저작자표시 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

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

A large, light gray watermark of the UNIST logo is centered in the background. It features a circular emblem with a stylized 'U' and 'I' inside, surrounded by the text 'UNIST NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY'. Below the emblem is a trapezoidal shape with the word 'UNIST' inside.

Nak-ku Kim

Computer Systems and Network Major
School of Electrical & Computer Engineering
Graduate school of UNIST

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

A thesis

submitted to the School of Electrical and Computer Engineering
and the Graduate School of UNIST

in partial fulfillment of the
requirements for the degree of
Master of Science

Nak-ku Kim

20. 6. 2011

Approved by

Major Advisor

Euiseong Seo

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

Nak-ku Kim

This certifies that the thesis of Nak-ku Kim is approved.

20. 6. 2011

Thesis Supervisor: Euseong Seo

Jongeun Lee

Younho Lee

Abstract

Virtualization enables flexible resource provisioning and improves energy efficiency through consolidating virtualized servers into a smaller number of physical servers than that of the virtualized servers. Therefore, it is becoming an essential component for the emerging cloud computing model. Currently, virtualized environment including cloud computing systems bills users for the amount of their processor time, or the number of their virtual machine instances. However, accounting based only on the depreciation cost of server hardware is not an economically proper model because the cooling and energy cost for datacenters has already exceeded the cost to own servers. This paper suggests an estimation model to account energy consumption of each virtual machine without any dedicated measurement hardware. Our estimation model estimates the energy consumption of a virtual machine based on the in-processor events generated by the virtual machine. Based on the estimation model, this paper also proposes the virtual machine scheduling algorithm that is able to provide computing resources according to the energy budget of each virtual machine. The suggested schemes are implemented in the Xen virtualization system, and the evaluation shows the suggested schemes estimate and provide energy consumption with errors less than 5% of the total energy consumption.

Contents

I . Introduction -----	1
II . Motivation and Related Works-----	5
2.1 Motivation -----	5
2.2 Related Works -----	8
III. Energy Accounting -----	12
3.1 Observation -----	12
3.2 Profiling and Estimation -----	18
3.3 Evaluation -----	21
IV. Energy-Aware Scheduling -----	27
4.1 Energy-Aware Scheduler -----	27
4.2 Evaluation -----	30
V. Conclusion -----	33
Acknowledgement-----	35

List of figures

Figure 2-1. Consolidation of heterogeneous virtual machines in a cloud node

Figure 3-1. Power consumption depending on varying number of retired

Figure 3-2. Power consumption depending on varying number of retired

Figure 3-3. Power consumption depending on the number of active cores and micro-ops. while executing *gromacs*

Figure 3-4. Power consumption depending on the number of micro-Ops. while executing diverse workloads simultaneously

Figure 3-5. Deadlocks may occur when cores call IPIs at every context switch

Figure 3-6. Detecting the number of active cores in fine time-granularity by using per-core array variables

Figure 3-7. Power consumption estimation errors of profile group

Figure 3-8. Power consumption estimation errors of CPU-intensive group when different variables are used

Figure 3-9. Power consumption estimation errors of Mem-intensive group when different variables are used

Figure 3-10. Normalized Errors of mixed workloads

Figure 3-11. Estimated values and measured values of power consumption during execution of diverse workloads

Figure 4-1. Flow chart of power-aware scheduling scheme implementation

Figure 4-2. 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 sec.)

List of tables

Table 3-1. Average number of last level cache misses of each workload during a 30 ms. execution interval

Table 3-2. Specifications of the system for the evaluation

Table 3-3. Benchmark applications used for experiments (* are benchmarks used for obtaining the coefficients)

Table 3-4. Workloads mixes and estimated power consumptions for each VMs

Table 4-1. Capping errors of various workloads and capping configurations

Nomenclature

I. INTRODUCTION

The energy cost of datacenters has been increasing rapidly.

