HPC CloudPills: on-demand deployment and execution of HPC application in cloud environments

Pietro Ruiu, Olivier Terzo
Istituto Superiore Mario Boella
Torino, Italy
(ruiu, terzo)@ismb.it

Giuseppe Carlino, Rossella Prandi
Simularia S.r.l.
Torino, Italy
(g.carlino, r.prandi)@simularia.it

Alberto Falzone, Paolo Maggi,
Livia Torterolo, Enrico Usai
Nice S.r.l.
Asti, Italy
(alberto.falzone, paolo.maggi, livia.torterolo,
enrico.usai)@nice-software.com

Gianpaolo Perego
Aethia S.r.l.
Torino, Italy
gianpaolo.perego@aethia.com

***Abstract*—High Performance Computing (HPC) is a valuable instrument in many areas of scientific research and industrial production. However, due to the high investment costs for its implementation, its diffusion is often restricted to large research centres or medium to large size industrial companies. The project *HPC CloudPills* aimed at exploring how the technologies of Cloud Computing can help to reduce the cost and complexity associated with creating, maintaining and managing IT infrastructure for high performance computing. The objective is to make the high-performance computing accessible to reality today not capable of supporting the high initial costs and management that it involves. In this paper the general architecture of *HPC CloudPills* will be described and some tests performed on the system with SPRAY (an HPC application).**

***Index Terms*—Cloud computing; HPC; automation; virtual cluster; HPCaaS;**

I. Introduction

High Performance Computing (HPC) is now common in many areas of scientific research and industrial production, such as aeronautical engineering, theoretical chemistry, meteorology or satellite data analysis where it is used to solve problems which cannot be addressed with ordinary mathematical tools. In other areas, moreover numerical simulation has become a valuable tool to reduce the number of experimental tests, often difficult to run and therefore expensive, such as in the case of designing buildings and vehicles, in the field of materials science, in microelectronics, telecommunications and in the pharmaceutical industry. Despite its widespread adoption in different sectors shows its undoubted utility, HPC technology is still difficult to access for small to medium-sized companies with insufficient economic resources. In fact the costs and complexities associated with the creation, the maintenance and the management of IT infrastructures for high-performance computing are still quite high. In addition, those organisations that hold an adequate IT infrastructure and can support the costs of management and maintenance are often faced with various problems such as:

- The proper sizing of the infrastructure: it is common over-sizing the infrastructure to its workload peak level, with hardware and software resources often unused, or also under-sizing it, causing losses in terms of performance;
- Optimization of configurations for multitenancy: the need to support a huge variety of HPC applications with very different requirements often forces IT administrators to implement trade-off solutions in the optimization of system configurations;
- Applications that require different operating systems: makes it necessary to implement complex strategies (often not fully automated) to reboot the physical nodes of the cluster or to split disks in homogeneous partitions for each operating system;
- The installation and management of multiple versions of the same software: HPC applications are often very complex to install and configure and it is frequently requested to support different versions of the same application in the same system.

In the recent years cloud computing has emerged as a cost effective alternative to dedicated infrastructure for HPC applications. HPC community has seen potential of cloud computing and more and more research groups from academia and industry started to use it, exploiting benefit of elasticity of resources and elimination of management costs and time [1], [2], [3], [4], [5]. The major idea of cloud computing lies in the flexible outsourcing of software and hardware services which can be accessed through standards protocols over the Internet. As defined by the National Institute of Standards and Technology (NIST) Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [6]. Thus one of the major benefits for cloud users is the possibility to access to flexible low cost compute resources, without the efforts and investments to set up and run an expensive local computing system.

However the quality of the current HPC offers in the cloud is not often fully satisfying, neither from the point of view of the performance, neither from the point of view of the complexity of the set-up and maintenance of the computing clusters. Regarding the performances issues, some providers are equipping their clouds with resources and tools specific for HPC, such as robust VMs with huge RAM and CPU, low-latency systems and GPU processors. Much remains to be done instead from the point of view of reducing the complexity of creating and managing the computing clusters.

