An Efficient Power-aware Resource Scheduling Strategy in Virtualized Datacenters

Yazhou Zu, Tian Huang, Yongxin Zhu
School of Microelectronics
Shanghai Jiao Tong University
800 Dongchuan Road, Shanghai, 200240, China
{memory_zu, zhuyongxin}@sjtu.edu.cn, huangtian@ic.sjtu.edu.cn

***Abstract*—In the era of cloud computing, datacenters are well-known to be bounded by the power wall issue. This issue lowers the profit of service providers and obstructs the expansions of datacenter's scale. As virtual machine's behavior was not explored sufficiently in classic datacenter's power-saving strategies, in this paper we address the power consumption issue in the setting of a virtualized datacenter. We propose an efficient power-aware resource scheduling strategy that reduces datacenter's power consumption effectively based on VM live migration which is a key technical feature of cloud computing. Our scheduling algorithm leverages the Xen platform and consolidates VM workloads periodically to reduce the number of running servers. To satisfy each VM's service level agreements, our strategy keeps adjusting VM placements between scheduling rounds. We developed a power-aware datacenter simulator to test our algorithm. The simulator runs in time domain and includes server's segmented linear power model. We validated our simulator using measured server power trace. Our simulation shows that compared with event-driven schedulers, our strategy improves datacenter power budget by 35% for random workloads resembling web-requests, and improve datacenter power budget by 22.7% for workloads exhibiting stable resource requirements like ScaLAPACK.**

Keywords*—*datacenter power consumption; virtual machine; cloud computing; resource provisioning; server power model; datacenter simulator

I. Introduction

Datacenter power consumption is a significant issue for internet service providers. Large-scale datacenter's power budget increases service providers' operational costs and reduces their business profit. Moreover, datacenters are a major contributor to the global warming effect [1]. Based on the two reasons above, reducing datacenter's power cost while preserving datacenter's compute capability is a highly profitable task for datacenter owners.

Nowadays, the rapid development of the cloud computing paradigm raised new frontier for this issue. Consumer's expanding demand for Internet-scale applications are forcing cloud providers to build more large-scale cloud datacenters, aggravating datacenter's power usage inefficiency. However, the wide adoption of virtualization technology [2, 3] in cloud datacenters enables datacenter operators to manage resources more efficiently. Through partitioning real servers into multiple virtual servers and live migrating entire VMs between physical hosts [4], datacenter resources are managed more conveniently compared with traditional process-level management. As a result, in cloud datacenters computing resources can be effectively managed with the goal of reducing the total power consumption of IT infrastructures. In our work, we address the power issue by scheduling computing resources to reduce datacenter's power budget in a virtualized environment.

Although there have been efforts in reducing datacenter's power consumption, modern cloud's VM situation is not taken into consideration sufficiently. In [5, 6] the authors studied power reduction in a HPC environment. However, their approach cannot be applied to general virtualized datacenters with mixed workload settings. In [7] connection-intensive workloads are considered for power budget refinement, but their work does not occur in a virtualized environment. [8] focused on resource management in virtualized datacenters. Their work targeted at improving load balancing instead of reducing power consumption. In [9] a power-aware resource management strategy is proposed in cloud datacenters. This strategy modifies VM placements only when VMs are created or destructed and have room for further power reduction. Compared with the works above, our scheduling strategy extracts the feature of virtualized datacenters more sufficiently, and is able to reduce datacenter power consumption more effectively.

Our work makes the following contributions:

1) We bring forward a segmented linear model for the power behavior of servers used in cloud infrastructures. Using this model, we identify that datacenter power consumption can be reduced by eliminating the number of turned-on servers.

2) We propose a power-aware virtual datacenter resource scheduling strategy. The algorithm consolidates servers periodically, and adjusts VM placements between consolidation rounds to honor SLA.

3) We constructed and validated a power-aware datacenter simulator. Using this simulator, we analyzed the performance of our scheduling algorithm. Simulations shows that our strategy is more power-efficient that event-based ones.

1521-9097/13 $31.00 © 2013 IEEE
DOI 10.1109/ICPADS.2013.27

4) Our further simulation results show that, in general, VM weight adjustment according to workload can enable power-aware schedulers to perform more efficiently under SLA constraints.

The rest of this paper is organized as follows: section II surveys related works on datacenter power saving strategies; section III discusses our server's power model; in section IV we present our scheduling methodology based on Xen and our server's power model; section V describes the framework of our datacenter simulator; in section VI we analyze the performance of our algorithm and discuss the influence of its executing environment; we conclude our work in section VII.

