PSim: A Simulator for Estimation of Power Consumption in a Cluster

Maaz Ahmed, Mohsin Khan, Waseem Ahmed, Rashid Mehmood, Abdullah Algarni, Aiiad Albeshri, Iyad Katib

***Abstract*—Emerging technologies have resulted in high end computing systems with many nodes with several processors per node. As we step into the exascale computing era, High Performance Computing (HPC) systems and technologies have provided an unprecedented level of computational capability for solving traditionally insolvable or extremely challenging social and scientific problems where experiments are impossible, dangerous, or inordinately costly. Achieving exascale computing requires millions of processors to be working in tandem which causes increased use of computational power to allow faster computing. One of the biggest challenge in HPC is to reduce power consumption as consumption of power has become a critical concern in HPC because power is a limiter for all computing platform as power is required to achieve performance as well as for cooling the system, especially in HPC cluster because, as the number of nodes in a cluster increases the total power consumption also increases. We have designed a simulator to estimate power and help designers to experiment execution of different applications on different HPC architectures and evaluate various power optimization techniques. A method which can be applied to manage the shutdown of CPUs when they are idle has been proposed, this method can be used in reducing power consumption. Our results show that when we run the applications using our proposed method of switching off the nodes when idle for more than user defined idle period threshold T_i seconds we achieve power saving of about 19.7 % when compared to running the applications without switching off the nodes when idle.**

***Index Terms*—HPC; Energy Efficiency; Simulation; Power Reduction**

I. Introduction

Most of the scientific applications which are compute intensive need to be solved using High Performance Computing (HPC) as high performance is delivered using HPC architectures. In HPC, for a long time the assessment of performance was done based on its speed. In Top500 list [1], speed decides the ranking of supercomputers. The past decade has seen the growth of computational power of High performance supercomputers of up to petaflops.

The emerging trends in technology may soon enable the use of exascale computing. As we step into the exascale computing era, HPC systems and technologies have provided an unprecedented level of computational capability for solving traditionally insolvable or extremely challenging social and scientific problems where experiments are impossible, dangerous, or inordinately costly. New technology innovations and breakthroughs facilitated or accelerated by the newest HPC technologies have been widely seen in the scientific fields of aerospace, astrophysics, climate modeling and combustion, fusion energy, nuclear engineering, nanoscience, and computational biology [2] .

As we are moving up from terascale computing to petascale computing and approaching towards exascale computing, power and energy have become critical concerns in HPC applications. Achieving exascale computing requires millions of processors to be working in tandem which causes increased use of computational power to allow faster computing. The amount of power/energy required by these computing systems may not be accessible every time, due to infrastructure unavailability. Also, the cost of running the hardware may outrun the cost of owning the hardware platform [3].

The Green500 list [4], ranks supercomputers based on their energy efficiency. Energy efficiency has increasingly become an important issue in HPC. In HPC, ignoring power consumption as a design constraint while manufacturing, results in a system with high operational costs for power and cooling and can detrimentally impact reliability as the components may ware out very fast, which translates into lost productivity.

Increased power consumption directly produces a substantial amount of operating cost, including electricity bills, expenses for cooling facilities and extra space [5], [6]. For instance, Sunway TaihuLight requires 15.371 MW of power to operate. With a utility rate of $0.10 per KW/hour, the annual electricity bill could reach as high as $13.5 million. This rough estimation doesn't include cooling expenses which can easily cost up to 40% of total system operating expenses [5], [6], [7], [8].

Increasing the scale of HPC systems to achieve better performance has the unwelcome consequence of reduced system reliability due to heat emissions caused by high power consumption. Increased power consumption can result in higher operating temperatures for parallel systems and dramatically reduce overall system reliability and availability. For example, a Google cluster with 450,000 processors has to be rebooted 60 times per day and experiences a 2 to 3 percent annual replacement rate [9].

Power reduction in HPC systems is extremely important but is challenging as it affects every part of a system. The cost of powering HPC systems has been steadily rising with growing performance, while the cost of hardware has remained relatively stable. If this situation continues to exist then energy cost of a large scale system could be more than the equipment itself during its life time [10].

Although a lot of research has been done for power reduc-

Manuscript received June 1,2018; revised July 19,2018.