The project *HPC CloudPills* intends to address the latter challenge by proposing a solution that reduces complexity associated with creating, maintaining and managing IT infrastructure for high performance computing. The aim is to build a system for creating dynamic on demand pre-configured virtual cluster which meet the minimum hardware and software requirement to run specific HPC applications. These virtual clusters are automatically sized, making available to end users the required amount of computing resources and storage. The HPC developers have only to provide the application, the *pill*, properly packed according to the *HPC CloudPills* format: it consists in a tar file containing the image of the VM and some configuration files. Then through a specific Graphical User Interface (GUI) the application can be published to the cloud. Since this moment the cloud user can deploy the application onto a cluster of VMs specifically configured for its execution, without the worry to set-up and configure the computing resources or to manage others infrastructure details. The *HPC CloudPills* solution is an optimized system for the automatic deployment of HPC computing in the cloud, composed by ad-hoc developed modules and open source solutions well integrated and combined together.

In order to evaluate the validity of the proposed solution, one pilot application will be analysed and will be delivered to the *HPC CloudPills* system to be executed through virtual cluster on cloud computing infrastructure.

The rest of the paper is structured as follows: Section II explains the related work. Sections III and IV describe the project architecture. Section V presents the testbed and system implementation. Section VI describes the SPRAY test application and system performance analysis. Eventually, Section VII draws the conclusion and presents future developments.

II. Related work

As stated above, the cloud technology can bring many advantages such as the benefits of virtualization, elasticity of resources and elimination of infrastructure setup cost and time. For these reasons, in recent years more and more tools and solutions, both commercial and open source, have been developed and appeared on the market, contributing to develop the concept of HPC in the cloud. However, as far as we know none of the solutions on the market owns the functionalities of the system designed in the project.

Even if today HPC in the cloud offers are very heterogeneous, from our point of view, can be regrouped in two main categories of offerings. The first includes virtualization-less solutions, that provide bare-metal computing models, with performances and complexity similar to in-house computing clusters [7], [8], [9]. Providers make available special hardware like GPUs, low latency interconnects and high-speed storage; resources can be reserved or shared with other users. These kind of systems have very good performances and are provided with pay as you go formula, but brings drawback of traditional supercomputing infrastructure that reside in a difficulty in configuring resources to suit needs and preferences of users, due to the lack of virtualization layer. Furthermore, during the set-up of the clusters, the provider interaction with the user is very high, due to the complexity of the system. In the second case, offers are characterized by the existence of a virtualization layer [10], [11] which give extreme flexibility but reduce drastically the performances of the systems. For this reason, these kinds of environments are appropriate for highly parallel applications which don't need special low latency systems. In the majority of cases users can access to the Infrastructure as a Service (IaaS) layer and have to deploy manually the needed resources, VMs or clusters. Also the installation of the scheduler, the queue manager and the application itself is on charge of the users. Recently are appearing on the market even some Platform as a Service (PaaS) solutions allowing users to automatically deploy Workflow Management Systems (WMS) on third party cloud platform [12]. These environments are linked through special adapters to all the leading public cloud and private cloud providers. But, even in this case the deploying of the clusters is in charge to the users.

From the analysis on the current offers of the marked it is possible identify some distinctive elements of the solution developed in the project. In particular on all the products observed the application layer is detached from the infrastructure layer of the solution and users should handle both in order to run the HPC application. The novelty of the developed solution lies precisely in this feature: *HPC CloudPills* reduces the complexity associated to the management of HPC applications in the cloud, providing a turnkey solution, in which the user does not have to care about the set-up of the infrastructure layer.

III. HPC CloudPills system

The main system components are described in the next paragraphs and are the following:

- OpenNebula, is the cloud computing platform that allows to manage in a dynamic and optimized way both physical and virtual resources;
- Openlava is the Workload Management System (WMS), able to manage clusters scheduling and executing application's jobs;
- Automation scripts, developed from scratch during the projects, that allow to create virtual clusters and launch the selected application from *Pill* application, in automatic way.

