(IJAER) 2013, Vol. No. 5, Issue No. II, February ISSN: 2231-5152

SURVEY ON ENERGY-EFFICIENT CLOUD COMPUTING SYSTEMS

***Mohammad Toufiq Ahmed, **AbrarHussain**

**Lecturer, Department of EEE,*
East Delta University, Chittagong, Bangladesh
***Lecturer, Department of EEE,*
Southern University, Chittagong, Bangladesh

ABSTRACT

The goal of the computer system design has been shifted to power and energy efficiency. To identify open challenges in the area and facilitate future advancements, it is essential to synthesize and classify the research on power and energy-efficient design conducted to date. In this study, we discuss causes and problems of high power/energy consumption, and present the taxonomy of energy-efficient design of computing systems covering the hardware, operating system, virtualization, and data center levels. We study various key works in the area and map them onto our taxonomy to guide future design and development efforts. This paper concludes with a discussion on advancements identified in energy-efficient computing and our vision for future research directions.

***Keywords**-energy-efficient computing, taxonomy*

INTRODUCTION

The primary focus of designers of computing systems and the industry has been on the improvement of the system performance. According to this objective, the performance has been steadily growing driven by more efficient system design and increasing density of the components described by Moore's law [1]. Although the performance per watt ratio has been constantly rising, the total power drawn by computing systems is hardly decreasing. Oppositely, it has been increasing every year that can be illustrated by the estimated average power use across three classes of servers presented in Table I [2]. If this trend continues, the cost of the energy consumed by a server during its lifetime will exceed the hardware cost [3]. The problem is even worse for large-scale compute infrastructures, such as clusters and data centers. It was estimated that in 2006 IT infrastructures in the United States consumed about 61 billion kWh for the total electricity cost about 4.5 billion dollars [4]. The estimated energy consumption is more than double from what was consumed by IT in 2000. Moreover, under current efficiency trends, the energy consumption tends to double again by 2011, resulting in 7.4 billion dollars annually. Energy consumption is not only determined by hardware efficiency, but it is also dependent on the resource management system deployed on the infrastructure and the efficiency of applications running in the system. This interconnection of the energy consumption and different levels of computing systems can be seen in Fig. 1. Energy efficiency impacts end-users in terms of resource usage costs, which are typically determined by the total cost of ownership (TCO) incurred by a resource provider. Higher power consumption results not only in boosted electricity bills but also in additional requirements to a cooling system and power delivery infrastructure, that is, uninterruptible power supplies (UPS), power distribution units (PDU), and so on. With the growth of computer components density, the cooling problem becomes crucial, as more heat has to be dissipated for a square meter. Apart from the overwhelming operating costs and the total cost of acquisition (TCA), another rising concern is the environmental impact in terms of carbon dioxide (CO2) emissions caused by high energy consumption. Therefore, the reduction of power and energy consumption has become a first-order objective in the design of modern computing systems.

Table I

Estimated Average Power Consumption per Server Class (w/u) from 2000 to 2006 [2]

Server Class	2000	2001	2002	2003	2004	2005	2006
Volume	186	193	200	207	213	219	225
Mid-range	424	457	491	524	574	625	675
High-end	5534	5832	6130	6428	6973	7651	8163

