Energy Saving Techniques in Datacenters

I. Grudenic and N. Bogunovic
Department of electronics, microelectronics, computer and intelligent systems
Faculty of Electrical Engineering and Computing, Unska 3, Zagreb, Croatia
{igor.grudenic@fer.hr, nikola.bogunovic@fer.hr}

Abstract – Number of internet users as well as number of large distributed computer systems has increased dramatically in the last decade. Large distributed computer systems serving lot of users are usually oversized in order to be able to keep up with user demand surges. However in many usage scenarios workload spikes are rare and systems operate on 20%-30% average utilization. Underutilized systems are shown to be power inefficient and different strategies are proposed to tackle this issue. In this paper we present an overview of different power saving techniques that target different types of systems and workloads.

I. Introduction

Internet services such as web search, mail, social networking and cloud computing are becoming increasingly popular. This increase in popularity generating a lot of revenue motivates companies to fight for their users by both improving their services as well as enhancing user experience. Improvement in user experience generally comes from high availability and performance guarantees. Performance guarantees and high availability are achieved by over engineering systems which in turn leads to a lot of underused system capacity over extended time periods. It is shown [1][2] that large datacenters are largely underexploited with 20%-30% average utilization. Oversized systems lead to increased power usage that can be leveraged by different saving schemes.

The problem of internet services growth causing increase in energy consumption was recognized by EPA in 2007 report to U.S. Congress [3]. The report estimated power consumption to increase twofold in five years time. More recent study by [4] shows that increase in energy consumption is actually about 36% in the U.S. and 56% worldwide. Lower than expected increase in electricity use is attributed to the 2008-2009 economic crisis and the use of virtualization technologies in data centers. Exploitation of virtualization technology is considered an energy preserving technique.

Energy preservation is not a new goal in computer system design. Miniaturization of the technology and rise of mobile devices in late 80s and through the nineties forced hardware manufacturers to improve battery technology and implement power management techniques into their products. However these techniques are of limited availability in modern server systems which make power efficient designs more difficult to come up with.

Different strategies for energy conservation are proposed on different optimization levels. Location of new data centers are being carefully selected according to climate conditions [5]. Natural sources such as wind and nearby rivers are used to prevent electricity to be spent on cooling the equipment. On the other hand hardware manufacturers can employ several energy saving techniques in order to make power efficient system design possible. On a software level, which includes operating systems and data center management software, different schemes are used to improve data center energy efficiency.

In this paper we give an overview of different system design techniques (both hardware and software) that can be used to decrease power usage in data centers. Section II contains description of hardware components, their energy consumption and designs that can be used (some coupled with appropriate control software) to preserve energy. In section III we present commonly used data centre architecture as well as some of the representative workloads used to estimate power efficiency of different techiques. In section IV we give classification and comparison of different energy saving techniques.

II. Hardware Based Power Optimization

Hardware design is a starting point for every energy optimization technique. Software based strategies are engineered to exploit on different hardware provisioned energy modes taking given workload into account. Energy consumption by a typical data center server made from commodity components is given in Table I [6]. Additional power conversion losses and cooling overheads in modern facilities can be approximately modeled as a fixed percentage of the computing power [6]. However some authors [7] specifically tackle power conversion losses

TABLE I. Server components energy consumption

COMPONENT	**POWER DRAW**		
	Peak Power	***Count***	***Total***
CPU	40W	2	80W
Memory	9W	4	36W
Disk	12W	1	12W
PCI Slots	25W	2	50W
Motherboard	25W	1	25W
Fan	10W	1	10W

and propose a more power efficient power supply design.

In this section we break down CPU, hard drive and power supply energy consumption and reveal hardware optimizations that can be exploited to optimize energy consumption of these components.

A. CPU

CPU is a main single power hog component in computer systems. It consumes about 40% of total energy (depending on server design) when fully loaded. The main problem is that CPUs rarely operate at 100% utilization and CPU power consumption is not proportional to utilization.

