HEP Specific Benchmarks of Virtual Machines on multi-core CPU Architectures

M. Alef

Forschungszentrum Karlsruhe, Karlsruhe, Germany

E-mail: manfred.alef@iwr.fzk.de

I. Gable

Department of Physics and Astronomy, University of Victoria, Victoria, Canada

E-mail: igable@uvic.ca

Abstract. Virtualization technologies such as Xen can be used in order to satisfy the disparate and often incompatible system requirements of different user groups in shared-use computing facilities. This capability is particularly important for HEP applications, which often have restrictive requirements. The use of virtualization adds flexibility, however, it is essential that the virtualization technology place little overhead on the HEP application. We present an evaluation of the practicality of running HEP applications in multiple Virtual Machines (VMs) on a single multi-core Linux system. We use the benchmark suite used by the HEPiX CPU Benchmarking Working Group to give a quantitative evaluation relevant to the HEP community. Benchmarks are packaged inside VMs and then the VMs are booted onto a single multi-core system. Benchmarks are then simultaneously executed on each VM to simulate highly loaded VMs running HEP applications. These techniques are applied to a variety of multi-core CPU architectures and VM configurations.

1. Introduction

Current Virtual Machine(VM) technologies can allow virtualization of complete software stacks including the operating system. This can allow hardware to simultaneously satisfy the disparate and often incompatible system requirements of different user groups in shared-use computing facilities. This can be particularly advantageous in the case of HEP applications, which often have a large number of specific requirements and often particular Linux distribution requirements. Most HEP applications are run on large Linux cluster systems and most HEP task are, so called, embarrassingly parallel. These properties lend themselves to allowing individual HEP jobs to be wrapped in a virtual machine running on cluster worker nodes. Previous work[1] has shown that it is also possible to deploy virtual machines in computing grid environment. However, the HEP community has not traditionally been a user of virtualization technologies because of the perceived overhead associated with the extra software layer between application and hardware.

If virtualization technology can perform well enough for HEP there could be the possibility of exploiting Cloud computing in an Infrastructure-as-a-Service (IaaS) model from commercial resource providers such as Amazon and their Elastic Computing Cloud (EC2). Indeed, there has already been work done by the STAR Experiment[2] in this area[1]. In addition to commercial providers, there are a number of open source projects such as Nimbus[3], Open Nebula[4], and Eucalyptus[5] developing middleware to allow research computing sites to provide cloud infrastructures. Should research computing resource providers begin offering cloud infrastructure there may be opportunities for HEP to exploit previously unavailable resources.

In order to exploit the benefits of virtualization we must show that its performance impacts are palatable to the HEP community. We choose to evaluate two methods for employing virtualization on a HEP worker node. The first method uses one VM per CPU core. This method allows the maximum flexibility such that multiple VM types could be running on a single physical worker node. The second method uses one VM per physical box, thus reducing the complexity of VM management. We ensure that all VMs are fully CPU loaded to simulate the ideal case of a HEP worker node. In order to arrive at a quantitative assessment we employ the HEP-SPEC06 Benchmark [6] produced by the HEPiX CPU Benchmarking Group [7] to evaluate different VM configuration on several multi-core worker nodes of different CPU architectures. We do not evaluate performance outside of the CPU.

2. HEP-SPEC06

HEP-SPEC06 is a HEP specific benchmark[2] derived from SPEC CPU2006[8] from the Standard Performance Evaluation Corporation[9]. HEP-SPEC06 came about from the 2007-2008 efforts of the HEPiX CPU Benchmarking working group to identify a suitable replacement for the now retired(February 2007) SPEC int 2000 benchmark[10] that has been popular in the HEP community.

HEP-SPEC06 is run by simultaneously executing an independent benchmarking run for every core on a particular machine. Each run consists of the all_cpp named set of benchmarks from the SPEC CPU2006 benchmarks, namely the benchmarks 444.namd, 447.dealII, 450.soplex, 453.povray, 471.omnetpp, 473.astar, and 483.xalancbmk. As the name would suggest, these are all the C++ benchmarks in the SPEC CPU2006 suite. HEPiX has shown[6] that this mix of benchmarks corresponds closely with the mix of floating point and integer operations performed in HEP codes. HEP-SPEC06 has also been shown to scale linearly with actual HEP application performance. It is therefore the ideal choice for measuring the CPU performance of virtual machines for HEP purposes.