Maaz Ahmed and Mohsin Khan are Research Scholars at HKBK College of Engineering, Bangalore, India

Waseem Ahmed, Rashid Mehmood, Abdullah Algarni, Aiiad Albeshri, Iyad Katib are with Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, KSA

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

tion in the HPC field, the objectives have not been achieved. On the other hand, research has been carried out in other fields like Embedded Systems and mobile computing where low power components and power reduction algorithms have been effectively used. HPC can benefit from these components and algorithms for energy efficiency. One of the biggest challenges in HPC is energy efficiency. Power and energy consumption has to be minimized to reduce the cost without much change in the performance of the HPC application. To address this issue the Scientific computing community are trying to build computer systems and applications that consume less energy.

There are many challenges which are faced in HPC. One of the biggest challenge in HPC is to reduce power consumption as consumption of power has become a critical concern in HPC because power is a limiter for all computing platform as power is required to achieve performance as well as for cooling the system, especially in HPC cluster because as the number of nodes in a cluster increases the total power consumption also increases. When utilization of the cluster is low, nearly same amount of energy is consumed as when it is fully utilized.

Power consumption can be managed by different power management techniques. Hypothetically, in these low utilization phases cluster hardware can be turned off or switched to a lower power consuming state. We have designed a simulator to estimate power and help designers to experiment execution of different applications on different HPC architectures and evaluate various power optimization techniques. If we carry out the experiments directly on these architectures they take a long time running from few days to several months, hence there will be difficulty in estimating the power and evaluating the different power management techniques and also there may be a possibility that something may go wrong. Hence, we use power simulators for estimating total power consumed and evaluate different ways of managing the power.

The main contribution of our work is as follows:

- We have developed a power simulator to estimate the total power consumed by the cluster for a given time period.
- We propose a method to manage the shutdown of the CPU's when they are in the idle state. This method helps in reducing the power consumption.

The rest of the paper is organized as follows. Section 2 describes the methodology. Section 3 contains the experimental setup used for our research. Section 4 discusses the results obtained and their analysis. Section 5 discusses the related work. Section 6 draws the conclusions and gives insight to the future work.

II. Methodology

A. Design of Power Simulator

We have designed a simulator to simulate applications on different nodes at different times. Block diagram of the power simulator is as shown in Figure 1. It consists of power data, user defined idle period threshold T_i, and config file as inputs, and total power consumed and graphs of power profiles as outputs. Power data of different applications present in Rodinia benchmark suite [11] is collected by averaging few runs of these applications. We have designed a config file as shown in Table III and IV, it is used to launch the application or switch the node to ON state or OFF state at a particular time and a particular node which is also specified in the config file.

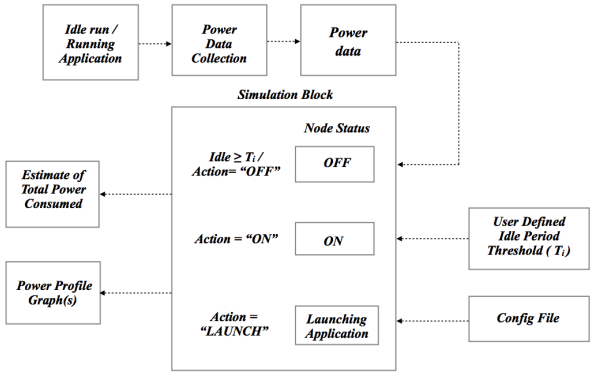

Figure 1. Block Diagram of Power Simulator

B. Power Saving by Shutting down Idle Nodes

In this section we present the proposed method. We describe how to use a selective shutdown technique for power saving. We have designed the simulator to shutdown the node which is idle for user defined idle period threshold T_i seconds. When a node is not doing any job and is in the idle state for more than T_i seconds, it will enter enter the OFF state by shutting it down. The system then stays in the power-saving OFF state until an ON or LAUNCH signal occurs. When the ON or LAUNCH signal occurs, the node will resume the running state.If the node does not enter the OFF state it will remain in the idle state when not busy, now the crucial issue is whether to shutdown the node and if it has to be shutdown when should it be done, so that we can reduce energy consumption. This issue is discussed as follows.

Let R be the running period, T_{off} the period when the node is OFF, T_{lag} is the delay for transition from OFF state to running state . P_R is the power consumption value of the node in the running state and P_{lag} is the power consumption value of the node when node transits from OFF state to running state. Finally, P_{saved} denotes the power saving of the node.