Generally electronic device power consumption according to Ohm's law is:

$$P = C \cdot V^2 \cdot f$$

where C is capacitance, V voltage and f operating frequency. Although this equation does not take utilization into account, hardware manufacturers came out with designs that are able to manipulate voltage and frequency and even turn off unneeded parts of the circuitry. This is done by employing hardware design techniques such as clock gating.

Some authors [8] argue that Ohm's law is oversimplification for power consumption in modern CPUs and they propose different empirically based power models such as:

$$P(f_i, U_i) = a_{i3} \cdot f_i \cdot U_i + a_{i2} \cdot f_i + a_{i1} U_i + a_{i0}$$

where f_i and U_i are discrete combinations of frequency and utilization and a_{ij} constants are obtained experimentally by curve fitting the model with experimental results.

Modern processors employ several power saving techniques which are abstracted by power states. Power states for Intel desktop processors are shown in Fig. 1 [9]. Computer system can be in one of the 3 main states: working, sleeping or mechanical off. If the system is working then CPU or some of its cores can enter one of the *C* states. Each *C* state defines parts of the CPUs that are inactive causing the energy consumption to decrease. Higher *C* states have lower electricity usage due to more CPU parts being disabled, but they also require longer wakeup time. Measurements on a Dell Vostro 430s system running at 2.93Ghz reveal entire system power consumption in *C0* CPU state to be about 110W [10]. Power usage decreases by 45%, 50% and 60% when CPU is forced to *C1E*, *C3* and *C6* states.

When CPU or some of its cores is in active mode (*C0*) it is possible to control its performance and power consumption by entering into one of the power states (*P0..Pn*). Power states are discrete states with predefined frequency and voltage combinations that are used to preserve energy when system is not fully utilized. Although power usage is decreased when CPU clock is scaled down it takes longer time to perform computations on lower frequencies and this must also be taken into account when simulating energy saving techniques.

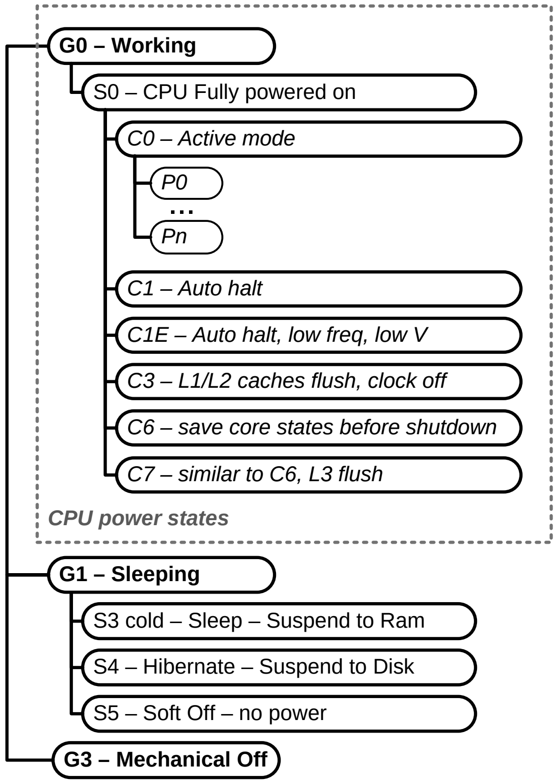

Figure 1. Sytem/CPU power states

B. Hard drives

Hard drive power consumption ranges from few watts up to 15 watts mostly depending on the platter rotation speed. Hard drive manufacturers have also recognized needs for power savings in large data centers and they have introduced several power states controllable by SATA or SAS interface. Seagate [11] modern hard drives consume from 23%-54% less energy in power saving states with wakeup times ranging from 0.5 to 8 seconds.

C. PSU

Power supply units are designed to convert AC to DC current that is used by the computer components. Modern power supplies that are most commonly used are very efficient when under full load, but this drops down to 50% efficiency when system is idling [7].

