HPC Performance and Energy-Efficiency of Xen, KVM and VMware Hypervisors

Sébastien Varrette*, Mateusz Guzek†, Valentin Plugaru*, Xavier Besseron‡ and Pascal Bouvry*
*Computer Science and Communications (CSC) Research Unit
†Interdisciplinary Centre for Security Reliability and Trust
‡Research Unit in Engineering Science
6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg
Sebastien.Varrette@uni.lu, Mateusz.Guzek@uni.lu, Valentin.Plugaru@gmail.com,
Xavier.Besseron@uni.lu, Pascal.Bouvry@uni.lu

***Abstract*—With a growing concern on the considerable energy consumed by HPC platforms and data centers, research efforts are targeting green approaches with higher energy efficiency. In particular, virtualization is emerging as the prominent approach to mutualize the energy consumed by a single server running multiple VMs instances. Even today, it remains unclear whether the overhead induced by virtualization and the corresponding hypervisor middleware suits an environment as high-demanding as an HPC platform. In this paper, we analyze from an HPC perspective the three most widespread virtualization frameworks, namely Xen, KVM, and VMware ESXi and compare them with a baseline environment running in native mode. We performed our experiments on the Grid'5000 platform by measuring the results of the reference HPL benchmark. Power measures were also performed in parallel to quantify the potential energy efficiency of the virtualized environments. In general, our study offers novel incentives toward in-house HPC platforms running without any virtualized frameworks.**

I. Introduction

Many organizations have departments and workgroups that benefit (or could benefit) from High Performance Computing (HPC) resources to analyze, model, and visualize the growing volumes of data they need to conduct business. Actually, HPC remains at the heart of our daily life in widespread domains as diverse as molecular dynamics, structural mechanics, computational biology, weather prediction or "simply" data analytics. Also, domains such as applied research, digital health or nano- and bio- technology will not be able to evolve tomorrow without the help of HPC. In this context, and despite the economical crisis, massive investments (1 billion dollars or more) have been voted last year (in 2012) by the main leading countries or federations (US, Russia, China, India or the European Union) for programs to build an Exascale platform by 2020.

This ambitious goal comes with a growing concern for the considerable energy consumed by HPC platforms and data centers, leading to research efforts toward green approaches with higher energy efficiency. At the hardware level, novel solutions or architectures are currently under investigation, typically in the direction of accelerators (Tesla K20, Intel Phi) or low-power processors (ARM) coming from the mobile or embedded device market. At an intermediate level (between software and hardware), virtualization is emerging as the prominent approach to mutualize the energy consumed by a single server running multiple Virtual Machines (VMs) instances. However, little understanding has been obtained about the potential overhead in energy consumption and the throughput reduction for virtualized servers and/or computing resources, nor if it simply suits an environment as high-demanding as a High Performance Computing (HPC) platform.

In parallel, this question is connected with the rise of Cloud Computing (CC), increasingly advertised as THE solution to most IT problems. Several voices (most probably commercial ones) emit the wish that CC platforms could also serve HPC needs and eventually replace in-house HPC platforms. In the secret hope to discredit this last idea with concrete and measurable arguments, we initiate a general study on Cloud systems featuring HPC workloads.

In this paper, we evaluate and model the overhead induced by several virtualization environments (often called *hypervisors*) which are at the heart of most if not all CC middlewares. In particular, we analyze the High Performance Linpack (HPL) benchmark performance and the energy profile of three widespread virtualization frameworks, namely Xen, KVM, and VMware ESXi, running multiple VM instances and compare them with a *baseline* environment running in native mode. Actually, this study extends our previous work in the domain proposed in [1]. This time, we are focusing on larger experiments (closer to an HPC environment) while our initial article used to model a single VM instance. As for our seminal paper, it is worth mentioning the difficulty to find in the literature fair comparisons of *all* these hypervisors. For instance, in the few cases where the VMWare suite is involved, the study is generally carried on by the company itself.

The experiments performed in this paper were conducted on the Grid'5000 platform [2], which offers a flexible and easily monitorable environment which helped to refine the holistic model for the power consumption of HPC components which was proposed in [1]. Grid'5000 also features an unique environment as close as possible to a real HPC system, even if we were limited in the number of resources we managed to deploy for this study. Thus, while the context and the results presented in this article do not reflect a true large scale environment (we never exceed 96 nodes whether virtual or physical in the presented experiments), we still think that the outcomes generated by this study are of benefit for the HPC community.

The contributions of this work cover three aspects. First, our original methodology allows to perform a reproducible and fair comparison of hypervisors thanks to our benchmarking framework that provides automated installation of hypervisors and automated test application executions. Then, we provide performance number for the HPL benchmark running on three major hypervisors (VMWare ESXi, Xen and KVM) and extend previous works to multiple node executions. Finally, the power consumption is measured and allow the comparison of the hypervisors and native platform according to the "Performance per Watt" metric.

This article is organized as follows. Section II presents the background of this study. The experimental methodology is described in Section III, in particular the benchmark workflow applied to operate the fair comparison of hypervisors. Then Section IV details and discusses the experimental results obtained with the HPL benchmark on the selected environments. Finally, Section V reviews the related works and Section VI concludes the paper and provides some future directions and perspectives opened by this study.