By 2012, it will become as much as twice of that in 2007 and it will exceed the cost to own the servers. (Kumar, 2007) In addition to the energy cost, the cooling cost is also increasing fast because the ever-increasing server power consumption induces escalating heat dissipation. Under this circumstances, the cloud computing model is expected to become the next generation enterprise computing infrastructure due to its energy efficiency.

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. (Armbrust et al., 2009) Because services and applications of users run in separate virtual machines, virtualization helps cloud platforms to accurately monitor and control the amount of resources being provided to each virtual machine. Because of that, virtualization also enables the guarantee of service-level agreements (SLAs) between users and cloud platform providers by resolving QoS crosstalks between services or applications. Moreover, the separation of both the execution context and data between virtual machines are supported by processors as well as hypervisors. Therefore, the users use their virtual machines as they own their dedicated server hardware. Due to these benefits, the a lot of commercial cloud systems including Amazon EC2 employ virtualization so that the users can freely configure their virtual servers from the kernel layers to the application layers.

Currently, these cloud services bill their users for the amount of resources being provided to the users. (Elmroth et al., 2009) In general, the amount of processor time or the number of allocated virtual machines are common criteria for accounting resource provisioning. (Armbrust et al., 2009) 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, if there are unconsidered significant factors that affect the management cost of servers, these billing schemes may not be fair and have room to be improved. As stated earlier, the energy and cooling cost will exceed the hardware cost. Therefore, in order to be fair, the billing system must imply the accounting of energy consumption by each user.

However, measuring 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.

First, most of servers do not equip real-time power measurement devices. Although some servers employ realtime power measurement components, the in-server power measurement devices sampling interval length significantly longer than the virtual machine scheduling interval length. Also, the in-server power monitors measures the power consumption of the entire server system. Therefore, it is hard to identify how much proportion each virtual machine contributes to the system power consumption.

The emergence and popularity of multicore processors enable multiple virtual machines to run simultaneously in a physical server. In a multicore processor system, significant differences in power consumption and performance may arise due to the resource sharing and contention. For example, by sharing large on-chip cache system, the energy efficiency improves when multiple virtual machines run simultaneously without resource contention such as conflict cache misses between different virtual machines. Because it is infeasible to measure the energy consumption characteristics of each virtual machine depending on the other virtual machines in the system in advance, the energy consumption of each virtual machine must be estimated based on realtime behavior analysis of each virtual machine.

The existing research result already approached the energy consumption accounting at the virtual-machine granularity by accounting and analyzing the amount of resources each virtual machine is using. However, these models assumed that the power consumption of each virtual machine is simply proportional to the length of the allocated processor time or the number of processed network packets.

In this paper, we reveal that the processor time is not sufficient to accurately estimate the power consumption of a processor, and that some in-processor events affect the processor power consumption significantly. Based on the observation made by diverse experiments, 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 the scheduling decision. Different from the traditional counterparts, the energy-credit scheduler employ 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.

Both the suggested estimation model and scheduler are implemented in the Xen virtual machine monitor. The implementation also provides the external interfaces so that they can be easily adopted to the real world systems. We evaluate our implementation 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, and based on the estimation model we propose the energy-credit scheduler as well as its implementation and evaluation in Section 4. We conclude in Section 5.

REFERENCES

1. ARMBRUST, M., FOX, A., GRIFFITH, R., JOSEPH, A. D., KATZ, R. H., KONWINSKI, A., LEE, G., PATTERSON, D. A., RABKIN, A. & STOICA, I. 2009. Above the clouds: A berkeley view of cloud computing. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28*.
2. KUMAR, R. 2007. US data centers: The calm before the storm. *Gartner RAS Core Research Note G*.
3. ELMROTH, E., MARQUEZ, F. G., HENRIKSSON, D. & FERRERA, D. P. Year. Accounting and billing for federated cloud infrastructures. *In*, 2009. Ieee, 268-275.

