Energy Efficiency and Server Virtualization in Data Centers: An Empirical Investigation

Yichao Jin and Yonggang Wen
Divison of Computer Communications
School of Computer Engineering
Nanyang Technological University
Email: {yjin3, ygwen}@ntu.edu.sg

Qinghua Chen
Dept. of Information Technology
Yangtze Delta Institute of
Tsinghua University, China
Email: chenbill@gmail.com

Abstract—**With a growing concern on the considerable energy consumed by data centers, research efforts are targeting toward green data centers with higher energy efficiency. In particular, server virtualization is emerging as the prominent approach to consolidate applications from multiple applications to one server, with an objective to save energy usage. However, little understanding has been obtained about the potential overhead in energy consumption and the throughput reduction for virtualized servers in data centers. In this research, we take the initiative to characterize the energy usage on virtualized servers. An empirical approach is adopted to investigate how server virtualization affects the energy usage in physical servers. Through intensive data collection and analysis, we identify a fundamental trade-off between the energy saving from server consolidation and the detrimental effects (e.g., energy overhead and throughput reduction) from server virtualization. This characterization lays a mathematical foundation for server consolidation in green data center architecture.**

I. INTRODUCTION

With the accelerating adoption of cloud computing, data centers, which serve a pivot role in cloud computing, are consuming a significant amount of energy. It was estimated data centers in the USA consumed 61 billion kWh electricity in 2006. This amount has doubled the consumption in 2000, and was expected to double again by the end of 2011 [1]. Hence, how to tame the explosive power consumption has been agreed as one of the key challenges for future data centers.

Recently, server consolidation is touted as an effective way to improve the energy efficiency for data centers. In this approach, applications running on multiple servers can be consolidated into one server via virtualization. As a result, idle servers in data centers could be turned off to reduce energy usage by server virtualization. It has been demonstrated that, by optimizing the data center operations via virtualization, up to 20% of energy consumption can be saved in data centers [2]. However, virtualization would also lead to potential hazard effects, such as a possible energy overhead or a possible reduction in maximum throughput. These detrimental effects, if not well understood, could offset the benefits of server virtualization. Therefore, clear understanding and precise modeling of server energy usage in data centers will provide a fundamental basis for data center operational optimizations.

In this paper, we investigate the impact of server virtualization on energy usage for data centers, with an objective to provide insights to optimize data center operations. We adopt an empirical approach to measure the energy consumed by servers under different virtualization configurations, including a benchmark case (i.e., physical machine) and two alternative hypervisors (i.e., Xen and KVM), in which a physical server is virtualized into multiple virtual machines (VM). We obtain statistics for CPU usage, task execution time, power and energy consumption, under both local computing intensive tasks and networking intensive traffic, corresponding to two important resources, computing and networking in cloud computing and data centers [3].

Our in-depth analysis on empirical characterization generates fundamental insights of server energy usage in the context of server virtualization, including:

- A significant amount of power is consumed even when the server is idle, thus opening an opportunity for server consolidation in data centers for reducing energy cost.
- Virtualized servers consume more energy than physical ones, for both computing and networking intensive traffic. The energy overhead from virtualized servers increases as the utilization of physical resources increases.
- The energy overhead resulted from server virtualization highly depends on the hypervisor used, which in turn is determined by the software architecture of the hypervisor.
- For a given traffic load, the energy cost can be minimized by launching an optimal number of virtual machines.
- Physical servers, if a multi-core optimization mechanism is absent, could consume more energy than virtualized servers, when running multi-process applications.

These empirical insights further suggest a fundamental trade-off in data center virtualization. This trade-off lies between the energy saving from shutting down idle servers and the detrimental effects (i.e., the energy overhead and the throughput reduction from hypervisor) due to virtualization.

The rest of this paper is structured as follows. Section II shows the related work. Section III describes the virtualization models. Section IV illustrates our detailed experimental setup. Section V presents the empirical results and the engineering insights. Section VI explains the fundamental trade-off in server virtualization and its application to server consolidation. Section VII concludes this work.