A. Cloud platform - OpenNebula

For the purpose of the project it has been decided to adopt OpenNebula (ON) [13] as cloud management platform, that is more suitable to satisfy *HPC CloudPills* requirements among all other solutions that have been considered. ON allows to implement IaaS (Infrastructure as a Service) solution for private, public and hybrid cloud through several tools for managing the main infrastructure aspects such as storage, networking, virtualization, monitoring and security. ON was developed to be customizable for any infrastructure and simplify the integration of new component.

Physical infrastructure for interconnection between node is based on a classical architecture cluster-like where are available a front-end and several cluster that hosted VMs. Main components are the following:

- Front-end provides ON services;
- Hosts supply of hypervisor that provides resources (CPU,RAM) used by VM;
- Image repository, storage for VM images;
- Networking infrastructure that support VLAN for VMs;

The complexity of the system is almost entirely focused in the front-end, while nodes belonging to the cluster should have only hypervisor. The main communication interfaces can be classified into two types: Cloud Interfaces for end-user and System Interfaces. Cloud interfaces can be used to develop useful tools for end users, providing high-level abstraction of cloud services. System Interfaces present all ON features and are used mainly to tailor and optimize infrastructure.

B. Workload Management System - Openlava

Openlava has been chosen as the Workload Management System (WMS) for the project [14]. It has been selected after an accurate analysis of the WMS technologies available in the market for clusters, grids and cloud platforms. A specific task of the project has been dedicated to the comparison of the most commonly used open-source and commercial WMS: Openlava, Oracle Grid Engine, PBS Professional, SLURM, LSF, Torque. The WMS has been compared according to a set of selected criteria, organized in these categories: supported platforms, installation, monitoring, ACL, GUI, nodes discovery, MPI jobs, other. The requirements of the project were fullfilled by almost all the WMS, so the open-source tools has been preferred according to the context of a research project, and, among them, Openlava has been considered the most innovative and promising WMS. Openlava is a software for the distributed resource management of large scale infrastructures, and provides an optimized management of the workload on the available computing resources. Openlava is a fork of Platform Lava, which was positioned by Platform Computing as an entry-level workload management system. Platform Lava is based on Platform LSF and supports clusters of 512 nodes. Scalable and robust, Openlava is suitable for heavy duty production clusters, with hundreds of nodes, thousands of cores and jobs. Comparing to its precursor, Openlava provides significant differences while providing the same interfaces in all aspects, configuration, command line and APIs. These concern installation and implementation, up to most common features like user job prioritization, dynamic cluster growth, storage of accounting data. Openlava provides cluster host load information, remote command execution and a distributed batch system.

C. Script for the deployment of clusters

For the creation and configuration of custom cluster on the cloud platform in automatic way were developed some script in bash language. The main tasks carried out concerning the network configuration, creation of template for the VM, the VM deployment, creation of the cluster, the VM configuration, the WMS configuration and provision of elaboration environment for the application . The code is collected in three separate files with the following syntax:

- create-vcluster.sh, to deploy VM;
- vmcontext-master.sh, master node configuration;
- vmcontext-compute.sh, compute node configuration.

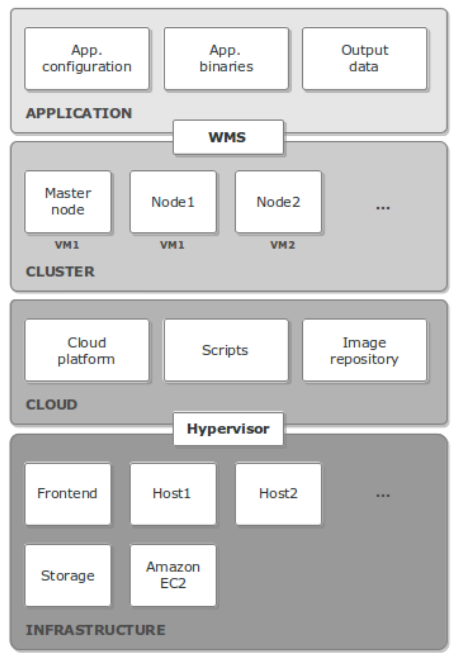

Fig. 1. Architecture of the platform and principal logical layers.

