Applied Mathematics & Information Sciences
An International Journal

PATPro: Power Aware Thin Provisioning of Resources in Virtualized Servers

Jiang Cong-Feng[1], *Zhao Ying-Hui*[2]

[1]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China
[2]Department of Hydraulic Engineering, Zhejiang Tongji Vocational College of Science and Technology, Hangzhou, China

Received: 25 Jul. 2012, Revised: 14 Nov. 2012, Accepted: 20 Nov. 2012
Published online: 1 Feb. 2013

Abstract: In traditional virtualization systems Virtual Machines (VMs) are usually over-provisioned for guarantying peak performance of hosting applications and thus waste a lot of computing resources. Although virtual machine consolidation can save power consumptions, it also increases the power intensity in a rack and makes the servers more prone to failures. It also increases the probability of Service Level Agreements (SLAs) violations under constrained power budget. Due to the autonomy, workload dynamics and performance requirements of Virtual Machines, it is nontrivial to tradeoff between the power and performance in virtualized servers. In this paper, we proposed a power aware thin provisioning approach for resource allocation, named PATPro, to coordinate and tradeoff the power and performance among multiple virtual machines. In order to provide enough information about power characteristics of individual VM and applications, we use agent for real time power collection and workload characterization. We evaluated our algorithm in a real virtualization environment with 8 VMs. The results show that the proposed approach can save 5.41% to 6.33% power consumption to provide the same performance in applications including computing-intensive, I/O intensive and hybrid workloads such as web servers and database operations. The results also show the potential of our approach to be used in real virtualized environments.

Keywords: Power, Virtual Machine, Dynamic Frequency Scaling, Energy Consumption

1. Introduction

Cloud computing improves the resource efficiency through economies of scale, and statistical multiplexing among numerous hosted workloads. However, large scale hosting cloud platforms also require enormous amounts of power, which results in high power provisioning costs. Power consumption is becoming an annoying problem to large scale server systems and cloud computing data centers, as well as laptops and mobile devices [1–3]. Energy related costs have become increasingly important to computing systems, since they directly impact the power provisioning cost for computing infrastructures and their operating expense. Higher power consumption results in more heat dissipation, cooling costs and makes servers more prone to failures. In Internet data centers, power saving technique is particularly valuable when there is a fixed power constraint or power budget[4]. Since the cooling costs of a server systems usually keep constant or change little, the dynamic proportion of the total power consumption are mainly correlated with the real time power consumption of the computing infrastructure, including processors, memories, disks, switches, etc.

In cloud computing environments, power budgeting is often utilized to manage and reduce power provisioning costs and modern server hardware provides budgeting mechanisms. Nowadays, various emerging industrial solutions address different aspects of the power consumption problem. For example, processors, memories, disks, and switches often have multiple power modes, such as maximum performance, normal operating, and sleep modes. However, per-component power gating or scaling can't always save power because power reduction in one component may increase power consumption in another component[5]. Therefore, power reduction in a system level is more likely to reduce the total power and energy consumption in real computing systems.

Virtualization has been widely deployed in internet data centers and cloud computing systems for power saving and flexible management of the physical resources through workload consolidation. Although virtual

* Corresponding author e-mail: cjiang@hdu.edu.cn

activities are limited within a virtual machine and seem that they have no influence on real power consumption, virtual activities can still affect real power consumptions when they are mapping to real activities operated on the physical hardware eventually. However, the distributed and autonomic nature of multi-layered virtual machine environments makes traditional power management approaches insufficient in virtualized systems because traditional power management techniques are based on the assumption that operating systems are with full knowledge of and full control over the underlying hardware. Moreover, reducing power consumption of data center in the presence of virtualization faces challenges include power consumption estimation and characterization, power management coordination among multiple Virtual Machines (VMs), and heterogeneity in hardware capabilities, which make it impossible to apply traditional power management techniques, such as coarser-grained device sleep or dynamic voltage/frequency scaling(DVS/DFS), directly to the virtualization environment without modification[6].

Traditionally, it is not applicable to measure the power of various components such as the CPU, screen, memory, and storage in commodity computers these components do not have a single wire supplying their power where a hardware power meter may be installed. Therefore, a software tool is needed to estimate the energy usage of an individual component. Moreover, in virtualized environment, the power consumption cannot be partitioned among shared hardware like CPU, caches and memory bandwidth.

