On the Interplay between Network Traffic and Energy Consumption in Virtualized Environment: An Empirical Study

Chi Xu
Simon Fraser University
British Columbia, Canada
Email: chix@cs.sfu.ca

Ziyang Zhao
Simon Fraser University
British Columbia, Canada
Email: ziyangz@cs.sfu.ca

Haiyang Wang
University of Minnesota at Duluth
Minnesota, USA
Email: haiyang@d.umn.edu

Jiangchuan Liu
Simon Fraser University
British Columbia, Canada
Email: jcliu@cs.sfu.ca

***Abstract*—Networking and virtualization are two key building blocks of modern cloud computing. The energy consumption of physical machines has been carefully examined in the past research, including the impact of network traffic. When it comes to virtual machines, the interplay between energy consumption and network traffic however becomes much more complicated. The traffic are now generated by and exchanged between *virtual machines* (VMs), which could reside in different physical machines with their respective *network interface cards* (NICs), or share the same physical machine. When multiple VMs share one physical NIC, their traffic can interfere with each other, causing extra overhead. Yet the VM's allocation can be dynamic and they can even migrate across different physical machines, thereby changing the traffic pattern. These factors combined make the network traffic highly diverse and dynamic, so is the corresponding energy consumption.**

In this paper, we present an initial measurement study on the interplay between energy consumption and network traffic in representative virtualization environments. Our study reveals a series of unique energy consumption patterns of the network traffic in this context. We show that state-of-the-art virtualization designs noticeably increase the demand of CPU resources when handling networked transactions, generating excessive interrupt requests with ceaseless context switching, which in turn increases energy consumption. Even when the physical machine is in an idle state, the VM network transactions will consume non-trivial energy. The energy consumption also varies significantly with different VM allocation strategies. Our close examination pinpoints the root cause, and offers new angles to revisit the existing resource usage and energy consumption models, so as to optimize the service provisioning as well as virtual machine placement and migration.

I. Introduction

Cloud computing has recently become the dominant paradigm for providing computing infrastructure as a utility through the Internet. The rapid adoption of the cloud computing has made modern datacenters grow at a fast pace in the amount of available resources and services. Such a dramatic and continued growth has unavoidably increased the total energy used by the servers. According to the U.S. Environment Protection Agency (EPA) congressional report on datacenter infrastructure [1], in 2008 alone, data centers consumed 61 billion kilowatt-hours (kWh), representing approximately \$4.5 billion and accounting for 1.5% of the total U.S. electricity use. A recent report further indicates that such cloud-based enterprise users as Facebook [2] used 153 million kWh of power in 2012 to host their datacenters, which is almost twice what they used the year before. Energy thus becomes a critical concern in the optimization of the current cloud infrastructures and the deployment of new infrastructures in the future.

Networking and virtualization are two key building blocks of cloud computing. Virtualization divides a high-performance physical machine into multiple *virtual machines* (VMs), allowing them to share the hardware resources with isolation and protection. It makes resource pooling and server consolidation for utility computing possible in modern clouds. The energy consumption of physical machines has been carefully examined in the past research, including the impact of network traffic [1] [3]. When it comes to the cloud context with virtual machines, the interplay between energy consumption and network traffic however becomes much more complicated. The traffic are now generated by and exchanged between VMs, which could reside in different physical machines with their respective *network interface cards* (NICs), or share the same physical machine. When multiple VMs share a physical NIC, their traffic can interfere with each other, causing extra overhead. Yet the VM's allocation can be dynamic and they can even migrate across physical machines, thereby changing the traffic pattern. These factors combined make the network traffic highly diverse and dynamic, so is the corresponding energy consumption. A close examination on the network traffic and energy consumption in virtualized environments is thus of urgent need and great significance.

In this paper, we present an initial measurement study on the interplay between energy consumption and network traffic in representative virtualization environments. Our study reveals a series of unique energy consumption patterns of the network traffic in this context. We show that such state-of-the-art virtualization designs as Xen and KVM noticeably increase the demand of CPU resources when handling networked transactions. In particular, allowing more virtual machines sending and receiving network traffic generates excessive interrupt requests, leading to ceaselessly context switching, which in turn increases energy consumption. Even when a physical machine is in an idle state, its VM's network transactions will incur non-trivial energy consumption. Furthermore, under identical number of VMs and amount of traffic on one specific physical machine, the energy consumption can vary significantly with different VM allocation strategies. Our close examination pinpoints the root cause which mainly comes from the different traffic multiplexing in virtualization. Our observations offer new angles to revisit the existing resource usage and energy models, providing opportunities for optimizing the service provisioning as well as energy-aware virtual machine placement and migration in the future.

The rest of this paper is organized as follows: In Section II, we closely investigate the state-of-the-art virtualization technologies, and discuss how Xen and KVM, two typical virtualization environments, handle network traffic. Section III shows detailed measurement setup. In Section IV, we present our measurement results and also the detailed analysis. In Section V, we examine whether a bridging scheme will have influence on our observation. In Section VI, we revisit the state-of-the-art resource usage model and energy consumption model. Section VII presents the related works and Section VIII concludes this paper.