II. MOTIVATION AND RELATED WORK

2.1 Motivation

The amount of energy consumed by a server varies greatly depending on the workload it runs. A lot of existing studies for prediction or control of the power consumption of servers are based upon an assumption that the the server's power consumption is determined mainly by processors, and that the power consumption of a processor is determined primarily by the processor utilization. (Govindan et al., 2009, Lefurgy et al., 2007, Zeng et al., 2002) Because the power consumption of a processor largely depends on the workload characteristics (Govindan et al., 2009), however, the estimation of energy consumption based on the processor utilization or the execution time is not generally accurate enough for the billing basis of energy cost.

Recently, vendors integrate power measurement hardware, which can monitor the server's power consumption on the fly, into servers. (Bhadri et al., 2008) By measuring power consumption via the power measurement hardware, energy consumption of servers can be measured accurately. However, the integrated power meter is able to measure only the power consumption of the whole server system while the energy-based billing system of a cloud system requires the measurement of energy consumption by each virtual machine because a virtual machine is the unit of billing in a cloud system. In spite of the dedicated measurement hardware as well as the existing research results, 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 second (Jenne et al., 2009) while the unit of scheduling virtual machines is very short from a few hundred sec. to a few tens of msec. Considering the difference between the time scales of measurement and scheduling, measuring the amount of energy consumption by a specific virtual machine with the integrated power meter or other general-purpose power meter is difficult.

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 simultaneously together. (Merkel et al., 2010) Therefore, measuring energy consumption of a virtual machine should not be blind arithmetic that only considers processing time of the virtual machine.

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 effects from the other virtual machines is necessary.

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.

Most of conventional virtual machine schedulers aim at providing processor-time to each virtual fairly and proportionally to its priority because processor time is one of the major billing criteria of conventional computing systems.

Figure 2-1. Consolidation of heterogeneous virtual machines in a cloud node

In other words, processor-time is the major scheduling basis in the conventional schedulers. (Ackaouy, 2006, Cherkasova et al., 2007). Therefore, a novel energy-aware scheduler as well as its notion should be designed from the scratch.

In order to utilize the energy-aware scheduling 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, and may range from a few tens millisec. to a few minutes. The fiscal interval should be carefully chosen because it may affect the throughput and responsiveness of virtual machines. When the interval is too short, excessively frequent context switches may occur. On the contrary, if the interval is too long, the suspension time of the virtual machine may become too long and, therefore, may harm the responsiveness of the services given by the virtual machine.

For example, as shown in Figure 2-1, when two virtual machines that are respectively for a web service and a data backup service run simultaneously together in a physical server, it is beneficial for the web service virtual machine to have a short-term plan such as 300mWh/0.5sec. and for the backup service virtual machine to have a long-term plan such as 18000mWh/1 min.

In order to realize this conceptual model, we suggest the estimation model that estimates the energy consumption of each virtual machine in a multicore server. The estimation model is aimed at providing accuracy sufficient for the billing basis of cloud computing services. In addition to that, we propose the energy-aware virtual machine scheduler that schedules virtual machines according to their energy budgets.

2.2 Related Work

A statistical method has been suggested for predicting the power consumption of a server with multiple consolidated virtual machines. (Choi et al., 2008) Based on this approach, Soft Fuse has been suggested. (Govindan et al., 2009) Similar to fuses in electric circuits, Soft Fuse limits the power consumption of a server by throttling the processor performance when the power consumption exceeds the predefined threshold. Their approaches, however, assume that the target system equips a single core processor, and exclude the interference and resource contention in their models. Therefore, they are not applicable to modern multicore systems.

The energy efficient virtual machines-to-physical server mapping method has been suggested. (Srikantaiah et al., 2008) The suggested mapping scheme observes the changes in energy efficiency and performance according to the different consolidation cases, and finds the energy-efficient consolidation mapping based on the observation. Although the energy efficient virtual machines-to-physical servers mapping improves the overall energy efficiency of the cloud systems, it does not count the energy efficiency of each virtual machine.

ECOSystem (Zeng et al., 2002), 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.

Joule meter (Kansal et al., 2010) 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 energy consumption estimation models.

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. When a new virtual machine is created, the system observes the changes in the power consumption patterns by measuring for the first 200 sec. This observation yields the power consumption characteristics of the new virtual machine after statistical processing such as linear regression. By using the obtained characteristics of the virtual machine's power consumption pattern, the second model estimates the power consumption of the virtual machine more accurately. 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 behavior of the virtual machine remains the same as that of its first 200 sec.