II. Background

With the advent of the Cloud Computing (CC) paradigm, more and more workloads are being moved to virtual environments. Indeed, virtualization is emerging as the prominent approach toward higher energy efficiency by mutualizing the energy consumed by a single server running multiple Virtual Machines (VMs) instances. Yet the issue of whether CC is suitable for High Performance Computing (HPC) workload remains unclear.
In this context, this work focuses on the underlying hypervisor or Virtual Machine Manager. Subsequently, a VM running under a given hypervisor will be called a guest machine. There exist two types of hypervisors (either *native* or *hosted*) yet only the first class (also named bare-metal) presents an interest for the HPC context. This category of hypervisor runs directly on the host's hardware to control the hardware and to manage guest operating systems. A guest operating system thus runs on another level above the hypervisor.

Among the many potential approaches of this type available today, the virtualization technology of choice for most open platforms over the past 7 years has been the Xen hypervisor [3]. More recently, the Kernel-based Virtual Machine (KVM) [4] and VMWare ESXi [5] have also known a widespread deployment within the HPC community such that we limited our study to those three competitors and decided to place the other frameworks available (such as Microsoft's Hyper-V or OpenVZ) out of the scope of this paper. Table I provides a short comparison chart between Xen, KVM and VMWare ESXi.

Hypervisor:	**Xen 4.0**	**KVM 0.12**	**ESXi 5.1**
Host architecture	x86, x86-64, ARM	x86, x86-64	x86-64
VT-x/AMD-v	Yes	Yes	Yes
Max Guest CPU	128	64	32
Max. Host memory	1TB	-	2TB
Max. Guest memory	1TB	-	1TB
3D-acceleration	Yes (HVM Guests)	No	Yes
License	GPL	GPL/LGPL	Proprietary

TABLE I. Overview of the considered hypervisors characteristics.

A. The Grid'5000 Testbed

To reflect a traditional HPC environment, yet with a high degree of flexibility as regards the deployment process and the fair access to heterogeneous resources, the experiments presented in this paper were carried on the Grid'5000 platform [2].

Grid'5000 is a scientific instrument for the study of large scale parallel and distributed systems. It aims at providing a highly reconfigurable, controllable and monitorable experimental platform to support experiment-driven research in all areas of computer science related to parallel, large-scale or distributed computing and networking [6]. The infrastructure of Grid'5000 is geographically distributed on different sites hosting the instrument, initially 9 sites in France (10 since 2011), but also abroad. In total, Grid'5000 features 7896 computing cores over 26 clusters. The infrastructure offers both Myrinet and Infiniband networks, as well as Gigabit Ethernet. The sites are interconnected through a dedicated 10 Gb/s wide area network operated by Renater in France and Restena in Luxembourg.

One of the unique features offered by this infrastructure compared to a production cluster is the possibility to provision on demand the Operating System (OS) running on the computing nodes. Designed for scalability and a fast deployment, the underlying software, named Kadeploy [7], supports a broad range of systems (Linux, Xen, *BSD, etc.) and manages a large catalog of images, most of them user-defined, that can be deployed on any of the reserved nodes of the platform. As we will detail in Section III, we have defined a set of common images and environments to be deployed to perform (and eventually reproduce) our experiment. As this study also focuses on the energy consumption, power measures were required such that we had to select a site where Power distribution units (PDUs) measurements were available. For this purpose, the site of Lyon was chosen.

B. The HPL benchmark and the Green500 Challenge

To evaluate the performance of the deployed environments in a way that reflects a typical HPC workload, the High Performance Linpack (HPL) reference benchmark [8] has been chosen. HPL is a software package that solves a (random) dense linear system in double precision (64 bits) arithmetic on distributed-memory computers. It can thus be regarded as a portable as well as freely available implementation of the High Performance Computing Linpack Benchmark. Despite the development of alternative packages [9], the Top500 project [10], which ranks and details the 500 most powerful publicly-known computer systems in the world, continues to rely on HPL for its sorting.

In parallel and for decades, the notion of HPC performance has been synonymous with speed (as measured in Flops – floating-point operations per second). In order to raise awareness of other performance metrics of interest (e.g. performance per watt and energy efficiency for improved reliability), the Green500 project [11] was launched in 2005. Derived from the results of the Top500 – and thus on HPL measures, this list encourage supercomputing stakeholders to produce more energy efficient machines. We will show in Section IV an energy-efficiency analysis based on the very same metric used in the Green500 project.

III. Experimental Methodology

In order to perform experimental tests on the hypervisors detailed in Section I, a novel methodology was devised that allows for the scalable deployment and benchmarking of hypervisors, based on the capabilities of the Grid'5000 platform. The objective of the methodology is to enable the automated creation of VMs in custom CPU/memory/disk configurations and the execution of CPU/memory/IO intensive applications that will stress the virtualization frameworks thus showing the added overhead of such solutions when used for HPC-style workloads. This benchmarking workflow is detailed in Figure 1 and shows the steps involved in the provisioning of VMs and benchmark execution.

The experiments were done at the Lyon site of the Grid'5000 platform, where we selected one of the most modern HPC clusters available *i.e.* the `Taurus` cluster. An overview of Taurus is provided in Table II. Each node features an Intel processor with a Sandy Bridge micro-architecture (thus each core performing theoretically a maximum of 8 double-precision floating point operations per cycle).

Site	Lyon
Cluster	`taurus`
Max #nodes	14
Processor	Intel Xeon E5-2630@2.3GHz
#cpus per node	2
#core per node	12
#RAM per node	32 GB
R_{peak} per node	220.8 GFlops
Operating System	Debian Squeeze
Linux Kernel	2.6.32 (baseline, XEN) 3.2 backported (otherwise)
HPL	2.1
OpenMPI	1.6.4
Intel Cluster Suite	2013.2.146
Intel MKL	11.0.2.146

TABLE II. Overview of computing nodes and the operating environment used in this study.