II. Overview of Virtualization

In order to provide abstraction of the physical resources, popular virtualization solutions use a *hypervisor* between different VMs and the underlying hardware to isolate performance, allocate device and ensure security between running VMs. To analyze energy consumption in a virtualized environment, we should carefully examine these widely adopted virtualization technologies. Drawn from current hypervisor-based virtualization solutions, we can present two main categories.

Paravirtualization (PVM). Paravirtualization is a widely deployed version of virtualization today. PVM does not try to appear as a real hardware to the guest kernel, but just lets the guest know that it is running in a VM. As a result, PVM does not require special hardware extension, instead, it achieves virtualization by implementing modifications on the VM's operating system (OS). Such modifications let guest kernel send privileged system calls and access hardware directly to the hypervisor, the hypervisor decides how to handle these requests. Therefore, the choice of guest OS used in PVM is restricted because the OS must allow modifications to work with the hypervisor. One of the typical example of PVM-based virtualization solutions is Xen [4].

Hardware Virtual Machine (HVM). HVM achieves virtualization through special hardware (often called virtualization extensions) that has the capability to trap privileged calls from guest systems. The hypervisor then decides what to do with the calls. HVM requires no special OS or drivers on the guest system, thus providing higher flexibility. However, when dealing with input/output (I/O), HVM can lead to higher overhead compared to PVM, since it must intercept each privileged call. KVM [5] and VMware [6] are two examples of HVM-based virtualization systems [7].

Two pioneer cloud platforms, Amazon AWS [8] and Rackspace Cloud [9], are both based on customized Xen hypervisors; another two cloud providers, Ubuntu Enterprise Cloud [10] and Eucalyptus Cloud Service are based on KVM hypervisors. Therefore, in this paper, we focus on Xen and KVM to evaluate their energy consumption on dealing with the network traffic in virtualized environments.

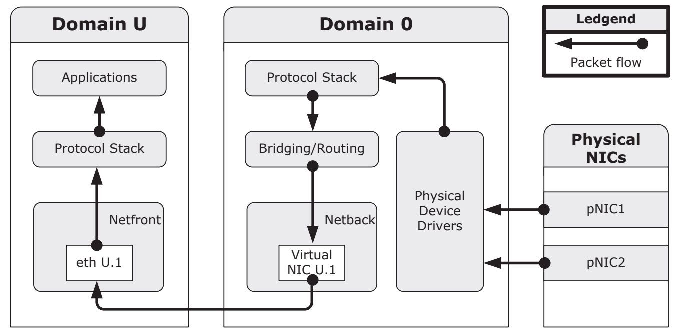


Fig. 1: Xen Network Architecture

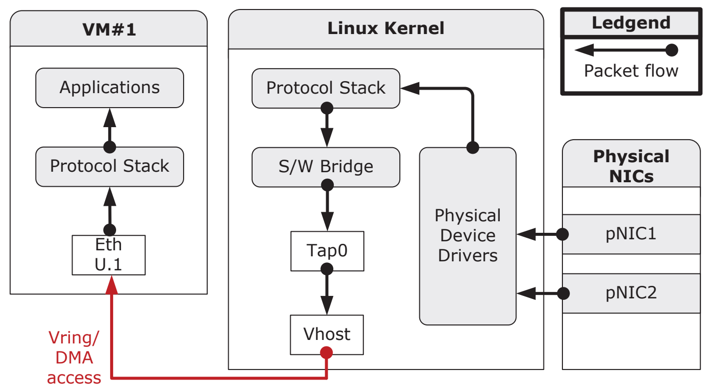


Fig. 2: KVM Network Architecture

A. Xen

A closer look into the network architecture in virtualized environment is given in Fig. 1, which lists the key steps involved in delivering packets to a Xen-based VM. In this figure, Domain-0 is the initial domain started by the Xen hypervisor up on boot. It runs the Xen management toolstack, and has special privileges. Domain-U refers to a set of unprivileged VMs. In Domain-U, the *netfront module* is designed to manage the network traffic from/to Domain-0. This module and its counterpart *netback module* (in Domain-0) are a pair of inter-linked drivers, bridging the network communications between different domains. For example, upon recipient of a packet in Xen, the physical NIC will deliver the packet to the physical device driver in Domain-0. Once the packet arrives at the bridge though the protocol stack, it will be transferred through the netback module and the netfront module. In particular, the netback module will allocate resources to process the packet and notify the the netfront module in Domain-U. Finally, the netfront module receives the packet and passes it to the guest's network layer.

B. KVM

The *vhost* architecture is a state-of-the-art architecture of KVM, as shown in Fig. 2. The vhost driver in Linux provides in-kernel virtual I/O device emulation. This architecture puts virtual I/O emulation codes into the Linux kernel, which enables the device emulation codes to directly call into kernel subsystems instead of perform system calls from the user space. Note that the vhost architecture does not emulate a complete virtual I/O PCI adapter. Instead, it restricts itself to virtual queue operations only. QEMU is used to perform virtual I/O feature negotiation and live migration, for example. This means a vhost driver is not a self-contained virtual I/O device implementation, but depends on the user space to handle the control plane while the data plane is done in-kernel. The vhost working thread waits for the virtual queue dumping and then handles buffers that have been placed in the virtual queue. The vhost architecture however is not tied to KVM in any way. It is a user space interface and has no dependency on the built-in KVM kernel module.