PSUs are not designed to be controllable by software in order to respond to energy since utilization can be automatically determined from the electrical current drawn by the system. Although power supplies are not controllable externally their models are sometimes used in simulation of power saving techniques.

III. Data Center Architecture

Data center systems are usually designed as a multitier architecture having load balancer on the front, followed by application servers as a second tier taking the requests and a data tier serving information to the application servers. In many cases both application and data tier have a degree of redundancy and parts of the system that are being idle can be temporarily put to sleep or shutdown. Application tier machines are usually easy to turn off since they contain no data possibly needed by subsequent requests. Data tier is more complicated to deal with since possibility of turning off the machines depends largely on data replication and redundancy schemes. These schemes may need to be altered in order to allow for certain subset of data tier machines to be safely turned off in low utilization scenarios.

Some of the energy saving schemes target a bit different or simplified versions of data center architecture, while others use a single machine to experiment on.

IV. Energy saving strategies

Classification of energy saving strategies is not straight forward and it can be done using various criteria. Some of the criteria include target system hardware and software architecture, type of workload used or a type of hardware optimization exploited. In this paper categorization is made depending on type of hardware optimization exploited. Most of the energy conservation strategies manipulate either CPU states or use system sleep states at off peak times. Server consolidation and virtual machine usage are also techniques used for power saving in computer systems. Following subsections describe different energy saving strategies. Summary of techniques, workload environments, software architecture and energy gains is presented in Table II.

A. CPU states manipulation techniques

First work on CPU frequency manipulation techniques used time intervals in which CPU frequency was fixed [12]. At the end of one interval workload in the next interval is estimated and CPU frequency is adjusted. Workload prediction is based on the previous time interval (PAST strategy) and energy savings are calculated using simulation of the single computer system. Energy consumption is reduced up to 50% depending on workload trace (several unix workstation user traces were used).

These theoretical results are put on trial in real hardware [13] using pocket computer with StrongArm CPU. CPU was configured to work on 2 base voltage levels (1.5V and 1.23V) and clock was adjusted accordingly. PAST strategy is used to predict workload in the next time interval. Several applications are used in workload trace and results are not promising. Real time applications such as MPEG decoding and text to speech conversion are shown to be choppy because system couldn't react fast enough on changes in load.

Simulation of web cluster with two different workloads (Winter Olympics 98 [14] and financial web site) reveal power savings of 22%-42% [1]. Savings were greater for the financial web site since Olympic web site has greater average utilization and more balanced workload over time.

Since CPU frequency/voltage control and C states transition is now built into modern OS schedulers research on energy savings on one Intel i7 based machine is performed [10]. In experiment different frequencies and different CPU C states were employed. Usage of different CPU C states was enabled by using custom Linux *cpuidle* governor [15]. Employing higher and more energy efficient C states is assumed to be beneficial in low utilization scenarios. In high load scenarios deeper C states would be of little use since longer wakeup times would cause observable power usage overhead. Results show that frequency and C state optimizations independently lead to lower power consumption. This was shown to be applicable to both MPEG decoding and web server workload scenarios with average system utilization between 7% and 28%. Overall power savings were up to 50% of total consumed power compared to the system that is always on and working on top frequency (*P0* state).

B. Optimization using single sleep state (turn on/off)

Transition of the machines between sleep states and powered on states can be exploited to conserve electrical energy. Sleep states differentiate on both power usage and wakeup times. Typical sleep states that are most frequently implemented and used are S3 (sleep) and S4 (hibernate). Transition to these power states takes about 5-10 seconds for S3 and 30 or more seconds for S4 depending on amount of memory that needs to be stored on the hard drive. Wakeup from both states to state S0 takes approximately same time. Power consumption of the entire computer system in both states is typically less then 5W so there is no justifiable difference to use S4 state due to its much longer transition state.

