International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019

Optimized Multi-threading To Balance Energy and Performance Efficiency

Sri Vidya B, Harini Sriraman, Rukmani P

Abstract*: *Energy efficiency is an important aspect of high performance computing today. Energy is the integration of power over time. Power consumption in a system depends on power consumption of processing units, memory and other peripherals. One of the recent advancements in energy efficiency is through parallel computing. Ideally in a system the number of software threads should be equal to the number of hardware threads. But in real time systems the ideal ratio cannot be always maintained. Moreover the ideal value will change depending on the workload and the dynamic characteristics of the system. In this work, a detailed study to understand the effect of multi-threading on power efficiency is carried out. The results of these benchmark analysis show optimal number of threads for different categories of workload, to achieve a fine balance between energy efficiency and performance. These results of the analysed benchmark applications are stored in secondary disks. When a new application is submitted for execution on the system, around 12 characteristics of the submitted application is compared with the analysed benchmark applications. Analysed benchmark application with the least hamming distance from the submitted application is chosen and its corresponding optimal thread value is read from the storage. This data is communicated to the compiler of the submitted application for improving the balance between energy efficiency and performance. For the experimental analysis the compilers of C and Java are used. The results show an improved power efficiency of up to 30% when optimal numbers of threads are used.

Index Terms*: *Energy Efficiency, Power dissipation, Multi-threading, benchmarks, power efficiency.

I. INTRODUCTION

Power and performance are two factors that has to be balanced for better cost. Power consumed is not always utilized in its entirety. There will always be loss of power that increases energy efficiency. Sustainable systems should reduce energy wastage. This becomes especially vital in high performance computing. High performance computing is referred to aggregate computing power in a way which delivers much high performance than the normal desktop systems. Cooling of high performance server has the major cost share compared to other physical and running costs. The total-liquid-cooled occupies a small footprint and it accepts the high inlet temperatures that enable the elimination of chiller plant. This is safer, low cost of infrastructure also reduce the electricity bill. In 1980, the number of processors used in a supercomputer began to increase. For the better performance, number of processors is used to achieve the better performance. At the same time, a processor may get many processes to execute, so in order to perform all the tasks at the same time, multithreading concept is used. Threads are the number of sub-processes being divided among processor cores by a process. This thread is of two types as software thread and hardware thread. Software thread is the number of thread count, user gives and hardware thread is the number of thread the processor uses for its process. In case of multithreading, the power factor is a major concern. The power values may differ with different number of threads.

A. Power Efficiency & Multi-threads

Power efficiency is the ratio of the output power divided by the input power: $\eta = 100\% \cdot Pout / Pin$. η is the efficiency in percent (%). Pin is the input power consumption. Pout is the output power or actual work. The power is calculated in watts (W).Thread is the sequence of programmed instructions which used to execute multiple processes. There is possible to using number of threads for performing a task in processor. Each and every applications using in the real time are having more number of threads that's help to perform number of tasks at a same time.

B. Energy

Energy is the main thing to operate the electrical devices, in such case computer is an electrical device and it need energy to performing tasks. According to our studies we are referring the energy as power. Our main objective for this literature survey is power management. If the power has been calculated in term of percentage then, if 100 % input power given to computer for performing a task, whole 100% power is not to be used for finishing the task only 75% of power is consumed and 25% of power is wasted. Now we have to manage the power but how can it possible. So we searched for possible ways to reduce the energy consumption then we started learning about the threads which consuming power while executing the processes. In some case more number of threads has consumed more power to perform the task and in some case less number of threads also consumed more power to perform the task. Only some optimal number of threads gives the exact result that which are consuming low power. At finally we found this is the way to reduce the power consumption and now we are going to find out how many optimal numbers of threads can help to reduce power consumption for different applications.

Revised Manuscript Received on 30 March 2019.
***** Correspondence Author
Sri vidya B*, MCA from Vellore Institute of Technology. She has completed her BCA from Madras University with distinction.
Dr. Harini Sriraman B.E from Madras University and M.E from College of Engineering, Guindy, Chennai, India.
Dr. P Rukmani B.Tech IT from Madras University and M.Tech IT from College of Engineering, Guindy, Chennai, India.