Although the virtualization technologies intend to tradeoff between isolation and efficiency, problems remain open to answer. There is no a perfect solution to achieve the best of the both. For example, network packets have to traverse through the hypervisor layer, and thus the extra overhead is inevitable. In our following experiments, we will evaluate such overheads in Xen and KVM.

III. Pre-measurement Setup

We have conducted measurements to examine the relationship between network traffic and power consumption, in the context of Xen and KVM. The detailed measurement setup is presented as follows.

A. Measurement Platform

Our test bed machine is a typical midrange server equipped with an Intel core i5 2400 3.09GHz quad core CPU, 8GB 1333MHz DDR3 RAM, The hard drive is a 500GB 7200RPM hard drive. Also, we have two Broadcom network interface cards attached to the PCI-E bus, each of which has a maximum throughput of 1000Mb/s. The reason why we choose Intel i5 as the CPU is that the Intel's x86 architecture is dominating the CPU market for a long period of time, and major cloud computing providers like Amazon also base their virtual machine implementation on the x86 architecture. It is noted that the core i5 CPU has low power consumption with a carefully designed power management strategy.

Since we focus on the relationship between network traffic and power consumption, we have configured a second machine to work as the other end of the network traffic. The CPU of the second machine is a 2.8GHz Intel Core2 Quad CPU. This machine is equipped with a 4GB DDR3 RAM. These two machines are directly connected by a 1000Mb/s Linksys SD2005 SOHO switch.

B. System Setup

1) Xen: We configured the Xen 4.3 Paravitualization Hypervisor on our test bed machine. We set the number of accessible virtual CPUs to be 4 and the amount of RAM to 2048MB for each virtual machine.

2) KVM: We compiled KVM version 1.2.0 from the official source repository and deployed on the test bed machine. Similarly, each virtual machine was given full access to all of the 4 processor cores as well as the 2048MB of the total memory. The disk interface was configured as a flat file on the physical host's file system. Since we hope to examine the network performance in best case, we enabled the vhost architecture to use a Linux kernel module implementations for network I/O devices.

C. Measurement Tools

It is noted that Intel introduced the Sandy Bridge line of processors, and embedded the Running Average Power Limit hardware counters (RAPL) [11] in their processor design. These accurate and versatile hardware counters allow users to extract the record of their CPU power consumption. We measured the power consumption using RAPL counters when we ran all the network-related experiments.

Since the RAPL counters only record the power consumption of CPU, we need other measurement tools to evaluate the overall power consumption of the machine. In general, there are modules other than CPU that consume considerable amount of power, namely, cooling fans, hard drivers, network interface cards. To measure the power spend on these parts, we wired a digital multi-meter (Mastech MAS-345) into the AC input power line of our machine. We obtained the data by collecting samples every second throughout our experiment from our power meter.

IV. Measurement and Analysis

In order to reveal the interplay between network traffic and power consumption, we performed detailed benchmarks on our test machine, measured the CPU power consumption by monitoring the internal RAPL counter and total power of the systems using the AC power meter. We also used the benchmark tool *Iperf* [12] to generate the network traffic.

Iperf is a widely used configurable network benchmark, which allows users to generate traffic and then gauge the performance of network flows, in terms of throughput, round trip time (RTT) and jitter. It is worth noting that the cost for Iperf to generate traffic are almost negligible when compared with the cost for VM to handle such traffic. We captured the virtual CPU utilization in Xen using *xentop*, which is a standard resource monitoring tool integrated in Xen distribution. While in KVM, we used the Linux hardware performance analysis tool *Perf* to collect system level statistics such as CPU cycles consumed and context switching information, which can help with measuring how much effort for CPU to deliver the network traffic. Both of our CPU benchmarks are set to use the lowest Linux scheduler priority by using the *NICE* command. The command ensures that the network task Iperf will be given priority to run on the virtual CPU. By doing this, we ensure that inside the VM, the CPU benchmark process does not take CPU cycles needed to process our network traffic.

To avoid the randomness in our data, we ran each experiment five times and calculated the average and their standard deviation. In our figures, they are shown in final results as error bar on the graph.