3. Benchmarking Testbed

3.1. Selecting an Operating System

Scientific Linux(SL)[11] is the most widely used Linux distribution in HEP. SL was therefore selected as the distribution of choice for making the various benchmark measurements in order to reflect as closely as possible a virtualization environment practical for widespread deployment. SL 5.X provides support for running the popular high performace Xen Virtual Machine Monitor (VMM) [12] originally developed at Cambridge University. Because of its ease of use with SL 5 we focus on measurements of the Xen VMM. The Kernel Virtual Machine(KVM)[13] is also a promising VMM technology, however there is presently no distribution support for KVM in SL5 up until at least SL 5.3.

3.2. Hardware

A broad spectrum of AMD and Intel machines was assembled from available resources at FZK Karlsruhe and the University of Victoria. The machines span CPU generations from 2003 until

[1] Monte Carlo Simulations using Cloud Computing by the Star Experiment: `http://www.isgtw.org/?pid=1001735`

[2] SPEC is a trademark of the Standard Performance Evaluation Corporation and HEP-SPEC06 is in no way endorsed by SPEC.

CPU Model	Mem (GB)	n Cores	Mainboard	Year
AMD Opteron				
246 (2.0 GHz SC)	4	2	MSI-9145	2003
270 (2.0 GHz DC)	8	4	MSI-9145	2005
2376 (2.3 GHz, QC, Shanghai)	16	8	Supermicro H8DMU+	2009
Intel Xeon				
Intel Xeon 3.00 GHz, SC (Nocona)	2	2	HP ProLiant DL360 G4, HT off	2004
5160 (3.0 GHz, DC, Woodcrest)	6	4	Intel Server Board S5000VCL	2006
E5345 (2.33 GHz, QC, Clovertown)	16	8	Supermicro CSE-812L-520CB	2007
E5405 (2.0 GHz, QC, Harpertown)	16	8	Dell PowerEdge 2950	2007
L5420 (2.5 GHz, QC, Harpertown)	16	8	Supermicro X7DCT	2008

Table 1. The benchmarking testbed is assembled from a sample of machines spanning the last 3-5 years of common CPU types for HEP worker nodes. Above 'year' refers to year of availability of that generation of CPU.

VM Type	Hypervisor Version	Kernel Version	Disk Access
i386 x86_64	Xen 3.0.3	2.6.18-92.1.13.el5xen i686 2.6.18-92.1.13.el5xen x86_64	tap:aio

Hardware Type	VM Memory Allocation
All AMD and Quad Core Intels	$n \times 1900$ MB
Intel Nocona	$n \times 870$ MB
Intel Woodcrest	$n \times 870$ MB

Table 2. VM configurations used for benchmarking. Each VM is given the memory listed above when running on a particular hardware type where n is the number of VCPUs.

2008. The full selection of hardware is listed in Table 1.

3.3. The Virtual Machines

All virtual machines benchmarked were para-virtualized domain-U Xen VMs. The benchmarked virtual machines were created in two varieties: i386 and x86_64 Xen VMs both with the 2.6.18-92.1.6.el5xen kernel. For easy portability between benchmarking physical hosts, VM were created as simple disk images stored as regular files on the domain-0 machine. The VMs accessed their disk using the blktap driver with asynchronous I/O achieved with the Xen configuration file option *'tap:aio'* [14]. For a full list of VM specifications please refer to Table 2.

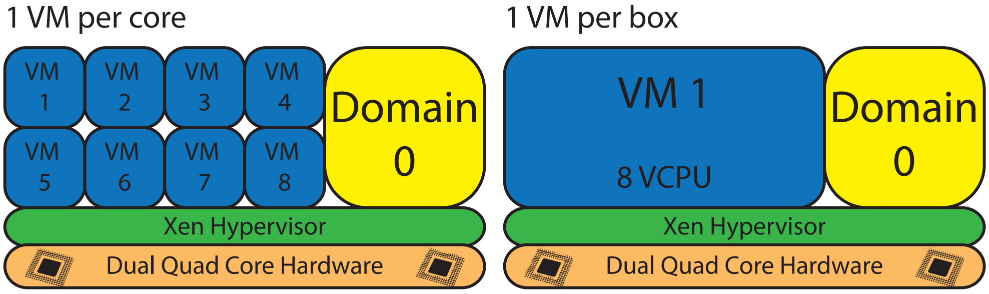


Figure 1. The three VM configurations used for benchmarking configuration used for benchmarking.