II. RELATED WORK

A. Impact of Increased Power Consumption on Sustainability

Energy consumption has become an important area. It is essential to know about the power and energy usage. Therefore to understand this, some of the related works were analyzed as follows.

Table 1. Existing Power Efficiency in Recent Parallel Systems

POWER AND MULTITHREADING	RELATED WORKS	COMMENTS
PERFORMANCE EFFIECIENCY & MULTITHREADING	1)Adaptive energy-efficient scheduling algorithm for parallel tasks on homogeneous clusters, et.al, Wei Liua,b,c,d,f	Study of power and energy usage of various benchmark applications on the Intel Xeon Phi.
	2) NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime. et.al, Basoglu, M	Shows how to utilize the guard band and it is a model to predict the processor power dissipation against NBTI without affecting the processor.
	3) Understanding the Interactions Hardware/Software Parameters on the Energy Consumption of Multi-Threaded Applications, et.al, Thiyagalingam, J.	The impact of parameters on runtime and energy performance in the multithreaded applications.
	4) Scheduled voltage scaling for increasing lifetime in the presence of NBTI, et.al Zhang L.	About a technique called scheduled voltage scaling to make the Integrated Circuit lifetime better in the presence of NBTI.
Power issues and multithreading	1) Speed scaling: An algorithmic perspective, et.al ,Wierman, A	Tells about speed scaling in an algorithmic perspective
	2)Power-aware job scheduling on heterogeneous multicore architectures, et.al Chiesi M	About power aware algorithm scheduling depending on effective distribution of resources to the workloads.
	3)Energy efficient task scheduling on a multi-core platform using real-time energy measurements.et.al, Singh, D	Concerning energy-efficient operating system with multiprocessor task scheduling.
	4)Power and energy implications of the number of threads used on the Intel Xeon Phi, et.al, Lorenzo, O.G	A study on energy and power consumption of parsec and splash-2X benchmarks on Intel Xeon phi processor for different number of threads.
Power efficiency for GPU, multiprocessors and multicore processors	1)Energy efficient scheduling of parallel tasks on multiprocessor computers, et.al, Li, K	Parallel tasks are scheduled on multiprocessor computers with dynamically variable speed and voltage is self-addressed as combinatorial improvement issues.
	2) Power efficient large matrices multiplication by load scheduling on multi-core and gpu platform with cuda, et.al, Ren, D	The energy efficient scheduling on CPU-GPU computers by adjustment of load and multi-threading.

III. PROPOSED WORK

The proposed work is carried out in a single node. The overall view of the proposed system is given in Figure 1.

As shown in Figur 1, benchmark applications of different categories are analyzed for their power effieciency against multiple threads.