An accurate processor power consumption model was suggested. (Snowdon et al., 2009) The processor power consumption model estimates the power consumption of a processor by monitoring its internal activities such as memory accesses, integer or floating point arithmetic operations and so on. However, their target processor is a single core embedded processor and their model does not consider the multicore effects either. Other power estimation models based on the observation of the interval activities have been suggested and shown their accuracy. (Sinha and Chandrakasan, 2001, Tiwari et al., 1996) Also, based on this approach, some system simulators estimate the power and energy requirements of applications. (Brooks et al., 2000, Stanley-Marbell and Hsiao, 2001)

REFERENCES

1. ACKAOUY, E. Year. The Xen Credit CPU scheduler. *In: Proceedings of 2006 Fall Xen Summit*, 2006.
2. BHADRI, S., RAMAKRISHNA & BHAT, S. 2008. *Enhanced power monitoring for Dell PowerEdge servers*, Dell Power Solutions.
3. BROOKS, D., TIWARI, V. & MARTONOSI, M. Year. Wattch: a framework for architectural-level power analysis and optimizations. *In*, 2000. ACM, 83-94.
4. CHERKASOVA, L., GUPTA, D. & VAHDAT, A. 2007. Comparison of the three CPU schedulers in Xen. *Performance Evaluation Review*, 35, 42.
5. CHOI, J., GOVINDAN, S., URGANONKAR, B. & SIVASUBRAMANIAM, A. Year. Profiling, prediction, and capping of power consumption in consolidated environments. *In*, 2008. IEEE, 1-10.
6. GOVINDAN, S., CHOI, J., URGANONKAR, B., SIVASUBRAMANIAM, A. & BALDINI, A. Year. Statistical profiling-based techniques for effective power provisioning in data centers. *In*, 2009. ACM, 317-330.
7. JENNE, J., NIJHAWAN, V. & HORMUTH, R. 2009. *Architecture (DESA) for 11G rack and tower servers*, Dell White Paper.
8. KANSAL, A., ZHAO, F., LIU, J., KOTHARI, N. & BHATTACHARYA, A. A. Year. Virtual machine power metering and provisioning. *In*, 2010. ACM, 39-50.
9. LEFURGY, C., WANG, X. & WARE, M. Year. Server-level power control. *In*, 2007. IEEE, 4-4.
10. MERKEL, A., STOESE, J. & BELLOSA, F. Year. Resource-conscious scheduling for energy efficiency on multicore processors. *In*, 2010. ACM, 153-166.
11. SINHA, A. & CHANDRAKASAN, A. P. Year. Jouletrack-a web based tool for software

- energy profiling. *In*, 2001. IEEE, 220-225.
12. SNOWDON, D. C., LE SUEUR, E., PETTERS, S. M. & HEISER, G. Year. Koala: A platform for OS-level power management. *In*, 2009. ACM, 289-302.
 13. SRIKANTAIAH, S., KANSAL, A. & ZHAO, F. Year. Energy aware consolidation for cloud computing. *In*, 2008. USENIX Association, 10-10.
 14. STANLEY-MARBELL, P. & HSIAO, M. Year. Fast, flexible, cycle-accurate energy estimation. *In*, 2001. ACM, 141-146.
 15. TIWARI, V., MALIK, S., WOLFE, A. & TIEN-CHIEN LEE, M. 1996. Instruction level power analysis and optimization of software. *The Journal of VLSI Signal Processing*, 13, 223-238.
 16. ZENG, H., ELLIS, C. S., LEBECK, A. R. & VAHDAT, A. 2002. ECOSystem: managing energy as a first class operating system resource. *ACM SIGPLAN Notices*, 37, 123-132.

III. ENERGY ACCOUNTING

The power consumption of a processor can be divided into the leakage power and dynamic power. (Seo et al., 2008) The leakage power of a processor is steadily drained away while electricity is supplied to the processor, and the dynamic power is consumed only when the processor is executing instructions.