IV. ARCHITECTURE

The architecture of the platform is based on four logical layers (Infrastructure, Cloud, Cluster, Application), as you can see in Figure 1.

Each layer completely relies upon the underlying one, and includes all the technologies used to efficiently manage its resources:

- Infrastructure: this is the physical layer and includes all the hardware components. All the servers are virtualized through the KVM open source hypervisor [15]. This layer includes also the resources coming from public clouds and partecipating to the computing capabilities of the system.

- Cloud: this is the resource management layer, and includes the cloud computing platform and the scripts developed during this project to automatically create virtual clusters. The layer is devoted to the design, creation, deployment and monitoring of virtual machines.
- Cluster: this is the computing layer, and includes the collection of virtual machines dedicated to the execution of a specific application. It also includes the Workflow Management System (WMS), which provides useful services to manage the jobs running inside the virtual cluster.
- Application: this is the top layer, and includes the final HPC application, its environment and configuration, and all the input and output data.

The storage provides a image repository to every host that mounts it through NFS. The images are used by the cloud platform for the instantiation of the virtual machines. The network infrastructure supports VLANs, and makes it possible to provide dedicate subnets to the virtual cluster nodes.

V. IMPLEMENTATION

The system consists of three servers and an external storage (DAS) unit (see Figure 2), interconnected by 1Gb Ethernet and Fiber Channel for the storage part (see Table I). Details of the system:

- Server hpcpills1 - Intel S5400SF, 2 x Intel Xeon 5472, 8 GB RAM, 1 x Hard disk 500GB, 2 x 1Gb Ethernet ports. Operating system: Centos 6.4 x86 64;
- Server hpcpills2 - Intel S5400SF, 2 x Intel Xeon 5472, 8 GB RAM, 1 x Hard disk 500GB, 2 x 1Gb Ethernet ports. Operating system: Centos 6.4 x86 64 (same as above);
- Server hpcpills3 - Supermicro PDSML, 2 x Intel Core2 6300, 3 GB RAM, 1 x Hard disk 250GB, 2 x 1Gb Ethernet ports. Operating system: Centos 6.4 x86 64;
- Storage - DotHILL E6550, dual FC controller, 6 x 300SAS Hard disk in RAID6;

The Ethernet connections are provided by a Netgear Prosafe 24 Gb ports switch.

A. Image preparation

For goals of this work it is needed to create from scratch new custom images of Virtual Machine (VM) to be assumed as reference image for every kind of chosen application. The chosen distribution, at development time, is Linux CentOS 6.2 64 bit. The ISO image used to perform the first installation was the minimal one, in order to get a virtual machine on which adding required packages only, optimizing the space needed for them.
The chosen hypervisor type is KVM; it was chosen for systematic management reasons; other OpenNebula supported hypervisor types can be used as valid alternatives. Working by remote on hypervisor hosts, a management dedicated VNC session was started. From virt-manager interface, the first installation of Linux CentOS 6.2 64 bit was performed from scratch, with 1 Virtual CPU, 512 MB RAM, 8GB Virtual disk

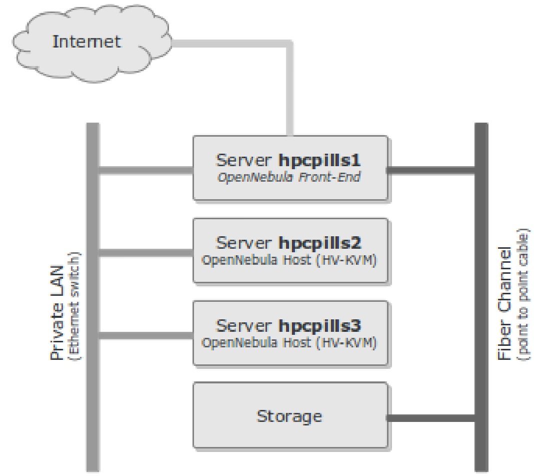

Fig. 2. Physical machines of the test bed and network interconnections.