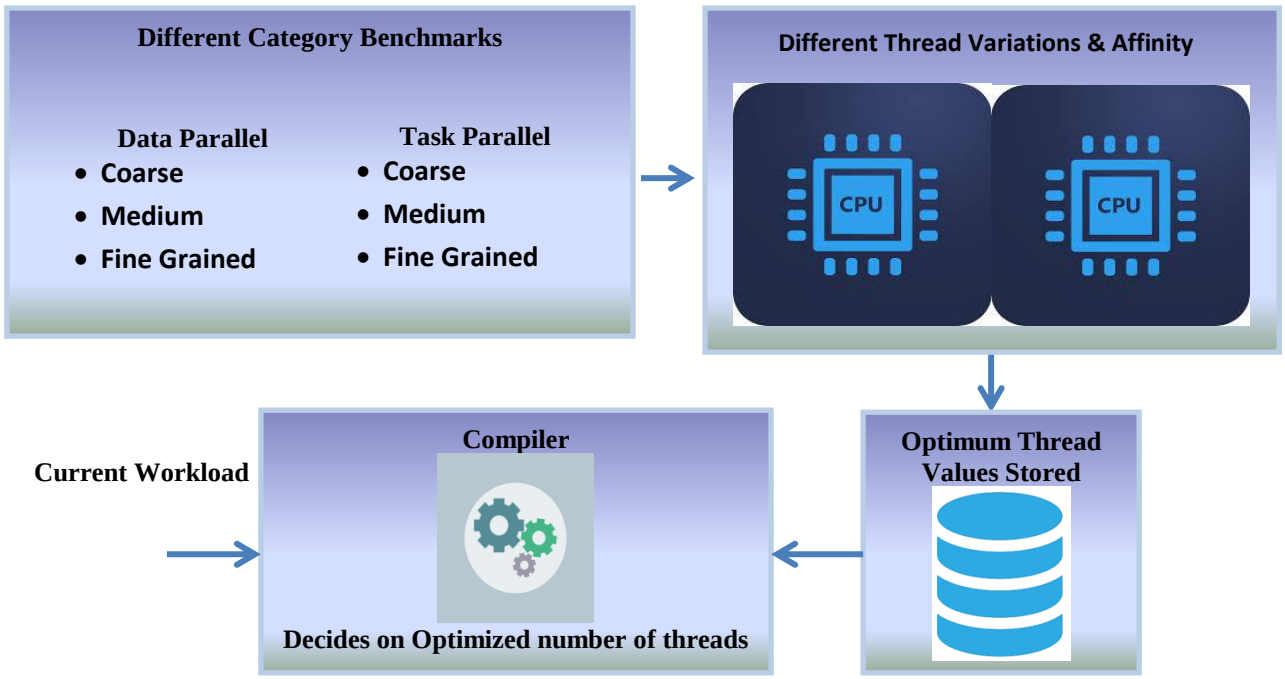

Figure 1. Overall view of the proposed System

The results are stored for future reference. When a new application has to be executed, the application is matched to the nearest benchmark applications and the optimal number of threads is suggested to the compiler.

IV. Experimental Analysis

The proposed environment in which the proposed work is carried out is given below.

A.Hardware Specification

System : 64 bit operating system
Hard disk : 500GB
Monitor : Generic pnp monitor
Ram : 4.00 GB(3.89 usable)
Processor : Multi-core

B. Software Specifications

Operating system : Linux
Tools used : Parsec-3.0,powertop-2.9,perf.

The benchmarks considered and their categories are listed in Table 2. The properties of each category are listed in Table 3. One of the PARSEC benchmark applications is loaded with varied thread count each time and also the power top tool is run simultaneously to note down the power values of each application for each thread value. Once a work load that belongs to one of the category is executed, the compiler can automatically set this as the number of threads for energy efficiency. The hardware runs the application with the optimized thread count and the corresponding power. As said, Power and energy consumption of various PARSEC-3.0benchmark applications were taken for varying thread count. A tool known as PowerTop-2.9 was used to monitor the power values. PARSEC-3.0 (Princeton Application Repository for Shared-Memory Computers) is a benchmark suite composed of multithreaded emerging workloads that is used to evaluate and develop chip-multiprocessors.

Table 2. Categorization of Benchmark Applications

Benchmark Application	Remarks	Category
Black Scholes	An application used in financial computations	Category 1
Body Track	To identify a human body which is marker less	
Facesim	To detect the motions of a human face for the process of visualization	Category 2
Fluid animate	Simulating the physics of fluid motion for animation purposes	
Dedup	To detect and eliminate redundancy in a data stream using a technique known as de-duplication	Category 3
Stream cluster	cluster the data points which is used in data mining	

Power-Top is a Linux tool used to measure power consumption and power management.

Not only about the power values, but also gives values of GPU ops/s, Disk IO/s, Wakeups/s. Perf is a tool used to take valued of some parameters such as memory miss rate, number of context switches.

Table 3. Properties of Categories Considered

Category of Benchmark	Thread level Granularity	Working Set	Data Usage
Category 1	Coarse – medium	Small – Medium	Low to medium sharing
Category 2	Medium – Fine	Large	Low
Category 3	Medium	Medium – Unbounded	High

V. Results and Discussions

This work has been implemented in software level for a single node to know the power consumption of the processor. Firstly, the power dissipation of the node's processor were taken by executing different parallel benchmark applications of PARSEC-3.0 for varied number of threads. As explained in the previous subsection, the applications are compared with the benchmarks with respect to the various key characteristics. The characteristics along with their values for different benchmark applications are given in Table 4 and Table 5.