Simulation of a web cluster using 8PCs with controllable outlets to perform switch on/of is performed to measure power savings achieved by the use of PID (proportional-integral-differential) feedback controller [16]. One feedback controller is used for each running PC in order to compute excess resource demand. Proportional, integral and differential components are used in order to reflect past and current resource demand as well as the latest demand trend. Feedback from all PCs is collected and decision for turning computers on or off is done using the highest excess resource demand. In order to accommodate for resource time wasted in switching cycles each PCs nominal capacity is reduced. Accommodation for workload spikes is done by keeping fixed percentage of spare resources active.

PID based technique is tested on two workloads – one involves load balanced lowly utilized web cluster and the other use of common single machine applications such as MPEG encoding and SPEC 2000 benchmark. For single machine applications it is assumed that there exists a mechanism for application migration between different machines. Results show 38% in energy savings for web server workload [17] with 30% spare resource capacity and up to 45% with only 15% spare capacity. Simulation of computer cluster workload running MPEG encoding and SPEC 2000 benchmark reveals 35% energy savings.

Strategy VOVO+DVS that combines switching machines on and off and CPU frequency manipulation techniques is evaluated on a simulation of a web cluster [1]. Main idea is to keep a number of machines at optimal frequency. If the average frequency of the system drops below theoretically calculated limit then some of the machines are turned off. Conversely if the average

TABLE II. COMPARISON OF ENERGY SAVING TECHNIQUES

Authors	Manipulation strategy	TA	Workload	Energy Savings	Hardware
Weisner 1994	CPU frequency	SC	Unix workstation trace	50%	Simulation
Grunwald 2000	CPU frequency	SC	MPEG ,text to speech	No*	Pocket pc
Elnozahy 2002	CPU frequency	WC	WO 98 and financial web site	20%-29%	Simulation
LeSueur 2011	CPU frequency	SC	MPEG, www ,SPEC 2005	up to 50%	i7 CPU system
LeSueur 2011	CPU C states	SC	MPEG, www, SPEC 2005	50%	i7 CPU system
Elnozahy 2002	On/Off	WC	WO 98 and financial web site	22%-42%	Simulation
Elnozahy 2002	CPU frequency+On/Off	WC	WO 98 and financial web site	33%-50%	Simulation
Horvath 2008	CPU frequency+On/Off+Opt	3TC	WWW (epa,sdsc,WCup 98)	17%-27%	12 Linux PCs
Horvath 2008	CPU frequency+On/Off+Dem.	3TC	WWW (epa,sdsc,WCup 98)	17%-21%	12 Linux PCs
Horvath 2008	CPU frequency+Multiple Sleep S. +Opt	3TC	WWW (epa,sdsc,WCup 98)	25%-34%	12 Linux PCs
Horvath 2008	CPU frequency+Multiple Sleep S+Dem.	3TC	WWW (epa,sdsc,WCup 98)	22%-36%	12 Linux PCs
Krioukov 2010	Multiple Sleep S./Hybrid architecure	WC	Wikipedia trace	27%	Simulation
Gandhi 2011	On/Off	CC	Syntetic trace	PPW increase	Simulation
Kusic 2008	Virtual machines	WC	WCup 98	22%	Dell PowerEdge
Bodik 2009	Virtual machines	WC	ebates.com trace	Cost decrease	Amazon ec2

TA-target achitercture, SC – single computer, WC – web cluster, 3TC – 3 tier cluster, CC computer cluster

WO 98 – World Olympics 1998 trace, Wcup 98 – World Cup 1998 trace

*Stuttering video/sound

frequency of the system needs to be increased beyond optimal power band (due to increased load) then additional machines are being turned on.

Evaluation is performed using simulation of two workload traces and energy savings reported are 33%-50%. In cases where CPU frequency manipulation was turned off energy savings achieved by solely turning machines off were in 22%-42% range.

The most general discussion on potential benefit of the sleep states simulates a number of hypothetical sleep states with different energy requirements and various wakeup times [18]. Analysis shows twofold increase in performance per watt for slow changing workloads and efficient sleep states. Efficient sleep states require low energy (few watts) and have small wakeup times (less then 20seconds).