as VIRT-IO device, one network card as VIRT-IO device.
The VIRT-IO feature is very important one in order to get the maximum of performance on virtual disk using internal buses of hypervisor, obtaining good speedup over I/O operations and throughput ones. On the installed VM up & running, next activities were: the update of OS packages from CentOS repositories, installation of software stack about HPL and P-SPRAY with all required libraries, installation of OpenLava, as choosen as LSRM. Starting from this image, ready right now to be used, it can be used as argument to build a new OpenNebula template for virtual machine instantiation. Referring to images ready for virtual clusters, in particular, this creation step can be executed automatically through bash script. Also, all of binaries, libraries and bash scripts installed on this image are self-consistent and not depending from other external tools.

B. Virtual Cluster and Pills

A Virtual Cluster is a pool of VMs in number greater than one, on which the LSRM is the glue middleware that group them into an unique logical entity. Using the image-disk file as building block, all of the rest of procedure about instantiation of Virtual Cluster has been automated using bash scripts. Both image-disk file, created into previous stage, and all configuration and script files make a *Pill*, provided as a tarball file. One of configuration files is the CloudPill descriptor file, in XML format, that allows the management from Web Portal services in order to assemble the properly command for Pill instance submission.
The workflow can be summarized as follows (Figure 3): the user does request a new Virtual Cluster instance filling all of required parameters, such as, i.e., cpu number, core per cpu number, node number, memory and others; the Pill tarball is extracted and new OpenNebula image is created; new OpenNebula templates are created; new Virtual Private Network is created; new VMs are instantiated in number of N

Hostname	RAM [MB]	Cores	Proc model	Freq [GHz]	Role
hpcpills1	8055708	8	Xeon X5472	3.00	front end
hpcpills2	8055712	8	Xeon X5472	3.00	host (HV-KVM)
hpcpills3	2955764	2	Core2 6300	1.86	host (HV-KVM)

TABLE I
HARDWARE COMPONENTS OF TESTBED.

requested node plus one VM as LSRM master node for job submission only.

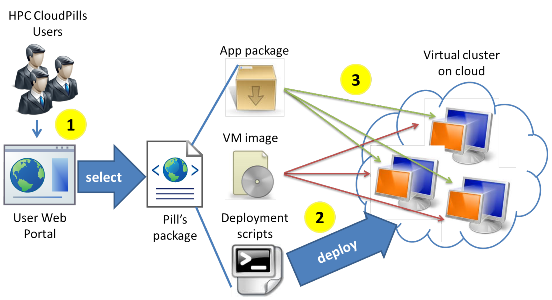

Fig. 3. Deployment of a Pill in the HPC CloudPills system.

VI. PERFORMANCE ANALYSIS

A. Pilot application

SPRAY is the application chosen to test the cloud platform feasibility and performance. It is a three dimensional Lagrangian particle model to simulate the dispersion of airborne pollutants in the atmosphere (see [16] and references therein). The code reproduces the transport, dispersion, dry and wet deposition and radioactive decay of chemically inert species released in complex meteorological conditions, such as low windspeed, flow over complex topography, together with inhomogeneous spatial and temporal meteo-diffusive variables, as vertical wind shear, breeze due to the presence of terrain discontinuities. SPRAY can also simulate the dispersion of particulate matter by taking into account the gravitational vertical settling phenomenon. Pollutants can be emitted by point-like, area-like or linear sources, either continuous or discontinuous. The model can perform simulations at local scale, with horizontal resolutions of about 100 m. It also allows to take into account the presence of obstacles to perform simulations at microscale with a resolution lower than 10 m. Tests for *HPC CloudPills* have been performed with the parallel version of SPRAY, a code written in FORTRAN 90 with MPI library. Several degrees of parallelization are allowed by the model:

- Level 0: SPRAY works on a single tile (spatial domain) with a single core. In this case, the behaviour of the model is the same as the scalar model;
- Level 1: SPRAY works on a single tile (spatial domain) with a number of cores greater than 1. In this case, the particles and/or the sources are distributed between the available cores.
- Level 2: SPRAY works on multiple tiles configuration with a number of cores greater than 1.