II. RELATED WORK

Physical server's power behavior has been studied in datacenters as a classic operational issue. In recent years, this issue arose again due to the wide adoption of VMs in virtualized servers. In [10, 11] researchers report that the server's power need increases as its workload rises. Rising workload may affect various subsystems of a machine, such as higher CPU usage rate, larger occupied memory, faster disk and network average I/O speed, etc. Heavier resource usage intensity causes larger server power need. In [7, 11-13], CPU usage rate is adopted as the main factor that affect a server's total power consumption. Moreover, [7, 13] report a linear relationship between server's CPU usage rate and its power consumption. In [13], power behavior is pinned to VMs instead of hosts by tracking each VM's virtual resource states. In this paper, we model our server's power behavior by its physical CPU usage rate. We find that the server's input power follows a linear function of its CPU usage rate well in different segments. Instead of a global linear model, a segmented linear function gives an accurate prediction of server's power need.

To understand the power-saving potential of servers, previous studies on server power models discover that servers consume up to half of its peak power under near-zero workload [7, 11, 14]. This idle power is the main cause of datacenter's resource usage inefficiency. In non-virtualized datacenters, workload is balanced on all servers to ensure QoS and each server hosts a small portion of workloads compared with the peak workload volume it supports. Through deploying virtual machines on hosing servers, today's cloud computing datacenter supports convenient resource management. Server consolidation strategies in cloud provider's datacenters can reduce the number of low-workload servers and improve datacenter's resource usage efficiency as well as its total power cost [15]. Our work adopts a similar approach. Moreover, we design the algorithm in time domain to give sustainable power efficiency under SLA constraint.

To implement power saving strategies, both software and hardware technologies were proposed and designed. Energy-efficient CPUs have been studied to provide energy-proportionality[14], such as DVFS and using embedded processors in datacenter servers [16]. However, power consumption from other sources, such as GPU, memory and NIC still produce server's idle power. As a result, it is necessary to manage datacenter resources in software to reduce its power cost.

To evaluate the software methodology that reduces large datacenter's total power consumption, researchers usually adopts the approach of computer simulation since it is not feasible to evaluate big varieties of strategies at physical datacenters. A well-modeled simulator, however, can fulfill our need while giving repeatable results at the same time. Current popular cloud datacenter simulators include CloudSim [17] and EEFSim [18]. CloudSim models a datacenter as collections of VMs and supports user-specified server power model. It outputs cloud power consumption and SLA violation costs. However, in CloudSim each VM serves a single request, which is not the real case where a VM may run mixed workloads. EEFSim models a datacenter as a set of identical servers and uses CPU usage rate as the only power model input. To improve simulation accuracy, EEFSim includes a feed-back step in its development process where simulation results are compared with experimental results. However, EEFSim does not support heterogeneous server deployment which is a common case in the cloud. As none of these existing simulators handles power of virtualized servers driven by multiple workloads, we develop our simulator which is based on our server's power behavior. Our simulator is designed on the Xen virtualization platform. Unlike CloudSim or EEFSim, our simulator runs in the continuous time domain instead of adopting a discrete-event based simulation engine. Our simulator supports workload record trace input. A user-specified workload characterization interface is needed to help enforce SLA. The simulator outputs datacenter's power consumption trace and total energy consumption for analysis. We'll discuss our simulator in detail in section V.

III. SERVER POWER MODEL

A server exhibits complex power behavior because each of its subsystem contributes to the total power in its own way, such as CPU, memory, disk, GPU, NIC, etc. Among these parts, we observe that the CPU mainly determines a machine's total power, while other resources' usage states do not affect server's total power noticeably. Moreover, accurate power modeling of every machines subsystem inevitably induces non-negligible system overhead, which is not desirable for the cloud management middleware. For the two reasons above, we treat server's CPU as the major factor that determines server's input power.

$$Total\_power = CPU\_power + Standby\_power \qquad (1)$$

A. Power Model Formulation

Accurate prediction of CPU's power consumption requires a cycle-level simulator. However, running these simulators is unacceptable for cloud management because they require intense compute resources and bring considerable prediction delay. As a result, we need a lightweight power model which predicts server's power accurately enough for datacenter-level management without bringing significant execution overhead.

We use the CPU usage rate of the server as the input parameter to estimate server's total power. The CPU usage rate reports the time share during which the CPU is activated to execute a certain task. Thus CPU's power consumption is linear to this variable. Between consecutive activated periods, the CPU resides in sleep mode, dissipating a fixed amount of sleeping power. Let u_{cpu} denote the total CPU usage rate of the server. Then, CPU's power consumption becomes:

$$CPU\_power = \alpha * u_{cpu} + sleep\_power_{cpu} \qquad (2)$$

where α and $sleep\_power_{cpu}$ are a model-specific parameters.

