2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing

Private IaaS Clouds: A Comparative Analysis of OpenNebula, CloudStack and OpenStack

Adriano Vogel*, Dalvan Griebler*†, Carlos A. F. Maron†, Claudio Schepke‡ and Luiz Gustavo Fernandes†

* Laboratory of Advanced Researches for Cloud Computing (LARCC),
Três de Maio Faculty (SETREM), 2405, Santa Rosa Av.– Três de Maio – RS – Brazil
†PUCRS, Department of Informatics, Computer Science Graduate Program, 6681, Ipiranga Av. – Porto Alegre – RS – Brazil
‡Federal University of Pampa (UNIPAMPA), Laboratory of Advanced Studies (LEA) – Alegrete – RS – Brazil

Email: adrianovogel03@gmail.com, {dalvan.griebler,carlos.maron}@acad.pucrs.br,
claudioschepke@unipampa.edu.br, luiz.fernandes@pucrs.br

***Abstract*—Despite the evolution of cloud computing in recent years, the performance and comprehensive understanding of the available private cloud tools are still under research. This paper contributes to an analysis of the Infrastructure as a Service (IaaS) domain by mapping new insights and discussing the challenges for improving cloud services. The goal is to make a comparative analysis of OpenNebula, OpenStack and CloudStack tools, evaluating their differences on support for flexibility and resiliency. Also, we aim at evaluating these three cloud tools when they are deployed using a mutual hypervisor (KVM) for discovering new empirical insights. Our research results demonstrated that OpenStack is the most resilient and CloudStack is the most flexible for deploying an IaaS private cloud. Moreover, the performance experiments indicated some contrasts among the private IaaS cloud instances when running intensive workloads and scientific applications.**

I. INTRODUCTION

Scientific researches are becoming even more necessary for cloud computing to find out new questions and current problem answers. The cloud popularity is a consequence of the well-consolidated architecture of service (IaaS, PaaS and SaaS) and deployment (private, public, hybrid and community) models [1]. This scenario also has changed the way as information technology deals with business and research, where resources are easily available by a shared pool and on-demand usage [2]. Moreover, although it has become a suitable model for enterprise and scientific applications, such environment integrates several technologies and distributed systems paradigms that makes it complex to manage and evaluate [3], [4].

Cloud computing has challenges for various reasons. We highlight software abstraction layer, deployment model, and different technology integration as the main aspects to focus our studies. The first aspect is the virtualization that abstracts the hardware layer [5]. The second is the workload impact and cloud performance variation, which may affect the overall system results (latency, throughput and energy consumption) [3]. The last are the Service Agreement Level (SLA) issues between the provider and client. This relationship refers for a correct combination and tools compatibility between the target application and cloud environment in a cloud deployment [6].

To increase performance, contingency, security and availability, many companies and research centers are also looking for private cloud deployment solutions. In the IaaS model, several open source tools provide elementary management capabilities over a data center infrastructure [7]. This interest motivates our study towards a comparative analysis of performance, flexibility and support for resiliency.

Choose an appropriate IaaS management tool for deploying a private cloud is important to achieve successful results. This decision is made according to the tool's features and imposed application constraints. The key points are support for flexibility and resiliency because it allows to estimate the tool robustness level. In this paper, we measured these aspects with our case study tools by using and extending the [7]'s taxonomy, which classifies IaaS management tools. Therefore, we can point out differences and challenges that match the application's needs and help the decision maker.

Another critical point for providing efficient services on cloud computing is the performance. In the last years, only few research papers investigated this topic, concerning IaaS and private deployment models. The literature (Section II) provides a small number of evaluation reports which puts such topic as a research challenge. Our experiments are the starting point towards a comparative evaluation and analysis of the tools.

The main goal of this paper is to discuss and compare IaaS management tools for addressing the challenges of private cloud deployments as well as to address flexibility and resiliency differences. Consequently, other tools can be analyzed in the future using our methodology. The secondary goal is to provide performance insights through intensive workloads and some scientific applications when running in the deployed environments. Therefore, the main contributions are:

- Support for Flexibility and resiliency evaluation of three IaaS management tools. This contribution extends the evaluations of [7], including the resiliency and CloudStack tool not considered in their taxonomy.
- A comparison of a private IaaS cloud deployment. Unlike previous work, this paper evaluates the tools considering three key aspects of cloud computing: flexibility, resiliency and performance.
- Performance evaluation of three private IaaS cloud environments. Considering the related work, we con-

978-1-4673-8776-7/16 $31.00 © 2016 IEEE
DOI 10.1109/PDP.2016.75

tribute with OpenStack performance view and scientific applications insights for these cloud pools.

Our paper is organized as follows. Section II discusses the closest related work. Section III performs a survey and measures IaaS tool's flexibility and resiliency. Section IV presents the performance experiments and analyzes the results. Finally, Section V provides the conclusion of the paper.

