Understanding tradeoffs between Power Usage and Performance in a Virtualized Environment

James William Smith, Ian Sommerville
School of Computer Science
University of St Andrews
St Andrews, Scotland
{james.w.smith, ian.sommerville}@st-andrews.ac.uk

Abstract— **The usage rate of hardware resources in a computing system is the single largest influence on power consumption that can be impacted by software. Approaches to energy efficient scheduling on private clouds are not suitable for all situations, i.e. particularly when that approach is to consolidate VMs on as few machines as possible, a strategy that while reducing energy usage also impacts performance and dependability. In this paper, we introduce the concept of Workload Mixes, a technique for assigning tasks to compute nodes with the aim of balancing the trade-off between system properties such as energy consumption and application performance. In this paper workload mixes are demonstrated through a series of experiments to affect both system energy consumption and application performance in a virtualized environment. The exploitation of these effects may lead to an energy efficient, performance and dependability aware scheduler for a private cloud computing platform.**

Virtual Machine power usage and performance; green cloud computing; virtualization; private clouds; energy usage

I. Introduction

Demand for computation is increasing faster than we can reduce the amount of energy required to do units of computation. In 2007, worldwide use of energy that could be directly attributed to IT systems was 1.5% of all energy use [1]. The current annual growth rate is 15%; clearly such a trajectory is unsustainable in the long-term [2]. The move to more efficient, energy-proportional equipment will slow this growth but it is not expected to cancel out the growing demand.

The cost of a single kilowatt-hour of energy is also rising, typically by 15% per annum based on historical data. This means that administrators must consider the operational cost of hardware along with the capital outlay when that equipment is purchased. It is likely that over a 3-5 year lifespan the operational expenditure will exceed the purchase price, mainly driven by energy costs. Organizations are therefore motivated to minimize their energy usage. One approach is to utilize software techniques to reduce the amount of energy consumed by hardware. However, this challenge is not simple. Traditional approaches to software based energy conservation attempt to minimize the number of servers used and powered down those that are idle, an approach that can impact system performance and dependability.

In the simplest terms, there are two approaches to traditional scheduling algorithms for a cloud; 'spread' and 'stack'. Spread balances workload across the available machines, ensuring no machine is overworked and therefore achieving the best possible application performance. Round-robin assignment would be an example. Stack is the opposite, where jobs are stacked on the least possible number of machines with the aim of reducing the amount of resources required to achieve the body of work and powering down those compute nodes that are not required. Bin-packing is the traditional approach here.

A bin-packing strategy where work is condensed onto as few machines as possible is not suitable when dependability is a key desired system property. Often administrators will deploy redundant copies of software applications and balance workload requests to them. In that case, certain tasks must be spread throughout a datacenter to ensure that a single hardware failure, such as a corrupt hard drive or faulty router, do not remove a large percentage of the application redundancy.

In addition, over working or repeatedly power cycling hardware may damage and reduce the lifespan of that hardware, particularly hard drives [3].

There is also the question of reaction time when the workload demands are increased, as computational hardware must be powered on and made ready to react to the incoming workload. This may be automated in a complex system deployment but may also involve having staff on standby to react to demand.

Therefore, it may be realistic to assume that reducing power demands in a datacenter by reducing the number of physical servers powered on at once is unlikely to be an acceptable option. Servers, realistically, are usually powered on and operating for essentially all of their recorded life [4]. As servers typically consume at least 50% of their peak power when idle a significant amount of energy is still being consumed. This is power usage is considered to the datacenter "base load".

The aim of our work, therefore, is to minimize power consumption through software so that it is as close to the baseline as possible and rely on other technological changes, such as more power efficient processors, that may come in time for base load reduction.

In such cases, a different approach to scheduling must be adopted, one that can take into consideration the need for workload spreading to maintain dependability. This new approach must attempt to reduce energy consumption as much as possible while still maintaining system requirements and restrictions. Workload mixing is such an approach.

A workload mix is the arrangement of a finite set of jobs on a finite set of computational hardware. The mix of those jobs is made up of their placement decisions and the characteristics of those jobs. It can be considered to be the profile of each job's characteristics, details of the hardware capabilities and the map of jobs to hardware as decided by some scheduling mechanism.