Table 4. Key Characteristic Values of Benchmarks

Bench mark	Stalled Cycles Front end	IPC	Time elapsed	Branch misses
Black Scholes	50.48%	1.47	9.77082	2.94%
Body Track	66.08%	0.82	12.6034	4.29%
Facesim	44.72%	1.96	4.53169	1.16%
Fluid animate	33.38%	1.79	11.3412	2.65%
Dedup	76.31%	0.48	8.95719	3.83%
Stream cluster	45.89%	1.41	11.2481	3.86%

While running those benchmark applications, the power dissipation of the processors were simultaneously taken by running the tool POWERTOP-2.9. This gave the power values of the processor. Those power values were like, till a certain point, the power values were decreasing. But after a certain point, the power values started increasing with increase in number of threads. The Power Values for benchmark applications are given in Figures 2, 3 and 4. The power values for these benchmark applications as said are different. It is like for each application, it is not ideal to use the same number of software threads as the number of hardware threads. Instead this is getting varied depending on the workload and its dynamic characteristics.

Table 5. Key Features with respect to Multi-Threading

Benchmark	Context Switches	Wakeup/sec
Black Scholes	1.053822	63.68611111
Body Track	3.483543529	146.6823529
Facesim	3.147660588	117.1470588
Fluid animate	2.163225882	76.51176471
Dedup	4.60717	192.8941176
Stream cluster	6.1234	206.3333

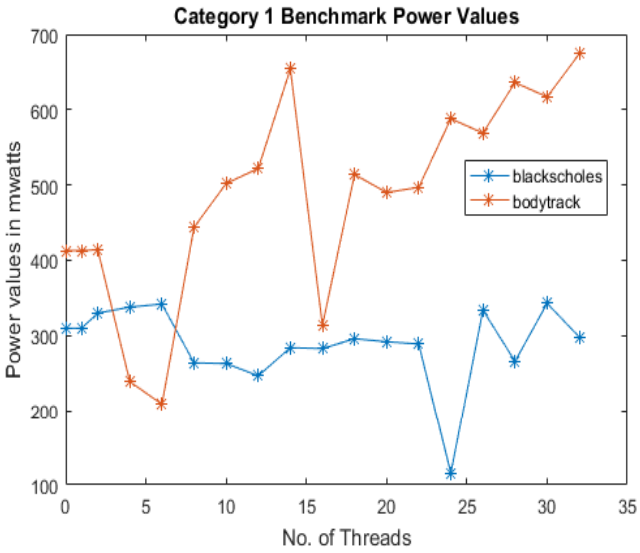

Figure 2. Category 1 Power Versus No. of Threads

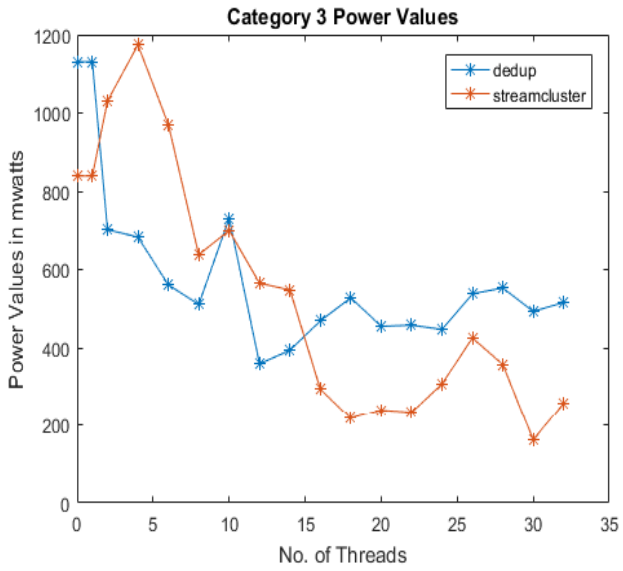

Figure 3. Category 2 Power Versus No. of Threads

This work has been implemented in software level for a single node to know the power consumption of the processor. Firstly, the power dissipation of the node's processor were taken by executing different parallel benchmark applications of PARSEC-3.0 for varied number of threads.

while running those benchmark applications, the power dissipation of the processors were simultaneously taken by running the tool POWERTOP-2.9.

