Energy-Based Accounting and Scheduling of Virtual Machines in a Cloud System

Nakku Kim
Email: nkkim@unist.ac.kr

Jungwook Cho
Email: jmanbal@unist.ac.kr

Euiseong Seo
Email: euiseong@unist.ac.kr

School of Electrical and Computer Engineering,
Ulsan National Institute of Science and Technology,
Ulsan, Republic of Korea

***Abstract*—Currently, cloud computing systems using virtual machines bill users for the amount of their allocated processor time, or the number of their virtual machine instances. However, accounting without cooling and energy cost is not sufficient because the cooling and energy cost is expected to exceed the cost for purchasing the servers eventually. This paper suggests a model to estimate the energy consumption of each virtual machine. Our model estimates the energy consumption of a virtual machine based on the in-processor events generated by the virtual machine. Based on the suggested estimation model, this paper also proposes a virtual machine scheduling algorithm that conforms to the energy budget of each virtual machine. Our evaluation shows the suggested schemes estimate and provide energy consumption with errors less than 5% of the total energy consumption.**

***Keywords*-virtualization, scheduling, energy-aware computing, cloud computing, resource accounting**

I. Introduction

In a cloud system, virtualization is an essential tool for providing resources flexibly to each user and isolating security and stability issues from other users. [1] Currently, a lot of commercial cloud systems including Amazon EC2 employ virtualization so that users can freely configure their virtual servers from OS kernels to applications.

Most cloud services bill their users for the amount of computing resources being provided. [2] In general, the amount of processor time or the number of allocated virtual machines are common criteria for such accounting. [1]

Such billing systems are based upon the common belief that the cost to own and maintain server hardware is proportional to the amount of the operation time. However, the energy and cooling cost will exceed the hardware cost. [3] Consequently, the billing system must account for the energy consumption of each user as well.

Estimating the amount of energy that each virtual machine consumes during a certain time interval in a server with multiple virtual machines is a technically challenging issue because a virtual machine is a non-physical entity that cannot directly connect to a measurement device.

In this paper, we reveal that the processor run time is solely not sufficient to estimate the energy consumption of a processor, and that some in-processor events affect the processor energy consumption significantly. Based on these observations, we suggest an energy consumption estimation model that estimates the amount of energy consumption by a virtual machine by monitoring processor performance counters.

We also propose the energy-credit scheduler. The conventional virtual machine schedulers only consider the processor time when it comes to scheduling decisions. Different from its traditional counterparts, the energy-credit scheduler employs energy consumption rates of virtual machines as its scheduling basis. It schedules virtual machines so that the energy consumption rates of virtual machines stay below user-defined values.

We implemented our schemes in the Xen virtualization system [4], and evaluated it with the SPEC CPU2006 benchmark suite.

The organization of this paper is as follows. The background and related works are introduced in Section 2. We suggest the energy consumption estimation model of virtual machines and evaluate the model in Section 3. Based on the estimation model we present the energy-credit scheduler as well as its implementation and evaluation in Section 4. We conclude in Section 5.

II. Motivation and Related Work

Server vendors integrate power measurement hardware, which can monitor the server's power consumption on the fly [5]. Despite having previous research to leverage the dedicated measurement hardware, determining the energy consumption of a virtual machine is still a difficult problem for the following reasons.

First, the sampling interval of the integrated power meters is generally a second or a tenth seconds [6], while the unit of scheduling virtual machines is very short, from a few hundreds μs. to a few tens of ms.

Second, virtual machines in a multicore processor system share and compete with other virtual machines for system resources. Therefore, the throughput as well as the amount of energy consumed by a virtual machine may vary due to the characteristics of other virtual machines that run concurrently [7].

Consequently, in order to identify the energy consumption of each virtual machine, which continually and rapidly changes, the hypervisor must equip the resource accounting scheme that can estimate the energy consumption of a virtual machine by observing its various activities as well as the effects from the other virtual machines.

Besides the virtual machine-level energy measurement schemes, in order to adapt billing systems based on the amount of energy consumption, energy-aware resource provisioning schemes are required so that users can limit the energy consumption of their virtual machines to stay within their energy budgets.