The amount of the leakage power is always the same as long as the system is activated. Therefore, it cannot be the target of accounting nor provisioning. 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.

3.1 Observation

There are a lot of factors that may affect the power consumption of a processor. The power is consumed by components executing instructions. It thus is expected that the energy consumption, which is the multiplication of the power and time, is proportional to the number of retired instructions. However, even the same number of retired instructions may induce the different amount of energy consumption depending on the types of the instructions. For example, memory access operations such as load and store use different set of execution units from that for arithmetic operations.

We conduct some experiments to observe that to what degree diverse in-processor events affect the processor power consumption.

Most of modern processors provide programmable performance counters. Each performance counter is a special register that is programmable to count the occurrences of a certain kind of activity in a processor like cache misses, branch prediction failures, retired floating point instructions, and so on. For example, the Intel Nehalem architecture processors provide four programmable performance counters. (2011)

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

A time tick of the Xen hypervisor, which is a minimum scheduling unit, is 10 ms. (Barham et al., 2003) Consequently, when CPU-intensive workloads are being executed, the Xen hypervisor can read the performance counters once a 10 msec. However, because the Xen hypervisor allows the same virtual machine to be scheduled up to 30 msec. without intervention, we use a 30 msec. interval, which consists of 3 consecutive time ticks, as the time unit of estimation.

30 msec. is too short for a measurement device to measure the power consumption of a system. Thus, in order to set up the estimation model, we measure the power consumption changes depending on the performance counter values at every 300 ms. interval.

Figure 3-1. Power consumption depending on varying number of retired

Figure 3-1 shows the average power consumption during 300 msec. time interval according to the number of retired micro-operations. The system run from one to four virtual machines simultaneously that executes the SPEC CPU2006 benchmark suite. The system equips a quad-core Intel i5 750 processor. Thus up to four virtual machines are able to run together concurrently.

Micro-operations 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.

Even within the same workloads, the number of retired micro-operations during a time interval varies greatly as time flows, and this tendency results in the fluctuation in the power consumption patterns. Generally, the average power consumption in the 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.

The workloads used in our experiments of Figure 3-1 have infrequent memory accesses in common. The results with an additional workload with frequent memory accesses are shown in Figure 3-2.

Figure 3-2. Power consumption depending on varying number of retired

deall is a benchmark with heavy last-level cache misses as shown in Table 3-1. When last-level cache misses are frequent, operations per a sec. tends to be lower than that with infrequent last-level cache misses. However, in comparison to the workloads with infrequent memory accesses, more energy is consumed to execute the same number of operations 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 to operate with. Consequently, in spite of the same number of _operations, the memory-intensive workloads tend to consume more power than the non memory-intensive workloads.

Workload	Avg. LLC misses	Workload	Avg. LLC misses
povray	1,476	gamess	1,615
tonto	2,042	hammer	23,585
gromacs	27,251	namd	30,948
deallI	526,830		

Table 3-1. Average number of last level cache misses of each workload during a 30 ms. execution interval

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.

$$P_{processor} = P_{shared} + \sum P_{core} \quad (1)$$

Consequently, we expect that a processor requires smaller 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 3-3 verify our assumption.

Figure 3-3. Power consumption depending on the number of active cores and micro-ops. while executing *gromacs*

Figure 3-3 shows the relationship between the power consumption and the number of retired operations when one to four cores are simultaneously used. The dots in the graph are grouped around four small regions according to the number of active cores. The difference in the power consumption between one and two, two and three, or three and four cores are less than the power consumption of only one active core. As we expected, this is because the shared components barely consume additional power when more cores are put into operation.

This tendency appears also when diverse heterogeneous workloads are concurrently executed together. Figure 3-4 shows the power consumption while multiple number of the virtual machines that run different benchmarks. Using more cores induces more micro-operations. Also, using more cores result in better energy efficiency due to the less steep inclination of power consumption in comparison to that of a single core.