Workload mixing attempts to look for complementary attributes amongst the profiles of virtual machines. A profile is made up of the resource usage data for a running VM. Complimentary profiles are profiles that, when paired, may result in increased performance or reduced energy consumption. The experiments outlined in this paper show that such complimentary attributes are possible, and that by exploiting these attributes it is possible to influence the characteristics of the overall system.

Our approach aims to minimize the power used by a set of active servers, without switching unused servers off. To do this we evaluate the allocation of workloads in different configurations to understand the tradeoffs between power and performance.

This evaluation is conducted over a series of experiments that will now be outlined. As part of our previous work [15, 16], we have developed tools to support decision makers during the adoption of cloud computing in their organizations. One of these is lightweight software based monitoring application, CloudMonitor, which is used to collect data for all experiments in this paper.

The remainder of this paper is divided into four sections: a discussion of related work, a description of five experiments, concluding remarks and an outline of future work.

II. RELATED WORK

Management of resources in cloud computing systems is an emerging area of research. Chen, et al [5] point out that it may be possible to rewrite load balancing algorithms to be more energy aware and introduce the concept of "load-skewing" as opposed to the more traditional approach of "load-balancing". In this context, skewing can be taken to be a workload consolidation approach, of which similar work has proven popular in energy efficient workload allocation research.

Srikantaiah, et al in [6] discuss consolidating applications or tasks on a lower number of physical machines, therefore allowing surplus machines to be switched off by employing powering down techniques. The goal of their work is to keep servers well utilized so that power costs are effectively amortized. This is balanced against over-utilization that can cause internal contentions such as cache contentions, conflicts at the functional units of the CPU, disk scheduling conflicts, and disk write buffer conflicts.

Other examples of consolidation-based approaches to reducing power consumption can be seen here; [7] introduces a switching penalty to incorporate the often ignored and undesirable cost of powering servers on and off. [8] further develops this research to include a dynamic configuration approach to power optimize the virtualized environment. In addition to a consolidation approach, [9] attempts to utilize the varying power states of modern compute servers in order to limit the amount of power being consumed. [10] extends these approaches by adapting machine learning techniques to deal with situations where information is incomplete or imperfect and still attempt to maximize the performance gains close to those that could be achieved with perfect knowledge of the system.

Alternative approaches do exist, such as that suggested by [11], where low power systems are mixed in a hybrid data center with high performance machines. This is impractical for most private cloud installations where the compute nodes are not likely to be so specialized.

Our approach adapts workload allocations by investigating the difference in task characteristics to reduce energy usage and improve performance. Other similar approaches in the field have focused on particular virtual machine weaknesses, such as I/O performance [12] and attempted to adjust scheduling algorithms accordingly. Our work differs by attempting to optimize system performance in a holistic approach, minimizing energy consumption while maintaining or improving application performance.

We do this in a static allocation process, without live migration. Alternative approaches that do employ live movement of virtual machines have been shown to have a negative impact on system performance [13] that can violate SLA's and reduce system dependability. It is for these reasons that our approach focuses on the improvements that can be met from static workload allocation with no real-time dynamic re-allocations.

The question of virtual machine energy power metering has also been addressed [14] to map resource usage to a power value. In our previous work [15] we showed that it is possible to create a power model for system resource usage data and accurately measure power utilization using software only.

III. EXPERIMENTS

Research was conducted to investigate the effect of workload mixes on energy consumption in a computing environment. If workload mixing can be demonstrated to have an effect, then it may be possible to find an optimal workload mix, which can be consistently demonstrated to require less energy to complete the same amount of work.

To that end, we conducted a series of five experiments, beginning with a null hypothesis experiment to demonstrate that workload mixes do have an effect the power consumption of a system.