In the desired scheme, each virtual machine should provide its energy budget to the scheduler explicitly. The energy budget of a virtual machine is the amount of energy the virtual machine may use during its *fiscal time interval*. If a virtual machine uses up its energy budget in a fiscal interval, then the virtual machine will be suspended until the current fiscal interval finishes and the next interval begins. The fiscal interval of a virtual machine is determined by the owner of the virtual machine in consideration of its purpose and characteristics.

Joule meter [8] is a software approach to measure the energy consumption of virtual machines in a consolidated server environment. Joule meter estimates the amount energy that each virtual machine consumes by monitoring its resource usage dynamically.

Joule meter uses two different energy consumption estimation approaches.

The first model is a simple model that calculates the amount of energy consumption by multiplying the processor time with the average power consumption of the processor, which is similar to that of ECOsystem. Due to the oversimplification, this model still shows poor accuracy.

The second model, the refined model, uses the integrated power measurement device to collect the power consumption patterns of the entire system. However, this approach requires the integrated power measurement device. Moreover, if the workload of a virtual machine fluctuates severely, this approach may perform poorly because this model is based on the assumption that the behaviour of the virtual machine remains the same as that of its first 200 sec.

ECOsystem [9], a prototype energy-centric operating system, considers the energy as a first-class operating system resource. it allocates energy to tasks according to their priorities and schedules the tasks within their energy budget in order to guarantee the battery lifetime. In their research, the processor is assumed to consume the fixed amount of power. This assumption is partially correct for mobile embedded systems, which are their targets, because the power consumption of processors in mobile embedded systems is significantly less than that in server systems and, therefore, less critical to the power of the overall system. Also, because their approach is for conventional single core embedded systems, it does not tackle the multicore issues either.

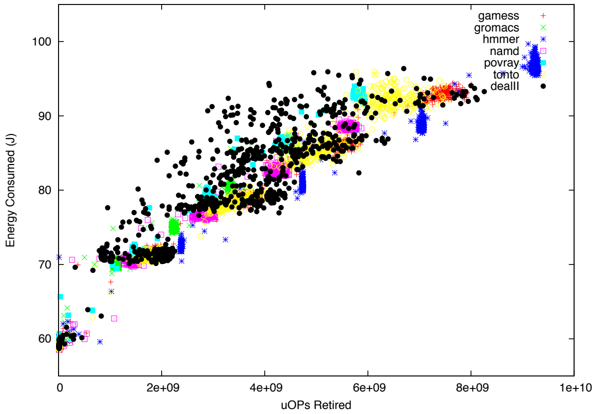

Figure 1. Power consumption depending on a varying number of retired μops-per-300 ms. under CPU and memory intensive workloads

III. Energy Accounting

In this paper, we will consider the dynamic power only, which is the manageable part by the virtual machine scheduler. From now on, "power" will be used to denote the dynamic power, unless otherwise stated.

A. Observation

We identified the relationships between each in-processor event count and processor power consumption through a series of experiments, and selected some meaningful relationships that will be used in our estimation model.

Figure 1 shows the average power consumption during 300 ms. time interval according to the number of retired micro-operations. The system run from one to four virtual machines simultaneously that execute the SPEC CPU2006 benchmark suite. Because the system equips a quad-core processor, up to four virtual machines are able to run together concurrently.

Micro-operations (μoperations) are a more fine-grained unit of execution in a processor than an instruction. Therefore, we believe that the number of retired micro-operations is likely to reflect the activities inside a processor more accurately than the number of retired instructions. Generally, the average power consumption in a time interval, which can be directly interpreted as the energy consumption of the interval, seems to be linearly proportional to the number of micro-operations.

In Figure 1, all workloads are CPU-intensive. *dealII* is a benchmark with relatively heavy last-level cache misses. In comparison to the workloads with infrequent memory accesses, more energy is consumed to execute the same number of μoperations of the workloads with frequent memory accesses. In order to access memory, processors have to stall for a while, and even during the stall cycles, the memory and bus systems continuously consume power. Consequently,

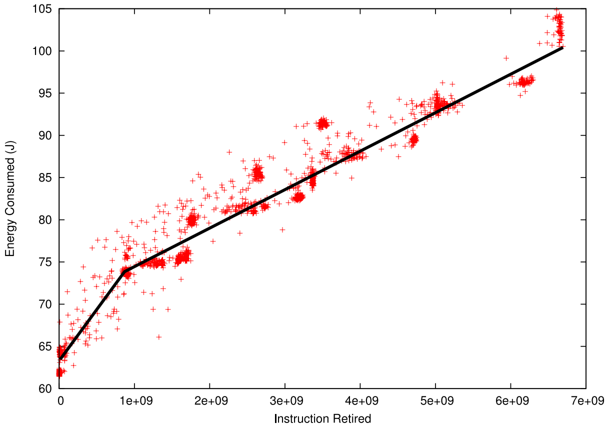

Figure 2. Power consumption depending on the number of μOps. while executing diverse workloads simultaneously

in spite of the same number of μoperations, the memory-intensive workloads tend to consume more power than the non memory-intensive workloads.