From (1) and (2), we derive a linear relationship between server's total power and u_{cpu}:

$$Total\_power = \alpha * u_{cpu} + Idle\_power \qquad (3)$$

where $Idle\_power$ stands for server's power consumption when its CPU is under zero workload.

B. Segmented Linear Power Model

Towards our server's behavior, we improved the power model in (3) by dividing the linear function into 3 segments. In each segment, server power follows a linear function as in (3):

$$\left\{ \begin{array}{l} power = \alpha_1 * u_{cpu} + \beta_1,\ u_{min} \le u_{cpu} < u_1 \\ power = \alpha_2 * u_{cpu} + \beta_2,\ u_1 \le u_{cpu} < u_2 \\ power = \alpha_3 * u_{cpu} + \beta_3,\ u_2 \le u_{cpu} < u_{max} \end{array} \right\} \qquad (4)$$

where β_1 equals server's $Idle\_power$.

The improved power model exhibits segmented linear function because our server adopts a variable-speed fan. When CPU usage reaches a critical value the fan accelerates to cool the CPU temperature down, causing abrupt power increase. For our server, the parameters in (4) are summarized in Table 1.

TABLE I. SEGMENTED POWER MODEL PARAMETERS

Parameter	α_1	α_2	α_3	β_1	β_2	β_3
Value	2.492	84.09	11.93	93.49	-188.2	72.8

a.

The boundaries for the 3 segments in (4) are $u_1 = 3.45$ and $u_2 = 3.615$. The limits for CPU usage rate are $u_{min} = 0$ and $u_{max} = 4$, which is equal to the core number of the CPU. Figure 1 shows our overall power model.

C. Parameter Extraction

We extracted the model parameters above using measured data from our server. By installing a Power-Bay in series to the server, we can monitor server's input power with a time resolution of 7s. The servers run Xen 3.0.3. We use "xentop" in the host OS to get each VM's resource usage state every 3 seconds. The server's total CPU usage rate is obtained by aggregating each VM's CPU usage rate.

Figure 1 shows our overall power model. In the following paragraphs, we discuss each segment of our power model in detail.

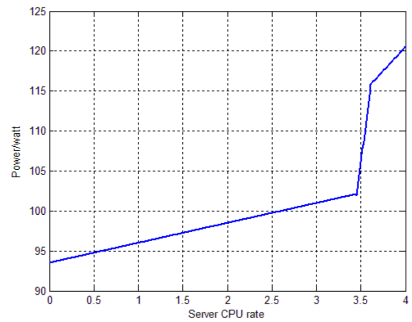

Fig. 1. Segmented linear server power model.

First, we create 4 VMs on a server and let each VM run a program that controls the VM's CPU usage rate. The program forces its VCPU to sleep for a while after every calculation phase. We increment each VM's CPU usage rate from 3% to 85% and measure the corresponding input power. Figure 2 displays the server CPU rate trace and power trace.

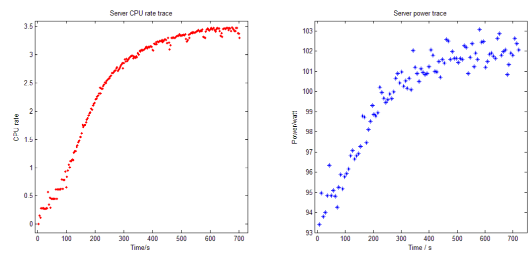

Fig. 2. CPU usage rate and server power trace, 0<CPU rate<3.45.

In Figure 2 we finds a strong relationship between server's CPU rate and its power consumption. We use linear fitting to form server's power model in this segment (Figure 3).

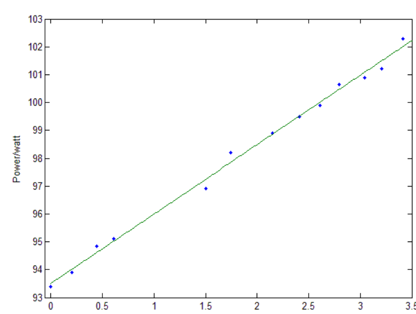

Fig. 3. Server power model, 0<CPU rate<3.45.

The coefficient and intercept fitted from Figure 3 are α_1 and β_1.

Next, we let 3 VMs occupies their entire VCPUs, which corresponds to 300% of server's CPU resource. The other VM runs the same program to control the rest of the CPU resources as before. Its VCPU rate increases from 3% to 80%. The measured server CPU rate trace and power trace are shown in Figure 4.