II. RELATED WORK

[4] presents a quantitative analysis of Xen and KVM, focusing on the overall performance, isolation and scalability of virtual machines running on them. An extensive empirical measurement on such evaluation was conducted in [5].

The architectural principles for energy-efficient management of resource allocation policies, scheduling algorithms were discussed in [6], demonstrating the immense potential to offer significant cost savings under dynamic workload scenarios. [7] reviewed the methods and technologies currently used for energy-efficient operation of computer hardware and network infrastructure. It also concluded that cloud computing with virtualization can greatly reduce the energy consumption.

Another area of study in green computing concentrates on the energy cost style. [8] characterized the energy consumption pattern of desktop PC. [17] created a power simulator for web serving workloads that is able to estimate CPU energy usage.

However, few researches are conducted for virtualized data centers and the tradeoffs in terms of energy consumption and service capability. So the novelty of this research is the empirical investigation from such perspective.

III. SERVER VIRTUALIZATION MODEL

This section compares two leading virtualization models, Xen and KVM, including their implementation mechanisms in I/O, CPU and networking resource management.

A. Virtualization Model Overview

Hypervisor, also refers to virtual machine manager (VMM), is one of the virtualization techniques that allows multiple operation systems (OSs) to run concurrently on one server. Existing hypervisors, based on their relationship with the hardware platform, can be classified into two alternative types [9]. Specifically, Xen is a type-1 hypervisor, which directly interfaces with the underlying hardware and uses a privileged domain 0 to manage other kernel modified guests [10]. KVM is designed as a type-2 hypervisor, in which the virtualization interface acts the same as the actual physical hardware [11].

B. Virtualized I/O Mechanism

Xen exposes a hypercall mechanism (also known as paravirtualization interface), that all guest OSs have to be modified to perform privileged operations (e.g., updating page table). Besides, event notification mechanism is proposed to deliver virtual interrupts derived from real device interrupts to VMs.

Oppositely, KVM typically uses full virtualization [11]. Guest OSs above KVM don't need to change, and they appear as normal Linux processes. When I/O instructions are issued by guest OSs, a process context switch in the hypervisor is enabled to allow I/O signals passing through.

The difference in virtualized I/O mechanisms for Xen and KVM directly impacts the energy consumption for virtualized servers. Xen allows guests to make system calls without invoking the kernel of host OS, whereas KVM incurs additional kernel operations to support I/O behaviors. The additional operations will probably translate to extra CPU cycles and memory access, which further lead to extra energy usage.

C. Virtualized CPU Model

The default CPU scheduler in Xen is a Credit-Based scheduler [13]. This scheduler, running on a separate accounting thread in the host, allocates certain credits to each virtualized CPU (VCPU). When a VCPU runs, it consumes its credit. Once the VCPU runs out of the credit, it only runs when other more thrifty VCPUs have finished their executing [14].

KVM uses the regular Linux CPU and memory scheduler [15]. It is known, by default, KVM makes use of a Completely Fair Scheduler (CFS) to treat every guest as a normal thread. Each task running on KVM has a priority, which determines the amount of CPU cycles and memory allocation on it.

In spite of the different mechanisms, the objectives of these two CPU schedulers are to balance global load on multi-cores to achieve better allocation, which will be verified by us.

D. Virtualized Networking Model

Xen, by default,uses bridging and virtual firewall router (VFR) within domain 0 to allow all domains to appear on the network as individual hosts. In comparison, KVM uses network TUNnel/network TAP based on networking virtualization in Linux kernel to create virtual network bridge and routing. The bridge essentially emulates a software switch.

This difference could lead to more energy consumed by KVM than that by Xen, when they are exposed to networking-intensive tasks, due to more software operations required by KVM. On the other hand, Xen takes advantage of its modified interface that needs relatively less software participation.

IV. EXPERIMENTAL SETUP

This section describes the setup for our measurements on energy usage in the context of server virtualization.