The five experiments are detailed in the table below:

Experiment	Description
Null Hypothesis	*Disprove that workload mixes have no effect on energy.*
Virtualization Effect	*Introduce a virtualization layer and confirm that energy is still affected by the workload mix*
Performance vs Energy	*Examine the effect workload mixes have on application performance*
VM Sizes	*Investigate the effect of different VM configurations on performance and energy consumption*
Mixes vs Bin-packing	*Compare the operation of a workload mix and a bin-packing strategy.*

The experiments were all conducted over a fixed period of time to ensure that only the mix, and not job completion time affected power consumption. Jobs were looped for a fixed period and the power consumption of the hosts for that period was measured.

A. Null Hypothesis

The Null Hypothesis, which will be disproved, is that workload mixes have no effect on the energy consumption of a private cloud system.

1) Experiment Design

The experiment involves four tasks of two types executed concurrently on two compute nodes for one hour.

In a virtualized environment such as private cloud system, tasks will typically outnumber computing hardware nodes. In this experiment there are two types of task, one that is CPU intensive and one that is Disk (HDD) intensive. These represent real world work such as a mathematical experiment that is CPU bound and a heavy-use database application that is I/O bound.

The CPU intensive task is designed to generate computational threads that will employ each multi-core CPU on the machine for the time required. The HDD intensive task writes a random integer to a random selection from 10 files, each of which exceeds the CPU and HDD cache size (12MB and 16MB respectively). These workloads are entirely synthetic and are intended to simply exercise the relevant resources for the time required.

The underlying hosts are two hardware nodes, both 2010 model Dell Poweredge R610 Servers with 2 Intel Xeon E5620 Processors, 16GB of RAM and Seagate Savvio 10K 6-Gb/s 146GB Hard Drives. The hosts run Ubuntu 12.04 OS.

Four tasks, two of each type, are executed under two workload mixes. For the first mix tasks of the same type are co-located on a single machine – HDD intensive tasks are located on Machine A and CPU intensive tasks are located on Machine B.

The second mix splits tasks apart and puts one HDD task and one CPU task on each machine.

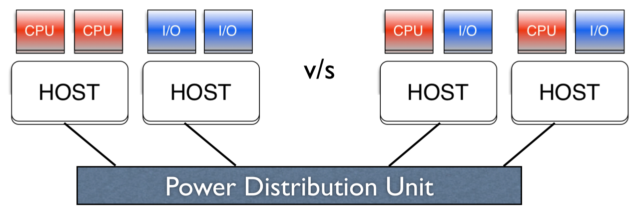

Figure 1. Diagram of first experiment in workload mixing

To limit the interference of mix runs, a square wave usage pattern was adopted. The machines began in an idle state prior to each stage of the experiment and were allowed to return to this idle state for at least 15 minutes between runs. This allow the machines to "cool down" and return to their idle energy consumption levels.

Data for the experiment was collected in real time using the CloudMonitor software application. Hardware utilization data was collected from each Virtual Machine (when used for later experiments) and host. In addition, for increased precision data from a billing-level Power Distribution Unit (PDU) was collected to monitor the hardware power usage.

2) Results

Mix	Host A	Host B	Total
A	196W•h	128 W•h	325 W•h
B	172 W•h	165 W•h	337 W•h

The mean watt-hour values for this experiment are shown in the table above. The experiment was repeated a number of times and on each occasion the individually reported figures varied by no more than a single watt-hour from the mean. Analysis of these results to determine their statistical significance revealed a p-value of 4.903e-07 from a Welch Two Sample T-Test, concluding that it is very unlikely these results could have arisen from a sampling variability. We therefore adopt the conclusion that the workload mix affected the power consumption.

It is clear from the table above that the two mixes produced different energy usage totals. The only variable that was changed was the workload mix, so the mix must have had an effect. The optimal workload mix (Mix B) required 96.4% of the energy required for the non-optimal mix (Mix A).

B. *Virtualisation Effect*

This second experiment introduces a virtualization layer to the same experimental design as before. The aim of this experiment is to confirm that the conclusion of the first experiment, namely workload mixes have an effect on energy consumption, is still valid when a virtualization layer is introduced.