A. Benchmarking workflow

The baseline benchmark uses a customized version of the Grid'5000 Debian `squeeze-x64-base` image, containing HPL 2.1 and OpenMPI 1.6.4 which have been compiled with the Intel Cluster Toolkit 2013.2.146 and Intel Math Kernel Library (MKL) 11.0.2.146. By using the Intel compiler and the MKL BLAS, we generate an optimized HPL that should achieve maximum performance on the target hardware.

A launcher script `configure-baseline` is used on the site's frontend to run Kadeploy3 [7] that deploys this image on a set of target nodes. The script is ran under a scheduler reservation and can be used to specify a delimited subset of the total nodes requested in the reservation. When run, it creates a hostfile which will be used by HPL's MPI processes, stores it along with a benchmark script on the head node, calculates appropriate values for the HPL run then launches the benchmark. The `benchmark` script runs HPL with the configuration-specific values, logging its progress and archiving the result file at the end. The archive is stored on the head node and is then retrieved by the launcher script which places it in user's home on the site frontend, in a directory structure which details the number of physical hosts used and date of the benchmark.

The benchmarking workflow for KVM and XEN is identical, although it is based on different scripts customized to work with these hypervisors. The KVM deployment image has been created from the baseline image, while the XEN image is based on `squeeze-x64-xen`. Both host images contain VM guest image files which incorporate the same benchmark suite as the baseline image.

The `configure-{kvm,xen}` launchers can be invoked with a set of parameters that specify, as in the baseline case, how many physical nodes are targeted and additionally how many guest VMs will be created on each. The launchers start the deployment of the appropriate host image on the set of selected nodes, create a hostfile for the VMs that will be running in a virtual subnet, then start specific guest preparation scripts for each physical host in parallel.

The `prepare-{kvm,xen}` scripts connect to a host node, copy and resize the virtual image, then start the VM also pinning the virtual cores to host cores one-to-one. This process is repeated until the required number of VMs are launched on the host. In the next step, the `configure-{kvm,xen}` script waits for the VMs to become available on the network, stores the hostfile and benchmark script on the head VM, calculates the HPL values for the selected configuration then launches the benchmark. When the `benchmark` script has finished, the results archive is retrieved from the head VM, and is placed on the site frontend, in the user's home directory in a directory structure which reflects the number of physical hosts and guests used for the test run and the date.

The workflow for the ESXi benchmark requires that the target host be booted (through the Grid'5000 kareboot application) with a specific PXE profile and configuration files so that the ESXi installer boots and configures the hosts according to a cluster-specific kickstart automated installation script. After the installation is done, the host automatically reboots and manual user intervention is required in order to ensure that the host will boot from the local drive by having an `ESXi-install` script reboot the host with another PXE profile that chainloads the newly installed MBR. The ESXi installer has been forced to use a MBR type partitioning scheme, as opposed to its default GPT in order to not interfere with the operation of the Grid'5000 platform. When the ESXi hypervisor has booted, the `configure-esxi` launcher is used to start the `prepare-esxi` script in parallel that will configure the user-specified number of guests on each physical host. This script creates the VM configuration files on the frontend, transfers them and the guest images to the host, then performs the VM initialization on the host node. After all the preparation scripts have finished, the launcher ensures that the VMs have started before copying the hostfile and benchmark script to the head VM, calculates appropriate HPL values for the selected configuration and starts the benchmark script. When the HPL run has completed, the result is collected and stored in the same manner as for the KVM and ESXi benchmark.

B. Power measurements

In parallel with the benchmarking process, power measurements are taken in order to quantify the energy efficiency of

