POWER MANAGEMENT FOR DESKTOP COMPUTER: A REVIEW

Ria Candrawati[1], Nor Laily Hashim[2], and Massudi Mahmuddin[3]
[1,2,3]*Universiti Utara Malaysia, Malaysia, riacandrawati@yahoo.com, laily@uum.edu.my, ady@uum.edu.my*

ABSTRACT. Computer manufacturers have offered various power saving functions to reduce the power consumption of their products. Yet, this function is not sufficiently used because it reduces user convenience in computer usage. Literature on this topic is fragmented: there are no categories in the literature to characterize this power management for desktop (PMD. To address this gap, this paper provides a review of existing studies on this domain focusing on hardware components.

Keywords: Power Management, Desktop Computer, Power Consumption, Autonomic Computing

INTRODUCTION

World population has increased as well as computer users and consequently, the power that was used has become one of the contributors to today's climate change (Candrawati, Hashim, & Mahmuddin, 2010). The needs to improve power management have become essential due to increment in power and bills when the power state needs to make a transition from active to idle and computer users need to activate the state in a short idle period (Irani, Shukla, & Gupta, 2003). Much research in computer components and techniques has invented assessment tools software to monitor and estimate the power consumption that is consumed by computers.

Joulemeter is a mechanism for Virtual Machines power metering where it shows the ability of a virtualized platform to use several power management mechanisms (power provisioning and power tracking) that have been proved beneficial on the physical servers (Kansal, Zhao, & Bhattacharya, 2010). However, this mechanism does not require any additional instrumentation of application workloads or operating systems within the virtual machines. This tool estimates the largest dynamic power consuming resources in a computer server which is composed of processor, memory and disk. Furthermore, this tool is open source software that monitors those components usage and needs an external power metering to measure power more accurately. Yet this software does not have an intelligent control feature to amend the power management in order to reduce the power consumption of computer.

On the other hand, Khargharia et al., (2008) proposed the use of an autonomous control to optimize power and performance. In their study, the autonomous control approach is applied in a domain of high performance distributed computing like e-business data centres, where an autonomic manager maintains changing circumstances impacted by the internal or external environment. This autonomic manager is consists of 4 main phases which are monitoring, analysis, planning and execution phases. Therefore, it can be concluded that the autonomic manager facilitates in improving the power management of computing system, and it must have four complete phases of autonomic manager.

Based on the literature review on this domain, the following research gaps have been identified. The improvement of power management technologies has become important as users want while the power management remain active at the background, they need a computer that still has a high performance with safe, reliable operations and low operating costs (Rajamani, Lefurgy, & Ghiasi, 2008). Joulemeter estimates the power usage without giving meaningful visual information to layman users. On the other hand, it needs the external power metering tools to operate and yet it just provides monitoring power function. EnePal PC Pack estimates power, monitor and learn user behaviours power usage patterns as the only phase for autonomous control. However, it is bundled together with a specific computer product. The existing autonomous control or autonomic manager has been applied in power management system at data centres and not on individual computer desktops.

The paper is organized as follows. The power management reviews are introduced in section 2. Section 3 shows a proposed method for this study. Finally, section 4 presents the conclusion and contribution of this study.

POWER MANAGEMENT CATEGORY

In general the studies on power management can be categorised into two which are system and component level. The following are the reviews on the existing studies of both categories.

Power Management for System Level

Power management is a prediction problem; it seeks to forecast whether an idle period will be long enough to compensate for the overhead of power state changes (Lu & Micheli, 2001). There are two areas of power management which are to improve technologies to be more power efficient and to increase people's use of power management options on computers (Chetty et al., 2009). Over the years, many approaches have been proposed in order to contribute to this area. Lorch & Smith (1997) suggested several heuristic techniques to decrease power use, such as never running a process that is still blocked and waiting on an event, delaying processes that execute without producing output or otherwise signalling useful activity, and delaying the frequency of periodic processes, most of which seem to wake up, look around, find nothing interesting to do, and go back to sleep. Then, Gupta & Singh (2003) have implemented Dynamic Power Management Algorithm that can optimize the battery life and at the same time can improve the resource usage without degrading the device performance. Furthermore, this study is executed on computers and routers. Also, Vahdat, Lebeck, & Ellis (2000) proposed a systematic re-examination of all aspects of operating system design and implementation from the point of view of power efficiency rather than the more traditional operating system metric of maximizing performance. Different from other researchers, Moshnyaga (2010) proposed a new application of power management by applying eye-tracking technology for PC power management. Nowadays, existing power management technologies used a PC user sense sensor through keyboard, touchpad and/or mouse, but Moshnyaga (2010) used watches sensor to his new technology through a single camera. More precisely, it tracks the users eyes to detect whether he or she looks at screen or not and based on that changes the display brightness and power consumption.