In the tests here presented, level 1 parallelization has been adopted. Since a single tile is present, the code does not need to treat the exchange of particles between adjacent tiles. Non interacting particles (i.e. heavy gas computation off) are emitted by the sources and randomly distributed to the available cores. At storage time, 3D concentration files computed independently by each single core are summed up. When the number of cores allocated to a single tile is greater than 1, MPI instructions are enabled and one master core is elected among the pool of available cores. The master core is dedicated to broadcast input data and does not directly process particles. Therefore, if N is the number of cores allocated to a level 1 parallel job, the number of true computing cores is N-1. The single tile mode requires very limited MPI communications, these being broadcasting of input data by master core at the beginning of the run and transmitting 3D concentration fields and optionally of 2D deposition fields at the end of every storage time step. In the standard situation, as long as the storage time step is large with respect to the particles displacement time step, the speedup obtained by increasing the number of computational cores is quasi-linear. The slope of the speedup curve tends to decrease only once the number of particles moved by each individual core becomes too small compared to the load of MPI communications. In the tests performed the parallel version of SPRAY has been compiled with Intel 12.3.3 fortran compiler (ifort) linked with MPICH2 1.4.1p1 libraries.

B. Performance test

In order to evaluate the solutions developed, performance analysis have been performed. It is worth mentioning that in this phase of the work best numeric performances, in terms of CPU time, have not been the main target of the described tests.

The performances of *HPC CloudPills* system has been assessed both with the HPL benchmarking tool (a portable implementation of the High-performance Linpack Benchmark for distributed memory computers) and with the SPRAY application, described in Sec. VI-A, applied to a realistic test case. The hardware configuration of the available test machine is described in Table I. Tests have been performed both on physical and virtual cluster. In Figure 4 and 5 the results of HPL benchmark are shown in the case of 8 (2x 4) and 9 (3x 3) cores for different sizes of the problem, for physical and virtual cluster. It is quite clear that the performances of the virtual cluster are worse by approximately one order of magnitude with respect to the physical cluster. In Table II, the CPU time needed to run a realistic test case with the SPRAY application in the case of virtual clusters with

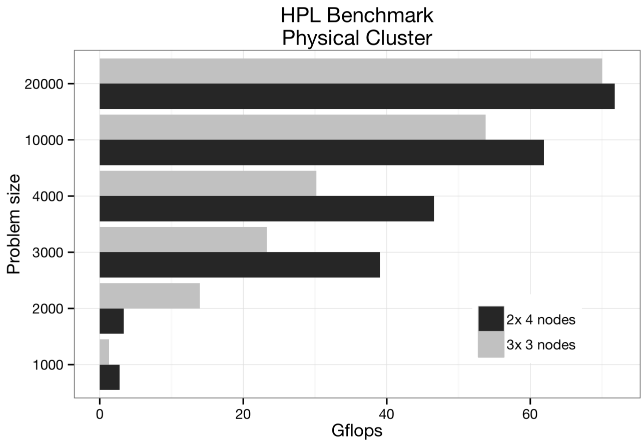

Fig. 4. Comparison of performances of HPL benchmark on physical clusters of 8 (2x 4) and 9 (3x 3) computational cores, as a function of the problem size (performance in Gflops).