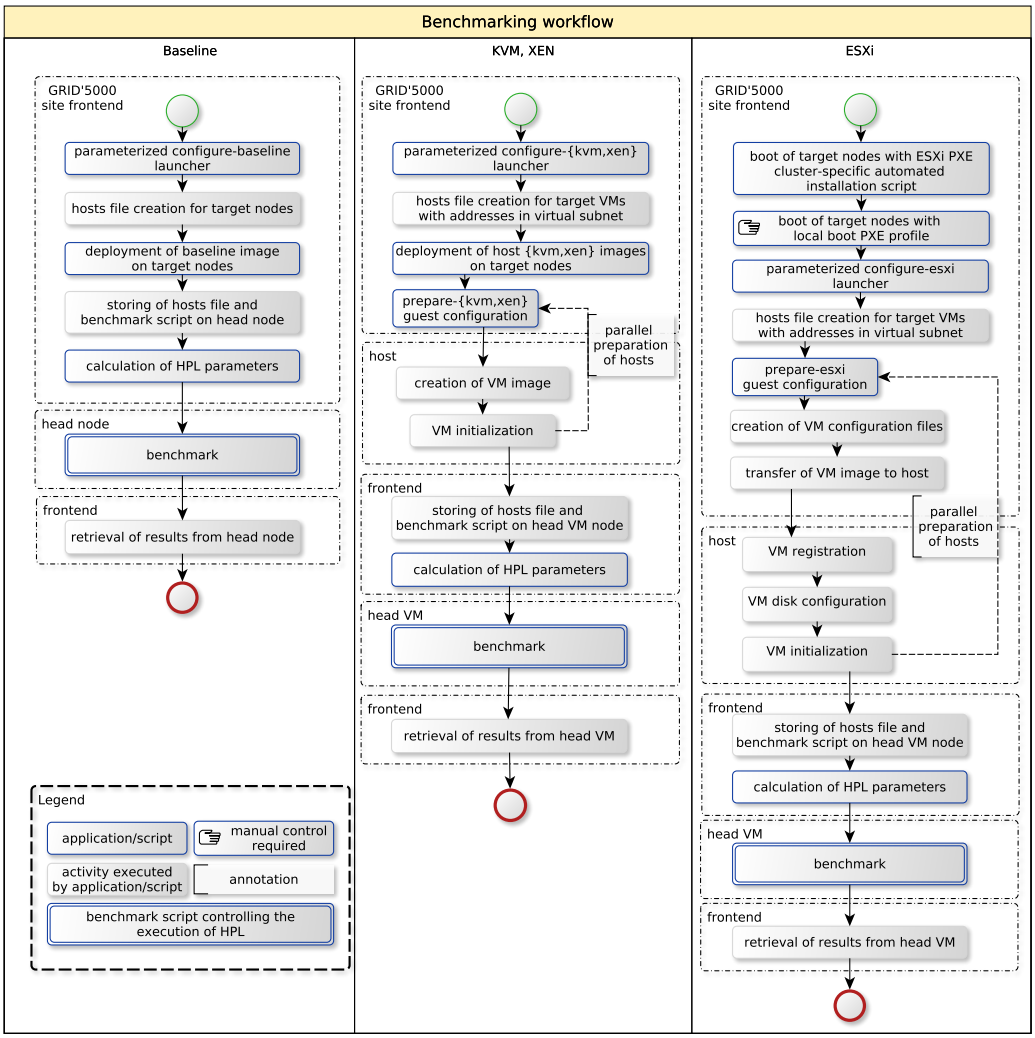

Fig. 1. Benchmarking workflow.

the tested hypervisors. The energy consumption data used in our analysis is collected from the Lyon site of the Grid'5000 platform where power consumption is measured every second using wattmeters manufactured by the OMEGAWATT company, having a precision of 0.125W. Power readings are gathered through the Grid'5000 Metrology API and continuously stored in a SQL database. The correlation of the benchmark execution with the compute node power consumption, post-processing and statistical analysis is done using the *R* statistical software.

IV. Experimental Results

This section presents the results obtained, either at the level of pure computing performance (as measured by HPL *i.e.* as it is done in the TOP500 project) or at the level of the Green efficiency. In an attempt to improve the readability of the article, we deliberately limit the number of displayed test results to the most significant ones.

A. HPL Performance Results Regarding Physical Resources

We have first compared the theoretical peak performance R_{peak} to the performance of the baseline environment for an increasing number of nodes. The results are displayed in Figure 2. As we can see, due to compiling with the Intel suite, we obtain a rather good efficiency (around 77%).

We have then compared the computing performance of the different hypervisors to the `baseline` environment (corresponding again to classical HPC computing nodes). In these first comparisons, we use the number of physical nodes as a basis to evaluate the performance reachable by the hypervisors and the baseline, i.e. hypervisor executions on N nodes with V VMs per nodes are compared to baseline executions on N physical nodes. This allows to clearly identify the overhead induced by the usage of the virtualization platforms on hardware offering the same computation capabilities. In this context, as explained in Section III, we have evaluated the scalability of each virtualization middleware under two perspectives: (1) for a fixed number of physical hosts that run an increasing number of VMs (from 1 to 12); (2) for a fixed number of VM (between 1 and 12), increasing the number of physical hosts (between 1 and 8). Both cases permit to artificially increase the total number of computing nodes.

The results of the first scalability checks are proposed in Figure 3 where we increase the number of VMs for a

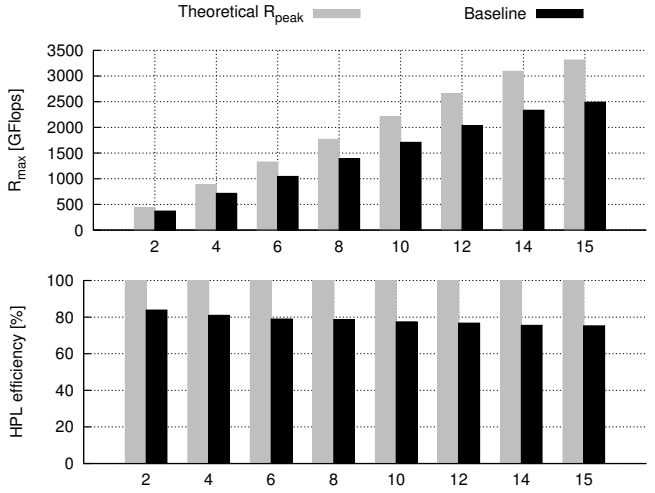

Fig. 2. HPL Efficiency of the baseline environment.