3.4. Benchmark Technique

We are most interested in investigating the practicality of running n VMs where n is the number of cores per machine. In order to evaluate this efficiency of this method we compare benchmarks in three configurations: (1) 1 vm per core, (2) 1 VM using all the physical cores (i.e 1 VM per box), (3) the physical machine. Please refer to Figure 1 for a diagram of the VM configurations.

1 VM per core - n VM are booted where n is the number of cores. Each VM is given a memory allocation as listed in Table 2. For example, an 8 core Intel Harpertown box with 16 GB of memory will have 8 VMs booted with 1900 MB of memory each, leaving 1184 MB of memory for the domain-0 machine. Each virtual disk contains the SPEC CPU 2006 code. The benchmarks are then pre-compiled with HEP-SPEC06 appropriate flags on each VM. The HEP-SPEC06 benchmark is then executed simultaneously on all 8 VMs. This causes all VMs to compete for the resources of the physical CPUs present on the box in much the same way that the individual threads of the HEP-SPEC06 benchmark compete for resources on a multi-core box.

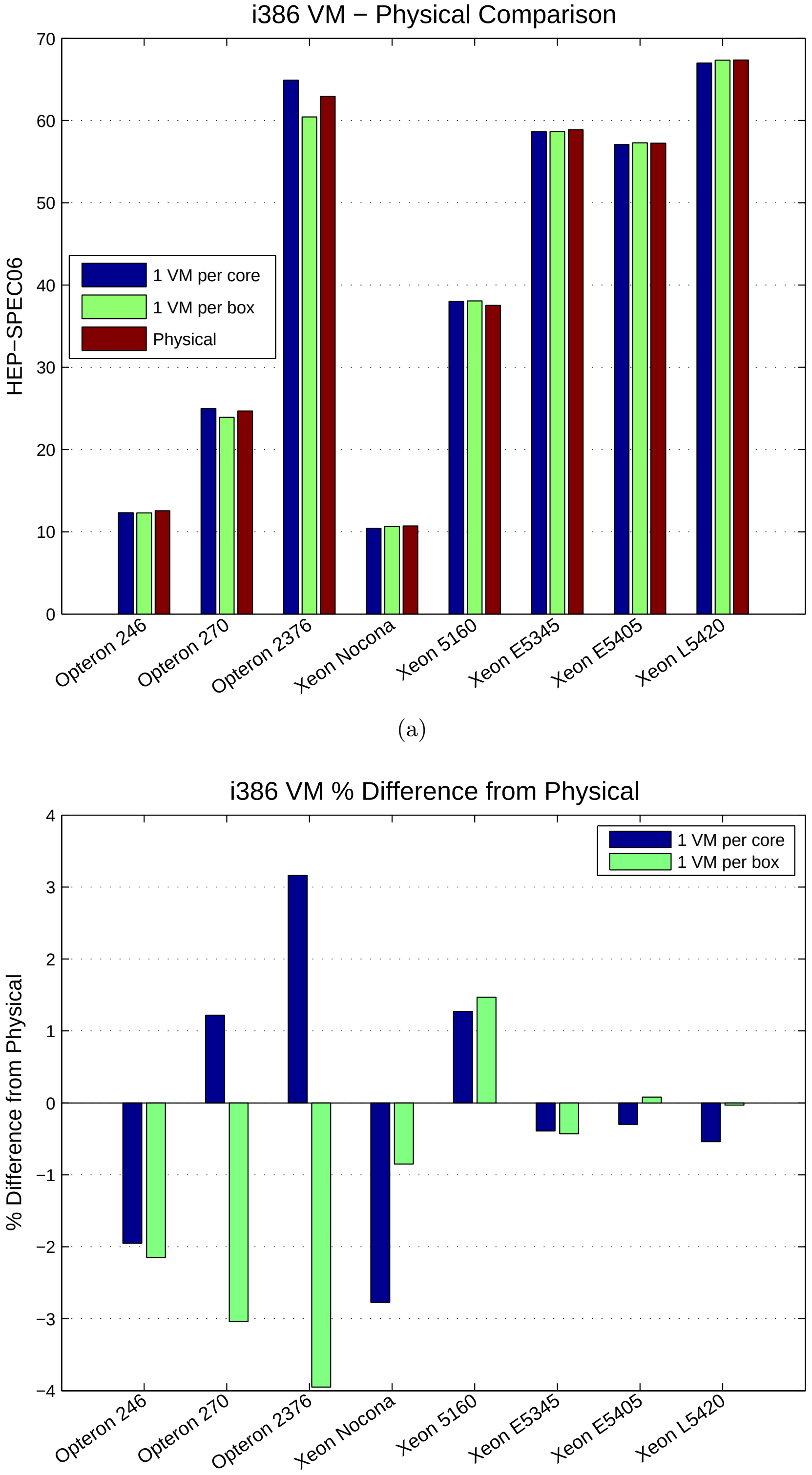
1 VM per box - In this case a virtual machine is created with VCPUs equal to the number of physical cores on the box. The benchmark is then executed in the same way as typical HEP-SPEC06 run with VCPUs standing in for physical cores.