Assuming the node is in the idle state for time equal to or more than t seconds, the node enters OFF state. If the node is in the OFF state for period longer than than the delay to transit from OFF state to ON or LAUNCH state (i.e., $T_{off} \geq T_{lag}$) then shutdown technique will be useful.

Mathematically this can be represented as:

Let S_d represent shutting down of nodes, then

$$(T_{off} \geq T_{lag}) \rightarrow S_d$$

If the node is in the OFF state for period lesser than than the delay to transit from OFF state to ON or LAUNCH state (i.e., $T_{off} \leq T_{lag}$) then it is better to keep the node in the idle state.

Mathematically this can be represented as:

$$(T_{off} \geq T_{lag}) \rightarrow \neg S_d$$

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

where $\neg S_d$ represents running the nodes in idle state (i.e., negation of shutdown)

State Diagram of Shutdown approach is as shown in Figure 2

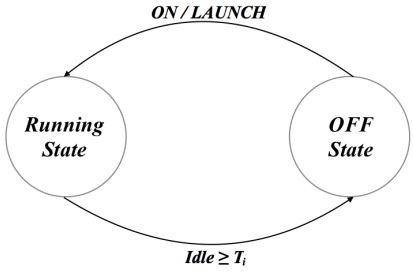

Figure 2. State Diagram of Shutdown approach

III. Experimental Testbed

The experiments were conducted on an Intel core i5-3210M. The i5-3210M is a dual-core processor. The core speed is 2.5 GHz. Each core has its own 32KB L1 cache and 256KB L2 cache. L3 cache is a shared cache of size 3MB. The size of the RAM is 4 GB. Thermal Design Power(TDP) is 35 Watts. The specification are as shown in the Table I.

Table I
System Specifications

Component	Details
System	Intel core i5-3210M
Micro-architecture	Ivy Bridge
Number of Cores	2
Number of Threads	4
Processor Speed	2.5 GHz
Ram type	DDR3
Ram size	4 GB
Memory bandwidth	25.6 GB/s
L1 Data Cache	2 x 32 KB
L1 instruction Cache	2 x 32KB
L2 Cache	2 x 256 KB
L3 Cache	3 MB
Minimum Idle Power	2 Watts
Maximum Idle Power	16 Watts
Thermal Design Power	35 Watt

We use Rodinia Benchmark Suite [11] for our validation. These benchmarks were developed at University of Virginia. It is collection of HPC applications. Rodinia was designed specifically to evaluate the efficiency of heterogeneous multi-core systems, including accelerators. The applications represent compute-intensive domains such as image processing and bioinformatics. The benchmark applications are available in different parallel programming languages such as OpenMP, CUDA and OpenCL. The Rodinia suite contains applications such as Heart Wall, LavaMD, LU Decomposition etc.

IV. Results and Analysis

We carried out our research with applications lud, heartwall and lavaMD present in Rodinia benchmark suite. We assume T_i to be 600 seconds. In Figure 3 we have a node running idle for 7200 seconds and consuming power of 22741 watts. In Figure 4 we have a node running idle for 600 seconds and turning OFF after 600 seconds, consuming power of 1897 watts. If we compare Figure 3 and Figure 4, we see that we have saved around 20844 watts of power when we switch OFF the node when it is idle. Similarly, we tried our experiments using 20 nodes as example, Figure 5 depicts running of 20 idle nodes for 7200 seconds, and Figure 6 depicts running of 20 idle nodes and turning them OFF after 600 seconds. Again, when we compare Figure 5 and Figure 6, we see that we have saved around 416246 watts of power by turning OFF the nodes when idle. Figure 7 and Figure 8 show running of benchmark on a single node and running of benchmark on a single node and turning OFF the node if idle for more than 600 seconds respectively. Comparing the power consumed in Figure 7 and Figure 8 we see that we can save around 7577 watts of power if we switch OFF the node when it is idle. We have also created a config file as shown in Table III and IV, using this file we can generate power profiles on different nodes by changing only the contents of the config file as shown in Figure 9 and Figure 10. The total power consumed by the node after each step is calculated and tabulated as shown in Table II.