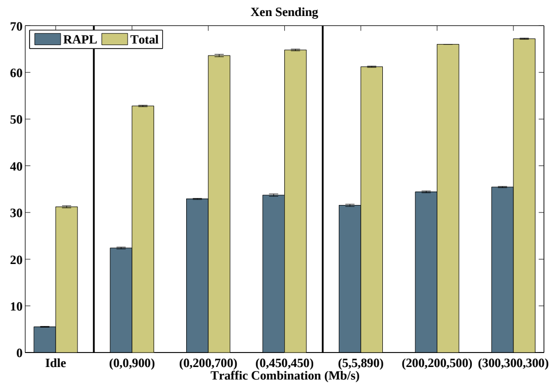


Fig. 3: Energy consumption when sending traffic (Xen)

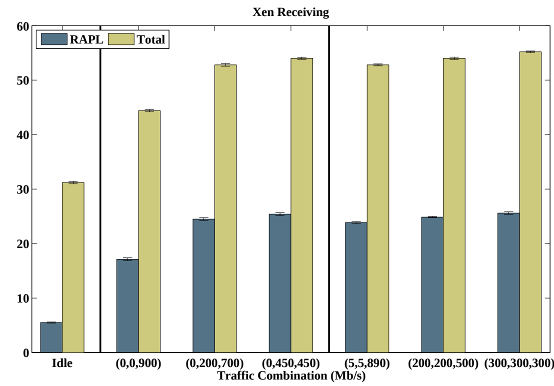


Fig. 4: Energy consumption when receiving traffic (Xen)

TABLE I: CPU Utilization of Each Domain When Sending Traffic (Xen)

	(0M,0M,900M)	(0M,200M,700M)	(0M,450M,450M)	(5M,5M,890M)	(200M,200M,500M)	(300M,300M,300M)
Sum of Domain-U Utilization (Percentage)	51	73	76.6	83	83	94
Domain-0 Utilization (Percentage)	55	81	87	62	73	95
Total CPU Utilization (Percentage)	106	154	163.6	145	156	189

TABLE II: CPU Utilization of Each Domain When Receiving Traffic (Xen)

	(0M,0M,900M)	(0M,200M,700M)	(0M,450M,450M)	(5M,5M,890M)	(200M,200M,500M)	(300M,300M,300M)
Sum of Domain-U Utilization (Percentage)	20	24	24	22	25	36
Domain-0 Utilization (Percentage)	38	45	46	39.5	45	60
Total CPU Utilization (Percentage)	58	69	70	61.5	70	96

A. Impact of Traffic Combination

Generally, one of the major principles for cloud providers is when achieving server consolidation, the traffic patterns should be diverse among different physical machines due to the divergence of each individual VM. In this section, we show that: even with identical number of VMs and amount of traffic on a physical machine, the energy consumption still could varies due to the traffic combination on each VM.

1) Measurements in Xen: In the environment, we set up three VMs on the test bed machine, all of which are sending traffic (or receiving traffic in another case); by fixing the cap of the total traffic amount, different traffic combinations are selected for our experiments. We fixed the physical machine's throughput to be 900Mb/s, and tested on different traffic combinations, namely, (0M, 0M, 900M), (0M, 200M, 700M), (0M, 450M, 450M), (5M, 5M, 890M), (200M, 200M, 500M), (300M, 300M, 300M). Note that the denotations (0M, 0M, 900M) mean that we simply let two of the virtual machines hold back on their traffic, leaving only one VM sending at 900Mb/s and fully occupying the throughput of one NIC. Our measurement results are drawn in Fig. 3. The first set of bars in Fig. 3 denote the power consumption when the machine is in an idle state. In this case, the physical machine consumes 31.2W and the CPU itself consumes 5.51W. Furthermore, we can observe in this figure that power consumption has relatively noticeable spikes between sending (0M, 0M, 900M) and sending (0M, 200M, 700M). It can be explained by that we enable another virtual machine sending network traffic and thus we have to maintain an additional set of network stack to deliver the packets; meanwhile, Network interrupts incur considerable context switching in between the two VMs. The increasing CPU utilization therefore leads to higher power consumption. This observation can be defined as a high-level traffic interference. By comparing another two cases, (0M, 450M, 450M) and (5M, 5M, 890M), we found that balancing the traffic on two VMs is not energy-efficient. Balanced traffic not only causes more overhead, but also consumes more power than bringing up a third VM sending. Our conjecture is that, balanced traffic will make each netback process compete with others, and this contention consequently leads to a worse situation. In fact, it is still curious why the power consumption still exhibits an increasing trend when comparing (5M, 5M, 890M), (200M, 200M, 500M) and (300M, 300M, 300M) in Fig. 3. As shown in [13], inside each individual VM, the network load and CPU utilization can be roughly described as a linear relationship, then CPU utilization can be directly mapped into power consumption [14]. Therefore, generating the same amount of traffic, we expect to detect identical amount of power consumption, even though we enabled multiple VM sending at the same time. In fact, this observation can also be explained by the high-level

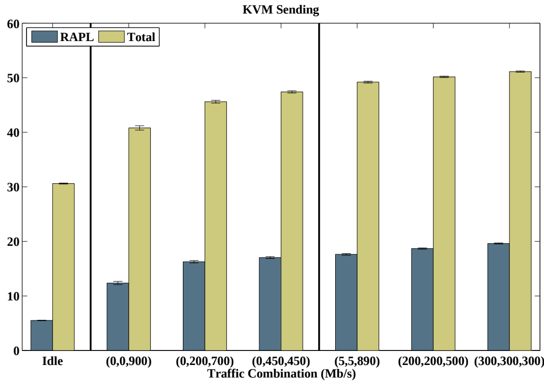


Fig. 5: Energy consumption when sending traffic (KVM)