The number of working cores in a multicore processor heavily affects the processor power consumption. Although each core is a separated and independent execution unit, they share a lot of components such as on-chip caches, buses, memory controllers, and so on. Therefore, the dynamic power of a processor can be formulated as Equation 1. P_{shared} is the power consumption by the shared components, and P_{core} is the power consumption by each core that is executing instructions.

$$P_{processor} = P_{shared} + \Sigma P_{core} \tag{1}$$

Consequently, we expect that a processor requires less energy when instructions are spread and executed over multiple cores than when the same number of instructions are executed on a single core or smaller number of cores. The experiment results shown in Figure 2 verify our assumption.

Contrary to popular belief, the energy consumption differences between floating point operations and integer operations at μoperation-level do not show significant difference in spite of the huge difference of the complexity at the macro operation level. In our experiments, the correlation coefficient between the proportion of floating point instructions and the energy consumption of a unit time is about 0.087, which is an extremely loose relation.

B. Profiling and Estimation

Although there are a lot of events that affect the power consumption of a processor, a processor is able to keep track of generally three events at the same time.[10] Therefore, we set up our estimation model by using a few event counters that are the most closely related to processor power consumption.

Usually, for the sake of performance, a virtual processor of a virtual machine is committed to a processor core. When there are virtual machines from VM_1 to VM_n and virtual machine k consumes E_{VM_k} during a time interval, our aim is to estimate E_{VM_k} based on the event counts of the processor cores that run VM_k.

E_{system}, the overall energy consumption of the entire system from its dynamic power during a fixed time interval, is defined in Equation 2.

$$E_{system} = \sum_{i=1}^{n} E_{vm_i} \tag{2}$$

Because the performance counter values are recorded at every context switch, It is possible to keep track of the number of events that occurred during the time interval. If the power consumption is determined by the events, E_{vm_i} at time t will have the tendency as shown in Equation 3. N_i is the number retired instructions during the time interval, M_i is the number of memory accesses, and S_t is the number of active cores at time t. S_t depends on time t because the number of active cores changes over time.

$$E_{vm_i,t} \propto C_1 N_i + C_2 M_i - C_3 S_t \tag{3}$$

By substituting E_{vm_i} in Equation 2 for Equation 3, we obtain Equation 4.

$$E_{system,t} = \sum_{i=1}^{n} (C_1 N_i + C_2 M_i - C_3 S_t) \tag{4}$$

We obtain the coefficients C_1, C_2 and C_3 by conducting multi-variable linear regression over data sets sampled under diverse circumstances. The obtained coefficients are used to estimate the energy consumption of a virtual machine interval by substituting N_i, M_i and S_t in Equation 3 for the measured performance counter values.

Each core records its current performance counter values with the time stamp in an array data structure, `ec_priv`, at every context switch. At the end of every predefined time interval, usually 30 ms., the master core aggregates the `ec_priv` data of all cores and reconstructs the time series of the utilization of cores as illustrated in Figure 3. As well as the core activity time table, the number of the monitored events occurred between every consecutive two time points in 3 is calculated.

Obtaining the coefficients through linear regression requires the energy consumption value $E_{system,t}$ of every time interval t. We use an external digital multimeter data acquisition device to measure the system power.

With this approach, the data for the linear regression are collected and obtain the coefficients. Once the coefficients are obtained, the external measurement is no longer required to estimate the energy consumption. Opposite to the linear regression, the estimation process obtains $E_{system,t}$ by replacing N_I, M_i and S_t in Equation 4.