In this paper, we proposed a power aware thin provisioning approach for resource allocation, named PATPro, to coordinate and tradeoff the power and performance among multiple virtual machines. In order to provide enough information about power characteristics of individual VM and applications, we use agent for real time power collection and workload characterization. We evaluated our algorithm on a real virtualization environment with 8 VMs. The results show that the proposed approach can save5.41% to 6.33% power consumption to provide the same performance in applications including computing-intensive, I/O intensive and hybrid workloads such as web servers and database operations. The results also show the potential of our approach to be used in real virtualized environments.

The remainder of this paper is organized as follows: In Section 2 provide some related works. In Section 3 we propose the methodology framework of the power aware thin provisioning approach. In Section 4 we present the system level power model and VM level power model in section 5. Then, in Section 6, we present experiment results of the proposed PATPro approach. Finally, we summarize the work in Section 7.

2. Related Works

When the scale of computing systems increases, the system performance, for example, availability, responsiveness, and throughput, do not scale with the number of processors but the power consumption does.

Traditionally, resource allocation and scheduling algorithms focus on job completion time and average waiting time of jobs[7, 8].

In modern data centers, power usage management can be realized by using circuit and outlet level measurements. Researchers have proposed various power adaptation and energy reduction approaches to reduce system level power and energy consumptions or different physical subsystems including CPUs, memories, and hard disks[9–15].

In virtualized systems, fine-grained power management among various VMs can also enable further savings in provisioning costs[5]. Researchers also proposed some approaches for power management in virtualization environment like Magnet[16], VirtualPower[17] and ClientVisor [18].

Most of current power management schemes are limited to OS with monolithic kernel and not applicable for virtualization environments and most of current virtualization solutions usually virtualize a set of standard hardware devices only, without any special capabilities or support for power management[6]. Therefore, VM power cannot be measured purely in hardware. In [19] the author presented a solution for VM power metering. They built power models to infer power consumption from resource usage at runtime. They use existing instrumentation in server hardware and hypervisors to build the required power models on real platforms with low error and low runtime overhead while providing practically useful accuracy. Their experiments show significant savings in power provisioning costs that constitute a large fraction of data center power costs.

In industrial area, in VMware vSphere suite, the vSphere Distributed Power Management (DPM) continuously optimizes power consumption in the datacenter. When virtual machines in a DRS cluster need fewer resources, such as during nights and weekends, DPM consolidates workloads onto fewer servers and powers off the rest to reduce power consumption. When virtual machine resource requirements increase (such as when users log into applications in the morning), DPM brings powered-down hosts back online to ensure service levels are met[20].

The Joulemeter[21] software provides a modeling tool to measure the energy usage of virtual machines (VMs), servers, desktops, laptops, and even individual software applications running on a computer. It can improve the energy efficiency of computing devices and infrastructures and can be used to improve power provisioning costs for data centers, virtualized power budgeting, desktop energy optimizations, and mobile battery management. In Joulemeter, the power consumption is divided into base, CPU, disk and monitor which are added to get total power consumption. Joulemeter currently is not measuring the GPU's power consumption because on modern computer systems, a GPU card uses lots of power.

Power metering and measurement is the first step for power aware thin provisioning in virtualized servers. Especially, per-VM power metering is inevitable for VM scheduling and resource allocation. Therefore, the power consumption of a physical host server must be divided into individual virtual machine power consumption residing on the host server. This can be done by generating a power model using the total power consumption of the host server and resource utilization for a virtual machine. With the per-VM power metering, we can use it for virtual machine power capping to allow power oversubscription in virtualized environments.

In [22], Laszewski proposed an approach to allocate virtual machines via the technique of Dynamic Voltage Frequency Scaling (DVFS) to reduce power consumption. They dynamically scale the supplied voltages of the compute cluster.

In cloud computing environments, reducing energy consumption is a critical step in lowering data center operating costs for various institutions. Jansen et al[23] analyzed the effects of virtual machine allocation on energy consumption by using a variety of real-world policies and a realistic testing scenario and found that by using an allocation policy designed to minimize energy, total energy consumption could be reduced by up to 14%, and total monetary energy costs could be reduced by up to 26%.