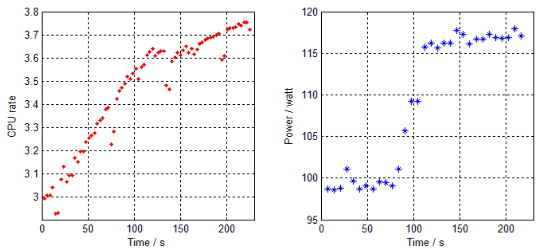

Fig. 4. CPU usage rate and server power trace, 3.45<CPU rate<3.8.

Unlike Figure 3, server input power displays 2 correlations segments with its CPU rate. We divide the 2 segments at the CPU rate of 3.615 and fit their linear functions in Figure 5.

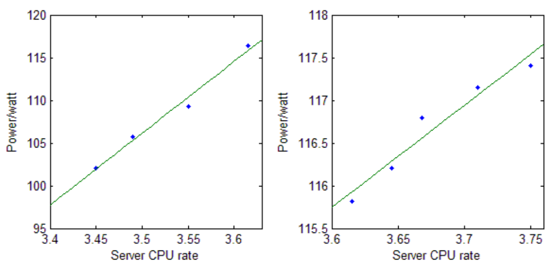

Fig. 5. Server power model, 3.45<CPU rate<3.8.

The parameters fitted from Figure 5 corresponds to α_2, β_2, α_3, β_3 respectively.

In heterogeneous environments, the parameters above can be obtained for different server configurations in the same way because linear models generally hold for all CPU designs.

IV. Scheduling Methodology

In this section we explain our scheduling algorithm that reduces virtualized server farm's power consumption.

A. Overview

Figure 1 shows that our server's idle power of is 93.5w, 77.6% of its peak power (120.5w). Consequently, a running server spends most of its input power on feeding idle power rather than executing workloads, leading to server resource usage inefficiency. Thus, large numbers of running servers, whether heavy-loaded or light-loaded, produce a significant amount of total idle power, causing a waste of datacenter power budget.

To improve this situation, we propose a datacenter resource scheduling strategy that reduces server farm's power budget by consolidating workloads into a smaller set of hosting servers. Our algorithm ensures that the VMs' SLAs are honored by allocating each VM with a sufficient amount of CPU resources. More specifically, our algorithm orchestrates a consolidation round periodically with a batch of VM live migrations (Algorithm 1, 3), and keeps adjusting VM placements to provide each VM with the necessary resource to meet its SLA (Algorithm 2, 3). An overview of our power-aware resource scheduling strategy is presented in Figure 6. We explain the details of our algorithm in the following of this section.

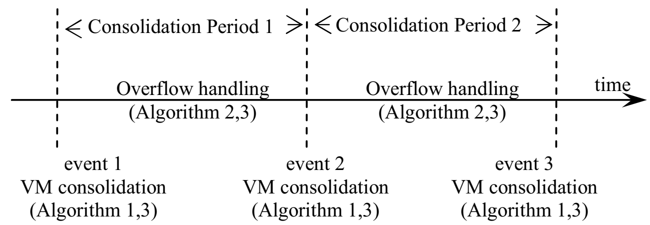

Fig. 6. Overview of our power-aware resource scheduling strategy.

While other power-aware scheduling algorithms also aim at consolidating VMs onto the smallest set of running servers, they usually adopt an event-driven approach instead of a time-driven one [9]. In our work, we do not distinguish the problem of scheduling incoming new VMs and re-distributing old VMs. We argue that running the algorithm in time-domain leverages the characteristics of workloads more effectively due to more frequent monitoring of server utilization status. Moreover, our algorithm introduces the assumption that the datacenter always hosts plenty of shut-down servers as re-distribution migration targets, which is usually the real case for large datacenters. Based on this condition, our algorithm is able to consolidate VMs to the largest extent every other period, and does not need to adjust existing VM placements for power budget improvement. As a result, our strategy produces smaller server farm power consumption than previous ones.

B. Algorithm 1: Consolidating VMs

The problem of allocating servers to VMs with the constraint of minimizing server farm's total power budget is viewed as the multi-dimensional bin-packing problem [9, 15]. We take CPU usage rate as the only measurement of the size of servers and VMs because other hardware parts usually do not restrict VM placements. When computing the solution to this problem we adopt the descending best-fit algorithm. Previous studies have demonstrated First-Fit and Best-Fit provide near-optimal solution for the bin-packing problem. The algorithm is described in Algorithm 1.