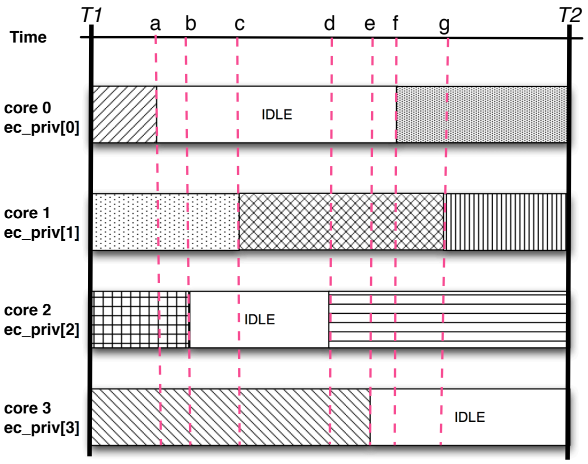

Figure 3. Detecting the number of active cores in fine time-granularity by using per-core array variables

Table I
SPECIFICATIONS OF THE SYSTEM FOR THE EVALUATION

Processor			
Num. of Cores	4	Clock Freq.	2.66 Ghz
System (approximate values)			
Peak Pwr.	332 W	Idle Pwr.	200 W

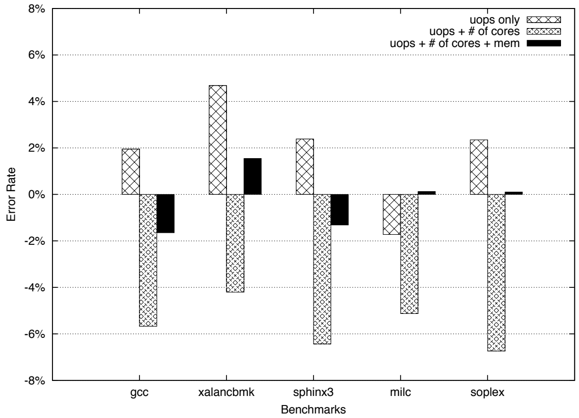

Figure 4. Power consumption estimation errors for homogeneous workloads when different variables are used

Table II
CONSTITUTION OF THE WORKLOAD MIXTURES

Set	Benchmarks
set1	gamess, GemsFDTD, calculix, gcc
set2	hmmer, bwaves, bzip2, xalancbmk
set3	gromacs, tonto, sphinx3
set4	h264ref, perlbench, milc
set5	dealII, wrf
set6	soplex, zeusmp

C. Evaluation

We implemented our estimation model in Xen 4.0 for evaluation. The hardware configuration and its characteristics used in our evaluation are introduced in Table I.

In our evaluation, we compared the aggregated values of estimated energy consumption of all virtual machines in a system with the measured energy consumption of the entire system because there is no practical method to measure the energy consumption of a virtual machine. Each virtual machine was configured to equip a single virtual processor and to run some benchmark programs chosen from the SPEC CPU2006 benchmark suite. We randomly chose six benchmark programs to sample data for conducting the linear regression to obtain the coefficients.

First, we measured the accuracy of our estimation model for the cases that the workloads of virtual machines are homogeneous. Every virtual machine was configured to execute the same benchmark program repeatedly. The number of concurrently running virtual machines varied from one to four depending on time.

Figure 4 show the average estimation errors of benchmarks. The left bars of each benchmark represents the error when only the number of retired instructions is considered. The middle bar represents the error when both the number of retired instructions and number of active cores are considered. The right bar shows the error when the memory access activities are considered together with the other two parameters. The errors are normalized to the total energy consumption, including the energy consumption from the leakage power and dynamic power.

In order to exclude the effects from resource sharing among multiple cores, we used only one virtual machine that utilize a single core when we collect data for the estimation model based only on the number of μoperations. When we applied the coefficients to the cases when multiple cores are in use, the energy efficiency enhancement from the shared processor components make our model overestimate the energy consumption. Due to this, the estimation accuracy improves dramatically when the model includes the active core number factor.

When the suggested model does not consider the memory access count, the model tends to underestimate the energy consumption of workloads. The μoperation count-only model shows better accuracy for the memory-intensive workloads than for the CPU-intensive workloads. However, this is because the underestimation tendency for the memory-intensive workloads offsets the overestimating property of the μoperation count-only model. The estimation model including the memory access count shows errors less than 2%.

We also evaluated the suggested scheme for the cases that multiple virtual machines with heterogeneous workloads run simultaneously. We randomly created six workload mixtures as listed in Table II.