In this paper, we first modeling the VM power consumption through historical power and performance data and then use this model to estimate the real time power consumption of individual VM. According to this estimation, we use methods to reduce VM power consumptions, including resizing CPU slice sharing and memory allocation, CPU pinning, and migration and re-consolidation while providing performance in terms of response time and ability to service requests. Our approach can be used to manage power consumption and performance by adaptive VM placement.

3. PATPro Framework and Methodology

3.1. Assumptions

In order to provide power aware thin provisioning in virtualized computing system, the first step is to accurately characterize the power consumption of individual VM. For simplicity, we assume that in a physical server, the power consumed by devices except for CPU, memory and disks are constant regardless of workload fluctuations. This assumption can easily hold because in modern computing systems and real application scenarios, these devices consumed most of the power during the machine running. Please also note that we do not consider the transformation loss in power source supply in the machine or mainboard. Although a high end graphics card may consume power as large as a CPU, we do not consider this situation because in typical server systems except for GPU cluster, its power consumption can be neglected compared to CPU or main memory. We also exclude the power consumption of CPU fans since they change rotation speed intermittently according to the processor workloads. Therefore, we can regard it as a part of a CPU.

3.2. Methodology

Although hardware performance counters based power profiling and characterization is more accurate, it is more complicated and hardware related. In this paper we first conducted experiments and characterize the relationship between processor utilization, memory utilization, disk I/O read/write performance, and the system level power consumption. We use a multi-meter to measure the system level power consumption.

In order to increase our model accuracy, we conduct different experimental scenarios that have different basic power consumption proportion and dynamic power consumption proportion.

We illustrate the PATPro framework in Fig.1.

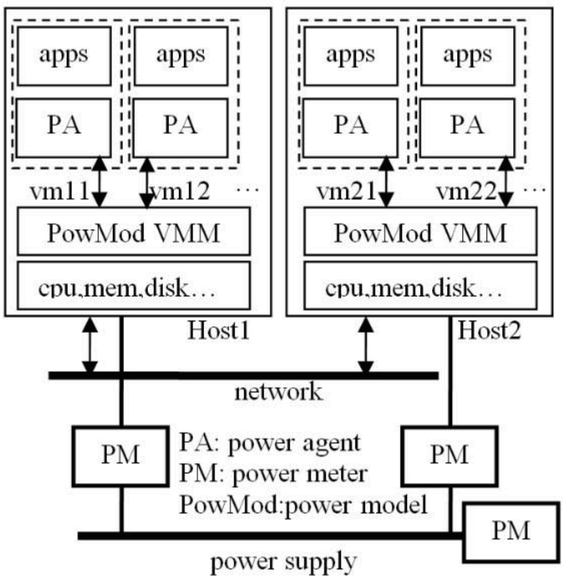


Figure 1 PATPro Framework

In order to better understand the potential relationship between the power consumption and VMs, we use a power meter (PM) to monitor power consumption on each host server and one power meter to report the power draw of the total system.

We use a power agent (PA) running in a VM to monitor the real time application requests and performance. The power agent communicate power and performance related information with a power model(PowMod) in the VMM. Then the power consumption of each VM in a host server can be calculated through the power model. Therefore the main challenge with this sort of accounting is to find a metric that reflects VM power consumption and tradeoff between calculation accuracy for power consumption and complexity. We are currently in the process of evaluating various such metrics and determine their accuracy.

4. System Level Power Model

Attempting to construct the power model, we conduct the experiments on a desktop server equipped with two 2.50GHz Intel Xeon E5420 processor, 8GB memory, and a disk raid with five 300GB SAS disks. In the experiments, we measure the power of the machine box (excluding the display monitor) and the processor utilization, memory utilization, disk I/O read/write performance. For generality, our system level power model is listed in Eq.1.

$$P_{host} = a \times u_{cpu} + b \times u_{mem} + c \times u_{dsk} + d \quad (1)$$

where:

u_{cpu} is the CPU utilization($0 < u_{cpu} < 100$) ;

u_{mem} is memory utilization($0 < u_{mem} < 100$);

u_{dsk} is disk read/write activity parameter(we use I/O waiting value to act as the disk read/write power indicator) ($0 < u_{dsk} < 100$).

a,b,c,d : coefficients to be determined.