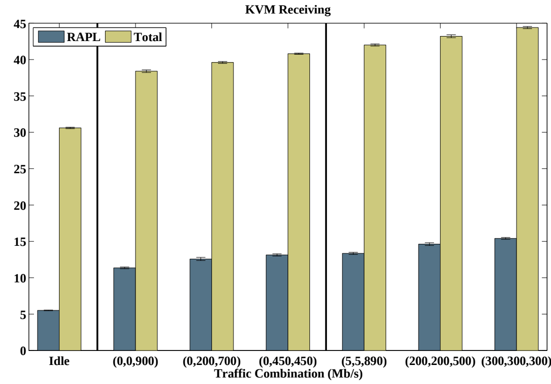


Fig. 6: Energy consumption when receiving traffic (KVM)

TABLE III: CPU Cycles And Context Switching Per Second When Sending Traffic (KVM)

	Idle	(0M,0M,900M)	(0M,100M,800M)	(0M,450M,450M)	(5M,5M,890M)	(200M,200M,500M)	(300M,300M,300M)
CPU Cycles	18.73M	1464.03M	2483.58M	2396.73M	2566.72M	3144.33M	3245.30M
Context Switching	289	57.804K	60.711K	65.444K	66.274K	101.743K	103.73K

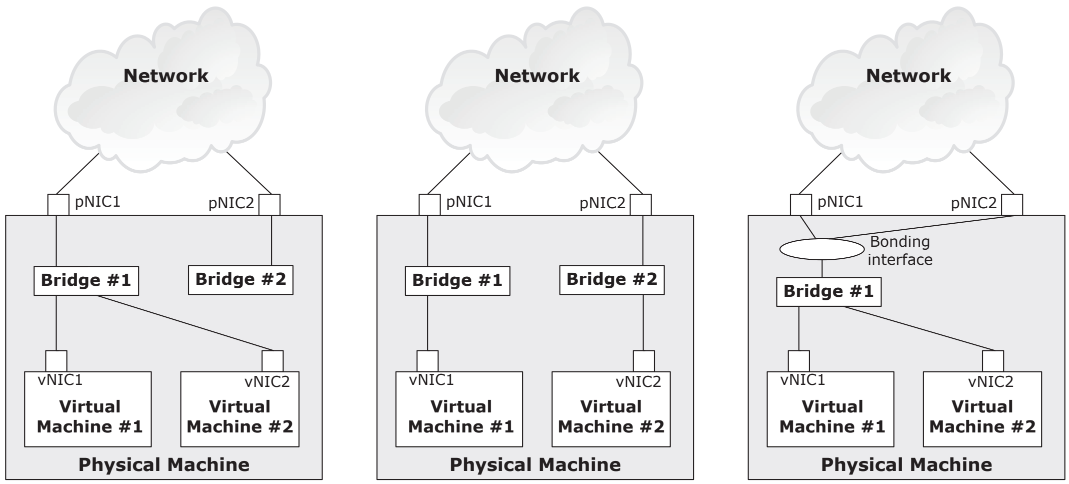
TABLE IV: CPU Cycles and Context Switching Per Second When Receiving Traffic (KVM)

	Idle	(0M,0M,900M)	(0M,100M,800M)	(0M,450M,450M)	(5M,5M,890M)	(200M,200M,500M)	(300M,300M,300M)
CPU Cycles	18.73M	934.00M	1129.50M	1149.81M	1300.00M	1315.48M	1355.77M
Context Switching	289	31.71K	33.85K	34.58K	36.42K	36.89K	38.34K

interference in managing multiple traffic. From the comparison of (5M, 5M, 890M), (200M, 200M, 500M) and (300M, 300M, 300M), we can further tell that a small traffic amount difference in between VMs leads to a higher interference. Therefore, from the angle of power consumption, it is preferred not to place those VMs with similar amount of network traffic on the same physical machine. We can infer that, in real world scenarios, the setting of more virtual machines with complex traffic combinations will lead to even more drastic difference on energy consumption. Also, we present the result of power consumption in the case of receiving networked traffic in Fig. 4. This result is also measured in Xen setting, and confirms our aforementioned observations that co-locating network-intensive VMs consumes massive power on the physical machine. Similar to the sending scenario, the gap between VMs' throughput on the contrary leads to lower power consumption. Balancing network traffic between each VM also causes more interference and consumes more power in consequence.

To further explain our observations, Table I and II measure the CPU utilization percentage on sending and receiving network traffic respectively. As can be seen, the variance of Domain-U sum is relatively small, and the major increase can be witnessed in Domain-0 in both sending and receiving cases. We have introduced that, as the representative for each Domain-U, Domain-0 is particularly in charge of handling the I/O operations. Therefore, the overhead generated in Domain-0 greatly contributes to the total CPU utilization increase and also the consumed power.