physical machine - The machine was benchmarked in exactly the same way any HEP-SPEC06 becnhmark run would be done; one benchmark process per physical core. When comparing against i386 VMs, benchmarks are done on i386 Linux kernel. When comparing against x86_64 VMs, benchmarks are done on x86_64 Linux kernel.

4. Results

The benchmarking results for the three configurations discussed in Section 3.4 can be seen in Figure 2(a) and 2(b). The immediate and most striking observation is the very small performance degradation, and in some cases, performance boost seen for the virtual configurations. Table 3 shows the benchmark results and the relative differences between the two virtualization strategies and the physical machine.

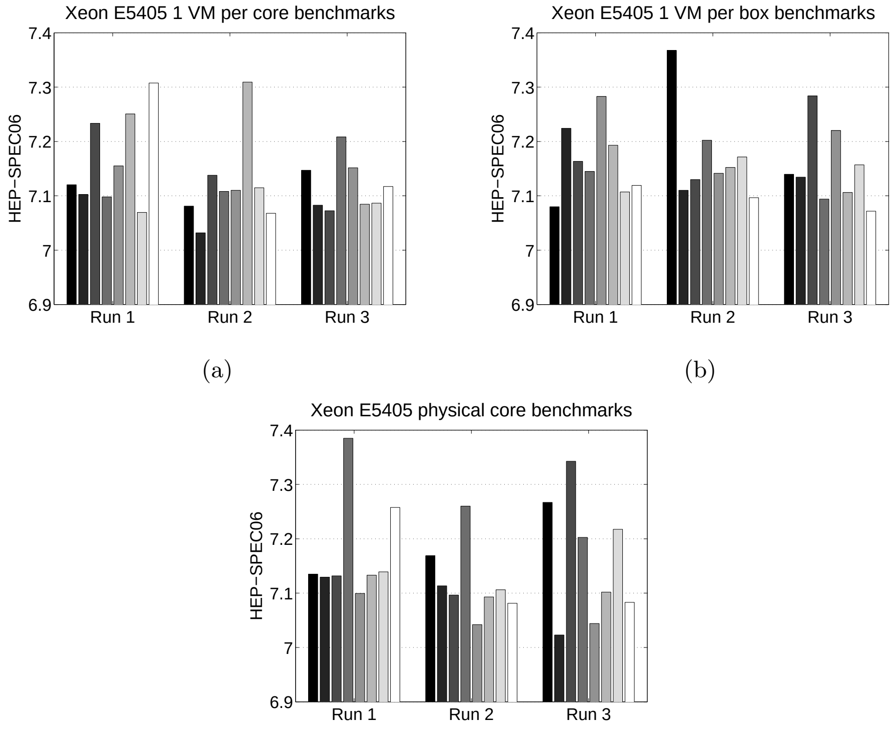
Both the Opteron 270 and the Opteron 2376 exhibit counterintuitive performance characteristics. In the 1 VM per core case the benchmark receives a 3.1% and 3.0% increase in performance, where as the 1 VM per box case receives -2.7% and -3.95% decrease in the Opteron 270 and Opteron 2376 respectively. One would expect that the extra overhead associated with running 8 VMs on the same physical hardware would result in lower performance, however the converse is true. Research at the University of California Santa Barbara (UCSB)[15] has shown similar performance improvements in single core Intel machines of up to 3.0 % with