For different workloads, including CPU intensive, I/O intensive and mixed workloads, the coefficients are different. For example, in a system that power of CPU and disks are dominant, b is set to zero. We measure the real-time power of the physical machine which hosts 8 VMs that runs different applications. The experiment settings are listed in Table1 and Table 2.

In the experiments, prime application is CPU intensive workload that finds prime numbers. The scp application is I/O intensive workload that simply copy large files from one VM to another. The httperf[24] application is a mixed workload for measuring web server performance by generating various HTTP workloads.

We list the trace data of experiment#1 in Fig.2 and experiment#2 in Fig.3.

We listed a, b, d values in Table 3(we set $b = 0$).

From table 3 we can see that in experiment #1, the system is lightly loaded and the base power is dominant. Therefore d in experiment#1 is larger than that of experiment#2. In the contrary, in experiment#2 the system is heavily loaded and highly dynamic(all VMs are pinned to 4 cores of the eight cores), therefore dynamic power is dominant in experiment#2 and d in experiment#2 is less than that of experiment#1.

Table 1 Settings of experiment #1

VM#	workloads	threads	CPU pinning
1	prime	1	No
2	prime	1	No
3	scp	1	No
4	scp	1	No
5	httperf	1	No
6	httperf	1	No
7	No	1	No
8	No	1	No

Table 2 Settings of experiment #2

VM#	workloads	threads	CPU pinning
1	prime	4	Core #0
2	prime	4	Core #0
3	prime	4	Core #1
4	prime	4	Core #1
5	prime	4	Core #2
6	prime	4	Core #2
7	httperf scp	4 2	Core #3 Core #3
8	httperf scp	4 2	Core #3 Core #3

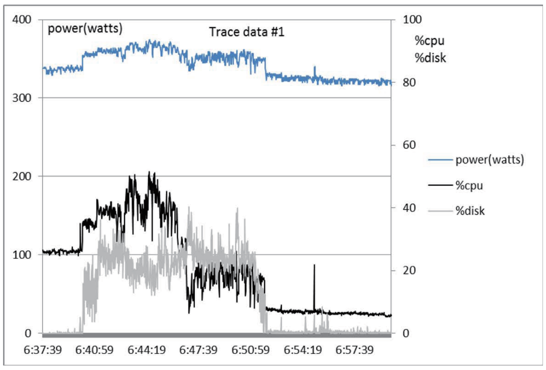


Figure 2 trace data of experiment#1(power in watts and cpu utilization and disk access are multiplied by 100)

Table 3 coefficients of experiment #1 and #2

experiment	a	b	d
#1	0.742543	0.656933	319.0726
#2	0.569055	1.347834	310.5591

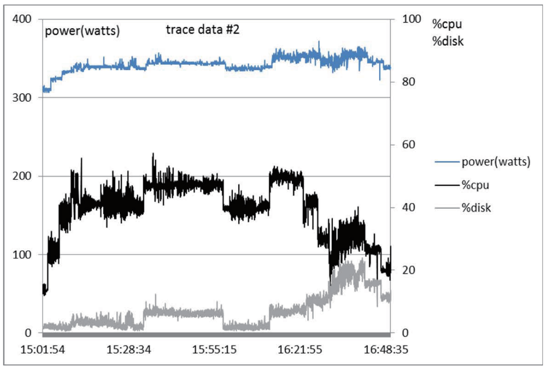


Figure 3 trace data of experiment#2(power in watts and cpu utilization and disk access are multiplied by 100)

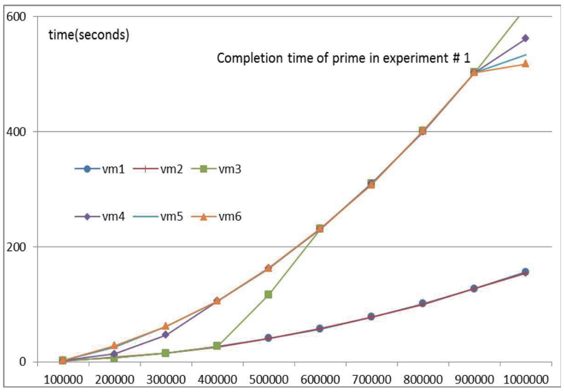


Figure 4 Completion time of prime in experiment#1(values of each VM)