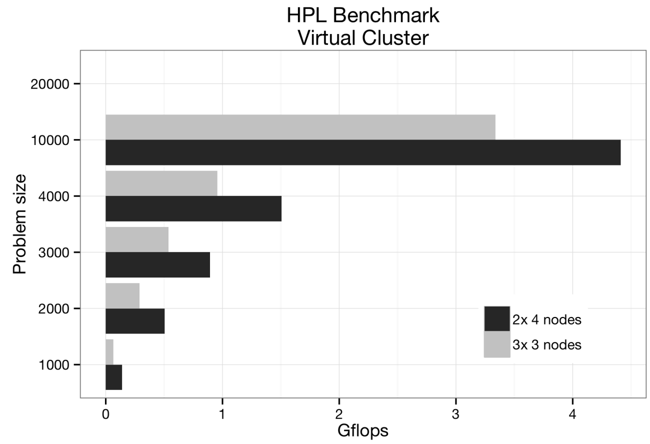

Fig. 5. Comparison of performances of HPL benchmark on virtual clusters of 8 (2x 4) and 9 (3x 3) computational cores, as a function of the problem size (performance in Gflops).

8 computational nodes and different configurations is shown. The results are extremely variable ranging from 264866.38 secs to 1200.01 secs depending on the selected architecture of the virtual clusters. For comparison, in the physical cluster with the specific configuration of 6 cores on nodes02 and 2 cores on the much slower nodes03 the average CPU time needed to complete the same test case is 2242.17 secs. In this case, the test application, run on a virtual cluster with multiple hosts with single node, has comparable performance with the physical cluster.

Test	CPU Time (sec)	Hosts
1	264866.38	8x node001
2	14123.73	4x node002 4x node001
3	5827.42	4x node002 4x node001
4	2558.87	2x node004 2x node003 2x node002 2x node001
5	2550.35	2x node004 2x node003 2x node002 2x node001
6	1534.78	2x node004 2x node003 2x node002 2x node001
7	1200.43	1x node008 1x node007 ... 1x node001
8	1200.01	1x node008 1x node007 ... 1x node001

TABLE II
PERFORMANCE OF SPRAY APPLIED TO A REALISTIC TEST CASE ON VIRTUAL CLUSTERS WITH DIFFERENT SETUP (PERFORMANCE IN CPU TIME).

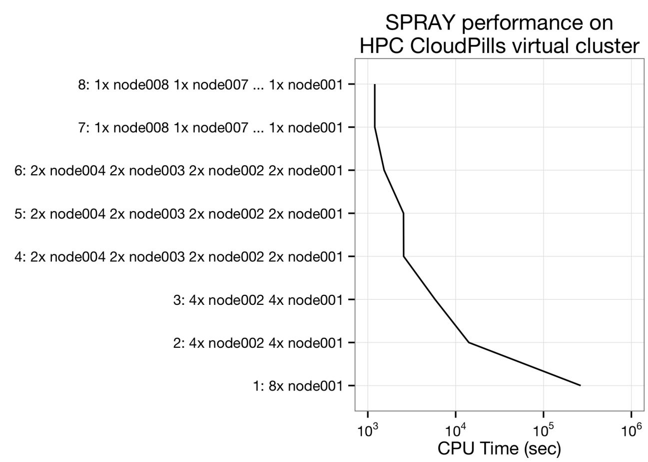

Fig. 6. Performance of SPRAY applied to a realistic test case on virtual clusters with 8 computational nodes, as described in table II (CPU time in seconds).

VII. CONCLUSION

The aim of the project is the study and the development of a set of technologies that, combined together, will help make it easy and inexpensive the on-demand deployment and running of HPC applications in the cloud. The aim is to build a system for creating dynamic on demand pre-configured virtual cluster which meet the minimum hardware and software requirement to run specific HPC applications. These virtual clusters are automatically sized, making available to end users the required amount of computing resources and storage.

At the actual stage of the development, the system allows the deployment of HPC applications in a private cloud based on OpenNebula. It is therefore a solution potentially usable only by companies provided with a datacenter. However, is in the intention of the authors to transform the solution in a public cloud service, accessible through a web GUI, which can also be accessed by users without the infrastructure. In addition, another goal of the project is the realization of a marketplace where developers can publish their HPC applications, packaged in the format HPC CloudPills, and make them available to all users of the cloud.