C. *Optimization using multiple sleep states*

Usage of multiple sleep states together coupled with CPU frequency control in multitier architecture is exploited in [8]. Theoretical model is calculated that determines number of machines needed in separate tiers in order to conform to predefined service level agreement (SLA). Two policies are used to control behavior of the system: active capacity policy and spare server policy.

Active capacity policy (ACP) determines number of machines and their frequencies needed for system to operate according to SLA and with lowest energy requirements. First step of ACP includes finding minimal number of machines so that any individual tier will conform to SLA. Afterwards additional machines are added needed for entire system to work within SLA. In the second step frequencies at each tier are adjusted to minimize energy consumption. Distribution of machines and CPU frequencies to tiers follows the theoretical model computed by minimizing total active power consumed. Variables over which Lagrange multiplier minimization is performed are frequency and number of machines in a tier.

Spare capacity policy (SCP) determines number of machines optimally needed in each of the sleep states. If current system status doesn't meet the requirements number of machines are transitioned to appropriate sleep states. Optimal and demotion are two SCP policies that put servers to sleep. Optimal computes theoretical optimal number of servers for predicted workload while demotion policy keeps timers for a number of machines in a given state and when timers run out transitions to deeper sleep states are triggered. Workload prediction is done by PI (Proportional-Integral) feedback controllers.

Measurements of both SCPs and single or multiple sleep states where made on a three tier architecture (web server, application server and database) managed by a front load balancer. This architecture is deployed on a 12 node computer cluster where both web server and application server tier are allowed to run on variable number of machines. Power savings are greatest when multiple sleep states (S0, S3, S4) are used (22%-36%). Optimal SCP policy is shown to almost always outperform demotion with average power savings of 25%-35%.

D. Optimization using multiple sleep states in heterogenous systems

Since multiple sleep state usage proved to be advantageous in power saving, this approach is extended to a heterogeneous architecture/multiple sleep state technique [19]. Motivation for multiple architecture solution is brought to light since typical server processors used in data centers such as Intel Xeon [20] have no support for sleep states. In order to make these systems perform in a power efficient manner hybrid system made of Intel Xeon and Intel Atom CPUs [21] is proposed. Intel Xeon processors are dedicated to handle the base load, and Atoms are used for quick reaction to load bursts since they expose a number of sleep states with short wake up times.

Algorithm for machine wake up iterates over each type of a machine (Xeon and Atom) and tries to predict amount of workload in time it takes for the observed machine type to wake up. In this way if surge in workload is expected shorly a lot of Atom machines will wake up. If there is a longer demand for computation power expected Xeon machines will power on too. Procedure for machine power down (putting to sleep) is similar in nature. Load prediction methods employed include Last Arrival, Moving Window Average and Exponentially Weighted Average. Last Arrival assumes workload in the next interval will be the same as current workload, Moving Window Average computes an average workload over a past time frame while Exponentially Weighted Average takes into account last predicted load rate and last seen load rate with different factors.

Simulation results show 27% improvement in energy efficiency running Wikipedia workload trace when baseline comparison is made to the computer cluster designed to handle peak load at most. Additionally it is noted that power used is proportional to the work done in any given time. Moving Window Average prediction method is shown to be the best by underperforming theoretical optimum for only 10%.

E. Virtual machine and server consolidation techniques

It has been shown that systems with low average utilization are not power efficient. Large systems are usually comprised of different services distributed among distinct machines. Consolidation of several services (such as web server and mail server) on a single machine can be beneficial if workload spikes for consolidated services are not likely to occur simultaneously. Main drawback of this approach is that consolidation of services within same host operating system involves significant human effort.

Natural extension of server consolidation approach is exploitation of virtualized environments where different services in separate virtualized OS environments (virtual machines) run on a predefined set of hardware. Allocation of virtual machines to hardware can be done by one of several virtualization technologies.