1) *Experimental Design*

Virtual Machines were created using QEMU on Ubuntu Hosts. Each Virtual Machine was assigned approximately one half of the host's physical resources. As each Virtual Machine held only one task, the resources given to each VM were designed as close as possible to mimic the previous experiment. Therefore each virtual machine was assigned 4 CPU cores and 8GB of RAM.

A further experiment (Experiment D) takes into consideration the effect of different virtual machine size on power consumption.

As before, there are two types of task, one HDD and one CPU intensive. The software tasks remained the same, as did the underlying hardware and monitoring solutions.

The experiment involves four tasks (two of each type) executed concurrently in single virtual machines for one hour in two different mix configurations. As before, In the first mix, co-located virtual machines are given tasks of the same type – CPU intensive tasks are located in VMs on Machine A and HDD intensive tasks are located in VMs on Machine B.

In the second mix the similar tasks are split between both machines, therefore, Machine A and B will now have VMs running one CPU task and one HDD task.

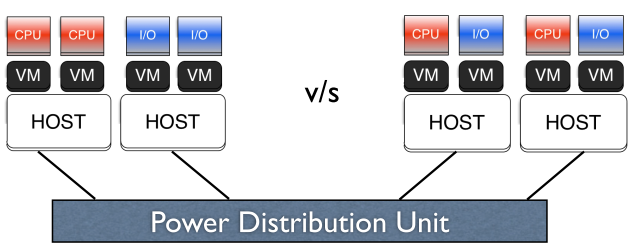

Figure 2 Diagram of workloads inside VMs and their allocations

2) *Results*

Mix	Host A	Host B	Total
A	190W•h	115 W•h	305 W•h
B	159 W•h	161 W•h	320 W•h

The mean watt-hour values for this experiment are shown in the table above. The experiment was repeated a number of times and on each occasion the individually reported figures varied by no more than a single watt-hour from the mean.

The same statistical analysis from the previous experiment was applied, giving a p-value of 0.001012 from the T-Test that it is very unlikely, about a one in one thousand chance, these results could have arisen from a sampling variability.

It is clear from the table above that the two mixes produced different energy usage totals. The only variable that was changed was the workload mix, so the mix must have had an effect. The optimal workload mix (Mix A) required 95.3% of the energy required for the non-optimal mix (Mix B).

As virtualization is a key characteristic of Private Cloud Computing platforms, these results show that workload mixing can be used to affect the energy consumption of such systems.

C. *Performance vs Energy*

This third experiment used the Phoronix Benchmark suite in place of the synthetic workload generating applications. Phoronix measures real world applications such as *gzip* and provides scores for their performances. This will allow us to compare the effect of mixes on both energy and performance.

The aim of this experiment is firstly to investigate if the previous conclusions regarding workload mixes are still applicable when real-world applications are used, and secondly, to determine the effect workload mixes have on the performance of those applications.

1) *Experimental Design*

In this experiment there are two types of task, one that is CPU bound and one that is Disk (HDD) bound. The CPU bound task is a benchmark of the *gzip* application – a standard Linux application that can compress files in memory. The I/O Bound task is a benchmark called *aio-stress* is an asynchronous I/O benchmark created by SuSE. It uses a single thread to consistently read and write a 1024MB test file to and from the hard disk.

As before, we use two VMs with half of the host server's physical resources to maintain consistency with the previous experiments. The underlying hardware and monitoring solutions remain the same.

The experiment involves four tasks (two of each type) executed concurrently in single virtual machines for one hour. In the first mix, co-located virtual machines are given tasks of the same type – CPU intensive tasks are located in VMs on Machine A and HDD intensive tasks are located in VMs on Machine B.

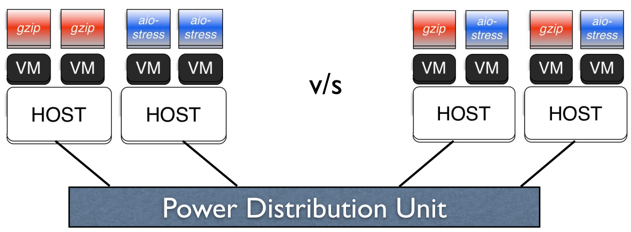

Figure 3 Diagram of real-world applications in two different workload mixes

As in the previous experiment, a square wave usage pattern was adopted.