Algorithm 1: Re-allocating VMs during Consolidation Rounds

```
Input: Host_List, VM_List     Output: VM allocations
sort VMs in their CPU requirement's descending order
for each VM in the sorted_VM_list
    minPower=MAX;   allocated_host=NULL;
    for each new (empty) Host in the datacenter
        if (this Host can accommodate this VM)
            estimateNewHostPower(this Host, VM)
            if ( estimated_power <  minPower )
                minPower = estimated_power;
                allocated_host = this Host;
    if no host is allocated:
        turn on a newHost;
        allocated_host = turned_on Host;
    migrate this VM to its allocated_host;
    if the VM's original host is empty after migration
        turn off original Host;
```

C. Algorithm 2: Overflow Handling

During consolidation rounds the algorithm squeezes the VMs into a set of hosts according to the VMs' temporal CPU requirements. However, workloads vary randomly between consolidation rounds, and a VM's allocated server resource may not satisfy its requirement defined by SLA when its wor kload surge up. Under such circumstance, we need to migrate some of the server's co-located VMs out to make room for the resource-overflowed ones. In our algorithm, we scan all VMs' resource requirements and choose the resource-overflowed one as the migration object. We use the First-Fit algorithm to find the migration target host for this VM. The overflow handling strategy is shown in Algorithm 2.

Algorithm 2: Handling Overflowed VMs

```
Input: Host_List, VM_List     Output: VM allocations
for each VM in the datacenter
    read VM's current CPU requirement;
    if the VM's CPU requirement > its CPU capacity
        for each Host in the datacenter
            if the Host can accommodate this VM;
                Migration target host = this host;
        if no such host is found
            turn on a new Host;
            Migration target host = turned_on Host;
        migrate this VM to its target host;
```

Since increasing the total number of turned-on servers is inevitable in this scenario, we abandon other sophisticated heuristics for VM selection and migration. The overflowed VMs' workloads are likely to surge up continuously in short term so migrating them into emptier host will reduce the possibility of having to handle them again. Server farm's total power is mainly decided by the number of turned-on servers, and the algorithm above suffice to achieve the near-optimal number with small performance overhead.

D. Algorithm 3: Scheduling Weight Adjustment

Xen uses the Credit Scheduler to divide server CPU for co-located VMs[19], and we can change each VM's size by adjusting its "weight" and "capacity" parameters. Our algorithm leverages this feature to scale each VM's VCPU share according to the resource requirement of its workload at the time of migration. Through adjusting each VM's weight, the algorithm allocates server's CPU to each VM in proportion to the VM's real CPU usage. The "weights" are set relative to Xen's default value (256). The adjustment method is explained in Algorithm 3.

Algorithm 3: Adjusting VM's Weight when Migrated

Input: VM_List **Output:** VM Weights
calculate the total CPU consumed on this server;
for each VM on this server
  toal_weight = 256*server's hosting VM_number;
  $$\text{weight} = \frac{\text{VM's CPU requirement}}{\text{Host's CPU consumed}} * \text{total\_weight};$$

By adjusting the VM's weight, we allocate each VM with a proper amount of CPU resource so that VMs with all sizes of CPU requirements will have high VCPU usage rate. This enables tighter VM consolidation and will further improve server's resource usage efficiency.

V. POWER-AWARE DATACENTER SIMULATOR

To test our scheduling algorithm, we developed a power-aware datacenter simulator. Our simulator runs in time domain and keeps reporting server farm's resource usage status as well as its total power consumption.

