

Performance of numerical simulations on the cloud

N. Seoane ¹, R. Valin ² A. Garcia-Loureiro ¹, T. F. Pena ¹ and I. Zablah ³

Abstract— There is a growing interest from companies and institutions in building and utilising Cloud infrastructures. Therefore, in the last few years, the number of projects aimed at developing solutions for managing virtualized enterprise data centers and cloud infrastructures has been booming. CloudStack is an open source solution for building a private, public or hybrid Cloud that provides support for several hypervisors like KVM.

In this paper, we analyse the influence of the number of virtual machines employed per host, as well as the I/O operations, on the performance of two benchmarks: a scientific application based on a one-dimensional semiconductor device simulation and the Linpack benchmark. Both benchmarks were executed under different configurations of the physical host, with and without hyperthreading, and with the virtual machines managed using the KVM hypervisor included in the CloudStack platform.

The obtained results show an important performance loss when more than four virtual machines are deployed in a four-cores host and the hyperthreading is disabled. The effect of the I/O operations on the performance is even more dramatic, showing that the execution time when four or more virtual machines are running on the same physical host can be more than twice the one when a single virtual machine is running.

Keywords— Cloud computing, virtualisation, KVM, benchmarking, CloudStack, semiconductor device simulation, Linpack

I. INTRODUCTION

Over the last years, there has been a very intensive development of Cloud infrastructures and services. There are several properties that give Cloud computing its appeal: the option of a short-term usage (which implies scaling up as well as down when resources are no longer needed), no up-front cost (no hardware investments or software licenses), and elastic capacity on-demand. However, at the same time there are some major obstacles that need to be tackled, such as data transfer bottlenecks due to the more and more data-intensive applications or performance unpredictability issues.

Cloud computing is based on sharing resources among service consumers, partners and vendors. This resource sharing results in various cloud offerings, such as infrastructure as a service (IaaS), which focuses on hardware and IT infrastructure management, or software as a service (SaaS), which deals with traditional software applications [1]. From the technical point of view, two pillars of cloud comput-

ing are service-oriented architecture (SOA) and virtualisation of hardware and software [1].

The important progress seen lately in virtualization is due to the development of several open-source virtual machine hypervisors [2], [3], [4], [5]. KVM [4] is a powerful but relatively simple hypervisor which has found its way into the Linux kernel from version 2.6.20, giving the Linux kernel native virtualization capabilities. It uses hardware-assisted full virtualization, and, therefore, it does not require modified guest operating systems. KVM can run on any Linux platform, provided that it is deployed on microprocessor that supports hardware-assisted virtualisation [6]. Every virtual machine (VM) is considered as a regular Linux process. Under KVM, a normal Linux process has three modes of execution: user, kernel and guest. The user model is used as default for applications, the kernel mode is used when an application needs some service from the kernel (for instance writing to disk) and the guest mode are used by processes that are run from inside the virtual machine [7].

To provide a basic abstraction of the underlying CPU to the VMs, the use of a Cloud management platform is required. For instance, CloudStack [8] is an open-source platform designed to deploy and manage large networks of virtual machines, as a highly available, scalable cloud computing platform. It offers three ways to manage cloud computing environments: a easy-to-use web interface, a command-line interface and a full-featured RESTful API. With regard to its software architecture we can distinguish four main types of components: management server, Pods, availability zones and compute nodes. A scheme representing the basic CloudStack architecture is shown in Fig. 1. The management server administrates the CloudStack infrastructure and the compute nodes are hypervisor-enabled hosts where the virtual machines will be executed. A collection of compute nodes is a Pod, which, at the same time, are grouped in availability zones. These zones are visible to end users who select one of them to initiate a VM.

To evaluate the performance of the Cloud infrastructure, we have selected a scientific computing application, a one-dimensional semiconductor device simulator [10] that has been developed in the USC, and a well known synthetic benchmark, the Linpack numerical library [11]. Using these applications we have evaluated the impact in performance of several key factors in cloud computing: the number of virtual machines per host, the influence of the hyperthreading and the hard disk I/O.