Table 4 Model errors of experiment #1 and #2

experiment	N	R^2	SE
#1	1341	0.904190	5.135579
#2	6489	0.653483	5.149805

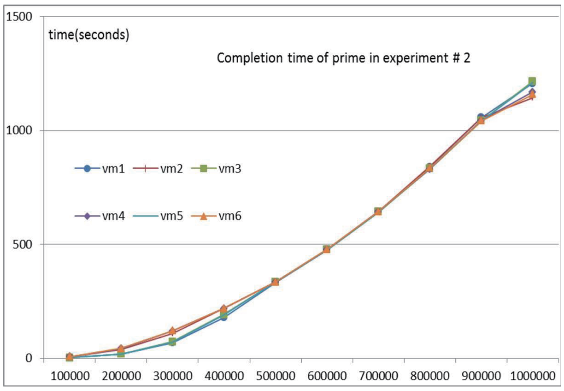


Figure 5 Completion time of prime in experiment#2(average values of four threads of each VM)

From the values in table 3, we can also see that a in experiment#1 is larger than that of experiment#2 and b in experiment#1 is less than that of experiment#2. This is because that in experiment#2 I/O access is much more than that of experiment#1. Because the power of disk raid ($48W * 5 = 240$ in max) is comparably larger than the total power of two processors ($80W * 2 = 160W$ in max), the power of disk raid is dominant although we add more prime threads in experiment#2.

The completion times of prime in experiment#1 and experiment#2 are listed in Fig.4 and Fig.5. From Fig.4 and Fig.5 we can see that the CPU is more heavily loaded in experiment#2 than that in experiment#2.

5. VM Level Power Model

5.1. Per-VM Power Consumption with CPU Intensive Workloads

In virtualized hosting environments, multiple VMs share the same hardware and they consume power even when there are no external requests to them. This is because that the hypervisor should provide an essential virtualized environment for VMs. Therefore, the power of VMs can also be partitioned into two parts: the base power and the dynamic power. The base power is calculated when the VM is idle. The dynamic power is correlated with the activities of the VM.

We conduct another two experiments to characterize the power of the VMs running CPU intensive workloads.

For consecutive reason, we mark the experiments as experiment#3, experiment#4, and experiment#5.

(1) Experiment#3: The host machine is idle and there is no VM running. The average power is 302.3W.

(2) Experiment#4: We start 8 VM instances in the host machine but the VMs do nothing. The average power is 303.4W and is very close to the average power when there is no VM running. The average CPU utilization is 4.3%, CPU I/O wait is 0.71%, and the disk throughput is 8.2 tps.

(3) Experiment#4: Firstly, we only start vm01 and run 4 prime search processes in vm01 for 2477 seconds, and the average power is 320.3W. Therefore a VM running 4 prime search processes will consume 16.9W. At the meantime, the CPU utilization is 14.7% during the application execution.

Secondly, we run 2 VMs (vm02 and vm03) and in each VM we run the same 4 prime search processes, and the average power is 332.7W. Therefore a VM running 4 prime search processes will consume 15.65W. At the

Table 5 httperf results in experiment #1 and #2

	Experiment#1		Experiment#2	
	vm01	vm02	vm07	vm08
num-conns	1000000	1000000	1000000	1000000
num-calls	1	1	10	10
rate	1500	1500	1500	1500
mcbl	10	6	1299	1507.5
conn	127515	127024	124373.25	124292.25
reqeust	68308	70859	42543.75	43148.75
reply	19465	19244	2785	3392
time	675.23	674.358	674.57725	675.05725
conn_rate	188.8	188.4	184.375	184.125
mincon	17.9	39	837.375	778.325
avg	3371.6	3341.4	4440.325	4332.675
maxcon	8233.9	8420.6	8914.6	8884.275
median	2935.5	2638.5	4409.25	4235.5
stddev	1727.5	1635.8	1930.6	1880.35
connect	1468.7	1468.9	1674.95	1600.25
connlen	1	1	1	1
req_rate	101.2	105.1	63.05	63.925
req_size	64	64	64	64
min_rep_rate	3.2	10.6	0	0
avg_rr	28.8	28.5	4.125	5
max_rr	111.2	59.2	29.9	46.325
stddev_rr	23.7	10.6	4.95	6.75
response	2157.9	2043.9	3047.975	2915.975
transfer	4.5	4.8	5.375	4.8
header	197	197	197	197
content	5043	5043	5043	5043
user	0.7	0.8	0.4	0.4
system	99.1	99.2	51.7	50.3
net_io	153.8	152.6	25.075	29.7
err_total	980535	980756	1000000	1000000
cli_timo	108050	107714	121587.75	120900.25
fd-unavail	872485	872976	875626.75	875707.75
connrefused	0	0	0	0
sock_timo	0	0	0	0
connreset	0	0	2785.5	3392