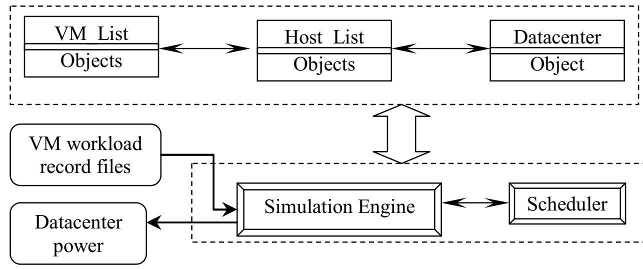

Fig. 7. Simulator Framework

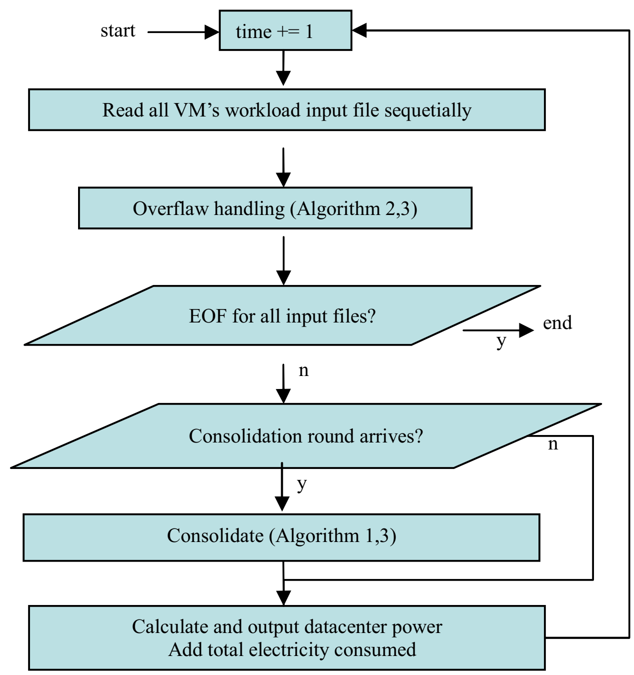

Fig. 8. Simulation Engine Workflow

A. Simulator Framework

The framework of our simulator is presented in Figure 7. The simulator views a datacenter as a list of servers, and views each server as a host for a list of VMs. Users can insert their workload schedulers between the simulated datacenter and the simulation engine.

Hosts can be heterogeneous in our simulator. A server's computing capability is measured by its CPU core number. Each host can include its own segmented linear power model. We simulate the division of host's CPU resource for VMs using Xen's credit Scehduler – Inside VM_List, each VM is allocated a "Weight" and a "Credit". VCPU is assigned according to VM's share of host's total CPU. As simulation input, a workload trace file is pinned to each VM. The simulated VM converts its workload records into resource requirements based on SLA and workload characteristics.

B. Simulation Engine

The simulation engine is illustrated in Figure 8. The simulator measures simulated time by counting total time steps incremented. At each time step, the simulation engine reads a workload record of all servers' hosting VMs. Based on this workload record, the simulator determines whether overflow handling for this VM is needed. A consolidation round is performed every time a consolidation period passes. Simulation ends when all VM's workload trace files encounter EOF.

VI. Simulation Results

In this section, we test our scheduling algorithm with our power-aware simulator.

A. Simulator Validation

First, we validate our simulator and our segmented linear power model. We let each VM's CPU rate oscillate between 5% and 85% and record the server's input power. We let 4 VMs run on a host. Figure 9 shows the simulated and measured server power trace.

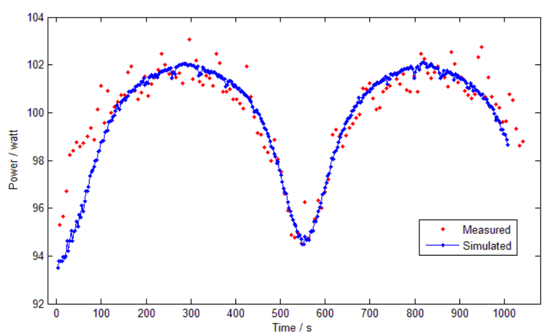

Fig. 9. Simulated and measured server power trace.

In Figure 9, we observe that the simulated result predicts server's power consumption very accurately. Prediction deviations from measured values occur sometimes within 2 watts because server subsystems such as caches and FPU are loaded differently and CPU usage rate cannot reflect this phenomenon. These small errors do not affect datacenter-level management. Therefore, we can use this simulator to evaluate and analyze the performance of our power-aware scheduling algorithm.

B. Simulated Results of Power-aware Scheduling Strategies

We compare our power-aware algorithm with an event-based energy-efficient datacenter management strategy. The event-based strategy consists of 2 parts: (1) admitting new VMs and provisioning host resources for them, and (2) optimizing exsiting VM allocation. For the first part, the algorithm consolidates VMs onto a smallest set of hosts. For the second part, the algorithm sets 2 thresholds for host's CPU usage rate. If the host CPU rate is below the lower threshold then all its hosting VM are to be migrated out, which eliminates low-workload server power dissipation. If the host rate is above the higher threshold then the algorithm select a co-located VM to migrate out to ensure SLA. We choose the "Highest Potential Growth Policy" for VM selection. This policy migrates out VMs with the lowest VCPU rate to reduce potential increase of hosts' utilization and honors SLA.

We set 500 cloud users for the simulation, with each of them leasing one VM. The VM's workloads are randomly generated and follow uniform distribution to resemble web requests workloads. The simulation lasts for 3 hours. Figure 10 shows the simulated datacenter power consumption trace.