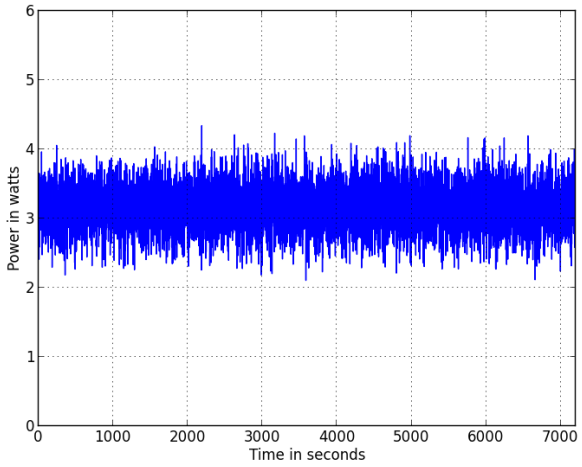

Figure 3. Single node running idle

Table III
Config_file_1

Time	Node	Action	Benchmark
1000	7	ON	
7000	9	ON	
6000	8	ON	
4999	1	ON	
4000	1	LAUNCH	lavaMD
2000	2	LAUNCH	lud
1010	13	ON	
1010	19	ON	
1003	5	ON	
1004	6	ON	
5000	20	LAUNCH	lud20
4000	11	LAUNCH	heartwall