A. Physical Setup

Figure 1 illustrates the physical setup of our experiment, which consists of three identical servers. The machines under test are Inspur 3060 servers, each of them contains a quad-core Intel 2.13 GHz Xeon processor, 2 GB RAM, 500 GB hard disk and an 1 Gigabit Ethernet card. All of them are connected to a test intranet over a D-link GDS-1024T 1000 Base-T switch. Kill-A-Watt power meters, with a standard accuracy of 0.2%, measure the energy usage of each server. CentOS 5.6-final-x86_64 with Linux kernel 2.6.18 is used as our OS platform for both host and guest systems. Xen 3.0.3 and KVM 83 are installed on server B and C respectively. The 3 guest virtual machines are allocated with 4 VCPUs, 512 MB RAM and 50 GB image. We leave all the software parameters intact.

Our experiment is controlled by another computer, which is also connected to the intranet as a monitor to obtain the benchmarking time, the energy and power consumption. And each server is responsible for gathering its average CPU usage.

B. Test Case Design

We begin with collecting the background energy consumption when all the servers are idle. Following that, a set of local and network traffics are launched to stress all three servers. Detailed test cases are explained as follows.

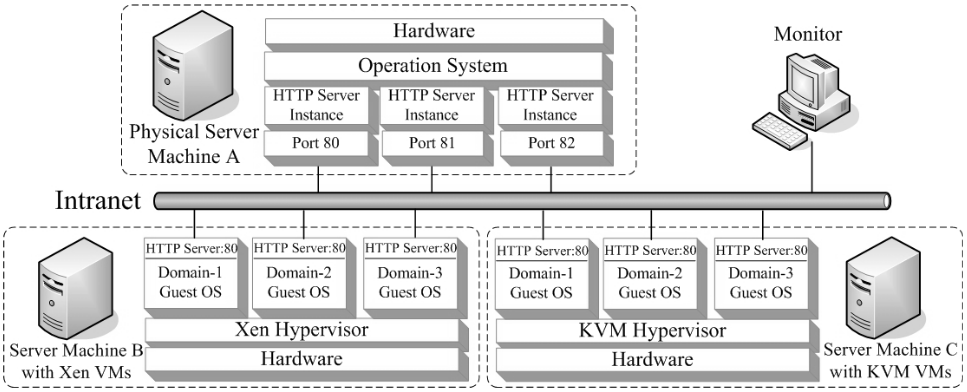


Fig. 1: Experimental Setup: 3 configured systems, a non-virtualized server, a Xen-virtualized server, a KVM-virtualized server

1) Local Computation Benchmark: bc command in Linux is used to calculate the constant π into an accurate level (100,000 digits after the decimal point). We simultaneously run multiple instances to generate computing-intensive loads.

Five cases with the number of concurrencies ranging from 3 to 7, are tested for 2 or 3 active domains. On the physical machine, all the instances are executed over the same OS, while on the virtualized servers, the concurrent instances are distributed evenly across all the active domains.

2) Http Request Benchmark: The network-intensive traffic benchmark is simulated through HTTP requests, similar to the web-server workload benchmark used in [16].

On the server side, three Apache servers are configured on all servers under test. On the physical one, the three HTTP servers are executed on 3 TCP ports for traffic segregation. For virtualized ones, the three instances are evenly distributed across all active guest domains for 2 or 3 active domains. The same TCP ports are used for fair comparison. The contents stored on the HTTP servers are 1000 unique files retrieved from a commercial web site, with mean file size of 10.8 KB.

On the client side, we use *ab* (Apache Bench) tool to simulate real web traffic. Three clients are configured to generate http GET requests at specific rates, each of which dedicates to one Apache server instance. Every client sends 5000 requests for each file. In this test profile, the overall size of data transferred can be as large as approximately 150 GB.

Our experiment generates various request rates to scope the energy usage as a function of the workload. Specifically, 2500 reqs/s, 5000 reqs/s, 10000 reqs/s and 15000 reqs/s are used to simulate low, moderate, high and peak web traffic loads, suggested by the workload of real commercial web server [17].