The estimation errors for the workload mixture sets is illustrated in Figure 5. For all the workload sets the error

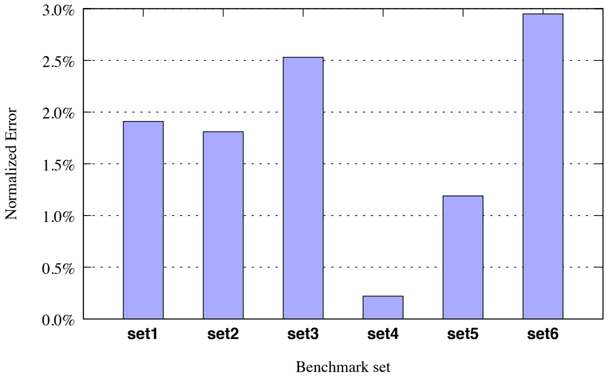

Figure 5. Normalized Errors of mixed workloads

rates are less than 3% and range from 0.5 Joule/s. to 7.0 Joule/s. Dividing the error range with the number of virtual machines yields the per-VM error range as much as from 0.5 Joule/s. to 3.5 Joule/s.

IV. Energy-Aware Scheduling

A. Energy-Credit Scheduler

The Credit Scheduler [11], which is the default virtual machine scheduler of Xen, is a kind of fair-share scheduler that distributes processor time to virtual machines according to their credit values. By modifying the Credit Scheduler, we suggest Energy-Credit Scheduler that schedules virtual machines according to their energy budgets instead of the processor time credits.

In the Energy-Credit Scheduler, each virtual machine is assigned its own energy credit at its own time interval, the energy fiscal interval. The unit of energy credit of a virtual machine is the *joule-per-fiscal interval*.

In the Credit Scheduler, the credit value of a virtual machine is interpreted as the guaranteed minimum processor time dedicated to the virtual machine. For the sake of throughput of the whole system, general schedulers including the Credit Scheduler employ work conserving scheduling that keeps the system busy by redistributing credit values when there are runnable tasks and none of them have available credit values.

However, the energy-credit of the Energy-Credit Scheduler works as the ceiling bounds of scheduling time, because the purpose of the Energy-Credit Scheduler is to limit the energy consumption rate of each virtual machine below its energy budget. When all runnable virtual machines use up their energy-credits, the system idles until the new fiscal interval of any runnable virtual machine begins. In other words, the Energy-Credit Scheduler is a non-work conserving scheduler.

The accounting function is executed every 30 ms. interval, which is T1-T2 in Figure 3. This time interval is different from the energy fiscal intervals of virtual machines, which may differ from each other's.

Table III
Capping errors of various workloads and capping configurations

Configurations (Joule/s.)				Measured Energy (Joule/s.)	Error Rate %
VM1	VM2	VM3	VM4		
leslie3d 20	astar 25	-	-	247.81	1.13
perlbench 20	sjeng 25	-	-	248.30	1.33
wrf 15	gobmk 20	sjeng 25	-	263.53	1.36
h264ref 15	calculix 20	perlbench 25	-	265.53	1.36
calculix 10	gobmk 15	sjeng 20	astar 25	273.93	1.46
h264ref 10	wrf 15	perlbench 20	leslie3d 25	274.19	1.55

At the end of every time interval, the accounting function subtracts the estimated consumed energy of a virtual machine during the interval from its remaining energy credit. If a virtual machine has no remaining credit, the scheduler takes the virtual machine out of the scheduler queue until it gets the credit again when the next fiscal interval of the virtual machine begins.

The energy consumption by a virtual machine during its fiscal interval is obtained by using the suggested energy consumption estimation model. The energy consumption estimation model in the Energy-Credit Scheduler utilizes all the three parameters.

The distribution of the energy credit to a virtual machine is done at the first execution of the scheduling algorithm after the new fiscal interval of the virtual machine begins. Therefore, the time granularity of energy credit provision is 30 ms.. When a virtual machine out of the scheduler queue earns energy credit, the scheduler put the virtual machine back in the scheduler queue.

B. Evaluation

The Energy Credit scheduler was implemented in the Xen hypervisor. The implemented scheduler was evaluated with the same environment introduced in Section III-C.