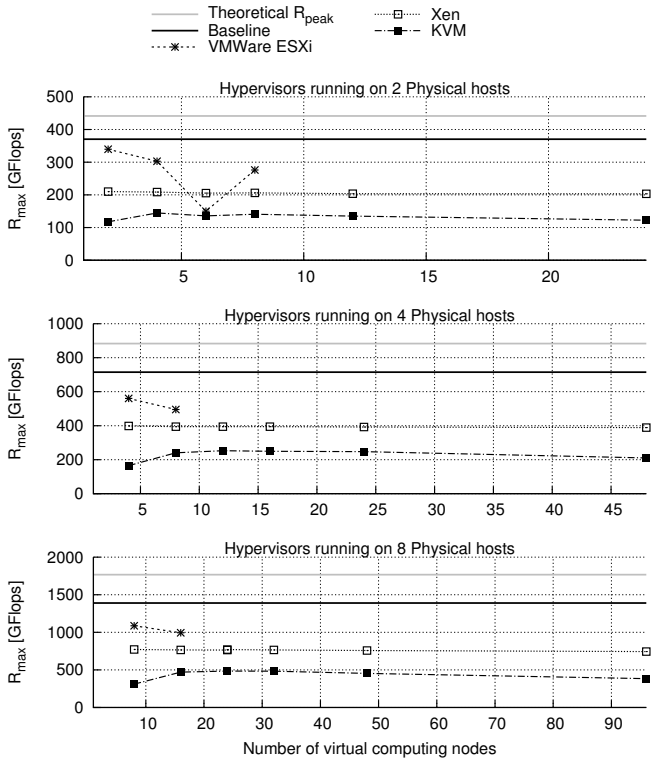

Fig. 3. HPL Performances for fixed numbers of physical nodes with increasing number of VMs per physical host. Baseline execution uses the number of actual physical nodes.

given number of physical hosts. It perfectly illustrates the obvious limitation raised by a multi-VM environment as the performance is bounded by the maximal capacity of the physical host running the hypervisors. Also, we can see that for a computing application as demanding as HPL, the VMWare ESXi hypervisor performs generally better than Xen and KVM. Yet this statement is balanced by the fact that the VMWare environment appeared particularly unstable, such that it was impossible to complete successfully runs for more than 4 VMs. On the contrary, Xen and KVM frameworks both offer

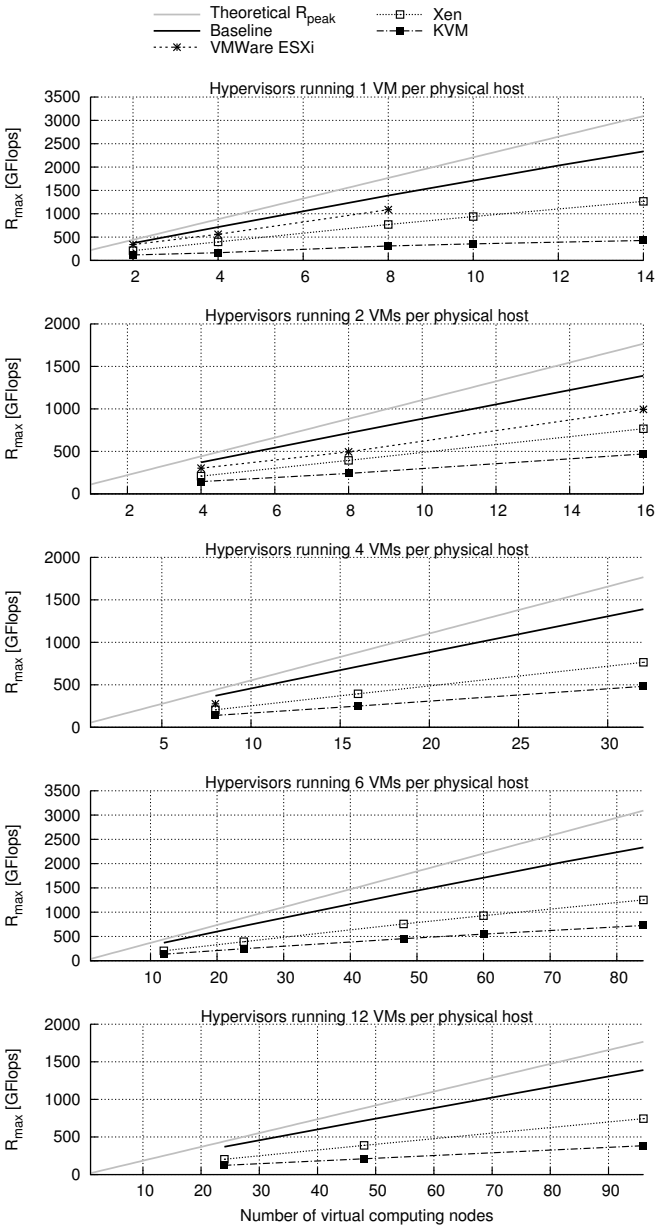

Fig. 4. HPL Performances for fixed numbers of VMs with increasing number of physical nodes. Baseline execution uses the number of actual physical nodes.

unmatched scalability, including the ability to run concurrently on each physical host 12 VMs each one executing HPL. Additionally, it clearly shows that having more than 2 VMs significantly degrades the overall performance for VMWare ESXi hypervisor. The same phenomenon occurs for the two other hypervisors Xen and KVM beyond 4 VMs per physical hosts.

Figure 4 illustrates the results of the second scalability tests where we increase the number of physical hosts for a fixed number of VMs per nodes. Here we can notice a rather good scalability of the hypervisors when physical nodes are added. The differences in the slope in due the hypervisor overhead illustrated in the Figure 3.

B. HPL Performance Results Regarding Virtual Resources

We now perform a new analysis using the virtual resources as a basis for the comparisons. It means hypervisor executions on N nodes with V VMs per nodes are compared to baseline executions on $N \times V$ physical nodes. As this approach might appear unfair as the hardware capabilities are not the same, this illustrates the point-of-view of the user that may not know the underlying hardware his application is running on in a virtualized environment.