V. EMPIRICAL RESULTS AND FUNDAMENTAL INSIGHTS

This section presents our empirical findings from the experiments, which further generalize a few fundamental insights.

A. Background Energy Consumption

In figure 2, we plot the background power consumption of the three configured servers, where the bar indicates the average power consumption, and the line refers to the fluctuation. The following findings are obtained from figure 2.

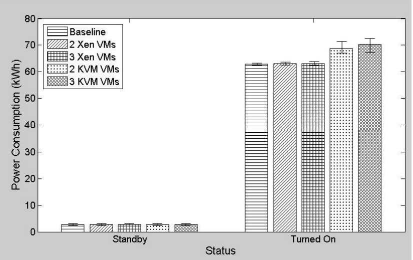


Fig. 2: Background Power Consumption

Finding (a): All the servers consume the same power (about 2.8W) when turned off, but plugged into the power supplies.

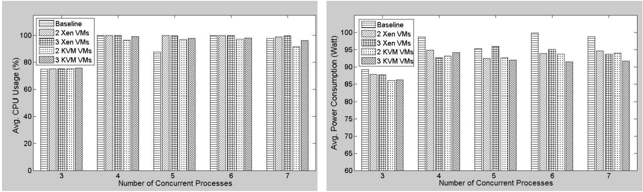
Finding (b): When servers are turned on but active VMs stay idle, the power consumption on different servers varies from each other. The power consumed by the Xen-based server and the physical server are almost the same. In particular, 63.1W (3 active VMs) and 63.0W (2 active VMs) of power is consumed by the Xen-based server, which is 0.47% and 0.32% more than 62.8W consumed by the physical server. While the KVM-based server incurs a much higher overhead, consuming 70.1W (11.6% overhead) for 3 active VMs and 68.8W (9.55% overhead) for 2 active VMs. Moreover, the power usage of the KVM-based server fluctuates within a wider range.

The *Finding (b)* can be explained by the different impact on CPU and RAM usage by Xen and KVM. The CPU utilization of the idle physical server is generally less than 0.3 %, compared to 0.7-0.8 % for the Xen-based server, and 0.8%-2.6% for the KVM-based server. The extra CPU usage of virtualized servers accounts for a portion of the energy overhead. The rest energy overhead for the KVM-based server can also be attributed to the large memory footprint in KVM, as indicated by the results of memory test in [5].

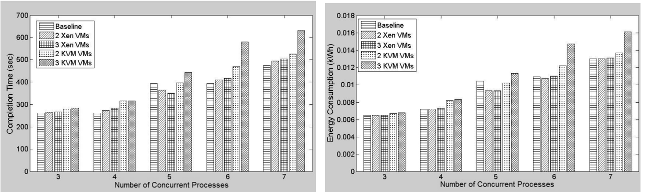
B. Local Calculation Benchmark

Results from local computation benchmark are depicted in figure 3 and 4. Observations on them are given as follows.

Finding (c): The virtualized server could consume less energy than the physical server does. Specifically, when 5 instances are executed (the instance number is one more than



(a) CPU Usage Comparison (b) Power Consumption Comparison



(c) Completion Time Comparison (d) Energy Consumption Comparison

Fig. 3: Statistics for Local Task Benchmark

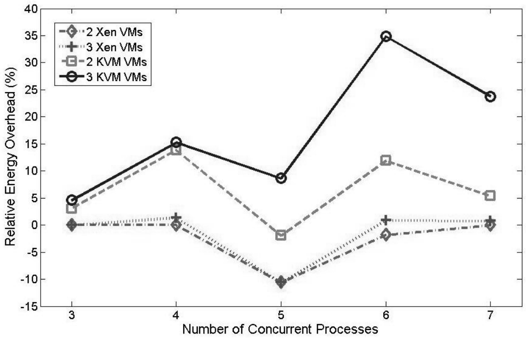
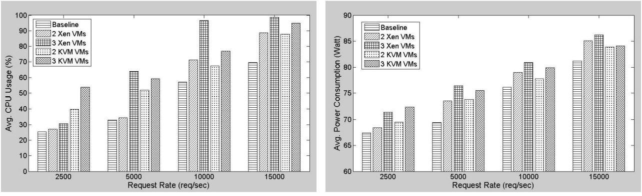