At system level, power management techniques are divided into two type of techniques, which are static techniques and dynamic. Static techniques called Static Power Management (SPM) techniques are applied at design time (off-line) and targeting different levels of hardware and software. In contrast, dynamic techniques called Dynamic Power Management (DPM) techniques are applied at runtime and these techniques are used to reduce power when systems are serving light workloads or idle (Chedid & Yu, 2002). Moreover, SPM techniques are divided into two target areas; first targeting the CPU and investigate the power consumption of the cycle and instruction levels; second approach is a high level approach targeting different or all system components. However, DPM techniques are applied in three different areas; first applied at the CPU level, using DVS which allows a processor to dynamically changes speed and voltage at run time, and it is saving the power by spreading run cycles into idle time; second area is targeting the system level which considers all of the system components such as memory, hard disk, input and output devices, display and others. As a final point, DPM is used on multiple systems like a server cluster, where is more than one system collaborates to save overall power (Chedid & Yu, 2002).

Furthermore, a power management policy is a procedure that takes decisions upon the state of operation of system components and on the state of the system itself (Benini, Bogliolo, Paleologo, & Micheli, 2000), or a standard that needs to decide when to perform operation state transition and which transition should be performed (Jiang, Xi, & Yin, 2010). Power management policies can be classified into three categories based on the methods to predict whether a device can sleep long enough. These categories are time-out, predictive, and stochastic (Lu & Micheli, 2001). First, time-out policy is used to turns the system components to lower power state whenever the time last at idle state reaches an assigned timeout value. Second, predictive policy will predict the duration of future idle periods by examining the past history, and turns the system components to lower power state whenever it becomes idle if the next idle is predicted longer enough. Lastly, stochastic policy is a policy that models the arrival of requests and device power-state changes as stochastic processes, such as Markov processes (Jiang et al., 2010).

A comparative analysis was conducted on seven (7) algorithms that were reviewed and executed by previous researchers. A brief description and limitation of each algorithm were tabled and shown in Table 1 below.

Table 1. Comparative Analysis of Power management Algorithm(system level)

Algorithm	Definition	Limitation
Timeout	This algorithms wait for Time Before Shutdown (*Tbs*)=Timeout Value (*T*), we can set for the timeout value, such as 1 minute or thirty second or etc (Y. Lu et al., 2000)	This algorithm wastes an amount of power for waiting the idle time to reach the idle value and the timeout to expire (Benini, Bogliolo, & Micheli, 2000).
L-Shape	This algorithm is performed when the busy period is short enough and the idle period is long. The device should shut down after short busy period (Lu et al., 2000).	This algorithm will keep a hard disk in the working state during idle period. This algorithm is not sufficient to determine the length of an idle period when a large group of short idle periods follow short busy periods enclosed by the circle (Lu et al., 2000).
Stochastic Algorithm	In this algorithm, time is divided into small intervals of length *L*. It is assumed that the system can only change its state at the beginning of a time interval. During an interval (jL, $(j+1)L$), the transition probability of the system depends only on the state of the system at time jL (hence, the Markovian property) and the command issued by the power manager (Qiu & Pedram, 1999).	This algorithm assumes the characteristic probability distribution of input or job arrivals and the service time distribution of the device (Irani et al., 2003).
Competitive Algorithm	This algorithm compares the performance of online algorithm with an optimal offline algorithm (Karlin, Manasse, Mcgeochj, & Owicki, 1994).	This algorithm consumes at most twice the minimum power consumed by an offline algorithm (Lu et al., 2000).

Learning Tree	This algorithm predicts the value of next idle period based on the sequence of recent idle period length observed (Chung, Benini, & Micheli, 1999).	This algorithm do not offer any guarantee on optimality and does not take performance overhead into account (Dhiman & Rosing, 2006).
Adaptive Algorithm	This algorithm will set the system automatically detects the idle periods based on past idle history. The idle period will change dynamically and then automatically disable the system. This algorithm can perform for both hardware and software (Ramanathan, Irani, & Gupta, 2002).	This algorithm is easily accomplished and reported accuracy is high when prediction only for single parameter (Benini et al., 2000).
Non-Adaptive Algorithm	This algorithm will set the idle periods statically determined. This algorithm also can be used for hardware and software system (Ramanathan et al., 2002).	This algorithm can guarantee the result from hardware performance, but there is no guarantee for software system (Ramanathan et al., 2002)

Power Management at Component level

According to Bray (2006), the power consumption of computer is determined by the amount of power that is required to operate and how they are used. This consumption is influenced by two main factors; (i) Power draw, the power required to run devices and (ii) Usage Pattern, how and when the device is used.

