016)
13. Subramaniam, B.: Metrics, models and methodologies for energy-proportional computing. In: *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 575–578, May 2014. <https://doi.org/10.1109/CCGrid.2014.119>