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

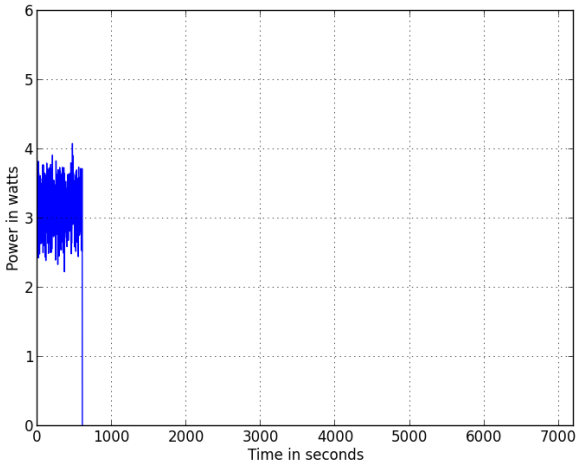

Figure 4. Single node running idle and turning OFF after 600 seconds

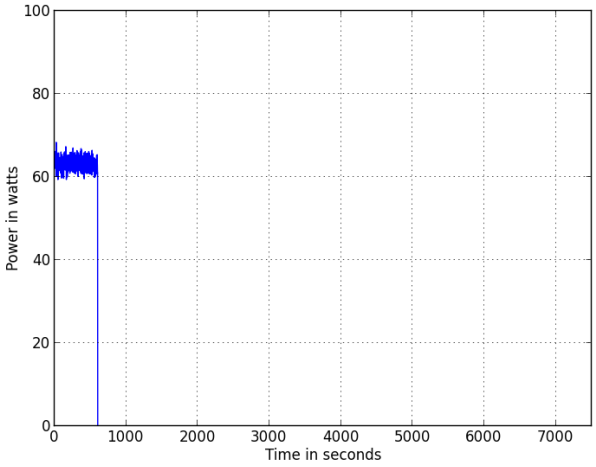

Figure 6. 20 nodes running idle and turning OFF after 600 seconds

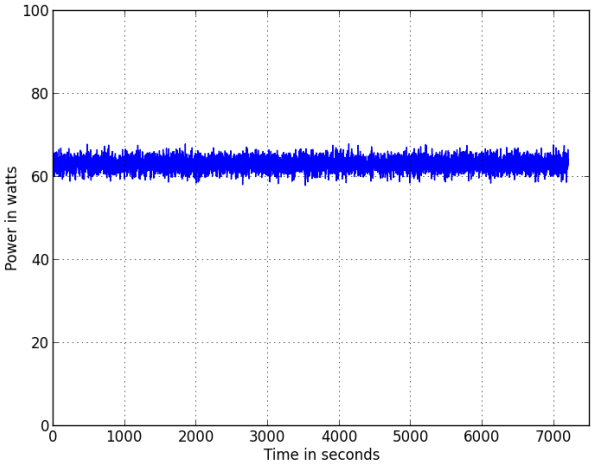

Figure 5. 20 nodes running idle

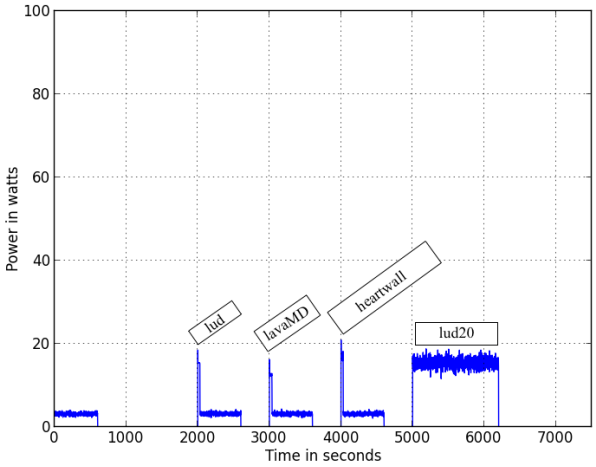

Figure 7. Running benchmark on single node

Table IV
CONFIG_FILE_2

Time	Node	Action	Benchmark
1000	7	ON	
7000	9	ON	
6000	8	ON	
4999	1	ON	
3000	1	LAUNCH	lavaMD
2000	2	LAUNCH	lud
1010	13	ON	
1010	19	ON	
1003	5	ON	
1004	6	ON	
5000	20	LAUNCH	lud20
4000	11	LAUNCH	heartwall

V. RELATED WORK

We have different Simulation approaches which use the power consumption characteristics that are derived from measurement of sample components and embedded into a simulator [12]. power and performance are both found by tracing the execution of applications by using these simulators. The framework in [13] uses simulation for profiling the power consumption of a microprocessor. A simulator for a complete system is presented in [14]. Simulation can be successfully used for memory and disk related operations [15], [16]. We also have some software tools like Joulemeter [17] which estimates the energy usage of an application, virtual machine (VM), a computer and a server. Joulemeter measures the hardware resources such as CPU utilization, disk, memory, screen brightness, etc. It converts the resource usage to actual power usage by using automatically learned realistic power models. The authors of [18] have developed pTop which is a power profiling tool, used at process level, providing information of the energy consumption of the process in real time. In most of the above works we can see that there may be a mismatch between the simulation and the real systems due to the inaccuracy of the simulation models.

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

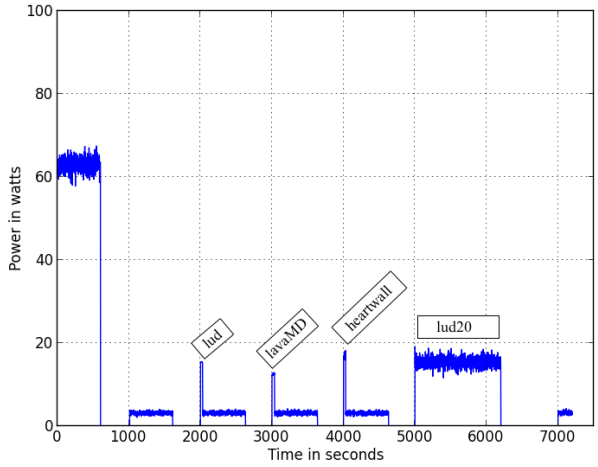

Figure 8. Running benchmark on single node and switching OFF if node idle for equal to or more than 600 seconds