¹Departamento de Electrónica e Computación, Universidade Santiago de Compostela, e-mails: natalia.seoane@usc.es, antonio.garcia.loureiro@usc.es, tf.pena@usc.es

²Supercomputing Center of Galicia (CESGA), e-mail: rvalin@cesga.es

³Universidad Nacional Autónoma de Honduras, e-mail: mrzablah@unah.tv

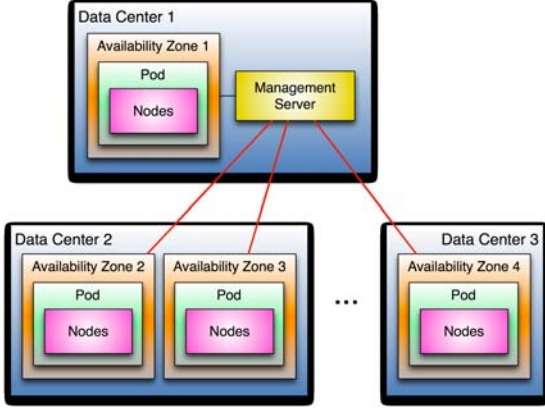


Fig. 1. Scheme representing the CloudStack architecture.

This paper is organized as follows. Section II describes the main characteristics of the two applications used as benchmarks. Section III presents the obtained results and the main conclusions of this study are drawn up in Section IV.

II. BENCHMARKS

To evaluate the influence of the hyperthreading and the I/O on the performance of the virtual machines we have selected two benchmark tests: a 1D semiconductor device simulator (1D-SIM) [10] and the Intel Linpack [11], [12], as synthetic benchmark. The core of both applications is the solution of linear systems of equations, which are sparse matrices for the 1D-SIM and dense for Linpack. Next, we are going to describe the main features of each application.

1D-SIM is a one-dimensional drift-diffusion simulator for semiconductor devices. A simple flow diagram of the simulation process is shown in Figure 2. The basic equations to be solved in the drift-diffusion model are the Poisson equation and the electron and hole continuity equations [13]. The finite element method is applied to discretise the device. For a device discretised in N nodes, a set of $3N$ coupled non-linear equations need to be solved. The unknowns of the problem are the electrostatic potential (ψ) and the quasi-Fermi potentials for electrons (ϕ_n) and holes (ϕ_p). The Gummel method is applied [14] to decouple these equations, which are then linearised using the Newton method [15]. The resulting linear system of equations is solved using the BiCGSTAB iterative method preconditioned with an incomplete LU factorisation, which depends on both, a certain fill-in and a numerical threshold. In this work, a heterojunction bipolar transistor has been used as benchmark test, including the thermoionic emission and tunneling effects.

The Intel Linpack benchmark solves a dense system of linear equations, measures the amount of time it takes to factor and solve the system, converts that time into a performance rate, and, finally, tests the results for accuracy. This benchmark, developed in 1976 by Jack Dongarra at the Argonne National Lab-