(a)

(b)

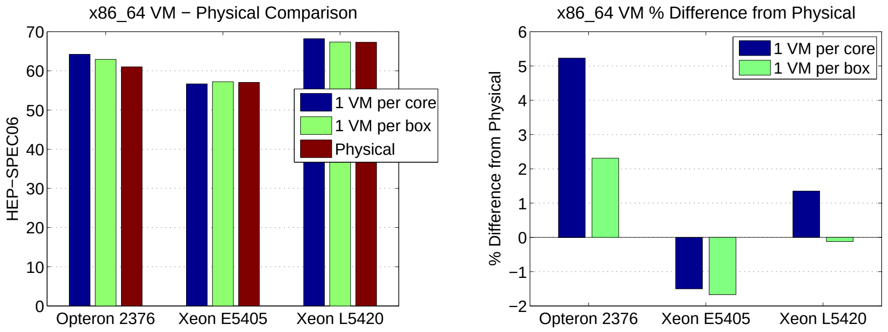
Figure 2. (a) shows the HEP-SPEC06 score of i386 VM configurations defined in Section 3.4 in addition to the score of the physical machine. (b) shows the relative difference in performance relative to the physical machine.



(a) (b)

(c)

Figure 3. Variability of individual benchmark runs on an 8 core Xeon E5405. Each bar represents the benchmark score for a particular VM. It is important to note the small scale of the y-axis here, as the scale is selected to accentuate small differences.



(a) (b)

Figure 4. (a) shows the HEP-SPEC06 score of x86_64 VM configurations defined in Section 3.4 in addition to the score of the physical machine. (b) shows the relative difference in performance relative to the physical machine.

CPU Type	n Cores	VM Type	Physical	1 VM per core	1 VM per box	% diff per core	% diff per box
Opteron 246	2	i386	12.56	12.32	12.29	-1.95	-2.15
Opteron 270	4	i386	24.68	24.98	23.93	1.22	-3.04
Opteron 2376	8	i386	62.94	64.92	60.45	3.16	-3.95
		x86_64	61.05	64.24	62.46	5.23	2.31
Xeon Nocona	2	i386	10.71	10.41	10.62	-2.77	-0.85
Xeon 5160	4	i386	37.52	38.00	38.07	1.27	1.47
Xeon E5345	8	i386	58.88	58.65	58.63	-0.39	-0.43
Xeon E5405	8	i386	57.25	57.08	57.30	-0.30	0.08
		x86_64	57.54	56.68	56.58	-1.50	-1.67
Xeon L5420	8	i386	67.36	66.99	67.34	-0.54	-0.03
		x86_64	67.31	68.22	67.23	1.35	-0.12

Table 3. The benchmark results on all CPU architectures. Note that n Cores indicates the number of cores per machine and not per CPU.

Red Hat Enterprise Linux Kernel (RHEL) from 4.X series (Kernel 2.6.9). UCSB's work points to the highly efficient Borrowed Virtual Time(BVT)[16] VM scheduler used by Xen 3.0 as being a possible source for the performance improvement over the regular Linux Symmetric Multiprocessing (SMP) SL kernels.

All the intel chips exhibit negligible performance differences between the physical box and both the VM configurations. The biggest difference is in the case of the 2003 generation of Xeon Nocona where a 2.7% percent decrease in performance is seen in the 1 VM per core case. All the 2008 and 2009 generation Intel Harpertowns show less then 0.6% difference between the virtual and physical configurations, with the edge in performance going to the 1 VM per box case by less then 1%.

Figure 3 compares the variability of individual cpu core and individual VM benchmarks on a 8 core Xeon E5405 box (hence 8 bars per run). The Xeon E5405 was selected as a typical example. Figure 3(a) shows the results of the 1 VM per core case. Figure 3(b) show 1 VM per box case where each one of the bars represents the score of an individual VCPU. Figure 3(c) shows running directly on hardware. We can see that variability of individual VM benchmarks (3(a) and 3(b)) is roughly similar to that of physical core benchmarks(3(c)).Insufficient statistics were collected to comment further on the distribution of benchmark results.