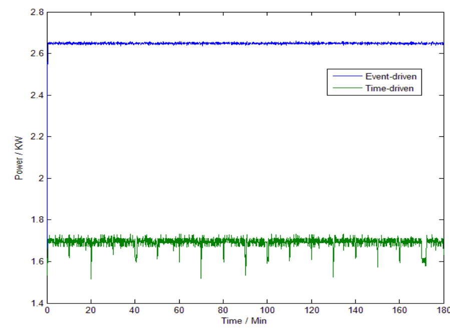

Fig. 10. Datacenter power trace under event- and time-driven scheduler.

Our power-aware scheduling strategy gives a steady power consumption of around 1.7 KW, about 0.9 KW lower than the event-based strategy. We believe that 2 factors contribute to this improvement:

1) Our strategy performs VM consolidation periodically (corresponding the sinks in the green curve) to make sure that VM placements return to a consolidated pattern. Under random workload, VM's resource requirements surge up frequently. To honor SLA, the number of turned-on servers rebounds quickly and is then stabilized, with each server hosting fewer number of VMs. Consolidating periodically prevents VMs from being allocated dispersedly all the time.

2) Our strategy adjusts VM's weight parameter in the Xen Credit Scheduler to promote more efficient VM resource utilization. Modifying VM size in proportion to its resource requirement defined by workload and SLA helps each VM extract its resources to a larger extent. Without this technique, VMs share server resources equally, leading to more frequent overflow handling and less efficient server resource usage. Consequently, more servers are turned on to satisfy SLA.

For another case, we generated a workload according to Gaussian distribution to resemble a computing-intensive workload such as ScaLAPACK. The workload and VM's resource requirements are stable in this scenario. Figure 11 presents the power trace.

When the event-based scheduler is adopted, datacenter power consumption goes up when workload jumps up because it orders the datacenter to turn more servers on to host selected VMs to abide by SLAs. VM consolidation does not occur thereafter because hosts' CPU usage rate usually stays between the two thresholds. On average, our strategy achieves 22.7% power reduction than the event-based scheduler in this case.

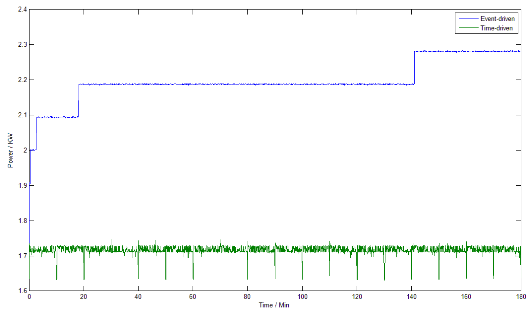

Fig. 11. Datacenter power trace with computing-intensive workload.

C. Simulted Analysis of VM settings and Workload Variance

The performance of power-aware resource scheduling strategies depends on VM settings and workload characteristics. For more stable workloads, power-aware strategies generally achieve better results because VM placements do not need to be frequently modified to satisfy SLAs. Workloads in Figure 10 and 11 have the same averaged resource requirements. However, in the latter case a server farm's power consumption is lower for both scheduling strategies because the workload is more stable and it enables both strategies to produce efficient VM allocation profiles continuously.

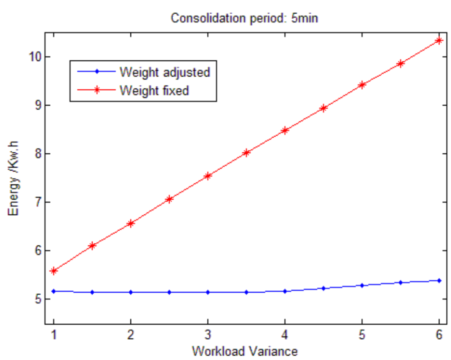

Fig. 12. Datacenter Energy Consumption with Gaussian workload variance.

Figure 12 shows the relationship between a datacenter's total electricity consumed and its Gaussian workload's variance when our resource scheduling strategy is adopted. Other workload distribution patterns can also be simulated to study their impact on power-aware scheduling algorithm's performance. In our simulation, we distinguish VM platforms that allow VM's weight to be changed and those that do not (earlier Xen platforms do not support user-specified VM weight and capacity). In the weight-fixed case, we find that datacenter's power consumption is very sensitive to workload variances - the energy-variance relationship nearly follows a linear function. However, in the weight-adjustable case datacenter's power consumption remains stable when weight is adjusted using Algorithm 3.

We further explore the effect of adjusting VM's weight at runtime on our scheduling strategy's performance (Figure 13).