Similarly to what has been presented strictly for the baseline environment, the HPL efficiency of the runs involving hypervisors is displayed in Figure 5. In this context and for each considered hypervisor, we have selected for a given number of computing nodes the combination (number of hosts / number of VMs) that resulted in the best performance. For instance, the experiments performed on Xen that feature 8 computing nodes give the following results:

#hosts	#VMs	GFlops
2	4	2.061e+02
4	2	3.947e+02
8	1	7.730e+02

Consequently, we selected in this case for the Figure 5 the value 773 GFlops. Note that we had to extrapolate the baseline results involving more than 14 nodes due to the limited number of resources within the `taurus` cluster.

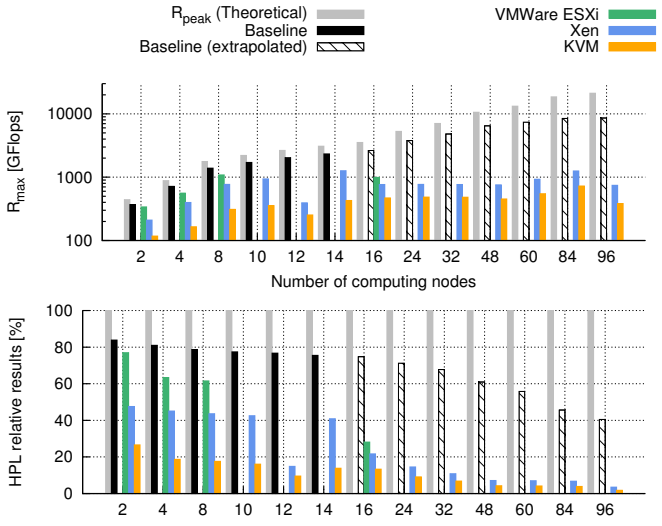

Fig. 5. HPL Efficiency of the selected hypervisors compared to the baseline environment.

From a general perspective, Figure 5 demonstrates the fast degradation in the computing efficiency when the number for computing nodes is artificially increased through virtualization.

Finally, we wanted to highlight the relative performance of each computing node. To achieve this goal, we define the *iso-efficiency* $ISO_{\text{effic.}}(n)$ metric for a given number of computing nodes n. This measure is based on the following definitions:

- $\text{Perf}_{\text{base}}(n)$: HPL performance of the baseline environment involving n computing nodes;
- $\text{Perf}_{\text{base}}(1)$: Normalized performance of a bare-metal single node. For this study, as we only started our measures with two hosts, we approximate this value by $\text{Perf}_{\text{base}}(1) = \frac{\text{Perf}_{\text{base}}(2)}{2}$
- $\text{Perf}_{\text{hyp}}(n)$: Maximal HPL Performance of the virtualized environment based on the hypervisor `hyp` that feature a total of n computing nodes.

Then for a given hypervisor `hyp`:

$$ISO^{\text{hyp}}_{\text{effic.}}(n) = \frac{\text{Perf}_{\text{hyp}}(n)}{n \times \text{Perf}_{\text{base}}(1)}$$

Our definition should not be confused with the classical iso-efficiency metric used in parallel programs. Our objective here is simply to normalize the hypervisor performance with regards to the best performance that can be obtained on a baseline environment *i.e.* $\text{Perf}_{\text{base}}(1)$. Figure 6 expounds the evolution of $ISO^{\text{hyp}}_{\text{effic.}}(n)$ with n. Again, this measure confirms that HPC workloads do not suit virtualized environments from a pure computing capacity point of view. The virtualized environment shows more available processors to the application. However, this computing resources have reduced performance compared to actual physical processors because they are shared for different VMs. This perfectly highlighted by the HPL benchmark whose performance are mainly bounded by the performance of the processors.

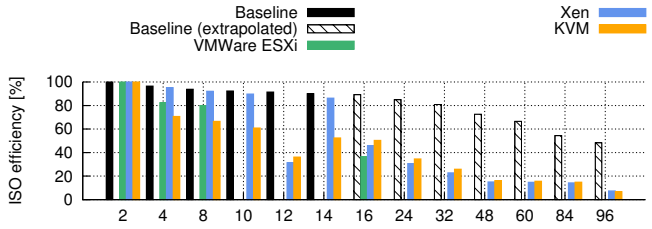

Fig. 6. Iso-efficiency evaluation for an increasing number of computing nodes.

C. HPL Energy-Efficiency Results

If the evaluation presented in the precedent section confirms what other studies suggested in the past, *i.e.* that the overhead induced by virtualized environments do not suit HPC workloads, we wanted to complete our analysis and eventually tamper this conclusion by performing a deep analysis of the power consumption. For instance, Figure 7 illustrates the total power profile of a run involving each considered environment in a large scale execution. As a consequence, the peaks in power draw are amplified proportionally to the number of nodes, which can lead to stressing the data center infrastructure. An uneven load in a data center may lead to an increase of the required cooling, which can result in increased operational costs or an even higher risk of cooling system failure. A remedy for such behavior could be the placement of VMs that perform unrelated computations on the same physical host, as their power draws are not synchronized.