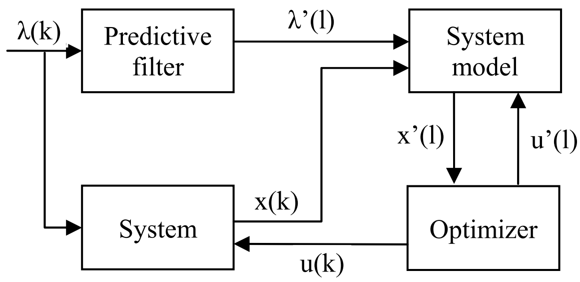

Figure 2. Limited lookahead controller

The goal of virtualization efforts is to minimize hardware/energy costs while maintaining specified SLA. There is a set of parameters needed for dynamical tuning of virtual machines [22]. Parameters include number of virtual machines (VM) provided to each service (application), number of hosts on which to allocate the virtual machines, CPU share of the host to be given to each VM and the number of host machines to power on.

In order to dynamically determine given set of parameters future load is predicted and different possible futures calculated using limited lookahead controller (Fig. 2) [22]. Prediction of input $\lambda'(l)$ in some future time l is performed by predictive filter. Sytem model calculates new system state $x'(l)$ according to predicted input and system control $u'(l)$ in some time l. Optimizer tries several different parameter settings $u'(l)$ for any calculated future system state. This leads to a construction of a tree of possible futures and trajectory with best predicted outcome is chosen.

LLC is tested on a heterogenous 6 machine computer cluster using 2 different services competing for hardware resources. Results obtained by simulating World Cup 98 workload [17] reveal 22% power savings over 24 hours while still maintaining service level agreement.

Simplified approach to VM scheduling is performed on the amazon ec2 testbed [23]. Since usage of amazon services doesn't provide information on power savings, minimization of total costs is set as optimization goal. Costs depend on VM rental costs and penalties for not conforming to SLA. Load prediction is done by linear regression over 15 minute system history. Future system load and needed machines are predicted by adjusting hysteresis parameters. Hysteresis is used to prevent system for overreacting to workload change.

System model is constantly evaluated and adjusted to resemble real system. Simulated system is shown to meet SLA as well as maintain cost as a linear function of system load.

V. Conclusion

In this paper we presented an overview of different energy saving techniques. These strategies enable significant power savings in underutilized systems. Since most of the systems are overdesigned to handle peak loads we can conclude that usage of energy saving techniques can be widely adopted.

Strategies for reducing energy consumption analyzed in this paper include CPU state manipulation techniques, sleep state exploitation and usage of virtual machines. CPU state manipulation techniques are implemented in most operating systems and deployed on mobile and desktop systems. Reported power savings obtained by CPU manipulation techniques range from 20%-50%.

Sleep state exploitation can be employed for systems where machines are not required to be constantly turned on. In low utilization scenarios machines can be put to different sleep states in order to save energy and be woken up in case of load increase. Synchronization of workload and number of active machines is not easy to make and involves keeping spare machine capacity as well as tuning of load prediction algorithms. Power savings reported in using sleep states range from 17%-50%.

Usage of virtual machines is the newest and probably most used power saving technique. It is usually employed by hosting companies that rent out virtual machines. Power savings reported by small scale simulation reveal 22% power savings made by shutting down unneeded hardware. Real world power savings of a virtualized data center are hard to estimate because there is no easily identifiable baseline scenario.

The main problem of power saving is compatibility with existing system architectures. Some large systems require most of the machines to be constantly available since they contain data that can be accessed at any time regardless of system utilization. Power savings of greatest magnitude are possible in cases in which systems are designed from scratch targeting energy optimization.

REFERENCES

[1] E.N. Elnozahy, M. Kistler and R. Rajamony, “Energy-Efficient Server Clusters”, In Proceedings of the 2nd Workshop on Power-Aware Computing Systems, 2002, pp. 179-196.