1) Energy Results

Mix	Host A	Host B	Total
A	127W•h	126 W•h	253 W•h
B	107 W•h	139 W•h	246 W•h

The mean watt-hour values for this experiment are shown in the table above. The experiment was repeated a number of times and on each occasion the individually reported figures varied by no more than a single watt-hour from the mean.

p-value was 0.012 from a T-Test, concluding that it is unlikely these results could have arisen from a sampling variability. We therefore adopt the conclusion that the workload mix affected the power consumption.

It is clear from the table above that the two mixes produced different energy usage totals. The only variable that was changed was the workload mix, so the mix must have had an effect. The optimal workload mix (Mix B) required 97.2% of the energy required for the non-optimal mix (Mix A).

2) Performance Results

Mix	*gzip*	*aio-stress*
A	18.83s	32.02 MB/s
B	18.71	111.45 MB/s

gzip scores are measured as seconds to compress a 2GB binary file. Therefore a lower score is better. *aio-stress* is measured as I/O throughput per second meaning a higher score is better.

The mean performance scores for this experiment are shown in the table above.

Analysis revealed a p-value of 0.035 for *aio-stress* and 0.026 for *gzip*, concluding that the workload mix also affected the application performance scores.

For this particular workload, on these particular servers, one workload mix (B) was demonstrated to reduce the energy required to power the servers and hold or increase the performance. Mix A required more energy and produced a slower *gzip* score and less *aio-stress* throughput. This is an important result as it demonstrates that energy utilization can be improved without loss of performance.

D. VM Sizes

This fourth experiment deviates from the standard pattern of Virtual Machine sizes to investigate the effect of different VM configurations on application performance and energy consumption.

Three Virtual Machine sizes are used:
m1.small - 1 CPU core and 1GB of RAM
m1.medium - 2 CPU cores and 2GB of RAM
m1.large- 4 CPU cores and 8GB of RAM

These sizes are based on the virtual machine sizes offered by the St Andrews Cloud Computing co-laboratory's (StACC) OpenStack private cloud.

1) Experimental Design

This experiment was conducted as a repeat of a co-located mix similar to the previous experiment with varying virtual machine sizes employed.

As before, we use two types of phoenix benchmark test suite applications, *gzip* and *aio-stress*, to give examples of different workloads. The underlying hardware and monitoring solutions remain the same as the previous experiments.

The experiment involves four tasks (two of each type) executed concurrently in single virtual machines for one hour. Only one mix is used; co-located virtual machines are given tasks of the same type – CPU intensive on Machine A and HDD intensive tasks on Machine B. This layout is the same for each virtual machine size.

2) Energy Results

VM Size	Host A (*aio-stress*)	Host B (*gzip*)	Total
Small	109.7W•h	139.7 W•h	249.4 W•h
Medium	110W•h	138W•h	248 W•h
Large	107 W•h	139 W•h	246 W•h

The mean watt-hour values for this experiment are shown in the table above. The size of the virtual machine had a very small impact on the amount of energy used and instead, the type of work performed dominated the power profile as the previous experiments had demonstrated.

3) *Performance Results*

VM Size	*aio-stress*	*gzip*
Small	17.55 MB/s	27.63s
Medium	18.20 MB/s	27.44s
Large	96.41 MB/s	18.73s

The mean performance scores for this experiment are shown in the table above. p-value was 7.651e-07 for *gzip* and 0.018 for *aio-stress*, concluding that the virtual machine sizes affected the application performance scores.

It is intuitive that a VM with more resources will be able to give better application performance at the expense of requiring more energy to run.

However, these results show that energy usage is almost constant across all Virtual Machine sizes, yet performance results vary considerably, with larger VMs giving increased performance.

This would suggest that constraining VM size for energy-saving purposes in a private cloud is ineffective, especially given the large detriment to performance that is experienced. In a situation where a larger virtual machine size is chosen the superior performance may mean a finite piece of work is completed quicker, thereby reducing the energy consumption in a different manner.