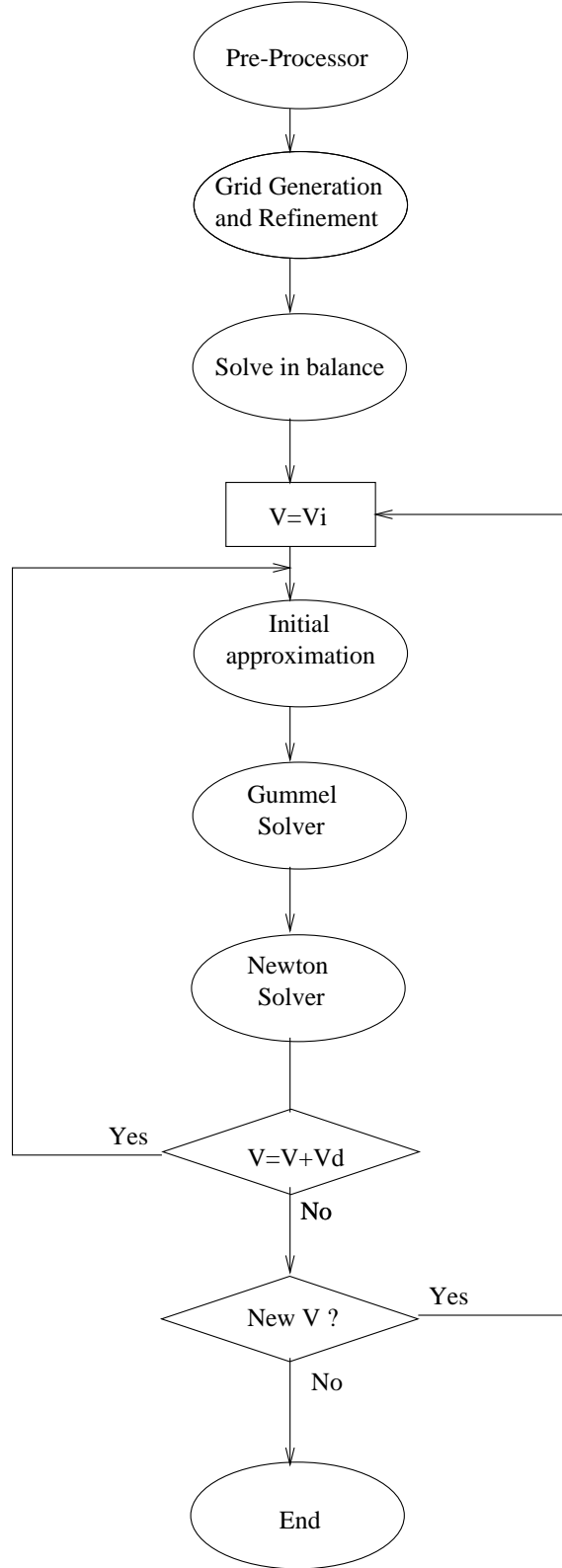


Fig. 2. Flow diagram of the 1D drift-diffusion simulator.

oratory, is well known and it is being used for instance, to create the TOP500 list [16] which gives a ranking of the most powerful supercomputers in the world.

III. NUMERICAL RESULTS

In this section we show the simulation results obtained from the two different benchmark tests. Ini-

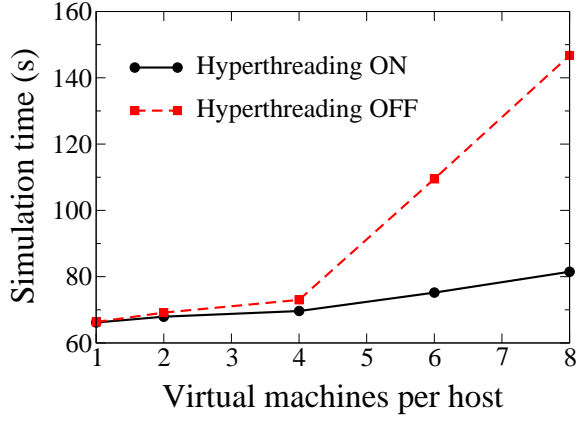


Fig. 3. 1D drift-diffusion simulation time versus the number of virtual machines per host when no I/O is generated. Results with hyperthreading activated or deactivated are compared.

tially, we describe the main hardware and software used in the host and in the VMs.

A. Hardware and software description

The machine used as host is an Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz processor with 4 cores and 8 GB of RAM. The interconnection network is a GigaBit Ethernet. The operative systems of the host and of the virtual machines are CentOS 64Bit release 6.1 and Debian 6.0.4 64Bit, respectively.

The network file system used is NFS V3 with the following configuration:

```
nfs rw, relatime, vers=3, rsize=1048576,
wsize=1048576, namlen=255, hard, proto=tcp,
timeo=600, retrans=2, sec=sys,
```

To perform our analysis, we have deployed several virtual machines in the same host using the KVM hypervisor (qemu-kvm-0.12.1.2). The virtualisation API [17] is libvirt (version 0.9.4). Each one of the VMs has 1 core and 1GB of RAM available. The virtual CPU is QEMU Virtual CPU (version cpu64-rhel6).

B. Simulation tests

Initially, we have modified the 1D-SIM to remove the writing operations to hard disk. In the physical host, with the hyperthreading disabled, we execute the code and it takes 64.7 s to obtain the solution. Under the same circumstances, a virtual machine deployed with the CloudStack agent requires just 2.5% more time to reach the same solution. To evaluate the impact of the number of virtual machines deployed per host on the performance of the one-dimensional simulator, we consider n VMs with $n=2,4,6,8$. The first $n-1$ VMs execute the simulator for a certain number of times and the n -th one runs the code and estimates simulation times.

Fig. 3 shows the simulation times versus the number of VMs deployed per host when there is no I/O generated. The influence of the hyperthreading is also shown in the figure. As expected, there is an increase in the simulation time when the number of

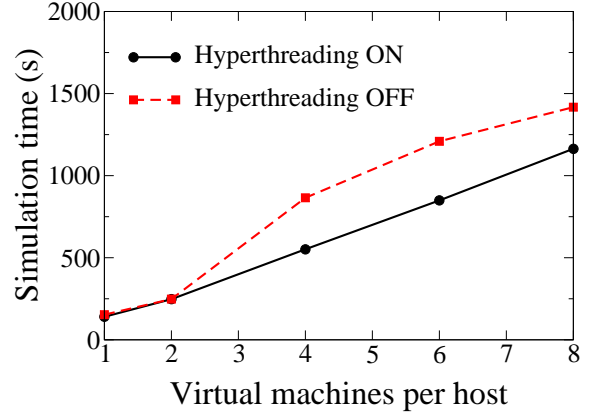


Fig. 4. 1D drift-diffusion simulation time versus the number of virtual machines per host when a realistic I/O is generated. The influence of the hyperthreading is also shown.