2) Measurements in KVM: In the KVM environment, similarly, we configured three VMs inside one physical machine and let them send or receive data from the remote hosts. For the sending case, the power consumption is presented in Fig. 5. The machine consumes 25.3% more power when we let each of the three VMs send at 300Mb/s, in comparison with only letting one VM send at 900Mb/s. Fig. 6 depicts the receiving scenarios. We also present the power consumption when machine is idle, the whole machine consumes 31.88W and CPU itself consumes 5.51W. If we set one VM receiving at 900Mb/s and keep the other two idle, which is then an unbalanced pattern. In this case the whole machine consumes 40W and the CPU itself consumes 11.36W. It is very similar to the Xen scenario. Obviously, the power consumption is closely related to the number of network-intensive VMs and also the variance of the throughput from all VMs. On one hand, increasing the number of VMs transferring data at the same time leads to much more power consumed by the physical machine. On the other hand, the power consumption grows as the variance of throughput decreases. When it comes to the fully balanced situation, that is, all the VMs receive at 300Mb/s, the overall machine power consumption reaches the peak of 46.25W, that is 15.63% higher than that of the unbalancing case, at the same time, power consumption of CPU itself increase to 15.41W, which contributes a large portion of the total increase.



(a) Sharing One Virtual Bridge (b) Using Two Virtual Bridges (c) Utilizing Bonding Interface

Fig. 7: Three Typical Network Settings

The details of CPU cycles and context switching numbers captured by Perf are listed in Table III and Table IV respectively. We can observe stepwised increases on both CPU cycles and context switching numbers. It further verifies our previous conclusion that the increase number of network-intensive VMs in the physical machine, together with the balanced throughput between each VMs, will consume extra power of the physical machine due to increased interference. For example, in Table IV, the unbalanced traffic pattern consumes 934M CPU cycles in 10 seconds, and the system takes 31.71K context switches during this period. In the balanced situation, the CPU cycle rises to 1355.77M and the context switch number has an increase of 6.63K.

V. Bridging Scheme

We also performed experiments under different network configurations to generalize our observations. It should be noted that a physical machine can commonly attach multiple NICs, especially when it is a commercial high-end server. Furthermore, cloud providers also enable multiple virtual machines co-locating on the same physical machine. Therefore, there will be a variety of choices on how to map the virtual NICs to physical NICs. Fig. 7 shows three typical network settings. Co-located VMs have chances to share the same virtual bridge, or they can be attached to their individual virtual bridges. Another choice is to enable bonding interfaces; that is, we can bond two (or more) physical NICs together, which look like and act as a single interface. This operation is transparent to VM users. Bonding interfaces can provide fail-over or load balancing for a cloud provider. By default, we set the bonding mode as the *balance-alb* mode. This mode achieves load balancing for IPV4 traffic, both in sending and receiving. The *balance-alb* mode does not require any special network switch support. Load balancing is achieved by ARP negotiation when receiving network traffic. The bonding driver intercepts the ARP replies sent by the local system and overwrites the source hardware address with the unique hardware address of one NIC slave of the bonding interface.

We noticed that different bridging schemes might lead to unexpected variations on our observation results. Therefore, during our experiment, we carefully examined whether different network configurations will have impact on power consumptions. Using the simplest case, Fig. 8, 9, 10 briefly present that one VM sending certain amount of traffic will cause how much extra overhead on CPU under different network settings; similar results can also be observed in receving tests, in both Xen and KVM environments. For all of the sending tests, the error bars, which represent the standard deviation between tests, are relatively small. It indicates that our measurement results are fairly stable. Further, it can be concluded that different network configurations do not cause much difference on the CPU overhead, neither on the power consumption for delivering network traffic. This straightforwardly implicated that it is beneficial to implement link aggregation, as the bonding interface allows the maximum throughput for each virtual machine to be the twice of one single NIC's throughput with no extra overhead. Fig. 8,9,10 also prove the stability of our observations, that is even with different mapping schemes, the CPU utilization still follows the identical increasing pattern.

VI. Further Discussion

We next discuss how our observations lead to potential optimization opportunities in future work.

A. Resource Usage Estimation

There have been extensive the research on estimating resource usage in virtualized environment. Typical work like [15], they use linear regression model to relate the virtualized overhead with bare-metal micro-benchmarks for one specific application on one virtual machine. Our work however points out that we should not only consider the virtual machine where the application located in, but also take account into its co-located VMs. For example, we have multiple applications running inside several virtual machines, if these