Figure 3-4. Power consumption depending on the number of micro-Ops. while executing diverse workloads simultaneously

Contrary to popular belief, the energy consumption difference between floating point operations and integer operations at micro-operation-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 extremely loose relation. This is because the floating point instructions tend to be translated into more number of micro-operations than the non-floating point instructions. Based on this observation, our model does not categorize the instruction type.

3.2 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. {, 2011 #362} Therefore, we set up our estimation model by using few event counters that are most closely related to the 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 as Equation 2.

$$E_{system} = \sum_{i=1}^n E_{VM_i} \quad (2)$$

Because the performance counter values are recorded at every time context switch, it is possible to keep track of the number of events 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.

$$E_{VM_t,t} \propto C_1 N_i + C_2 M_i - C_3 S_t \quad (3)$$

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

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

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_i in Equation 3 for the measured performance counter values.

Figure 3-5. Deadlocks may occur when cores call IPIs at every context switch

The performance counters are read at every context switch to keep track of their changes. In order to read performance counters of other cores, a core issues inter-processor interrupts to the other cores so that they jump to the code that read their performance counters. However, if every core is allowed to invoke the inter-processor interrupts at any time, deadlock may occur when two or multiple cores invoke the inter-processor interrupts simultaneously as shown in Figure 3-5.

In order to prevent this catastrophic situation, our approach employs a tweak for collecting time series of 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 300 ms., the master core aggregate the `ec_priv` data of all cores and reconstruct the time series of the utilization of cores as illustrated in Figure 3-6. As well as the core activity time table, the number of the monitored events occurred between every consecutive two time

points in Figure 3-6 is calculated.

Figure 3-6. Detecting the number of active cores in fine time-granularity by using per-core array variables

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.

3.3 Evaluation

We implement our estimation model in Xen 4.0 for evaluation. The hardware configuration and its characteristics used in our evaluation are introduced in table 3-2.

Processor			
Num. of Cores	4	L2 Cache	256 KB per core
Clock Freq.	2.66 Ghz	L3 Cache	8 MB per processor
ISA	X86_64	TDP	95 W
System (approximate values)			
Peak Pwr.	332 W	Idle Pwr.	195 W

Table 3-2. Specifications of the system for the evaluation

In order to collect the power values, we use a digital multimeter (DMM), National Instrument PXI 4070, with PXI data acquisition interfaces. The DMM is set to sample the power input to the target server at every 300 msec.

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 is configured to equip a single virtual processor and to run some benchmark programs chosen from the SPEC CPU2006 benchmark suite. Because there are four physical cores in the system, up to four virtual machines can run concurrently without sharing cores.

	Benchmark applications
CPU-intensive	gamess*, calculix*, namd*, tonto*, gromacs*, h264ref, hmmer, bzip2, perbench, wrf, dealII*
Mem-intensive	gcc, xalancbmk, sphinx3, milc, soplex
Non-resource saturating	GemsFDTD, bwaves, zeusmp

Table 3-3. Benchmark applications used for experiments
(* are benchmarks used for obtaining the coefficients)

The benchmark set for our evaluation is listed in Table 3-3. The benchmark programs are categorized into three groups. The CPU-intensive group consists of the programs that have relatively infrequent memory accesses or last-level cache misses. They show high micro-ops./sec. in common. The benchmark programs in the memory-intensive group frequently access memory and have low last-level cache hit ratio. Both the CPU- and memory-intensive groups are of resource saturating workloads. In other words, the processor stays busy and does not idle while the workloads are in execution. The workloads of the last group, the non-resource saturating group, contain a lot of disk accesses. The processor usually idles while the workloads wait for the disk access to finish.

If our assumption that the interval activities of processors directly induce the energy consumption is correct, the selection of workloads for obtaining the coefficients in our model will barely affect the estimation accuracy. We randomly chose six benchmark programs to sample data for conducting the linear regression to obtain the coefficients, and the benchmark programs are marked in Table 3-3.

First, we measure the accuracy of our estimation model when the workloads of virtual machines are homogeneous. In this experiment, every virtual machine is configured to execute the same benchmark program repeatedly. The number of concurrently running virtual machines varies from one to four depending on time.