In order to evaluate the accuracy of energy provisioning, we used the workload configurations listed in Table III. One to four virtual machines run simultaneously and each of them executed a different workload with a different energy budget. Although a virtual machine is free to choose its own fiscal interval, all virtual machines in our experiment were set to have 1 s. energy fiscal interval for ease of analysis.

Like the estimation model evaluation, we compared the aggregated value of the designated energy consumption rate of all virtual machines with the measured energy consumption rate of the whole physical machine. The two rightmost columns in Table III show the measured energy consumption per a second and the energy provisioning error, respectively.

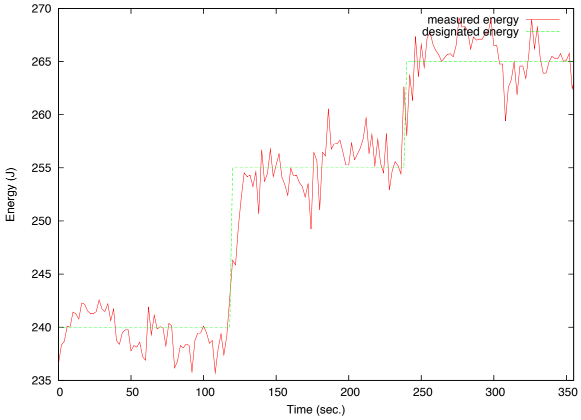

Figure 6. Time series of the designated energy consumption rate and actual energy consumption rate when the energy budget of four virtual machines dynamically changes (The fiscal interval is set to 1 s.)

According to the experiment results, the error rates are less than 2% of the total energy consumption. The errors are mostly positive values, which means that the measured energy consumption exceeds the energy budget. This is because our approach is reactive, not proactive. Our scheduler suspends a virtual machine only after it uses up its energy budget. Therefore, a virtual machine tends to consume little more energy than its budget. If the energy budget has to be strictly guaranteed, the algorithm must be improved to be proactive.

Figure 6 shows the time series of the designated energy budget changes and measure energy consumption rate. Four virtual machines ran concurrently while their energy budget changed dynamically.

The energy consumption rate rapidly followed the energy budget changes. However, on a short time scale such as 5 s., the difference between the designated energy provision and actual usage varied significantly. This is because the constitution of processor activities that affect the energy consumption rate changes according to the phases of the workloads. In spite of the short-term inaccuracy, our scheme provides the energy according to the energy budget in the long run.

V. Conclusion

In this paper, we proposed the energy consumption estimation model that can estimate the amount of energy consumption of each virtual machine without any dedicated measurement devices, and based on the estimation model, we suggested and implemented the energy-credit scheduler, which limits the energy consumption rate of each virtual machine below a user-defined budget.

The evaluation results showed that our model estimates the energy consumption with errors less than 5%. This error rate will be reduced as we refine our model by adding more event counters from other components such as NIC or disks, and refine the relationships between the parameters during the following research.

Acknowledgement

This work was supported by the MKE [KI002088, Development of Virtualization Technology based on Open Source Software].

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," UC Berkeley, Technical Report UCB/EECS-2009-28, 2009.

[2] E. Elmroth, F. G. Marquez, D. Henriksson, and D. P. Ferrera, "Accounting and billing for federated cloud infrastructures," in *Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing*, 2009.

[3] R. Kumar, "US Datacenters: The calm before the storm," Gartner White Paper, 2007.

[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, 2003.

[5] S. Bhadri, Ramakrishna, and S. Bhat, "Enhanced power monitoriing for Dell PowerEdge servers," *Dell Power Solutions*, vol. August, 2008.

[6] J. Jenne, V. Nijhawan, and R. Hormuth, "Architecture (DESA) for 11G rack and tower servers," Dell White Paper, 2009.

[7] A. Merkel, J. Stoess, and F. Bellosa, "Resource-conscious scheduling for energy efficiency on multicore processors," in *Proceedings of the 5th European conference on Computer systems*, ser. EuroSys '10, 2010, pp. 153–166.

[8] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing*, 2010.

[9] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, "ECOSystem: managing energy as a first class operating system resource," in *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, 2002, pp. 123–132.

[10] *Intel 64 and IA64 Architectures Software Developer's Manual Vol. 3B*. Intel Corporation, 2011, ch. 30.

[11] E. Ackaouy, "The Xen Credit CPU scheduler," in *Proceedings of 2006 Fall Xen Summit*, 2006.