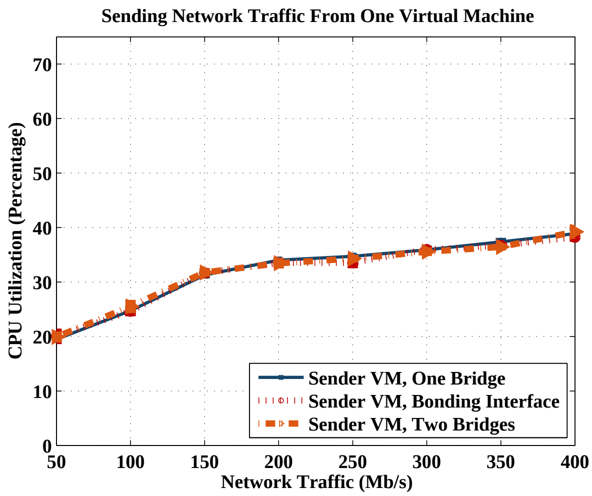


Fig. 8: Domain-U CPU Utilization

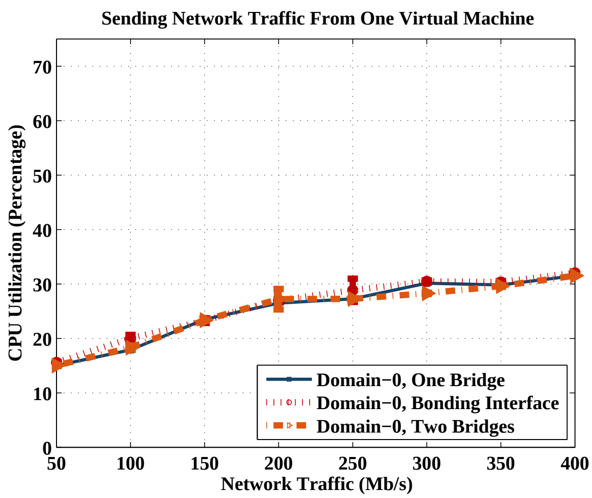


Fig. 9: Domain-0 CPU Utilization

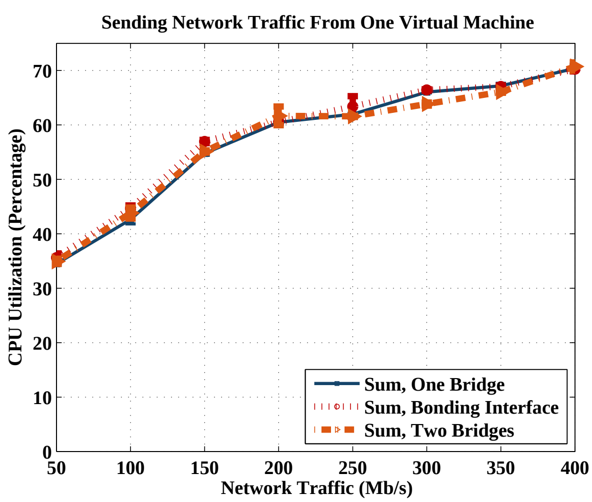


Fig. 10: Sum of CPU Utilization

virtual machines are located on the same physical machine, it can hardly estimate how much resource will Domain-0 consume. In [13], how network traffic with affinity influences on both end of the transmission is investigated. In our work, we further explain how the traffic combination on each VM will cause high-level interference.

B. *Energy Consumption Model*

Energy consumption is always a major concern for cloud providers. If they do not carefully design their cloud infrastructures, extra utility fees will be charged. Previous work like [16] and [17] have confirmed that power consumption is directly related to the resource utilization. When it refers to CPU energy consumption, there are two typical methods for the calculation: one is a linear regression between CPU utilization and consumed power. Recently, since CPU providers Intel and AMD enable advanced power-saving technology, researchers have proposed another polynomial model, as pointed out in [17]. In [16], the authors also mentioned that except for the energy consumed by the CPU, the power consumption on other parts of the system has only limited variance. Therefore, the major part for optimizing power consumption in cloud service should be focused on on CPU utilizations. In our work, we present that, CPU utilization varies due to traffic allocation on the co-located virtual machines, and this observation may lead to potential optimization opportunities. We also consider this kind of interference will have bias on the precision of previously proposed energy consumption model.

VII. Related Work

Our work mainly focuses on the power consumption caused by different combination of network traffic among VMs. Though it is the first time to pinpoint this problem, there have been a few previous works addressing similar problems, including CPU usage estimation, power consumption estimation and traffic interference.

A. *CPU utilization estimation*

Wood *et al.* [15] aim at estimating the CPU utilization by exploiting eleven different resource metrics related to CPU, network activity and disk I/O on a given hardware platform. They use regression modelling techniques and propose a linear model that maps the native system usage profile into a virtualized one. Sudevalayam *et al.* [13] use the same set of micro-benchmarks to build a linear model and capture the relationship between the resource utilization profiles and the resultant CPU utilization when a pair of VMs transfer between dispersed scenarios and co-located scenarios. Their work is based on Xen and infers the impact of inter-VM communication on CPU resource usage when the VM migration strategies are implemented. Their another work [18] extends the result to KVM. Besides, CPU usage is further examined in [19].

B. *Power consumption estimation*

Kim *et al.* [14] propose a model for estimating the energy consumption of a VM based on the in-processor events. They further propose an algorithm conforming the energy budget of each VM. Aman *et al.* [16] present a software named *Joulemeter* to measure the energy consumption of virtual machines in a consolidated server environment. They also build a power model to infer power consumption from resource usage. Chen *et al.* [20] work on profiling the energy consumption in VM based on computation-intensive, data-intensive and network-intensive tasks. The virtual machine power consumption has been estimated in [17] [21].