[2] L.A. Barroso and U. Hölzle, “The Case for Energy-Proportional Computing”, IEEE Computer, vol. 40, pp. 33-37, 2007.

[3] U.S. Environmental Protection Agency, “Report to Congress on Server and Data Center Energy Efficiency”, 2007.

[4] Koomey, Jonathan. 2011. Growth in Data center electricity use 2005 to 2010. Oakland, CA: Analytics Press. August 1, 2010.

[5] Emerson, “Economizer Fundamentals: Smart Approaches to Energy-Efficient Free-Cooling for Data Centers”, 2010.

[6] 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,vol. 35, 2007.

[7] D. Meisner, B.T. Gold and T.F. Wenisch, “PowerNap: eliminating server idle power”, In Proceedings of the 14th international conference on Architectural support for programming languages and operating systems, vol. 44, pp. 205-216, 2009.

[8] T. Horvath and K. Skadron, “Multi-mode energy management for multi-tier server clusters”, In Proceedings of the 17th international conference on Parallel architectures and compilation techniques, pp. 270-279, 2008.

[9] “2nd Generation Intel® Core™ Processor Family Desktop, Intel® Pentium® Processor Family Desktop, and Intel® Celeron® Processor Family Desktop”, Intel, Datasheet, vol. 1, 2011

[10] E. Le Sueur and G. Heiser, “Slow Down or Sleep, that is the Question”, In Proceedings of the 2011 USENIX Annual Technical Conference, USA, 2011

[11] “Reducing Storage Energy Consumption by up to 75%”, Seagate, 2011.

[12] M. Weiser, B. Welch, A. Demers, and S. Shenker, “Scheduling for reduced cpu energy”, In First Symposium on Operating Systems Design and Implementation, pp. 13–23, 1994.

[13] Dirk Grunwald, Philip Levis, Charles B. Morrey III, and Michael Neufeld, "Policies for Dynamic Clock Scheduling." In Proceedings of the 4th Symposium on Operating System Design and Implementation ,2000.

[14] J. Challenger, P. Dantzing and A. Iyengar, “A scalable and highly available system for serving dynamic data at frequently accessed web sites”, In Proceedings of 1998 ACM/IEEE Supercomputing, USA, 1998.

[15] V. Pallipadi, S. Li and A. Belay, “cpuidle—Do nothing, efficiently”, In Proceedings of Ottawa Linux Symposium, 2007.

[16] E. Pinheiro, R. Bianchini, E.V. Carrera and T. Heath, “Dynamic cluster reconfiguration for power and performance”, Compilers and operating systems for low power, Kluwer Academic Publishers, USA, 2003.

[17] M. Arlitt and T. Jin, “Workload Characterization of the 1998 World Cup Web Site”, HP Labs Technical Reports, 1998.

[18] 2011 A. Gandhi, M. Harchol-Balterm and M.A. Kozuch, “The case for sleep states in servers”, In Proceedings of the 4th Workshop on Power-Aware Computing and Systems, USA, 2011.

[19] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler and R. Katz, In Proceedings of the First ACM SIGCOMM Workshop on Green Networking, New Delhi, India, Aug 2010.

[20] “Intel® Xeon® Processor 5500 Series Datasheet”, Intel, vol. 1, 2009.

[21] “Intel® Atom™ Processor Z540/Z530/Z520/Z510/Z500 Datasheet”, Intel, June 2010.

[22] D. Kusic, J.O. Kephart, J.E. Hanson, N. Kandasamy, G. Jiang, “Power and Performance Management of Virtualized Computing Environments Via Lookahead Control”, In Proceedings of the International Conference on Autonomic Computing, USA 2008.

[23] P. Bodik, R. Griffith, C. Sutton, A. Fox, M. I. Jordan, and D. Patterson, “Statistical machine learning makes automatic control practical for Internet datacenters”, Workshop on Hot Topics in Cloud Computing (HotCloud), San Diego, CA, 2009.