A subset of the benchmarking testbed was selected for further examination with x86_64 VMs. Results are shown in 4(b). Once again the Intel chips perform very near the physical case. The Xeon E5405 suffered performance degradation of less than 2%, while the Xeon L5420 saw a slight boost of 1.35% in the 1 VM per core case. The largest performance gain of all the bechmarks was realized with the Opteron 2376 which saw a boost of 5.23% running 1 VM per core.

In total the results show that there is very little difference in performance between the virtual machines and the physical machines. The performance changes are such that minor changes in kernel and possible small code optimizations could easily eclipse them. We believe the spectrum of CPUs surveyed to be representative of common HEP worker node types. Therefore we see no CPU performance barrier to adoption of Xen and likely other high performance VMMs.

5. Conclusion

Our results have shown that, in terms of CPU performance, it is indeed very feasible to run highly CPU loaded virtual machines on current generation multi-core CPUs. We established this using the HEP-SPEC06 benchmark which has been proven to map to real HEP application performance. No benchmark suffered more then a 5% decrease in performance. The 2008 generation Opteron 2376 (Shainghai) showed striking performance gains 3.16% and 5.23% when running i386 and x86_64 VMs, 1 per core. Recent generation quad core Intel CPUs appear mostly unaffected by Xen virtualization. Each new VMM will have to validated in a similar fashion, however it is now apparent that virtualization technology has reached a level of maturity such that it's CPU performance impact can be ignored.

6. Acknowledgements

The financial support of the National Sciences and Engineering Research Council of Canada (NSERC) and the German Federal Ministry of Education and Research (BMBF) is acknowledged.

References

[1] Agarwal A, Charbonneau A, Desmarais R, Enge R, Gable I, Grundy D, Penfold-Brown D, Seuster R, Sobie R and Vanderster D C 2008 Deploying HEP Applications Using Xen and Globus Virtual Workspaces *Proc. of Computing in High Energy and Nuclear Physics 2007 J. Phys.: Conf. Ser.* **119** 062002 doi:10.1088/1742-6596/119/6/062002

[2] STAR Collaboration, 2003 STAR detector overview, Nuclear Instruments and Methods in Physics Research Sec. A **499** pp 624-632

[3] Nimbus Project `http://workspace.globus.org/`

[4] Open Nebula Project `http://www.opennebula.org/`

[5] Eucalyptus Software `http://www.eucalyptus.com/`

[6] Michelotto M, Alef M, Bly M J, Benelli G, Brasolin F, Degaudenzu H, De Salvo A, Gable I, Hirstius A, Hristov P, Iribarren A, Meinhard H, Wegner P 2009 A comparison of HEP code with SPEC benchmark on multicore worker nodes, to appear in *Proc. of Computing in High Energy Physics 2009*

[7] HEPiX organization website: `http://www.hepix.org/`

[8] Spradling, C D, 2007 SPEC CPU2006 benchmark tools Computer Architecture News March 2007 130-4 **35** doi:10.1145/1241601.1241625

[9] Standard Performance Evaluation Corporation website `http://www.spec.org/`

[10] Henning J L 2000 SPEC CPU2000: measuring CPU performance in the New Millennium *Computer* , **33**, 7, pp 28-35

[11] The Scientific Linux Distribution: `http://www.scientificlinux.org/`

[12] Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I and Warfield A, 2003 Xen and the art of virtualization. *Proc. of the 19th ACM Symp. on Operating Systems Principles* (Bolton Landing, NY, USA) pp 164-177

[13] The Kernel Virtual Machine `http://www.linux-kvm.org/`

[14] Matthews J N 2008 *Running Xen: A Hands-On Guide to the Art of Virtualization* (Boston: Prentice Hall)

[15] Youseff L, Wolski R, Gorda B and Krintz C, 2006 Evaluating the performance impact of xen on MPI and process execution for HPC systems *Proc. of Virtualization Technology in Distributed Computing 2006* doi:10.1109/VTDC.2006.4

[16] Cherkasova L, Gupta D and Vahdat A 2007. Comparison of the three CPU schedulers in Xen. SIGMETRICS *Perform. Eval. Rev.* **35** pp 42-51 doi:10.1145/1330555.1330556