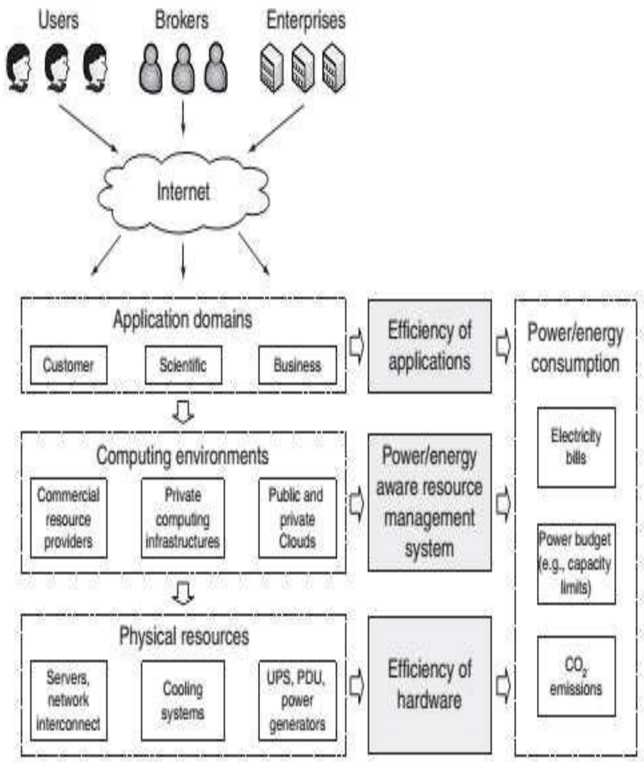

Figure 1: Energy Conversion in different levels in computing systems

PROBLEMS OF HIGH POWER & ENERGY CONSUMPTION

A. *High powerconsumption*

The main reason of the power inefficiency in data centers is low average utilization of the resources. To show this, we have analyzed the data provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab. We have used the data of the CPU utilization by more than a thousand servers located at more than 500 places around the world. The data have been collected in 5 minute intervals during the period from 10 to 19 May 2010. The distributions of the data over the mentioned 10 days along with the characteristics of the distribution are presented in Fig. 2. The data confirm the observation made by Barroso and Holzle [9]: the average CPU utilization is below 50%. The mean value of the CPU utilization is 36.44% with 95% confidence interval from 36.40% to 36.47%. The main run-time reasons of underutilization in data centers are variability of the workload and statistical effects. Modern service applications cannot be kept on fully utilized servers, as even non-significant workload fluctuation will lead to performance degradation and failing to provide the expected quality of service (QoS). However, servers in a non-virtualized data center are unlikely to be completelyidlebecauseofbackgroundtasks(e.g.,incrementalbackups),ordistributeddata bases or file systems. Data distribution helps to tackle load-balancing problem and improves fault tolerance. Furthermore, despite the fact that the resources have to be provisioned to handle theoretical peak loads, it is very unlikely that all the servers of a large-scale data centers will be fully utilizedsimultaneously.

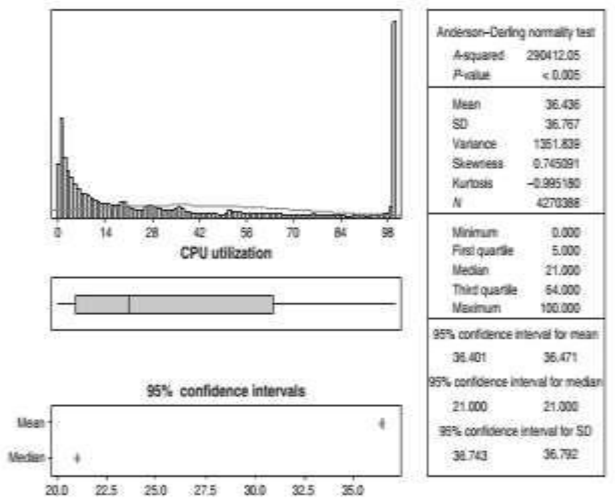

Figure 2: The CPU utilization of more than 1000 PlanetLab nodes over a period of 10 days

Another problem of high power consumption and increasing density of server's components (i.e., 1U, blade servers) is the heat dissipation. Much of the electrical power consumed by computing resources gets turned into heat. The amount of heat produced by an integrated circuit depends on how efficient the component's design is, and the voltage and frequency at which the component operates. The heat generated by the resources has to be dissipated to keep them within their safe thermal state. Overheating of the components can lead to a decrease of their lifetime and high error-proneness. Moreover, power is required to feed the cooling system operation. For each watt of power consumed by computing resources, an additional 0.5–1 W is required for the cooling system [6].

B. High energyconsumption

The way to address high power consumption is the minimization of the peak power required to feed a completely utilized system. In contrast, the energy consumption is defined by the average power consumption over a period of time. Therefore, the actual energy consumption by a data center does not affect the cost of the infrastructure. However, it is reflected in the cost of electricity consumed by the system, which is the main component of data center operating costs. Furthermore, in most data centers, 50% of consumed energy never reaches the computing resources: it is consumed by the cooling facilities or dissipated in conversions within the UPS and PDU systems. With the current tendency of continuously growing energy consumptionandcostsassociatedwithit,thepointwhenoperatingcostsexceedthecostof computing resources themselves in few years can be reached soon. Therefore, it is crucial to develop and apply energy-efficient resource management strategies in data centers.