meantime, the CPU utilization is 27.04% during the application execution.

Thirdly, we run 3 VMs (vm04, vm05 and vm06) and in each VM we run the same 4 prime search processes, and the average power is 345W. Therefore a VM running 4 prime search processes will consume13.9W. The CPU utilization is 39.3% during the application execution.

Therefore a VM running 4 prime search processes will consume15.5W approximately.

However, when we start 2 VMs that each runs 8 prime search processes, the average power is 334.8W. Therefore one VM consumes almost 15.2W. And the average CPU utilization is 27.5%. This suggests that when there are 2 VMs that can exhaust two processor cores, the maximum power consumption almost keeps constant. This is also because that when one VM is pinned to one processor, the maximum power consumption of the VM is close to that of the processor core.

5.2. Per-VM Power Consumption with I/O Intensive and Web Workloads

We start two VMs(vm05 and vm06) and each VM copies files from outer machines. The average power is 312.3W. Thus each VM consumes 4.45W.

We start two VMs(vm07 and vm08) and run two httperf benchmarks which generate web requests to vm07 and vm08.The results show that one VM receiving httperf requests consumes only 5.2W.

5.3. Per-VM Power Consumption with Mixed Workloads

We start 8 VMs and 4 VMs of them (vm01, vm02, vm03, and vm04) run 4 prime search processes each and one file copy process from other VM. vm01, vm02, vm03 copy files from vm05. vm04,vm07 and vm08 copy files from vm06. We also run two httperf benchmarks which generate web requests to vm07 and vm08. The average power is 391.7W. The peak power is 396.7W.

Since the workloads ends at different time, we only calculate the average power when all the workloads are run simultaneously. The average CPU utilization is 54.8%. The IO wait is 11.4%.

Thus we can get the VM level power consumption model in Eq.2.

$$P_{host} = \sum_{i=1}^{M} P^{i}{}_{vm\_cpu} + \sum_{j=1}^{N} P^{j}{}_{vm\_io} + \sum_{k=1}^{Q} P^{k}{}_{vm\_web} + P_{BASE} \quad (2)$$

where:

p_{vm_cpu} is the power of CPU intensive VM, and $p_{vm\_cpu} = 15.5$;

p_{vm_io} is the power of I/O intensive VM, and $p_{vm\_io} = 4.45$;

p_{vm_web} is the power of mixed loaded VM, and $p_{vm\_web} = 5.2$;

p_{BASE} is the base power of the computing system, and $p_{BASE} = 303.4$.

M, N, Q are the numbers of different VMs with different types of workloads.

Therefore,

$$P_{host} = \sum_{i=1}^{M} P^{i}{}_{vm\_cpu} + \sum_{j=1}^{N} P^{j}{}_{vm\_io} + \sum_{k=1}^{Q} P^{k}{}_{vm\_web} + P_{BASE}$$
$$= 15.5 \times 4 + 4.45 \times 6 + 5.2 \times 2 + 303.4$$
$$= 402.5$$

However, we observed that the peak power of the system is 396.7W and it is less than 402.5W. This is because that when more VMs are multiplexed in the same host machine, the more consolidated they are and hence they can consume less power than the summary of each VM when the VM runs the workloads alone.

6. Experiment Results with PATPro

Based on the system level and VM level power model, we use PATPro to reduce the system's power consumptions. In our implementation, we use the power model to estimate the power consumption of the system and VMs. To provide performance guarantees under power budget, we merge applications into VMs to reduce power consumption while sacrifice the performance (i.e., job completion time) lightly because we found that less VMs results in much less power consumption if we multiplex workload proper according to the applications power characteristics.