E. Mixes vs Bin-packing

For the fifth and final experiment, we compare the operation of a workload mix where four virtual machines containing four tasks in a spread configuration are compared against a stacked bin-packing configuration where the virtual machines are placed on a single server.

We take the results of the m1.medium virtual size from the previous experiment and compare those with the operation of a bin-packing configuration of four m1.medium VMs on a single host.

1) *Experimental Design*

This experiment utilized four m1.medium VMs on a single compute node and compared the results to that from the previous experiment.

As before, we use two types of phoenix benchmark test suite applications, *gzip* and *aio-stress* to give examples of different workloads. The underlying hardware and monitoring solutions remain the same as for previous experiments.

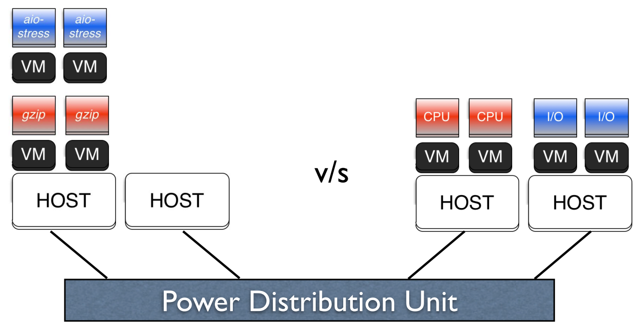

Figure 4. Diagram of stacked, bin-packed workload mix against a spread configuration

As in the previous experiment, a square wave usage pattern was adopted.

2) *Energy Results*

Allocation	Host A	Host B	Total
Bin-packed	124 W•h	74.7W•h	198.7 W•h
Workload Mix	138W•h	110W•h	248 W•h

Analysis of these results revealed a p-value of 0.018, concluding that it is unlikely that these results could have arisen from a sampling variability. We therefore adopt the conclusion that the bin-packing layout had a different affect on power consumption than the alternative mix layout.

The bin-packing layout used, on average, 19.9% less energy than the workload mix alternative.

3) *Performance Results*

Allocation	*aio-stress*	*gzip*
Bin-packed	6.55 MB/s	45.33s
Workload Mix	18.20 MB/s	27.44s

The mean performance scores for this experiment are shown in the table above. Analysis revealed a p-value of 0.00034 for *gzip* and 0.0087 for *aio-stress*, concluding that the bin-packing layout affected the application performance scores compared to the workload mix effect.

The bin-packing stacked virtual machine layout impacts application performance in a negative way but requires less energy to execute. It is intuitive that a host with stacked VMs contending for resources will result in worse application performance than a configuration where the virtual machines are spread out.

The bin-pack *aio-stress* throughput was only 35% of the mix spread and required 65% longer to complete the *gzip* trials on average. However it required only approximately 80% over the energy over the same time period.

Bin packing requires less energy to run than a spread layout. If workload mixes are to be employed, it must be accepted that an more optimal energy usage pattern can be achieved using a bin-packing scheduling algorithm, but at significant detriment to system performance and dependability.

IV. CONCLUSION

This work is presented as a real world validation of a theoretical framework which could be adapted going forward into a concrete implementation. The actual effects of various mixes on different hardware would need to be investigated if that hardware changed, but in future work this would be done automatically for heterogeneous compute nodes.

We have demonstrated that Workload Mixes can affect application performance and system energy. Workload mix scheduling is a spread approach to assigning work that takes into consideration how virtual machine placement will effect both energy and performance.

Traditional approaches such as bin packing may use less energy but are not concerned with system performance and dependability. If such attributes are important to system administrators, workload mixes can help to optimize workload layout so that energy usage is minimized while still meeting these requirements.

All tasks, scripts and data from these experiments are available to download at http://jws7.net/workload-mixes

V. FUTURE WORK

The next stage in our work is to develop a cloud scheduler that exploits VM workload characteristics to mix jobs in such a way that system energy usage is minimized while maintaining VM performance and application dependability.

Such a scheduler will automatically adapt mixes depending on the read-world application performance and system energy use. This would allow it to be deployed on heterogeneous hardware and take advantage of gains that may be found when workload allocations are studied.