VMs per host is increased. When the hyperthreading is enabled, this increase is lower than 6% when 4 or less VMs are running and it reaches 13% and 23% respectively, for 6 and 8 VMs. However, if the hyperthreading is disabled, the loss in the performance is more noticeable. In this case, when 2 or less VMs are deployed the increase in the simulation time is lower than 5% but it reaches 10% when 4 VMs are used. For a higher number of VMs per host, there is an important deterioration in performance, observing on 6 and 8 VMs, simulation times around 65% and 120% higher, respectively, than the ones obtained on 1 VM.

Now, the influence of the I/O on the performance of the 1D-SIM is studied. In order to this, during each iteration of the simulation process the values of the different variables which are calculated during the simulation (electrostatic potential, and electron and hole concentrations) are written to a file for every node of the mesh. This is a more realistic test and it is performed in the same way as the previous one, but including an additional VM in another host that exports a NFS file system that is shared among the other VMs to store the simulation results. This additional VM has 4 cores and it is deployed in an Intel(R) Core(TM) i7-2600 CPU @ 3.4GHz host.

Fig. 4 shows, for this case, the simulation times versus the number of VMs deployed per host. The impact of the hyperthreading is also shown in the figure. When the number of processes that use the disk NFS storage system is increased, there will be more interruptions in the application due to the number of processes competing for getting access to the disk. Hence, there is a dramatic increase in the simulation time when the number of VMs is increased. When the hyperthreading is on, it ranges from an increase of 76% when 2 VMs are deployed to 730% with 8 VMs. When the hyperthreading is disabled and 8 VMs are being used in the same host, we observe an increase in the simulation time of around 825% when compared to the value obtained when 1 VM is used.

We have performed a similar study using the Intel Linpack benchmark. In this test we have used

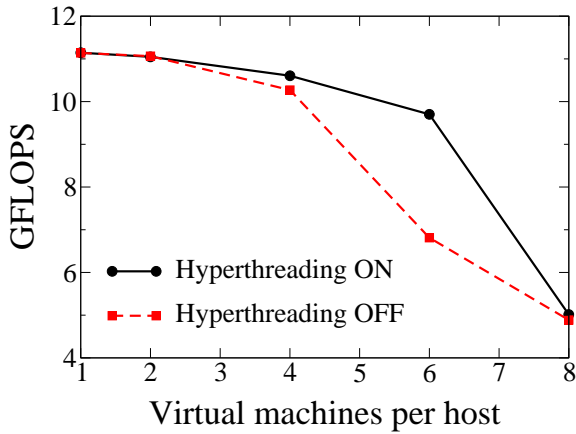


Fig. 5. Floating-point operations per second versus the number of virtual machines per host when using the Linpack software. The influence of the hyperthreading is also shown.

a problem size of 5000, with alignment values of 4KBytes. Linpack provides the average performance in GFLOPS by all the Linpack runs (5 in our case) for a single test. These results are shown in Fig. 5, for different virtual machines per host. In order to make a proper comparison between the results obtained when using Linpack and 1D-SIM, Fig. 6 shows the execution time versus the number of VMs for the Linpack test. In these figures, the impact of the hyperthreading is also shown. When just 1 VM is deployed per host Linpack requires around 43 s to solve the test and it gives a performance of 11.1 GFLOPS. Note that, when the hyperthreading is disabled, the simulation times follow exactly the same trend observed in Fig. 3. When compared to the results obtained in 1 VM, there is a minimum increase in the simulation time for 2 VMs that rises to 9% for 4 VMS. For more than 4 VMs per host, we notice the same degradation of the performance as in the 1D-SIM, observing on 6 and 8 VMs simulation times around 63% and 124% higher respectively, than the ones obtained in 1 VM. Therefore, while the decrease in the performance is less than 1 GFLOPs when 4 or less VMs are being deployed, there is a drop of around 4 or 6 GFLOPs when 6 or 8 VMs are used, respectively. When the hyperthreading is enabled, the simulation results up to 6 VMs also follow the same behaviour as in Fig. 3, showing similar percentages of the increase in the simulation time when the number of virtual machines is increased. This penalty in the simulation time is lower than 5% when 4 or less VMs are being used, and it reaches 13% for 6 VMs. For all these configurations, the decrease in the performance is less than 1.5 GFLOPs. However, when 8 VMs are being deployed in one host, the efficiency drops very quickly, leading to an increase of 112% in the simulation time and to a decrease of 6.1 GFLOPs in the performance.