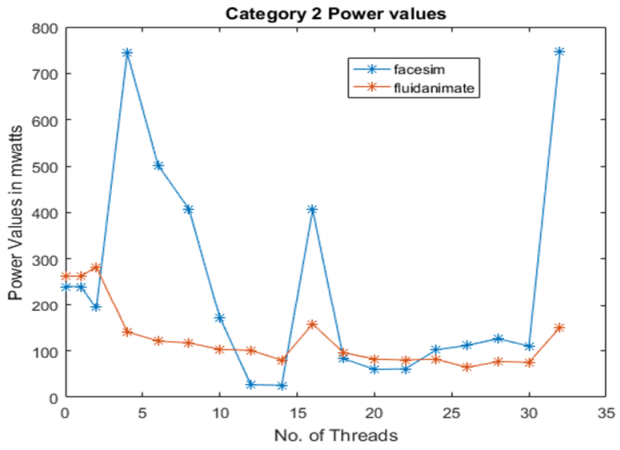

Figure 4. Category 3 Power Versus No. of Threads

This gave the power values of the processor. Those power values were like, till a certain point, the power values were decreasing. But after a certain point, the power values started increasing with increase in number of threads.

A.Balancing Thread Cost and Performance

To infer the number of optimal threads for each category of workload, an analysis of the results obtained was carried out. The analytical analysis of the results are shown in the below sections. From the analysis, we infer thread cost is a function of three main parameters. They are given as below in the equations 1 to 5. The proposed system will help in reducing the power consumption of the processors significantly. So, if the power dissipation of the processor gets reduced, the lifetime of the processor will be considerably increased and the overall lifetime of the system will be increased. Therefore, the system will not be easily thrown as E-waste. The system will be used by the user for some more time thus leading to reduction in E-waste. In this way, the proposed systems reduce power consumption.

$$\textit{Thread Cost Function} \propto \text{Base cost of the thread, No. of Context Switch, Memory Miss rate} \quad (1)$$

$$\text{Thread cost} = \sum \text{Base Cost of Thread, Thread Context Switch Cost, Thread Memory Miss Cost} \quad (2)$$

$$\text{Base Cost of Thread} = \text{Power Consumed by Idle Thread} \quad (3)$$

$$\text{Thread Context Switch cost} = \frac{\text{No. of Context Switches} * \text{Power Consumed by one context Switch}}{\text{Performance improvement due to context switch}} \quad (4)$$

$$\text{Thread Memory Miss Cost} = \text{No. of Misses} * \text{Power Consumed by one memory}_{\frac{\text{miss}}{\text{Performance improvement due to Memory Miss}}} \quad (5)$$

VI. CONCLUSION

In this work, a detailed study has been done to understand the effect of multi-threading on power efficiency .The results shows the optimal number of threads that balance power and performance of different varieties of benchmark applications. The benchmark applications considered for this analysis falls into three categories. Once the optimal thread for each category is identified it is stored in the secondary disk for future use. When an application is submitted for execution, the characteristics of the three categories are compared with the application's characteristics. The parallel scheduling will happen by creating the optimal number of threads for that application. This is done by setting the corresponding environment variable in the compiler. With this proposed technique, a power-aware scheduling algorithm is implemented. The proposed technique was able to save 30% power without any compromise in the performance. In future, the variety of benchmark pool can be increased for power aware scheduling.