TAXONOMY OF POWER/ENERGY MANAGEMENT IN COMPUTING SYSTEMS

A large volume of research has been done in the area of power and energy-efficient resource management in computing systems. As power and energy management techniques are closely connected, from this point we will refer to them as power management. As shown in Fig. 3, the high-level power management techniques can be divided into static and dynamic. From the hardware point of view, SPM contains all the optimization methods that are applied at the design time at the circuit, logic, architectural, and system levels [17]. Circuit level optimizations are focused on the reduction of the switching activity power of individual logic gates and transistor level combinational circuits by the application of a complex gate design and transistor sizing. Optimizations at the logic level are aimed at the switching activity power of logic-level combinational and sequential circuits. Architecture level methods include the analysis of the system design and subsequent incorporation of power optimization techniques in it. In other words, this kind of optimization refers to the process of efficient mapping of a high-level problem specification onto a register-transfer level design. Apart from the optimization of the hardware-level system design, it is extremely important to carefully consider the implementation of programs that are supposed to run in the system. Even with perfectly designed hardware, poor software design can lead to dramatic performance and power losses. However, it is impractical or impossible to analyze power consumption caused by large programs at the operator level, as not only the process of compilation or code generation but also the order of instructions can have an impact on power consumption. Therefore, indirect estimation methods can be applied. For example, it has been shown that faster code almost always implies lower energy consumption [18]. Nevertheless, methods for guaranteed synthesizing of optimal algorithms are not available, and this is a very difficult researchproblem.

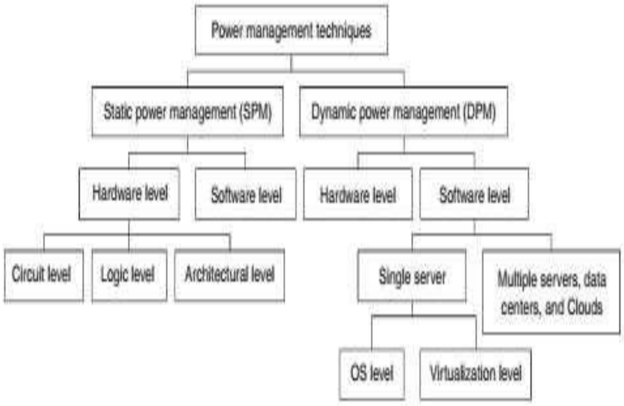

Figure 3: High-level taxonomy of power and energy management

C. Hardware and FirmwareLevel

The problem could be easily solved if transitions between power states would cause negligible power and performance overhead. However, transitions to low power states usually lead to additional power consumption and delays caused by the reinitialization of the components. For example, if entering a low-power state requires shutdown of the power supply, returning to the active state will cause a delay consisting of turning on and stabilization of the power supply and clock, reinitialization of the system, and restoring the context [23]. In the case of non-negligible transitions, efficient power management turns into a difficult online optimization problem. A transition to low-power state is worthwhile only if the period of inactivity is longer than the aggregated delay of transitions from and into the active state, and the saved power is higher than the required to reinitialize thecomponent.

.

D. Operating SystemLevel

The taxonomy of the characteristics used to classify the works is presented in Fig. 4.