We conducted experiments on the former testbed and list the results in Fig.6 and Fig.7.

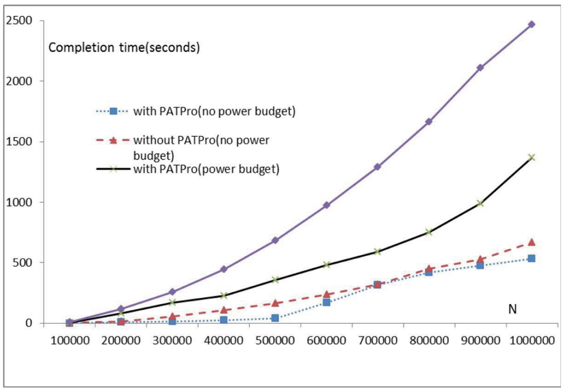


Figure 6 PATPro performance with mixed workloads(job completion of prime search)

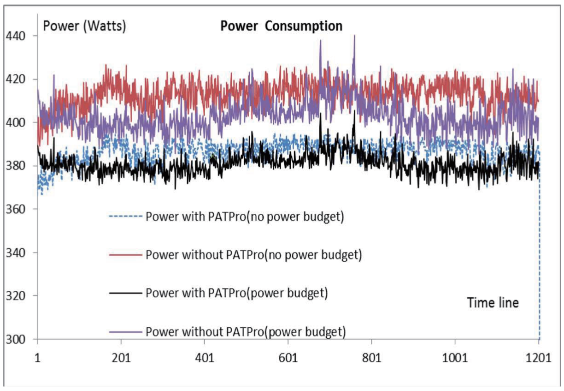


Figure 7 PATPro Power consumption with mixed workloads

In our experiments, PATPro can save 5.41% power in average under power budget and 6.33% without power budget.

7. Conclusions

In virtualized systems, VMs are consolidated on the same hardware and they are usually over-provisioned for best performance guarantees for peak workloads. However, this over-provisioning wastes lots of power and energy. In this paper we use power aware thing provisioning approach to reduce power consumption while providing satisfied performance guarantees and coordinate and tradeoff the power and performance among multiple virtual machines. In our approach, we characterize the power behavior of individual VM and applications and modeling the system power and per-VM power. In our implementation, we use agent for real time power collection and estimation. Our experiments show that PATPro can save 5.41% power under power budget and 6.33% without power budget in average. The results show that our power model can be used in a real virtualized system.

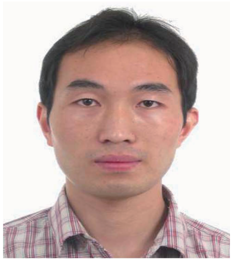
Acknowledgement

The funding supports of this work by Natural Science Fund of China (No. 61003077), Natural Science Fund of Zhejiang Province (Y1101092,LY12E09009), Research Fund of Department of Education of Zhejiang Province (No. GK100800010), and Research Fund of Department of Water Resources of Zhejiang Province (No.RC1218) are greatly appreciated.

References

[1] D. J. Brown and C. Reams., Toward energy-efficient computing, ACM Queue, **8** (2010) 30-43.
[2] L. A. Barroso, U. Holzle, The case for energy-proportional computing, Computer, **40** (2007) 33-37.
[3] C. Jiang, X. Xu, J. Wan, J. Zhang, X. You, R. Yu. Power Aware Job Scheduling with QoS Guarantees Based on Feedback Control .Proceedings of 18th IEEE International Workshop on Quality of Service (IEEE IWQoS 2010).
[4] C. Jiang, J. Wan, X. Xu, Y. Li, X. You, D. Yu, Dynamic voltage/frequency scaling for power reduction in data centers: enough or not? Proc. 2009 ISECS International Colloquium on Computing, Communication, Control, and Management(CCCM), (2009) 428-431.
[5] C. Jiang, J. Wan, X. Xu, Y. Li, X. You, Power management challenges in virtualization environments, Communications in Computer and Information Science, **71** (2010) 1-12.
[6] C. Jiang, J. Wan, X. Xu, Y. Li and X. You. Power Management Challenges in Virtualization Environments. Third International DMTF Academic Alliance Workshop on Systems and Virtualization Management. Standards and the Cloud (SVM 2009) (2009) 1-12.
[7] M. Maheswaran, S. Ali, H.J. Siegal, D. Hensgen,R.F. Freund.Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems.In Proceedings of Eighth Heterogeneous Computing Workshop(HCW '99), pp. 30-44.
[8] K. Li. Job scheduling and processor allocation for grid computing on metacomputers. Journal of Parallel and Distributed Computing, **65** (2005): 1406-1418.