The last metric we were interested in was the one used in the Green500 project [11]. More precisely, the Green500 List

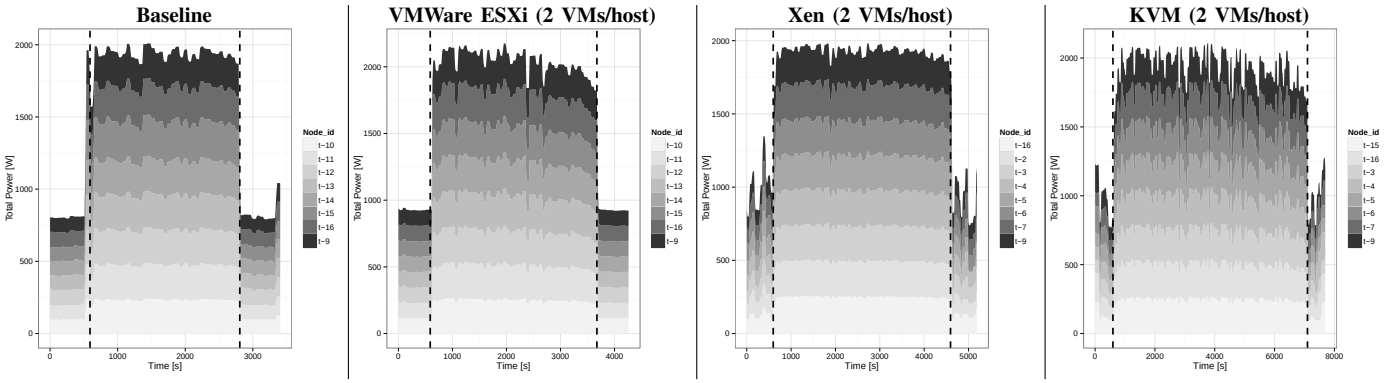

Fig. 7. Stacked traces of the power draw of hosts for selected runs with 8 physical hosts.

uses "Performance per Watt" (PpW) as its metric to rank the energy efficiency of supercomputers, defined as:

$$\text{PpW} = \frac{R_{\max} \quad (\text{in } \mathbf{MFlops})}{\text{Power}(R_{\max}) \quad (\text{in } \mathbf{W})}$$

This metric is particularly interesting because it is somehow independent of the actual number of physical nodes. Indeed, as shown in Figures 3 and 4, the performance is proportional to the number of physical nodes, and the same applies for the power consumption. Thus, this metric highlights the energy consumption overhead due the distributed execution (on virtual hosts or not) and the usage of hypervisors.

Figure 8 details the evolution of the PpW metric over the baseline environment for an increasing number of computing nodes, thus limited to maximum 14 hosts. We have compared these values with the cases where we have the corresponding PpW measure in the hypervisor environments. This figure outlines many interesting aspects. First of all, with a PpW measure comprised between 700 and 800 MFlops/W, the baseline platform would be ranked between the 93 and 112 position. While surprising at first glance, this result is easily explained by the usage of cutting-edge processors (Sandy-bridge) and the limited number of involved resources – the linear decrease is evident in the figure. The second conclusion that can be raised from this figure is that virtualized environments do not even ensure a more energy-efficient usage during an HPC workload.

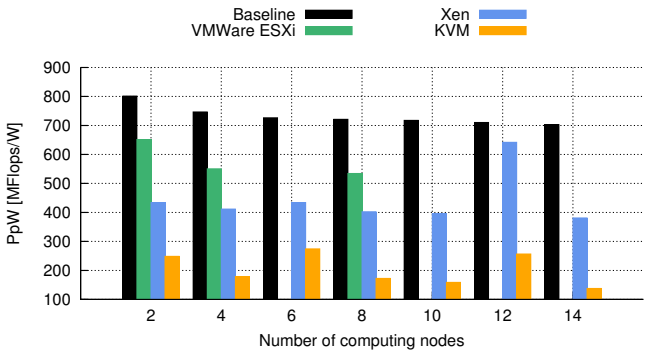

Fig. 8. PpW Metric for the HPL runs.

V. Related Work

At the level of the pure hypervisor performance evaluation, many studies can be found in the literature that attempt to quantify the overhead induced by the virtualization layer. Yet the focus on HPC workloads is recent as it implies several challenges, from a small system footprint to efficient I/O mechanisms.

A first quantitative study was proposed in 2007 by A. Gavrilovska et al. in [12]. While the claimed objective was to present opportunities for HPC platforms and applications to benefit from system virtualization, the practical experimentation identified the two main limitations to be addressed by the hypervisors to be of interest for HPC: I/O operations and adaptation to multi-core systems. While the second point is now circumvented on the considered hypervisor systems, the first one remains challenging.

Another study that used to guide not only our benchmarking strategy but also our experimental setup is the evaluation mentioned in the previous section that was performed on the FutureGrid platform [13]. The targeted hypervisors were Xen, KVM, and Virtual Box and a serious performance analysis is proposed, with the conclusion that KVM is the best overall choice for use within HPC Cloud environment.

In [14], the authors evaluated the HPC Challenge benchmarks in several virtual environments, including VMware, KVM and VirtualBox. The experiments were performed on a rather limited platform (a single host featuring two four-core Intel Xeon X5570 processor) using Open MPI. They demonstrate a rather low (yet always present) overhead induced by the virtual environments, which becomes less significant with larger problem sizes.