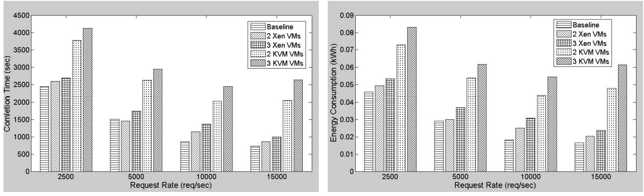
Fig. 4: Relative Energy Overhead of Local Task Benchmark

the number of CPU-cores), the energy overhead is negative for Xen-based servers, as the valley point shown in figure 4.

Such an observation can be understood as the inter-play between the concurrent processes and the CPU cores in a multi-core server. For the physical server, we observe that 4 instances are finished first and the last instance is completed much later. This is further verified by the observation that the CPU usage maintains nearly 100% until the first 4 instances are completed, and afterwards it drops to around 25%. In comparison, in the virtualized servers, all the instances are completed almost at the same time. The CPU usage on the Xen-based server maintains at a high level of 99.8%, compared to 87.7% for the physical server. In this case, the Xen-based server, either running 2 or 3 VMs, takes around 10% less time and consumes 11% less energy than that of the physical server. For the KVM-based server, the advantage of the CPU scheduler is reversed by the extra penalty of hypervisor in most cases, except for the case when 2 active VMs are configured, resulting in a saving of 2% energy than the physical server. This finding suggests that, if there is no binding between running processes and CPU-cores, native operation system can not truly take advantage of multi-core architecture; in contrast, virtualized systems, based on either Xen or KVM, is able to partition computing resources into smaller pieces to achieve better resource allocation across active VMs to save energy.



(a) CPU Usage Comparison (b) Power Consumption Comparison



(c) Completion Time Comparison (d) Energy Consumption Comparison

Fig. 5: Statistics for HTTP Benchmark

Finding (d): The KVM-based server consumes more energy than that of the Xen-base server. For example, when processing 7 parallel tasks, 2 KVM VMs consumes 5.4% energy more than that based on 2 Xen VMs, and the gap reaches 23% between 3 KVM VMs and 3 Xen VMs. It is because the KVM hypervisor consumes more CPU cyles and occupies higher memory footprint, compared to the Xen hypervisor. The additional requirement translates into higher energy consumption.

Finding (e): The number of active VMs affects the energy usage for the KVM-based server. Particularly, when configuring 3 active VMs, the KVM-based server consumes more energy than that consumed by 2 active VMs configured on the same server. This can be attributed to the frequent Lock Holder Preemption (LHP) mechanism, investigated by [18]. A guest VCPU in the KVM-based server may be preempted when the host de-schedules the VCPU threads. If the preempted VCPU is running in a critical section, the lock will be held a certain time from the perspective of the guest VM. The probability of LHP is higher with more active VMs. Once LHP occurs, CPU resources are simply wasted in the lock holding period, which in turn increases the task completion time. So the average power consumption for the KVM-based server with 3 active VMs is the lowest, but the task completion time is the longest.

C. HTTP Request Benchmark

Results from the HTTP benchmark are plotted in figure 5 and 6. Some findings based on them are highlighted as follows.

Finding (f): The virtualization overhead for network-intensive traffic is much larger than that for computing-intensive traffic. For the Xen-based server, the energy overhead for computing-intensive traffic is less than 5%, while the overhead for network-intensive traffic could rise up to 70%. The same situation happens to the KVM-based server.