[9] A. Kansal, F. Zhao, Fine-grained energy profiling for power aware application design, ACM SIGMETRICS Performance Evaluation Review, **36** (2008) 26-31.
[10] Tridib Mukherjee, Ayan Banerjee, Georgios Varsamopoulos, Sandeep K.S. Gupta, Sanjay Rungta.Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. Computer Networks, **53** (2009): 2888-2904.
[11] X. Wang, M. Chen, C. Lefurgy, and T.W. Keller, SHIP: scalable hierarchical power control for large-scale data centers, Proc. International Conference on Parallel Architectures and Compilation Techniques(PACT'09), (2009) 91-100.
[12] Slo-Li Chu ,Geng-Siao Li ,Ren-Quan Liu . DynaPack: A Dynamic Scheduling Hardware Mechanism for a VLIW Processor,Applied Mathematics & Information Sciences, No. 6-3S (Nov. 2012), PP:983-991
[13] R. Ge, X. Feng, and K. W. Cameron, Modeling and evaluating energy-performance efficiency of parallel processing on multicore based power aware systems, Proc. IEEE International Parallel and Distributed Processing Symposium (IPDPS2009), (2009) 1-8.
[14] Y. Sun, L. F Wanner, M. B. Srivastava, Low-cost estimation of sub-system power, Proc. the Third International Green Computing Conference (IGCC'12) (2012).
[15] Wu Ji-yi, Zhang Jian-lin, Wang Tong, Shen Qian-li. Study on Redundant Strategies in Peer to Peer Cloud Storage Systems,Applied Mathematics & Information Sciences, No. 5-2S (Mar. 2011), PP: 235S-242S.
[16] L. Hu,H. Jin,X. Liao,X. Xiong, H. Liu. Magnet: A Novel Scheduling Policy for Power Reduction in Cluster with Virtual Machines. In: the 2008 IEEE International Conference on Cluster Computing (Cluster 2008) (2008) 13-22.
[17] R. Nathuji, K. Schwan. Virtualpower: Coordinated Power Management in Virtualized Enterprise Systems. ACM SIGOPS Operating Systems Review. **41** (2007) 265-278.
[18] H. Chen, H. Jin, Z. Shao, K. Yu, K. Tian. ClientVisor: Leverage COTS OS Functionalities for Power Management in Virtualized Desktop Environment. In: the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments (VEE09) (2009) 131-140.
[19] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. Bhattacharya. Virtual Machine Power Metering and Provisioning. in ACM Symposium on Cloud Computing (SOCC)(2010).
[20] http://www.vmware.com.
[21] Joulemeter: Computational Energy Measurement and Optimization. http://research.microsoft.com/en-us/projects/joulemeter/.
[22] G. Laszewski, L. Wang, A. Younge and X. He, Power-Aware Scheduling of. Virtual Machines in DVFS-enabled Clusters, in 2009 IEEE Cluster Computing conference (Cluster2009) (2009).
[23] R. Jansen, P. R. Brenner. Energy efficient virtual machine allocation in the cloud. International Green Computing Conference and Workshops (IGCCW2011) (2011).
[24] www.hpl.hp.com/research/linux/httperf/.



Jiang Cong-Feng received his Ph.D. degree from Huazhong University of Science and Technology in 2007. He is now an associate professor in School of Computer Science and Technology in Hangzh-ou Dianzi University, China. He is with the Grid and Services Computing Lab. His current research interests include power aware computing system, cloud computing, and system virtualization.

Zhao Ying-Hui received her Master degree of Engineering from Huazhong University of Science and Technology in 2007. She is now a Lecturer in Department of Hydraulic Engineering, Zhejiang Tongji Vocational College of Science and Technology, Hangzhou, China. Her current research interests include Remote Sen-sing Image Processing and Geographical Information Systems.