REFERENCES

1. Widmer, R., Oswald-Krapf, H., Sinha-Khetriwal, D., Schnellmann, M. and Böni, H., 2005. Global perspectives on e-waste. Environmental impact assessment review, 25(5), pp.436-458.
2. Plambeck, E. and Wang, Q., 2009. Effects of e-waste regulation on new product introduction. Management Science, 55(3), pp.333-347.
3. Saha, B., 2014. Green computing. International Journal of Computer Trends and Technology (IJCTT), 14(2), pp.46-50.
4. Murugesan, S., 2008. Harnessing green IT: Principles and practices. IT professional, 10(1).
5. Debnath, B., Roychoudhuri, R. and Ghosh, S.K., 2016. E-Waste Management–A Potential Route to Green Computing. Procedia Environmental Sciences, 35, pp.669-675.
6. Adaptive energy-efficient scheduling algorithm for parallel tasks on homogeneous clusters
7. Basoglu, M., Orshansky, M. and Erez, M., 2010, August. NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime. In Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design (pp. 253-258). ACM.
8. Chiesi, M., Vanzolini, L., Mucci, C., Scarselli, E.F. and Guerrieri, R., 2015. Power-aware job scheduling on heterogeneous multicore architectures. IEEE Transactions on Parallel & Distributed Systems, (1), pp.1-1.

9. Wierman, A., Andrew, L.L. and Lin, M., 2012. Speed scaling: An algorithmic perspective. Handbook of Energy-Aware and Green Computing-Two Volume Set, pp.385-405.
10. Ren, D. and Suda, R., 2009, August. Power efficient large matrices multiplication by load scheduling on multi-core and gpu platform with cuda. In Computational Science and Engineering, 2009. CSE'09. International Conference on (Vol. 1, pp. 424-429). IEEE
11. Zhang, L. and Dick, R.P., 2009, January. Scheduled voltage scaling for increasing lifetime in the presence of NBTI. In Proceedings of the 2009 Asia and South Pacific Design Automation Conference (pp. 492-497). IEEE Press
12. Singh, D. and Kaiser, W.J., 2014, August. Energy efficient task scheduling on a multi-core platform using real-time energy measurements. In Proceedings of the 2014 international symposium on Low power electronics and design (pp. 271-274). ACM
13. Lorenzo, O.G., Pena, T.F., Domínguez, J.C.C., Campos, J.C.P., Rivera, F.F. and Nikolopoulos, D.S., 2015. Power and energy implications of the number of threads used on the Intel Xeon Phi. Annals of Multicore and GPU Programming: AMGP, 2(1), pp.55-65.
14. Li, K., 2012. Energy efficient scheduling of parallel tasks on multiprocessor computers. *The Journal of Supercomputing*, *60*(2), pp.223-247.
15. Ren, D. and Suda, R., 2009, August. Power efficient large matrices multiplication by load scheduling on multi-core and gpu platform with cuda. In Computational Science and Engineering, 2009. CSE'09. International Conference on (Vol. 1, pp. 424-429). IEEE
16. Thiyagalingam, J. and Trefethen, A.E., 2014. Understanding the Interactions Hardware/Software Parameters on the Energy Consumption of Multi-Threaded Applications

AUTHORS PROFILE

B Srividya is currently pursuing her MCA from Vellore Institute of Technology. She has completed her BCA from Madras University with distinction. Her research interest includes Energy Efficiency, Parallel and distributed computing and PHP programming.

Dr. Harini Sriraman completed her B.E from Madras University and M.E from College of Engineering, Guindy, Chennai, India. She has completed her PhD in VIT University in the area of fault tolerant computer systems. She has published articles in reputed international peer reviewed conferences and journals. She is currently working with Vellore Institute of Technology, Chennai, as Assistant Professor(senior) in the School of Computing Science and Engineering. Her teaching interests include Sustainable Computer Architectures for the Dark Silicon Era, High performance Computing accelerated by FPGAs and GPUs and Industry 4.0

Dr. P Rukmani completed her B.Tech IT from Madras University and M.Tech IT from College of Engineering, Guindy, Chennai, India. She has completed her PhD in VIT University published articles in reputed international peer reviewed conferences and journals. She is currently working with Vellore Institute of Technology, Chennai, as Assistant Professor (senior) in the School of Computing Science and Engineering. Her teaching interests include wireless computer networks, Queuing Theory, Cyber Physical Systems (CPS) and Security in CPS.