The cause of this finding is at least two-fold. First, for networking traffic, the CPU usage of virtualized server is much

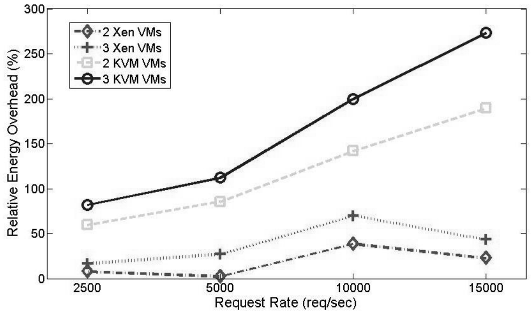
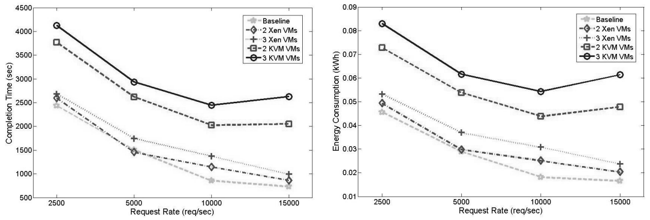


Fig. 6: Relative Energy Overhead of Networking Benchmark



(a) Completion Time Curve (b) Energy Consumption Curve

Fig. 7: Elapsed Time and Energy Usage for HTTP Benchmark

higher than that of the native server; while for local computing tasks, the CPU usages of all the servers are almost equal. This difference suggests that dramatic CPU cycles are budgeted for VFR/VIF in Xen or TUN/TAP in KVM. Second, according to [19] the probability of Lock Hold Preemption (LHP) for I/O-intensive workloads is 39% for virtualized server. The high frequency of LHP translates into high energy cost.

Finding (g): The energy overhead for the virtualized server is correlated with the number of active VMs. For 3 active KVM VMs, the energy overhead is around 1.5 times higher than that for 2 active VMs; similarly, 3 active Xen VMs consumes almost twice overhead of that for 2 active VMs. Moreover, the gap for the KVM-based server is higher. For example, in the case of 15,000 req/s, the overhead gap between 3 active VMs and 2 active VMs for KVM is more than 80%; while it is around 20% for Xen.

Finding (h): The network throughput for the KVM-based server reaches its maximum between 10,000 reqs/s and 15,000 reqs/s. Figure 7 makes this finding quite clear that when the request rate is 15,000 reqs/s, the KVM-based server takes longer time to complete the task and thus consumes more energy, compared to the case of 10,000 reqs/s. As a comparison, the task completion time and the energy cost for the physical machine and the Xen-based server monotonically decrease as the request rate increases up to 15,000 req/s.

This observation is largely due to the extra memory footprint for KVM. In Apache server, each serving request takes certain memory. The maximum number of requests that can be served simultaneously is thus proportional to the amount of available resources. For the case of KVM, the extra memory footprint shrink the amount of available memory for request serving.

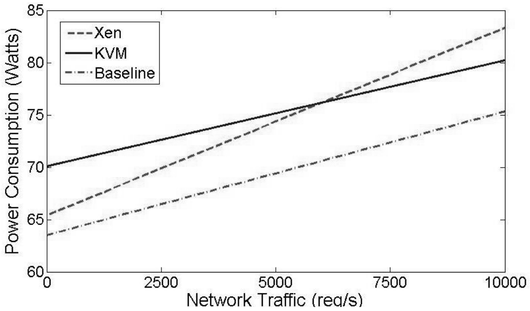


Fig. 8: Expansive Energy Overhead due to Virtualization

Finding (i): The marginal power consumed by the server under different load conditions is limited, compared to the power consumed when the server is idle. Specifically, the additional power consumed by the server under different level of networking requests is at most 37.3 % against the idle state, and the maximum additional power consumption for the local computation benchmark is 57.6 %. Moreover, the marginal power consumption is highly correlated with the CPU usage as observed. As a result, our experiment verifies a previously power consumption model for the server, in which the power consumption of the server can be viewed almost as an affline function of CPU usage with the idle power consumption as the y-intercept [12]. It is desirable for the y-intercept to be as small as possible, to achieve an energy-proportional architecture.