ACKNOWLEDGMENT

The authors would like to thank Ali Khajeh-Hosseini, Jonathan Ward and Graham Kirby for their comments, help and advice. We would also like to thank the UK's Large Scale Complex IT Systems initiative and the Scottish Informatics and Computer Science Alliance for funding this work.

REFERENCES

[1] Report to Congress on Server and Data Center Energy Efficiency, Public Law 109-431, U.S. Environmental Protection Agency, ENERGY STAR Program, August 2, 2007

[2] Green Computing: Reducing the Carbon Footprint of Information Processing Systems in an Energy-Constrained World, Massoud Pedram, University of Southern California, Dept of Electical Enginering, ICCCN 2009, August 3rd 2009

[3] PARID: A Gear-Shifting Power-Aware RAID, Charles Weddle, Mathew Oldham, Jin Qian, An-I Andy Wang, Florida State University, 5th USENIX Conference on File and Storage Technologies, p 245-260, February 13–16, 2007, San Jose, CA

[4] Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX conference on File and Storage Technologies* (FAST '07). USENIX Association, Berkeley, CA, USA, 2-2.

[5] Gong Chen, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao, and Feng Zhao. 2008. 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*(NSDI'08), Jon Crowcroft and Mike Dahlin (Eds.). USENIX Association, Berkeley, CA, USA, 337-350.

[6] S. Srikantaiah, "Energy Aware Consolidation for Cloud Computing," USENIX HotPower'08: Workshop on Power Aware Computing and Systems at OSDI, 2008

[7] V. Petrucci, O. Loques, and D. Mosse. A dynamic conguration model for power-efficient virtualized server clusters. In 11th Brazillian Workshop on Real-Time and Embedded Systems (WTR), 2009

[8] V. Petrucci, O. Loques, B. Niteroi, and D. Mosse. Dynamic configuration support for power-aware virtualized server clusters. In WiP Session of the 21th Euromicro Conference on Real-Time Systems. Dublin, Ireland, 2009.

[9] A. Verma, P. Ahuja, and A. Neogi. Power-aware dynamic placement of hpc applications. In ICS '08: International Conference on Supercomputing, pages 175-184, New York, NY, USA, 2008. ACM.

[10] Josep Ll. Berral, Inigo Goiri, Ramon Nou, Ferran Julia, Jordi Guitart, Ricard Gavalda, and Jordi Torres. 2010. Towards energy-aware scheduling in data centers using machine learning. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking* (e-Energy '10). ACM, New York, NY, USA, 215-224. DOI=10.1145/1791314.1791349

[11] B.-G. Chun, G. Iannaccone, G. Iannaccone, R. Katz, L. Gunho, and L. Niccolini. An Energy Case for Hybrid Datacenters. HotPower, 10 2009.

[12] Hwanju Kim, Hyeontaek Lim, Jinkyu Jeong, Heeseung Jo, and Joonwon Lee. 2009. Task-aware virtual machine scheduling for I/O performance.. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments* (VEE '09). ACM, New York, NY, USA, 101-110. DOI=10.1145/1508293.1508308

[13] Voorsluys, William.; Broberg, James.; Venugopal, Srikumar.; Buyya, Rajkumar.; "Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation", Lecture Notes in Computer Science, Cloud Computing, 2009-01-01, pages 254-265

[14] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. 2010. Virtual machine power metering and provisioning. In *Proceedings of the 1st ACM symposium on Cloud computing* (SoCC '10). ACM, New York, NY, USA, 39-50. DOI=10.1145/1807128.1807136

[15] Smith, J.W.; Khajeh-Hosseini, A.; Ward, J.S.; Sommerville, I.; , "CloudMonitor: Profiling Power Usage", *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on* , vol., no., pp.947-948, 24-29 June 2012, doi: 10.1109/CLOUD.2012.112

[16] A. Khajeh-Hosseini, D. Greenwood, J.W. Smith, I. Sommerville. "The Cloud Adoption Toolkit: Supporting Cloud Adoption Decisions in the Enterprise". Software: Practice and Experience. 2011