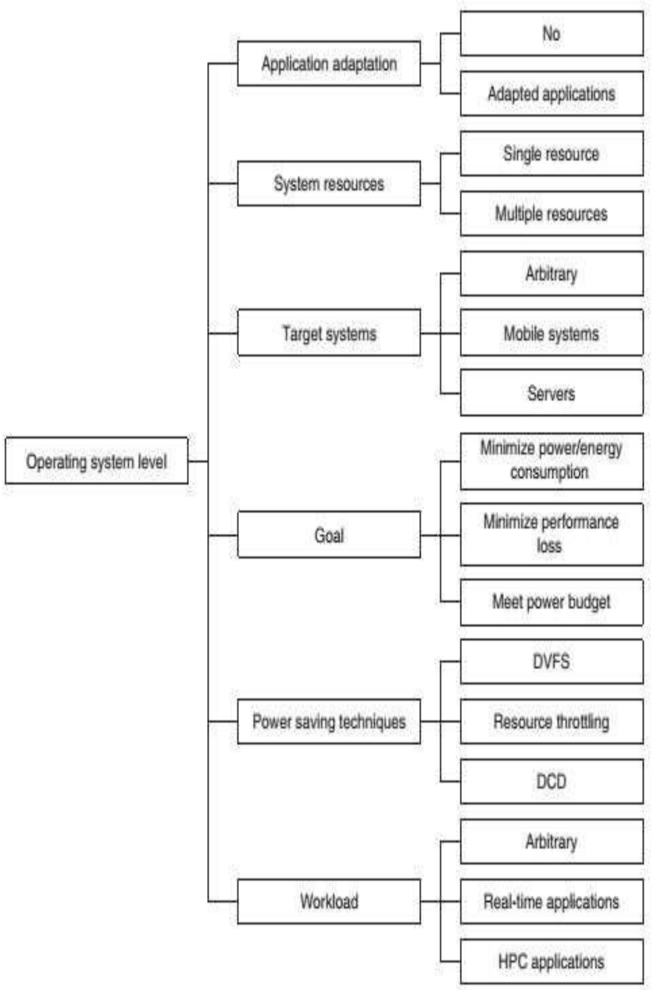

Figure 4: Operating system level taxonomy

Pallipadi and Starikovskiy [19] have developed an in-kernel real-time power manager for the Linux OS called the on-demand governor. The manager continuously monitors the CPU utilization multiple times per second and sets a clock frequency and supply voltage pair that corresponds to current performance requirements keeping the CPU approximately 80% busy to handle fast changes in the workload. The goal of the on-demand governor is to keep the performance loss due to reduced frequency to the minimum. Modern CPU frequency scaling technologies provide extremely low latency allowing dynamic adjustment of the power consumption matching the variable workload with almost negligible performance overhead. For example, Enhanced Intel Speed step Technology enables frequency switching with the latency as low as 10 ms. To accommodate different requirements of diverse systems, the on-demand governor can be tuned via the specification of the rate at which the CPU utilization is checked and the value of the upper utilization threshold, which is set to 80% bydefault.

There are a number of improvements that are currently under investigation, including parallel calculation of the utilization and a dedicated work queue. The original governor samples the utilization of all of the processors in the system in a centralized way that can become a significant overhead with increase in the number of CPUs. To overcome this problem, the authors in [19] have proposed a parallel sampling independently for each CPU. Another improvement that can increase the performance for multiprocessor systems is to have dedicated kernel threads for the governor and do sampling and changing of frequencies in the context of a particular kernel thread.

E. EcoSystem

Zeng et al. [37,38] have proposed and developed ECO system—a framework for managing energy as a first-class OS resource aimed at battery-powered devices. The authors' fundamental assumption is that applications play an important role in energy distribution opportunities that can be leveraged only at the application level. ECO system provides an interface to define a target battery lifetime and applications' priorities used to determine the amount of energy that will be allocated to applications at each time frame.

The authors split OS-level energy management into two dimensions. Along the first dimension, there is a variety of system devices (e.g., CPU, memory, disk storage, network interface) that can consume energy concurrently. The other dimension spans applications that share the system devices and cause the energy consumption. To address the problem of accounting the energy usage by both devices and applications, the authors have introduced a new measurement unit called currency. One unit of currency represents the right to consume a certain amount of energy during a fixed period of time. When the user sets the target battery lifetime and prioritizes the applications, ECO system transforms these data into an appropriate amount of currency and determines how much currency should be allocated to each application at each time frame. The length of the timeframe has been empirically determined as 1 s that is sufficient to achieve smooth energy allocation. An application expends the allocated amount of currency by utilizing the CPU, performing disk and memory accesses and consuming other system resources. An application can accumulate currency up to a specified limit. When an expenditure of an application exceeds the allocated amount of currency, none of the associated processes are scheduled or otherwiseserviced.

The system has been implemented as a modified Linux kernel and has been experimentally evaluated. The obtained results show that the proposed model can be effectively used to meet different energy goals, such as achieving a target battery lifetime and proportional energy distribution among competing applications.

VIRTUALISATION LEVEL

The virtualization level enables the abstraction of an OS and applications running on it from the hardware. Physical resources can be split into a number of logical slices called VMs. Each VM can accommodate an individual OS creating for the user a view of a dedicated physical resource and ensuring the performance and failure isolation between VMs sharing a single physical machine. The virtualization layer lies between the hardware and OS and, therefore, a virtual machine monitor (VMM) takes the control over resource multiplexing and has to be involved in the system's power management. There are two ways of how a VMM can participate in the power management:

1. A VMM can act as a power-aware OS without distinction between VMs: monitor the overall system's performance and appropriately apply DVFS or any DCD techniques to the systemcomponents.
2. Another way is to leverage OS's specific power management policies and application-level knowledge, and map power management calls from different VMs on actual changes in the hardware's power state or enforce system-wide power limits in a coordinatedmanner.