Finding (j): The energy overhead for virtualized servers is expansive. As shown in figure 8, where the lines are curved by one degree polynomial fitting based on the power consumption of different configurations, the power gap between the baseline and the virtualized servers for both Xen and KVM increases as the throughput increases, before the maximum throughput of the KVM-based server is reached. When no network traffic occurs, the gap between Xen and baseline is around 1%(0.8 Watt), and the gap between the KVM-based server and the baseline server is approximately 10% (6.9 Watt). When the throughput grows to 10000 reqs/s, the gap becomes 15.2% (10.8 Watt) for Xen and 11.2% (7.9 Watt) for KVM.

D. Fundamental Insights

Based on those empirical findings, we present the following insights on the impact of server virtualization on energy usage.

1) The server is still far from energy-proportional. The idle server even consumes approximately two thirds of the energy when its computing resource is fully occupied. As a result, it will be advantageous to consolidate applications from multiple servers to one and turn off those idle servers to save energy.

2) The virtualized server in general consumes more energy than the physical server does. The energy overhead for virtualized servers increases as the resource utilization increases. When the virtualized servers are idle, Xen incurs nearly 1% energy overhead, and KVM contributes around 10% extra energy cost. For networking benchmark, Xen's virtual firewall router, virtual network interface, and virtual event mechanism add an energy overhead ranging from 2.8% to 70.2% for

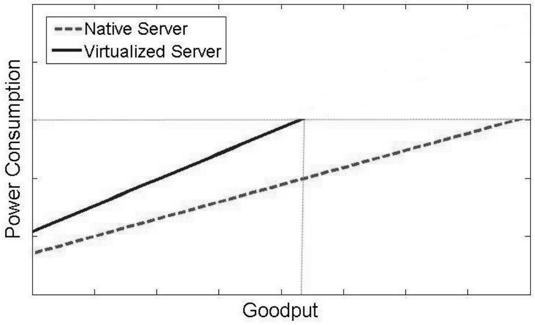


Fig. 9: Fundamental Trade-Off In Server Virtualization

different workloads and VM configurations; and KVM's Linux kernel based virtual network bridge and x86 VTx invoking results in an energy overhead between 59.6% and 273.1% for various combinations of configuration and workload.

3) The 2 types of hypervisors exhibit different characteristics in energy usage for various tasks. KVM consumes more energy (on average 12%) than Xen under the same test case. Specifically, KVM embeds the native Linux kernel with virtualization capability, but requires more software operations and accordingly consumes more energy. Moreover, KVM's networking model consumes more energy (on average nearly twice) than that of Xen. On the other hand, the benefit of using KVM is there is almost no modification required to the host OS; while such modifications are necessary for Xen.

4) Energy saving can be achieved by launching an optimum number of virtual machines. In our measurement, the virtualized server with 2 active VMs consumes less energy than the one with 3 active VMs. Specifically, about 20% energy for KVM and 15% energy for Xen on average could be conserved for all cases under networking benchmark, by migrating tasks from one VM to another and turning off the idle VM.

5) When a multi-core server is running multi-process applications, the physical machine could consume more energy than virtualized servers. It is due to a lack of the multi-core optimization in the physical machine. While both Xen and KVM are able to distribute physical CPU cores into virtual CPU cores, avoiding starvation. This demonstrates an essential advantage of virtualization in improving resource utilization.

VI. FUNDAMENTAL TRADE-OFF

Figure 9 presents a fundamental trade-off, which dictates how server consolidation should be designed to reduce energy usage for green data centers. Specifically, there are two competing forces that server consolidation should balance.

On one hand, the energy usage in data centers can be reduced by consolidating applications from multiple servers to one server and shutting down the rest. This is based on our observation of the power consumption model for native server.