As well as hardware, power management inside the operating system also has several conditions that contribute to power consumption calculation process. Foster & Calwell (2003) classified two main conditions: on mode and low power (monitors sleep, hardware sleep, and off). On mode is when the computer is performing only minimal computing, with or without user inputs. Low power consists of three modes. First, monitor sleep is when the computer is on and the screen is powered down. Second, hardware sleep happens either by choosing "Standby" option from the Windows Start menu, or by enabling an automatic timer in the Power Management software that sends the computer into hardware sleep mode after the user has not been typing or otherwise using the computer for some period of time. Last, off mode happen when the computer is switched off by manually pushing the power switch or by choosing "Shut Down" from the Windows start menu.

Bray (2006), on the other hand, adopted the terms 'active', 'low power', and 'off' to describe different power states of computers and monitors. These terms are used by a number of the studies. 'Active' refers to when the computer is turned on ready for use, but not necessarily being used. 'Low power' encompasses commonly used terms such as 'sleep' or 'suspend', where a device has multiple low power settings, 'low power' refers to the lowest of these settings. 'Off' describes a device that is turned off but still connected to main power. Bray (2006) classified the power draw result based on computer and monitor. Power draw is the power that is required to running the device.

COMPARATIVE ANALYSIS OF DESKTOP COMPONENTS POWER CONSUMPTION

Several power consumption systems as mentioned in Table 2 have different approaches and focus on different monitored components. The analysis in table below shows the six (6) PC components: Processor, Graphic Card, Memory, HDD, Monitor and Power Supply.

Table 2. Power management systems and computer hardware

Power Management System	Description	PC Components					
		1	2	3	4	5	6
Power Consumption Monitoring System (Hirao et al., 2005)	This system developed a method of monitoring power consumption by analysing PC's operating states. This system constantly monitors the power consumed by specific PC components and calculates the electric energy consumed.	√	√	√	√	√	√
JouleTrack (Sinha & Chandrakasan, 2001)	This system is a software energy profiling which is a web based tool that characterize and estimate a processor.	√	-	-	-	-	-
Joulemeter (Kansal et al., 2010)	Joulemeter is a software tool that can estimate the computer power consumption by tracing the computer resources used like screen brightness, CPU utilization and memory, then estimates the power usage of the computer	√	-	√	√	-	-
VMeter (Bohra & Chaudhary, 2010)	This system is a power consumption model that profile the resource and power consumed by an individual VM based on the monitored data obtained by hardware performance counters and a disk monitoring utility program	√	-	√	√	-	-

1. Processor; 2. Graphic Card; 3. Memory; 4. HDD; 5. Monitor; 6. Power Supply

In examining the existing power management systems, some were found to use more components than others, but overall all those system is to estimate the power usage by monitoring the computer resources used. However, none of these systems measure all components that consume much energy as mentioned by Higgs (2007) and Moshnyaga (2010).

CONCLUSION

This paper has provided a review on current research conducted in this domain, where the research gaps and trends in power management for system and component levels have been discusses. After all the previous researchers' effort to raise the awareness of power consumption is less applied in daily activity based on those limitations. Therefore, this research proposed an enhanced model of power management that operates as open source power management software that has similar features of EnePal PC Pack such estimates power usage, visualizes the total usage for previous day usage, monthly usage and yearly usage in Watt and Ringgit Malaysia. In addition to that, this model is able to handle extended components of computer hardware that contribute to computer power consumptions and also have its own autonomic manager features. This does not only monitor and learn behaviours of user patterns like Joulemeter and EnePal PC Pack but, it also has a complete phase of autonomic system architecture which are monitoring phase, analysis phase, planning phase and execution phase has been mentioned in (Khargharia et al., 2008).

REFERENCES

Benini, L., Bogliolo, A., & Micheli, G. De. (2000). A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *8*(3), 299–316. doi:10.1109/92.845896

Bohra, A. E. H., & Chaudhary, V. (2010). VMeter: Power modeling for virtualized clouds. *2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, 1–8. doi:10.1109/IPDPSW.2010.5470907

Bray, M. (2006). *Review of Computer Energy Consumption and Potential Savings* (pp. 1–35). Retrieved from http://www.dssw.co.uk

Candrawati, R., Hashim, N. L., & Mahmuddin, M. (2010). A Proposed Method Optimizing Energy Usage for Software Process. *Annual International Conference on Green Information Technology (Green IT 2010)* (Vol. 10, pp. 130–132). doi:10.5176/978-981-08-7240-3 G-48 130

Chedid, W., & Yu, C. (2002). *Survey on power management techniques for energy efficient computer systems. Mobile Computing Research Lab* (pp. 1–18). Retrieved from http://137.148.49.106/engineering/ece/research/techReport/2002/tr01.pdf

Chetty, M., Brush, A. J. B., Meyers, B. R., & Johns, P. (2009). It ' s Not Easy Being Green : Understanding Home Computer Power Management. *CHI 2009* (pp. 1033–1042). Boston, Massachusetts, USA. doi:978-1-60558-246-7/09/04

Chung, E.-Y., Benini, L., & Micheli, G. De. (1999). Dynamic Power Management Using Adaptive Learning Tree (pp. 274–279). IEEE.