Implications of Cloud Computing

Cloud computing has become a very promising paradigm for both consumers and providers in various areas including science, engineering, and not to mention business. A Cloud typically consists of multiple resources possibly distributed and heterogeneous. Although the notion of a Cloud has existed in one form or another for some time now (its roots can be traced back to the mainframe era [66]), recent advances in virtualization technologies and the business trend of reducing the TCO in particular have made it much more appealing compared to when it was first introduced. There are many benefits from the adoption and deployment of Clouds, such as scalability and reliability; however, Clouds in essence aim to deliver more economical solutions to both parties (consumers and providers). By economical, we mean that consumers only need to pay per their use and providers can capitalize poorly utilizedresources.

CONCLUSIONS

In recent years, energy efficiency has emerged as one of the most important design requirements for modern computing systems, ranging from single servers to data centers and Clouds, as they continue to consume enormous amounts of electrical power. Apart from high operating costs incurred by computing resources, this leads to significant emissions of CO_2into the environment. For example, currently, IT infrastructures contribute about 2% of the total CO_2 footprints. Unless energy-efficient techniques and algorithms to manage computing resources are developed, IT's contribution in the world's energy consumption and CO_2emissions is expected to rapidly grow. This is obviously unacceptable in the age of climate change and global warming. To facilitate further developments in the area, it is essential to survey and review the existing body of knowledge. Therefore, in this chapter, we have studied and classified various ways to achieve the power and energy efficiency in computing systems. Recent research advancements have been discussed and categorized across the hardware, OS, virtualization, and data centerlevels.

REFERENCES

[1] G.E. Moore, Cramming more components onto integrated circuits, Proc. IEEE 86 (1) (1998)82–85.

[2] J.G. Koomey, Estimating Total Power Consumption by Servers in the US and the World, Analytics Press, Oakland, CA,2007.

[3] L. Barroso, The Price of Performance, ACM Press, Queue, 2005, vol. 3 (7), p.53.

[4] R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, et al., Report to Congresson
Server and Data Center Energy Efficiency: Public Law 109–431, Lawrence Berkeley National Laboratory, Berkeley, CA, USA, 2008.

[5] L. Minas, B. Ellison, Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers, Intel Press, USA,2009.

[6] P. Ranganathan, P. Leech, D. Irwin, J. Chase, Ensemble-level power management for denseblade
servers, in: Proceedings of the 33rd International Symposium on Computer Architecture (ISCA 2006), 2006, pp. 66–77.

[7]S.Rowe,Usenetarchives.http://groups.google.com/group/comp.misc/browse_thread/thread/5c4db94663b5808a/f99158e3743127f9, 1992.

[8] V. Venkatachalam, M. Franz, Power reduction techniques for microprocessor systems, ACM Comput. Surv. CSUR 37 (3) (2005)195–237.

[9] L.A. Barroso, U. Holzle, The case for energy-proportional computing, Computer 40 (12) (2007)33–37.

[10] X. Fan, W.D. Weber, L.A. Barroso, Power provisioning for a warehouse-sized computer, in: Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA 2007), ACM New York, NY, USA, 2007, pp.13–23.

[11] M. Blackburn, Five Ways to Reduce Data Center Server Power Consumption, The Green Grid, USA,2008.

[12] American Society of Heating and Refrigerating and Air-Conditioning Engineers, Thermal Guidelines for Data Processing Environments, ASHRAE, Atlanta, GA, USA,2004.

[13] G. Dhiman, K. Mihic, T. Rosing, A system for online power prediction in virtualized environments using Gaussian mixture models, in: Proceedings of the 47th ACM/IEEE Design Automation Conference, Anaheim, CA, USA, 2010, pp.807–812.

[14] G. Koch, Discovering multi-core: extending the benefits of Moore's law, Technology 1 (2005).

[15] F. Petrini, J. Moreira, J. Nieplocha, M. Seager, C. Stunkel, G. Thorson, et al., What are the future trends in high-performance interconnects for parallel computers? in: Proceedings of the 12th Annual IEEE Symposium on High Performance Interconnects, 2004, p.3.

[16] C. Pettey, Gartner estimates ICT industry accounts for 2 percent of global CO2 emissions, http:// www.gartner.com/it/page.jsp?id¼503867,2007.

[17] S. Devadas, S. Malik, S. Devadas, S. Malik, A survey of optimization techniques targeting low power VLSI circuits, in: Proceedings of the 32nd ACM/IEEE Conference on Design Automation, 1995, pp.242–247.

[18] V. Tiwari, P. Ashar, S. Malik, Technology mapping for low power, in: Proceedings of the 30th Conference on Design Automation, 1993, pp. 74–79.