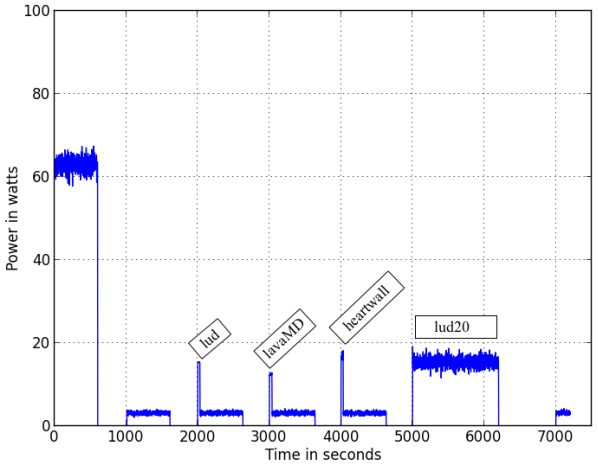

Figure 10. Launching benchmarks based on config_file_2

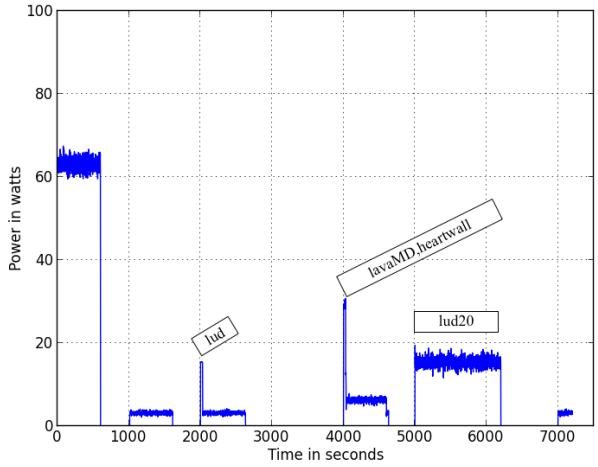

Figure 9. Launching benchmarks based on config_file_1

VI. Conclusions and Future work

One of the biggest challenges in HPC is to reduce power consumption as consumption of power has become a critical concern in HPC because power is a limiter for all computing platform as power is required to achieve performance as well as for cooling the system, especially in HPC cluster because as the number of nodes in a cluster increases the total power consumption also increases. When utilization of the cluster is low, nearly same amount of energy is consumed as when it is fully utilized. Power consumption can be managed by different power management techniques. Hypothetically, in these low utilization phases cluster hardware can be turned off or switched to a lower power consuming state. Our results show that when we run the applications using our proposed method of switching off the nodes when idle for more than T_i seconds we achieve power saving of about 19.7 % when compared to running the applications without switching off the nodes when idle. In future we want to incorporate power management techniques such as DVFS in the simulator for applications present in the benchmark.

References

[1] "The top500 list," *http://www.top500.org (December 2017)*.
[2] T. Agerwala, "Exascale computing: The challenges and opportunities in the next decade," in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*. IEEE, 2010, pp. 1–1.
[3] S. Mittal, "A survey of techniques for improving energy efficiency in embedded computing systems," *International Journal of Computer Aided Engineering and Technology*, vol. 6, no. 4, pp. 440–459, 2014.
[4] "The green500 list," *http://www.green500.com (December 2017)*.
[5] S. Murugesan, "Harnessing green it: Principles and practices," *IT professional*, vol. 10, no. 1, 2008.
[6] O. Sarood and L. V. Kale, "A'cool'load balancer for parallel applications," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011, p. 21.

Table II
Power Savings using Shutdown Technique

		Total Power Consumed (Watts)	Power Saved Using the Shutdown Technique (%)
Single Node Running Idle	Without using Shutdown Technique	22748	91.7
	Using the Shutdown Technique	1889	
20 Nodes Running Idle	Without using Shutdown Technique	454227	91.7
	Using the Shutdown Technique	37796	
Executing Benchmark Applications on Single Node	Without using Shutdown Technique	38545	19.7
	Using the Shutdown Technique	30925	

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[7] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmal, and R. Friedrich, "Smart cooling of data centers," in *Proceedings of IPACK*, vol. 3, 2003, pp. 6–11.
[8] R. Sullivan, "Alternating cold and hot aisles provides more reliable cooling for server farms. 2002," *The Uptime Institute*.
[9] W.-c. Feng and K. Cameron, "The green500 list: Encouraging sustainable supercomputing," *Computer*, vol. 40, no. 12, 2007.
[10] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 13–23.
[11] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*. Ieee, 2009, pp. 44–54.
[12] F. N. Najm, "A survey of power estimation techniques in vlsi circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 446–455, 1994.
[13] D. Brooks, V. Tiwari, and M. Martonosi, *Wattch: a framework for architectural-level power analysis and optimizations*. ACM, 2000, vol. 28, no. 2.
[14] S. Gurumurthi, A. Sivasubramaniam, M. J. Irwin, N. Vijaykrishnan, and M. Kandemir, "Using complete machine simulation for software power estimation: The softwatt approach," in *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*. IEEE, 2002, pp. 141–150.
[15] E. Brockmeyer, B. Durinck, H. Corporaal, and F. Catthoor, "Layer assignment techniques for low energy in multi-layered memory organizations," in *Designing Embedded Processors*. Springer, 2007, pp. 157–190.
[16] E. Pinheiro and R. Bianchini, "Energy conservation techniques for disk array-based servers," in *ACM International Conference on Supercomputing 25th Anniversary Volume*. ACM, 2014, pp. 369–379.
[17] "Joulemeter," *http://research.microsoft.com/en- us/projects/joulemeter/ (March 2016)*.
[18] T. Do, S. Rawshdeh, and W. Shi, "ptop: A process-level power profiling tool," in *Proceedings of the 2nd workshop on power aware computing and systems (HotPower 2009)*, 2009.

ISBN: 978-988-14048-1-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)