C. Network traffic performance

Shea *et al.* [22] reveal that degradation and variation for network traffic performance in VM are mainly caused by CPUs' scheduling policies. This work mainly focuses on the optimization for each individual VM. The research of Mei *et al.* [23] shows that, when there are multiple VMs in a single physical machine, hypervisor does not provide sufficient performance isolation. Yuan *et al.* [24] address the problem that CPU-intensive and network-intensive tasks will interfere the performance for each other and meanwhile reduce the system efficiency.

VIII. Conclusion and Future Work

In this paper, we for the first time look into the interplay between power consumption and network traffic combinations in the virtualized environment. We conducted systematic experiments to measure the CPU utilization and power consumption of one physical machine with one VM and multiple VMs, in both the representative virtualization environments Xen and KVM. The results show that such state-of the-art virtualization designs noticeably increase the demand of CPU resources when handling networked traffic, and the traffic combination on each VM plays key roles in estimating the power consumption. We analyze the root cause and further point out that existing CPU utilization models and power consumption models hardly capture such kind of interference. Our future work will focus on proposing a generalized energy model, and discussing energy-aware virtual machine placement and migration strategies based on our measurement results.

Acknowledgement

J. Lius research is supported by a Canada NSERC Discovery Grant, an NSERC Strategic Project Grant, and a China NSFC Major Program of International Cooperation Grant (61120106008). H. Wang's work is supported by a Start-up Grant from the University of Minnesota at Duluth.

References

[1] R. Brown *et al.*, "Report to congress on server and data center energy efficiency: Public law 109-431," 2008.

[2] "Facebook," http://www.facebook.com/.

[3] R. Bianchini and R. Rajamony, "Power and energy management for server systems," *Computer*, vol. 37, no. 11, pp. 68–76, 2004.

[4] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.

[5] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "kvm: the linux virtual machine monitor," in *Proceedings of the Linux Symposium*, vol. 1, 2007, pp. 225–230.

[6] B. Walters, "Vmware virtual platform," *Linux Journal*, vol. 1999, no. 63es, p. 6, 1999.

[7] J. R.Shea, "Network interface virtualization: challenges and solutions," *Network, IEEE*, vol. 26, no. 5, pp. 28–34, 2012.

[8] "Amazon web services," http://aws.amazon.com/.

[9] "Rackspace cloud," http://www.rackspace.com/.

[10] S. Wardley, E. Goyer, and N. Barcet, "Ubuntu enterprise cloud architecture," *Technical White Paper, UEC*, 2009.

[11] "Running average power limit," https://lwn.net/Articles/545745/.

[12] "Iperf," http://iperf.fr/.

[13] S. Sudevalayam and P. Kulkarni, "Affinity-aware modeling of cpu usage for provisioning virtualized applications," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*, July 2011, pp. 139–146.

[14] N. Kim, J. Cho, and E. Seo, "Energy-based accounting and scheduling of virtual machines in a cloud system," in *Green Computing and Communications (GreenCom), 2011 IEEE/ACM International Conference on*. IEEE, 2011, pp. 176–181.

[15] T. Wood, L. Cherkasova, K. Ozonat, and P. Shenoy, "Profiling and modeling resource usage of virtualized applications," in *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*. Springer-Verlag New York, Inc., 2008, pp. 366–387.

[16] A. Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, "Virtual machine power metering and provisioning," in *Proceedings of the 1st ACM symposium on Cloud computing*. ACM, 2010, pp. 39–50.

[17] I. Waßmann, D. Versick, and D. Tavangarian, "Energy consumption estimation of virtual machines," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 1151–1156.

[18] S. Sudevalayam and P. Kulkarni, "Affinity-aware modeling of cpu usage with communicating virtual machines," *Journal of Systems and Software*, vol. 86, no. 10, pp. 2627–2638, 2013.

[19] H. Mousa, K. Doshi, and E. Ould-Ahmed-Vall, "Characterizing performance in virtualized execution," in *2nd Workshop on Statistical and Machine learning approaches to ARchitectures and compilaTion (SMART)*, 2008.

[20] F. Chen, J. Grundy, Y. Yang, J.-G. Schneider, and Q. He, "Experimental analysis of task-based energy consumption in cloud computing systems," in *Proceedings of the ACM/SPEC international conference on International conference on performance engineering*. ACM, 2013, pp. 295–306.

[21] F. Chen, J. Schneider, Y. Yang, J. Grundy, and Q. He, "An energy consumption model and analysis tool for cloud computing environments," in *Green and Sustainable Software (GREENS), 2012 First International Workshop on*. IEEE, 2012, pp. 45–50.

[22] R. Shea, F. Wang, H. Wang, and J. Liu, "A deep investigation into network performance in virtual machine based cloud environments," *INFOCOM*, 2014.

[23] Y. Mei, L. Liu, X. Pu, and S. Sivathanu, "Performance measurements and analysis of network i/o applications in virtualized cloud," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. IEEE, 2010, pp. 59–66.

[24] Y. Yuan, H. Wang, D. Wang, and J. Liu, "On interference-aware provisioning for cloud-based big data processing," in *Quality of Service (IWQoS), 2013 IEEE/ACM 21st International Symposium on*. IEEE, 2013, pp. 1–6.