Figure 3-7, 3-8 and 3-9 show the average estimation errors of benchmarks which are belonged to each group according to the number of parameters used for the estimation, respectively. The left bars of each benchmark represent 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

Figure 3-7. Power consumption estimation errors of profile group when different variables are used

Figure 3-8. Power consumption estimation errors of CPU-intensive group when different variables are used

Figure 3-9. Power consumption estimation errors of Mem-intensive group when different variables are used

As shown in Figure 3-3 some components are shared between cores and the power consumed by these components are not negligible. Thus, when accounting power consumption of CPU in per-core basis we have to take the power consumed by these shared components account in. By counting the number of active cores we could estimate the power consumed by shared components and the errors are dramatically decreased on profile and CPU-intensive groups.

As shown in Figure 3-9 when the suggested model does not consider the memory access count, the model tends to underestimate the energy consumption of workloads. The micro-operation 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 micro-operation 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 measured total power consumption and compared it with estimated total power consumption using our estimation model for several sets of benchmarks. Obviously, there is no physical power meter which can measure the power consumption of a VM.

Only the total power consumption can be measured. However we can expect that if all the power consumptions of each VM are estimated correctly, the sum should be matched to the measured total power consumption.

Set	Benchmarks	Est. power
set1	gamess(24), GemsFDTD(4), calculix(24), gcc(29)	282.7480
set2	hmmer(22), bwaves(2), bzip2(7), xalancbmk(34)	265.9609
set3	gromacs(25), tonto(21), sphinx3(27)	266.7107
set4	h264ref(26), perlbench(23), milc(14)	263.9729
set5	dealII(26), wrf(21)	240.4443
set6	soplex(31), zeusmp(8)	237.7680

Table 3-4. Workloads mixes and estimated power consumptions for each VMs

Figure 3-10. Normalized Errors of mixed workloads

Table 3-4 shows 6 combinations of benchmarks which run simultaneously at each experiment. On the right of benchmark names, estimated power consumption of a respective VM is shown and on the last column estimated total power consumptions are shown.

The estimation errors for the workload mixture sets is illustrated in Figure 3-10. For all the workload sets the error rates are less than 3% whose range is 0.5Watts to 7.0Watts. By dividing its number of VMs which ran simultaneously we expect 0.1Watts to 3.5Watts to be the per-VM error range.

Figure 3-11. Estimated values and measured values of power consumption during execution of diverse workloads

Figure 3-11 shows some portion of time series of measured power consumption and estimated power consumption by using proposed power model while running randomly generated workloads with benchmarks listed in table 3-3. Red line represents the measured power consumption and blue dotted line represents estimated power consumption. As in the graph, on most cases our model estimates power consumption with low errors despite the existence of drastic power consumption changes. The average error on this experiment is 0.96 watts.

IV. Energy-Aware Scheduling

4.1 Energy-Credit Scheduler

The Credit Scheduler, 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, similar to the Credit Scheduler, each virtual machine is assigned its own energy credit at its own time interval, the energy fiscal interval. The fiscal interval of a virtual machine is provided by users and should be determined depending on the characteristics of the virtual machine. A fiscal interval that is too short will induce excessively frequent context switches, and too long interval may cause the virtual machine to suspend for significant time. Because the time slice of the virtual machine scheduling is as long as 30 ms., the fiscal interval must be longer than 60 ms., which is the twice of the time slice. The unit of energy credit of a virtual machine is the *joule-per-fiscal interval*.

In the conventional Credit Scheduler, the processor time dedicated to a virtual machine is determined by the proportion of the credit of the virtual machine to the sum of all virtual machine credit values. Therefore, 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.

Figure 4-1. Flow chart of power-aware scheduling scheme implementation

The scheduling algorithm is illustrated in Figure 4-1. The left hand side shows the flow chart of the scheduling algorithm and the right hand side shows the corresponding scheduler function names. This algorithm is executed every 30 ms. interval, which is T1-T2 in Figure 3-6. This time interval is different from the energy fiscal intervals of virtual machines, which may differ from each other's.