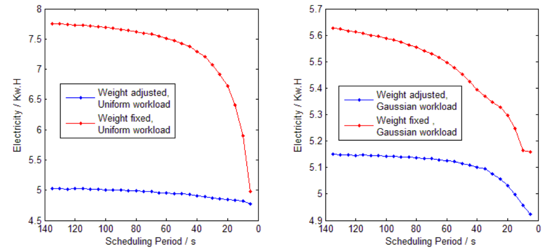

Fig. 13. Datacenter electricity consumed with adjusted and fixed VM weights.

Adjusting VM's weights at runtime reduces datacenter's electricity consumed for both random workload (the uniform distributed one) and stable workload (the Gaussian distributed one). Fixing VM's weight causes inefficient VCPU usage because VM sizes are not determined according the workload variations. VM overflow handling occurs more frequently under this situation. Consequently, more servers are turned on to satisfy their SLAs.

In general, adjusting VM weight and running stable workloads enable power-aware resource scheduling strategies to produce more efficient VM distribution profiles under SLA constraints.

VII. CONCOLUSION

In this paper, we proposed a power-aware resource scheduling strategy. Based on VM live migration[4], our scheduling strategy consolidates VMs periodically and re-distributes VMs between consolidation rounds to satisfy SLAs. When VMs are re-located, our algorithm leverages Xen's Credit Scheduler to resize VM's VCPU to improve VM resource utilization rate.

We developed a datacenter simulator based on the Xen[2] platform. The simulator runs in time domain and simulates a virtualized datacenter's power behavior using server's segmented linear power model. Our power model is formed by linear fitting measured server power data. We validated our simulator and further used it to analyze our scheduling strategy.

Our simulation shows that: compared with event-based scheduling algorithms[9], our strategy achieves 35% power saving improvement for workloads exhibiting random resource requirements like web requests, and achieves 22.7% power saving improvement for workloads with stable resource requirements like ScaLAPACK. Further simulations indicate that, in general, power-aware datacenter scheduling algorithms perform better when workloads are stable and when VM weights are adjustable according to the needs of VMs for SLA fulfillment.

VIII. ACKNOWLEDGEMENT

This work was supported in part by the Shanghai International Science and Technology Collaboration Program under Grant 13430710400.

REFERENCES

[1] A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang*, et al.*, "Energy-efficient cloud computing," *The Computer Journal,* vol. 53, pp. 1045-1051, 2010.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho*, et al.*, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review,* vol. 37, pp. 164-177, 2003.

[3] C. A. Waldspurger, "Memory resource management in VMware ESX server," *ACM SIGOPS Operating Systems Review,* vol. 36, pp. 181-194, 2002.

[4] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach*, et al.*, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*, 2005, pp. 273-286.

[5] C. Cai, L. Wang, S. U. Khan, and J. Tao, "Energy-Aware High Performance Computing: A Taxonomy Study," in *Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on*, 2011, pp. 953-958.

[6] A. Kipp, J. Liu, T. Jiang, D. Khabi, Y. Kovalenko, L. Schubert*, et al.*, "Approach towards an energy-aware and energy-efficient high performance computing environment," in *Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on*, 2011, pp. 493-499.

[7] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao*, et al.*, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, 2008, pp. 337-350.

[8] Z. Zhang, L. Xiao, Y. Li, and L. Ruan, "A VM-based Resource Management Method Using Statistics," in *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, 2012, pp. 788-793.

[9] R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," *arXiv preprint arXiv:1006.0308,* 2010.

[10] L. A. Barroso and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures on Computer Architecture,* vol. 4, pp. 1-108, 2009.

[11] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News,* vol. 35, pp. 13-23, 2007.

[12] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proceedings of the 2008 conference on Power aware computing and systems*, 2008.

[13] 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, pp. 39-50.

[14] L. A. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer,* vol. 40, pp. 33-37, 2007.

[15] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *Services Computing, IEEE Transactions on,* vol. 3, pp. 266-278, 2010.

[16] V. Janapa Reddi, B. C. Lee, T. Chilimbi, and K. Vaid, "Web search using mobile cores: quantifying and mitigating the price of efficiency," in *ACM SIGARCH Computer Architecture News*, 2010, pp. 314-325.

[17] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience,* vol. 41, pp. 23-50, 2011.

[18] F. Julia, J. Roldan, R. Nou, J. O. Fitó, A. Vaqué, Í. Goiri*, et al.*, "EEFSim: Energy Efficiency Simulator," research report no. UPC-DAC-RR-CAP-2010-15, Dept. of Computer Architecture, BarcelonaTech2010.

[19] Credit Scheduler. http://wiki.xensource.com/xenwiki/CreditScheduler