Finally, note that when Linpack is run concurrently with processes that make an intensive use of the hard disk, we do not expect any important effect in the result, since this benchmark does not perform

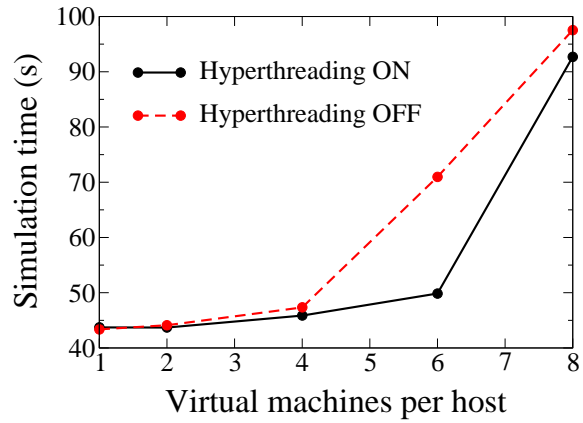


Fig. 6. Simulation time versus the number of virtual machines per host when using the Linpack software. The influence of the hyperthreading is also shown.

writing operation on hard disk. Therefore, we do not present a study of the influence of the I/O using this benchmark.

IV. CONCLUSION

The last few years have seen a boom in cloud technologies since they offer the possibility to virtualise services, providing virtual machines to users for their execution. There is a wide range of open-source solutions for building private, public or even hybrid clouds. Among them is the CloudStack platform. In this paper, we have used the KVM hypervisor included in this platform to manage the virtualised services. The objective is to analyse the influence of the hyperthreading, the number of virtual machines employed per host and the I/O on the performance of two benchmarks, a one-dimensional semiconductor device simulator and Linpack.

Results from both benchmark tests show a similar behaviour, with an important degradation in the performance when the hyperthreading is disabled and more than 4 virtual machines are deployed in the same host. The effect of the I/O operations on the performance is even more noticeable, with increases in the simulation time over 200% when 4 or more virtual machines are used.

ACKNOWLEDGEMENT

The work developed in this paper has been supported in part by the Ministry of Education and Science of Spain and FEDER funds under contract TEC2010-17320 and by the Xunta de Galicia under contracts 2010/28 and 09TIC001CT, and also with aid of the UNESCO/Keizo Obuchi Research Fellowships Programme (UNESCO/Japan Young Researchers' Fellowship Programme) cycle 2011.

REFERENCIAS

- [1] L. J. Zhang, J. Zhang, J. Fiaidhi and J. M. Chang, *Hot Topics in Cloud Computing*, IEEE IT Professional, Vol. 12, No. 5, pp. 17-19, 2010
- [2] VirtualVM, <https://www.virtualbox.org>
- [3] OpenVM, <http://openvz.org>
- [4] KVM, <http://www.linux-kvm.org>
- [5] Xen, <http://xen.org>

- [6] M. T. Jones, *Discover The Linux Kernel Virtual Machine*, EMULEX Corp. IBM-DeveloperWorks., <http://www.ibm.com/developerworks/linux/library/l-linux-kvm/>, 2007
- [7] I. Habib, *Virtualization with KVM*, Linux Journal, <http://www.linuxjournal.com/article/9764>, 2008
- [8] CloudStack, <http://cloudstack.org/>
- [9] F. Gomez-Folgar, J. López Cacheiro, C. Fernández Sánchez, A. Garcia-Loureiro and R. Valin, *An e-Science infrastructure for nanoelectronic simulations based on Grid and Cloud technologies*, Spanish Conference Electron Devices (CDE), pp. 1-4, 8-11, 2011
- [10] A.J. García Loureiro, T.F. Pena, J.M. López González and Ll. Prat *Parallel implementation of a simulator for heterojunction bipolar transistors*, VIII Symp. on Parallelism, Cáceres, pp. 41-50, 1997
- [11] J. J. Dongarra, C. B. Moler, J. R. Bunch and G.W. Stewart, *LINPACK Users' Guide*, SIAM publishing, 1979
- [12] Linpack, <http://software.intel.com>
- [13] S. Selberherr *An Analysis and Simulation of Semiconductor Devices*, Springer, 1984
- [14] D.L. Scharfetter and H.K. Gummel *Large-Signal Analysis of a Silicon Read Diode Oscillator*, IEEE Trans. on Electron Devices, pp. 64-77, 1969
- [15] R.E. Bank and D.J. Rose, *Parameter Selection for Newton-Like Methods Applicable to Nonlinear Partial Differential Equations*, SIAM J. Numer. Anal., Vol. 17, No. 6, pp. 806-822, 1980
- [16] Top 500 list, <http://www.top500.org/project/linpack>
- [17] The virtualization API, <http://libvirt.org/>