In our previous work [1], we try to reflect the main facets of an HPC usage by running the HPCC, IOZone and Bonnie++ benchmarks. We also compare the three most widespread virtualization frameworks, Xen, KVM, and VMware ESXi and examine them against a *baseline* environment running in native mode. It permits to evaluate both high-performance and high-throughput workloads. Also, to abstract from the specifics of a single architecture, the benchmarks were run using two different hardware configurations, based on Intel or AMD processors. The measured data was used to create a statistical holistic model for the power consumption of a computing node that takes into account impacts of its components utilization metrics,

as well as used application, virtualization, and hardware. For this initial study, we performed our experiments on single hosts running a single VM instance. Here, we extended that study by evaluating a more realistic HPC workload that involves far more nodes (whether virtual or physical).

VI. Conclusion

In this work, we evaluated the performance of virtualized environments for HPC-type workloads. Our experiments focused on the three most widespread hypervisor middleware, namely ESXi, XEN and KVM and they were performed on the Grid'5000 infrastructure. The HPC benchmark tool of choice HPL was ran under a varying number of physical (up to 14) and virtualized (up to 96) nodes and the results compared with the baseline runs that had no virtualization layer. Power consumption monitoring, enabled by precise PDU meters available for the Lyon Taurus cluster, was performed for the duration of the benchmarking process.

Our findings show that there is a substantial performance impact introduced by the virtualization layer across all hypervisors, which confirm again, if needed, the non-suitability of virtualized environments for large scale HPC workloads. A non-negligible part of our study includes the energy-efficiency analysis, using the typical metrics employed by the Green500 project [11]. Indeed, virtualization is emerging as the prominent approach to mutualize the energy consumed by a single server running multiple Virtual Machines (VMs) instances. Here again, we demonstrate the poor power efficiency of the considered hypervisors when facing an HPC-type application as demanding as HPL.

The future work induced by this study includes more large-scale experiments, involving other processor architectures. Also, an economic analysis of public cloud solutions is currently under investigation that will complement the outcomes of this work. In general, we would like to perform further experimentation on a larger set of applications and machines.

Acknowledgment

The experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr). This work was completed with the support of the FNR GreenIT project (C09/IS/05). M. Guzek acknowledges the support of the FNR in Luxembourg and Tri-ICT, with the AFR contract no. 1315254. The authors would like also to thank Hyacinthe Cartiaux and the technical committee of the Grid'5000 project for their help and advises in the difficult deployment of the ESXi environment.

References

[1] M. Guzek, S. Varrette, V. Plugaru, J. E. Sanchez, and P. Bouvry, "A Holistic Model of the Performance and the Energy-Efficiency of Hypervisors in an HPC Environment," in *Proc. of the Intl. Conf. on Energy Efficiency in Large Scale Distributed Systems (EE-LSDS'13)*, ser. LNCS. Vienna, Austria: Springer Verlag, Apr 2013.

[2] "Grid'5000," [online] http://grid5000.fr.

[3] 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," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, ser. SOSP '03. New York, NY, USA: ACM, 2003, pp. 164–177. [Online]. Available: http://doi.acm.org/10.1145/945445.945462

[4] A. Kivity and al., "kvm: the Linux virtual machine monitor," in *Ottawa Linux Symposium*, Jul. 2007, pp. 225–230. [Online]. Available: http://www.kernel.org/doc/ols/2007/ols2007v1-pages-225-230.pdf

[5] Q. Ali, V. Kiriansky, J. Simons, and P. Zaroo, "Performance evaluation of HPC benchmarks on VMware's ESXi server," in *Proceedings of the 2011 international conference on Parallel Processing*, ser. Euro-Par'11. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 213–222. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-29737-3_25

[6] R. Bolze, F. Cappello, E. Caron, M. Daydé, F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quetier, O. Richard, E.-G. Talbi, and I. Touche, "Grid'5000: A large scale and highly reconfigurable experimental grid testbed," *Int. J. High Perform. Comput. Appl.*, vol. 20, no. 4, pp. 481–494, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1177/1094342006070078

[7] E. Jeanvoine, L. Sarzyniec, and L. Nussbaum, "Kadeploy3: Efficient and Scalable Operating System Provisioning," *USENIX ;login:*, vol. 38, no. 1, pp. 38–44, Feb. 2013.

[8] A. Petitet, C. Whaley, J. Dongarra, A. Cleary, and P. Luszczek, "HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers," http://www.netlib.org/benchmark/hpl/.

[9] "Hpc2002," [online] http://www.spec.org/hpc2002/.

[10] "Top500," [online] http://www.top500.org.

[11] S. Sharma, C.-H. Hsu, and W. chun Feng, "Making a case for a Green500 list," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, 2006, pp. 8 pp.–.

[12] A. Gavrilovska et al., "High-Performance Hypervisor Architectures: Virtualization in HPC Systems," in *Proc. of HPCVirt 2007*, Portugal, Mar. 2007.

[13] A. J. Younge, R. Henschel, J. Brown, G. von Laszewski, J. Qiu, and G. C. Fox, "Analysis of virtualization technologies for high performance computing environments," in *The 4th International Conference on Cloud Computing (IEEE CLOUD 2011)*, IEEE. Washington, DC: IEEE, 07/2011 2011, Paper. [Online]. Available: http://www.computer.org/portal/web/csdl/doi/10.1109/CLOUD.2011.29

[14] P. Luszczek, E. Meek, S. Moore, D. Terpstra, V. M. Weaver, and J. Dongarra, "Evaluation of the HPC Challenge Benchmarks in Virtualized Environments," in *VHPC 2011, 6th Workshop on Virtualization in High-Performance Cloud Computing*, Bordeaux, France, 08/2011 2011.