Dhiman, G., & Rosing, T. (2006). Dynamic Power Management Using Machine Learning. *2006 IEEE/ACM International Conference on Computer Aided Design* (pp. 747–754). Ieee. doi:10.1109/ICCAD.2006.320115

Foster, S., & Calwell, C. (2003). *Laptop Computers: How much energy do they use and how much can we save?* (pp. 1–64). US.

Gupta, M., & Singh, S. (2003). Greening of the internet. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03* (p. 19). New York, New York, USA: ACM Press. doi:10.1145/863956.863959

Higgs, T. (2007). Energy Efficient Computing. *IEEE*, 210–215.

Hirao, E., Miyamoto, S., Hasegawa, M., & Harada, H. (2005). Power Consumption Monitoring System for Personal Computers by Analyzing Their Operating States. *Environmentally Conscious Design and Inverse Manufacturing, 2005. Eco Design 2005. Fourth International Symposium* (pp. 268–272). doi:10.1109/ECODIM.2005.1619220

Irani, S., Shukla, S., & Gupta, R. (2003). Online strategies for dynamic power management in systems with multiple power-saving states. *ACM Transactions on Embedded Computing Systems*, *2*(3), 325–346. doi:10.1145/860176.860180

Jiang, Q., Xi, H.-S., & Yin, B.-Q. (2010). Adaptive optimisation of timeout policy for dynamic power management based on semi-Markov control processes. *IET Control Theory & Applications*, *4*(10), 1945. doi:10.1049/iet-cta.2009.0467

Kansal, A., Zhao, F., & Bhattacharya, A. A. (2010). Virtual Machine Power Metering and Provisioning. *SoCC'10* (pp. 1–12). Indianapolis, Indiana, USA. doi:978-1-4503-0036-0/10/06

Karlin, A. R., Manasse, M. S., Mcgeochj, L. A., & Owicki, S. (1994). Competitive Randomized Problems Algorithms for and Results. *Algorithmica*, *11*(6), 542–571.

Khargharia, B., Hariri, S., & Yousif, M. S. (2008). Autonomic power and performance management for computing systems. *Cluster Computing*, *11*(2), 167–181. doi:10.1007/s10586-007-0043-6

Lorch, J. R., & Smith, A. J. (1997). Scheduling techniques for reducing processor energy use in MacOS. *Wireless Networks*, *3*, 311–324.

Lu, Y., Chung, E., Simuni, T., Benini, L., Micheli, G. De, & Informatica, D. (2000). Quantitative Comparison of Power Management Algorithms. *Design, Automation and Test in Europe Conference and Exhibition 2000* (pp. 20–26). IEEE.

Lu, Y.-H., & Micheli, G. De. (2001). Comparing system level power management policies. *IEEE Design & Test of Computers*, *18*(2), 10–19. doi:10.1109/54.914592

Moshnyaga, V. G. (2010). The use of eye tracking for PC energy management. *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications - ETRA '10*, 113. doi:10.1145/1743666.1743694

Qiu, Q., & Pedram, M. (1999). Dynamic Power Management Based on Continuous-Time Markov Decision Processes *. *The 36th annual ACM/IEEE Design Automation Conference* (pp. 555–561). ACM.

Rajamani, K., Lefurgy, C., & Ghiasi, S. (2008). Power management for computer systems and datacenters. *Proceedings of the 13th International Symposium on Low-Power Electronics and Design*. Retrieved from http://www.islped.org/X2008/Rajamani.pdf

Ramanathan, D., Irani, S., & Gupta, R. K. (2002). An analysis of system level power management algorithms and their effects on latency. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *21*(3), 291–305. doi:10.1109/43.986423

Sinha, A., & Chandrakasan, A. P. (2001). JouleTrack - A Web Based Tool for Software Energy Profiling. *DAC 2001* (pp. 220–225).

Vahdat, A., Lebeck, A., & Ellis, C. (2000). Every joule is precious: The case for revisiting operating system design for energy efficiency. *9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system* (pp. 31–36). New York, NY, USA: ACM. doi:10.1145/566726.566735