However, turning the system into a public cloud service, must be taken into into account that the pool of users may drastically increase, leading to struggle with new issues related to the management of computational resources. For this reason future work will include the implementation of a module for the up-scaling to public cloud. In this way it will be possible to respond effectively to any burst of requests that could cause saturation of computing resources.

The experiment described in Sec. VII demonstrated that the SPRAY application has significantly better performance when running on non-virtualized core, as shown by the HPL benchmark tool. The virtualization therefore introduces an overhead of approximately one order of magnitude, that is relevant in HPC field. However the virtualization itself is the cloud's primary benefits for HPC, because of its flexibility that let customize the environment, in contrast to supercomputers' strictly preserved system software. Moreover the degradation of performance are strictly related to the type of application and are greater for tightly coupled applications, like the one used for the experiment. In general, cloud could not be seen as an alternative to supercomputing but it can be considered an addition to it when internal resources are saturated and it is convenient for not tightly coupled applications, like embarrassingly parallel applications.

Acknowledgment

This research project has been funded by Programma Operativo Regionale "Competitivit regionale e occupazione" F.E.S.R. 2007/2013 of Regione Piemonte.

References

[1] Abhishek Gupta, Dejan Milojicic *Evaluation of HPC Applications on Cloud*, HP Laboratories, HPL-2011-132 [Online] Available at: http://charm.cs.illinois.edu/newPapers/12-18/paper.pdf, 2011.

[2] Katherine Yelick, Susan Coghlan, Brent Draney, Richard Shane Canon, *The Magellan Report on Cloud Computing for Science*, U.S. Department of Energy Office of Science Office of Advanced Scientific Computing Research (ASCR)[Online] Available at: http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Magellan_Final_Report.pdf, 2011.

[3] Carlyle, A.G., Harrell, S.L., Smith, P.M., *Cost-Effective HPC: The Community or the Cloud?*, Cloud Computing Technology and Science (CloudCom), IEEE Second International Conference on, 2010.

[4] Gupta, A., *Towards Efficient Mapping, Scheduling, and Execution of HPC Applications on Platforms in Cloud*, Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW), IEEE 27th International, 2013.

[5] Xiaorong Li, Palit, H., Yong Siang Foo, Hung, T., *Building an HPC-as-a-Service Toolkit for User-Interactive HPC Services in the Cloud*, Advanced Information Networking and Applications (WAINA), IEEE Workshops of International Conference on, 2011.

[6] *The NIST Definition of Cloud Computing, National Institute of Standards and Technology Special Publication 800-145*, [Online] Available at: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2011.

[7] *Penguin Computing on Demand*, [Online] Available at: http://www.penguincomputing.com/services/hpc-cloud/pod, 2014.

[8] *SGI*, [Online] Available at: http://www.sgi.com/products/?/cyclone/, 2014.

[9] *Sabalcore Computing*, [Online] Available at: http://www.sabalcore.com/, 2014.

[10] *Amazon Web Services High Performance Computing*, [Online] Available at: http://aws.amazon.com/hpc-applications/, 2014.

[11] *High Performance Computing Cluster in a Cloud Environment* [Online] Available at: http://www.rackspace.com/knowledge_center/article/high-performance-computing-cluster-in-a-cloud-environment, 2014.

[12] *UniCloud*, [Online] Available at: http://www.univa.com/products/unicloud.php, 2014.

[13] *OpenNebula.org*, [Online] Available at: http://opennebula.org/, 2014.

[14] *OpenLava*, [Online] Available at: http://www.openlava.org/, 2014.

[15] *KVM*, [Online] Available at: http://www.linux-kvm.org, 2014.

[16] G. Tinarelli, L. Mortarini, S. Trini Castelli, G. Carlino, J. Moussafir, C. Olry, P. Armand and D. Anfossi, *Review and Validation of MicroSpray, a Lagrangian Particle Model of Turbulent Dispersion*, in Lagrangian Modeling of the Atmosphere, AGU Geophysical Monograph, edited by Lin, J.C., D. Brunner, C. Gerbig, A. Stohl, A. Luhar, and P. Webley, 2012.