[19] V. Pallipadi, A. Starikovskiy, The on demand governor, in: Proceedings of the Linux Symposium, 2006, vol.2.

[20] E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, Power Aware Comput. Syst. 2325 (2003)179–197.

[21] E. Pinheiro, R. Bianchini, E.V. Carrera, T. Heath, Load balancing and unbalancing for power and performance in cluster-based systems, in: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, 2001, pp. 182–195

[22] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future Generation Comput. Syst. 25 (6) (2009)599–616.

[23] L. Benini, A. Bogliolo, G.D. Micheli, A survey of design techniques for system-level dynamic power management, IEEE Trans. VLSI Syst. 8 (3) (2000)299–316.

[24] S. Albers, Energy-efficient algorithms, Commun. ACM 53 (5) (2010)86–96.

[25] M.B. Srivastava, A.P. Chandrakasan, R.W. Brodersen, Predictive system shutdown and other architectural techniques for energy efficient programmable computation, IEEE Trans. VLSI Syst. 4 (1) (1996)42–55.

[26] C.H. Hwang, A.C. Wu, A predictive system shutdown method for energy saving of event-driven computation, ACM Trans. Des. Autom. Electron. Syst. 5 (2) (2000)241.

[27] F. Douglis, P. Krishnan, B. Bershad, Adaptive disk spin-down policies for mobile computers, Comput. Syst. 8 (4) (1995)381–413.

[28] G. Buttazzo, Scalable applications for energy-aware processors, in: Embedded Software, 2002, pp.153–165.

[29] M. Weiser, B. Welch, A. Demers, S. Shenker, Scheduling for reduced CPU energy, Mobile Comput. (1996)449–471.

[30] K. Govil, E. Chan, H. Wasserman, Comparing algorithm for dynamic speed-setting of a low-power CPU, in: Proceedings of the 1st Annual International Conference on Mobile Computing and Networking (MobiCom 2005), Berkeley, CA, USA, 1995, p.25.

[31] A. Wierman, L.L. Andrew, A. Tang, Power-aware speed scaling in processor sharing systems, in: Proceedings of the 28th Conference on Computer Communications (INFOCOM 2009), Rio, Brazil,2009.

[32] L.L. Andrew, M. Lin, A. Wierman, Optimality, fairness, and robustness in speed scaling designs, in: Proceedings of ACM International Conference on Measurement and Modeling of International Computer Systems (SIGMETRICS 2010), New York, USA,2010.

[33] A. Weissel, F. Bellosa, Process cruise control: event-driven clock scaling for dynamic power management, in: Proceedings of the 2002 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, Grenoble, France, 2002, p.246.

[34] K. Flautner, S. Reinhardt, T. Mudge, Automatic performance setting for dynamic voltage scaling, Wireless Netw. 8 (5) (2002)507–520.

[35] S. Lee, T. Sakurai, Run-time voltage hopping for low-power real-time systems, in: Proceedings of the 37th Annual Design Automation Conference, Los Angeles, CA, USA, 2000, pp.806–809.

[36] J.R. Lorch, A.J. Smith, Improving dynamic voltage scaling algorithms with PACE, ACM SIGMETRICS Perform. Eval. Rev. 29 (1) (2001) 61.

[37] H. Zeng, C.S. Ellis, A.R. Lebeck, A. Vahdat, Ecosystem: managing energy as a first class operating system resource, ACM SIGPLAN Notices 37 (10) (2002),132.

[38] H. Zeng, C.S. Ellis, A.R. Lebeck, Experiences in managing energy with ecosystem, IEEE Pervasive Comput. 4 (1) (2005) 62–68.

[39] R. Neugebauer, D. McAuley, Energy is just another resource: energy accounting and energy pricing in the nemesis OS, in: Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems, Elmau/Oberbayern, Germany, 2001, pp.59–64.

[40] D.G. Sachs, W. Yuan, C.J. Hughes, A. Harris, S.V. Adve, D.L. Jones, et al., GRACE: a hierarchical adaptation framework for saving energy, University of Illinois at Urbana-Champaign, Technical Report, UIUCDCS, 2003, pp.2004–2409.

[41] V. Vardhan, D.G. Sachs, W. Yuan, A.F. Harris, S.V. Adve, D.L. Jones, et al., Integrating fine grained application adaptation with global adaptation for saving energy, in: International Workshop on Power-Aware Real-Time Computing, Jersey City, NJ,2005