On the other hand, for the virtualized servers, there are two detrimental effects would hamper the energy efficiency. First, the hypervisor introduces a potential energy overhead over the physical machine, by occupying system resources for its execution. This overhead is expansive as a function of the 'goodput', which denotes the portion of computation capabilities used for support applications. Second, the maximum supportable goodput for virtualized server is reduced, compared to its native server. The combination of these two detrimental effects would offset the energy benefit of server consolidation. Moreover, the impact of these detrimental effects also depends on the type of hypervisor chosen.

This fundamental trade-off dictates how server consolidation should be designed for green data centers. Specifically, the decision of server consolidation should balance those two competing forces, to reduce the energy usage by data centers.

VII. CONCLUSION

This paper reported an empirical study on the impact of server virtualization on energy efficiency. Through intensive measurements, we obtained statistics for energy usage from native server and virtualized servers with Xen and KVM, as well as a few findings based on our motivations. Finally, we reveal the fundamental trade-off in virtualized servers, which would dictate how server consolidation should be designed and deployed to tame the explosive energy usage in data centers.

REFERENCES

[1] R. Brown, et al., *Report to congress on server and data center energy efficiency: Public law 109-431*. Lawrence Berkeley National Lab, 2008.
[2] Silicon Valley Leadership Group, *Accenture, Data Centre Energy Forecast Report-Final Report*. July, 2008.
[3] P. Mell and T. Grance. *NIST definition of cloud computing*. NIST Special Publication 800-145 (Draft), 2011.
[4] T. Deshane, et al., *Quantitative comparison of Xen and KVM*. Xen Summit, Boston, MA, USA, pp. 1-2, Jun, 2008.
[5] J. Che, et al., *A synthetical performance evaluation of OpenVZ, Xen and KVM*. IEEE Services Computing Conf., China, Dec, 2010.
[6] A. Beloglazov, et al., *Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing*. Future Generation Computer Systems, Elsevier, May, 2011.
[7] A. Berl, et al., *Energy-efficient cloud computing*. The computer Journal, vol 53, pp. 1045-1051, Aug, 2009.
[8] Y. Agarwal, et al., *Somniloquy: augmenting network interfaces to reduce PC energy usage*. In Proc. of the 6th symposium on networked systems design and implementation, Apr, 2009.
[9] IBM Corp., *IBM System Virtualization*. http://publib.boulder.ibm.com-/infocenter/eserver/v1r2/topic/eicay/eicay.pdf, Dec, 2005.
[10] P. Barham, et al., *Xen and the Art of Virtualization*. In Proc. of the 19th ACM symposium on operating systems principles, Dec, 2003.
[11] A. Kivity, et al., *kvm: the Linux Virtual Machine Monitor*. In Proc. of the Linux Symposium, Ottawa, 2007.
[12] X. Fan, et al., *Power Provisioning for a Warehouse-sized Computer*. In Proc. of the ACM Int'l Symposium on Computer Architecture, 2007.
[13] L. Cherkasova et al., *Comparison of the three CPU schedulers in Xen*. Sigmetrics Performance Evaluation Review 35(2): pp. 42-51, 2007.
[14] Xen Credit-Based CPU Scheduler, http://wiki.xensource.com/xenwiki/C-reditScheduler
[15] W. Jiang, et al., *CFS Optimizations to KVM Threads on Multi-Core Environment*. 15th Int'l Conf. on Parallel and Distributed Systems, 2009.
[16] A. Menon, et al., *Diagnosing Performance Overheads in the Xen Virtual Machine Environment*. 1st ACM/USENIX Conf. on Virtual Execution Environments (VEE'05), Jun, 2005.
[17] P. Bohrer, et al., *The case for power management in web servers*. Power Aware Computing, 2002.
[18] W. Jiang, et al., *CFS Optimizations to KVM Threads on Multi-Core Environment*. 15th Int'l conf. on Parallel and Distributed Systems, 2009.
[19] V. Uhlig, et al., *Towards Scalable Multiprocessor Virtual Machines*. In Proc. of Virtual Machine Research And Technology Symposium, 2004.