At the end of every time interval, `do_burn_energy` 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.

4.2 Evaluation

The Energy Credit scheduler is implemented in the Xen hypervisor. The implemented scheduler is evaluated with the same environment introduced in Section 3.3.

In order to evaluate the accuracy of energy provisioning, we use the workload configurations listed in Table 4-1. One to four virtual machines run simultaneously and each of them executes 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 are set to have 1 sec. energy fiscal interval for ease of analysis.

(J / sec)

Configurations					Measured Energy	Errors
VM1	VM2	VM3	VM4	Total Energy		
20 leslie3d	25 astar			250	247.81	2.19
20 perlbench	25 sjeng			250	248.30	1.70
15 wrf	20 gobmk	25 sjeng		265	263.34	1.66
15 h264ref	20 calculix	25 perlbench		265	265.53	0.53
10 calculix	15 gobmk	20 sjeng	25 astar	275	273.93	1.17
10 h264ref	15 wrf	20 perlbench	25 leslie3d	275	274.19	0.81

(Idle Power = 205 Watt)

Table 4-1. Capping errors of various workloads and capping configurations

Like the estimation model evaluation, we compare 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 4-1 show the measured energy consumption per a second and the energy provisioning error, respectively.

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 4-2. 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 sec.)

Figure 4-2 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 sec., 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

The use of cloud computing model is spreading like wild fire due to the energy efficiency improvement and infrastructure cost reduction. The conventional cloud computing billing systems are based upon that of the traditional multi-user server systems, such as the allocated processor time or the number of virtual machine instances. However, considering the circumstances that the energy cost exceeds the hardware cost in datacenters, this billing system should be modified to reflect the energy cost.

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 the estimation model estimates the energy consumption with the errors less than 5%. This error rate is expected to be reduced as we refine our model by adding parameters, such as the clock frequency changes from using DVFS (dynamic voltage and frequency scaling), and reform the relationships between the parameters during the following research.

The suggested schemes can be used as billing bases for the cloud systems, or can enable the cloud systems to provide computing resources to virtual machines according to their energy budgets.

Our research dealt only with the processor energy consumption. Although the processor is the biggest energy consumer in a server system, diverse components other than processors including storage devices and network interface cards should be taken into consideration in the estimation model because many cloud applications such as web servers or database management systems are I/O intensive workloads. Therefore, the accurate energy consumption estimation models for those kinds of components are required to set up the energy-based billing system.

We are currently working on these issues to build the holistic energy-aware resource provisioning schemes for the cloud systems.

REFERENCES

1. 2011. *Intel 64 and IA64 Architectures Software Developer's Manual*, Intel Corporation.
2. BARHAM, P., DRAGOVIC, B., FRASER, K., HAND, S., HARRIS, T., HO, A., NEUGEBAUER, R., PRATT, I. & WARFIELD, A. Year. Xen and the art of virtualization. *In*, 2003. ACM, 164-177.
3. SEO, E., JEONG, J., PARK, S. & LEE, J. 2008. Energy efficient scheduling of real-time tasks on multicore processors. *Parallel and Distributed Systems, IEEE Transactions on*, 19, 1540-1552.

Acknowledgements

My utmost gratitude goes to my thesis advisor, Dr. Euiseong Seo for his guidance, kindness and especially wholehearted supports so that, I could focus only on my own jobs. I would also like to express my appreciation to my thesis committee members: Dr. Jongeun Lee and Dr. Younho Lee. I thank to Dr. Beomseok Nam for his guidance and advices on general academic pursuits.

My gratitude also goes to all the CSE colleagues. Special thanks to Jung wook Cho for his help on writing this thesis. Besides, your hardworking attitude greatly inspired me. Suntae, I owe you by leaving the future work on you. I believe this work can be a really good starting point of your research. Jinwoong and Toan, I wish you a good luck on your research. It was a great pleasure to discuss about cloud computing.

Last but not least, all of my family, without your support and trust I would not be able to accomplish this.

Thank you.