II. RELATED WORK

The related work is classified in tool features evaluation and performance analysis.

A. *Surveys for Tool Features*

In this section, we investigate papers concerning the evaluation of IaaS cloud features.

Regarding cloud infrastructure provision and web hosting, [8]'s paper proposed a taxonomy to identify and classify cloud computing aspects (service type, resource deployment, hardware, runtime tuning, security, business model, middleware, and performance). They considered seven IaaS, Web hosting and PaaS providers.The paper also introduced a performance measurement (CPU) on cloud instances, which was continued in [9], using a larger number of applications and test scenarios. The results emphasized the weak performance (with variations) and SLA support on cloud providers.

[7]'s study proposed seven conceptual layers for a management tool of cloud infrastructure. In their classification, the IaaS management layer is composed by Command Line Interface Tools, APIs, Dashboard and Orchestrator. Such services perform essential tasks (instantiate, delete, and manage VMs, monitor resources, and power management APIs) on cloud environments for users and administrators. The authors surveyed private and public IaaS providers to demonstrate their taxonomy. As a result, it distinguishes support and capabilities for adapting the most suitable technology.

A survey of the state-of-the-art efforts was conducted by [10]. It investigated interoperability as well as practical cloud technologies of both the cloud provider's and user's perspectives to enable inter-operation. They concluded that innovations on a virtualized network should exploit software-defined networking capabilities to better facilitate inter-cloud cooperation. Advanced SLA mechanisms may be needed to constantly incorporate user feedback and customized features into the SLA evaluation framework. Finally, the question remains of how to implement cost analysis and pricing of various resources in interoperable clouds.

Considering these related works, [8]'s paper provides a survey of IaaS taxonomy for public clouds while our work concentrates on private clouds. [7]'s work creates a taxonomy for flexibility evaluation that we use as a baseline in our paper. On the other hand, [10]'s study provides a taxonomy for IaaS interoperability. **Therefore, we complement flexibility and contribute with resiliency and CloudStack tool analysis**.

B. *Experiments for Performance Evaluation*

For each performance related works, we analyzed aspects such as objectives, execution environment, adopted workloads, numeric and qualitative conclusions and limitations.

[9] and [3] analyze usefulness and performance of scientific applications for current cloud computing services. They analyzed Amazon EC2 platform using micro-benchmarks and synthetic kernels. The results indicated current public cloud services need performance improvements for scientific applications. Also, [3] concluded performance is a challenge for cloud environments due to resource sharing.

The goal in [11] was to evaluate HPC applications in the main public cloud providers, considering performance, cost efficiency and development. The article suggests cloud computing as an alternative for eliminating the maintenance of standard clusters. They ran NAS benchmarks for evaluating development and cost-effectiveness. The results show that clouds are a viable to run HPC applications with some development disadvantages such as the support for libraries. Considering performance and the cost-effectiveness, the authors estimated cloud environments 27% and 41% more efficient than clusters.

[5]'s paper studied six IaaS public providers using standard benchmark applications. The objective was to analyze performance and scalability variations when a multi-tier application is migrated from a traditional data-center environment to a cloud infrastructure. Also, they evaluated the performance on three mainstream hypervisors: XEN, KVM and CVM. The results show significant performance variations among the hypervisors. More precisely, Xen outperforms the commercial hypervisor (CVM) by 75% on the read-write RUBBoS workload and the commercial hypervisor outperforms Xen by over 10% on the Cloudstone workload.

All these papers validated the benefits to assess virtualization technologies or public IaaS clouds. [11] point out HPC challenges and usage on public cloud computing. By the other hand, [9], [5] and [3] contributed for public IaaS cloud provider performance evaluation when running scientific applications. **Our work differs from these previous studies on a performance analysis for three private IaaS cloud deployments**. Moreover, this paper contributes for the literature with new performance challenges and insights when running intensive workloads and scientific applications.

III. ANALYZING FLEXIBILITY AND RESILIENCY

The IaaS cloud comprises different layers, as illustrated in Figure 1. At the bottom are the physical resources (hardware). Above there is a Virtual Machine Manager (VMM) that supports the abstraction needed for the upper layers. The VMM is a hypervisor running within the native operating system (full virtualization) or a hypervisor running directly in the hardware (bare metal). Also, the Virtual Infrastructure Manager (VIM) is part of the IaaS cloud tools that takes advantage of virtual resources for offering services. The main tasks of resources scheduler, images, networks, volumes, templates and VMs

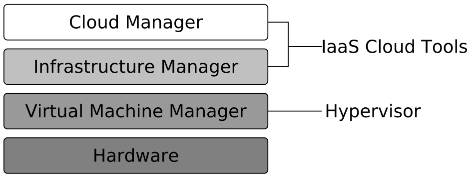

Fig. 1. IaaS overview.

creation are performed by the VIM, which are related to virtual instances and appliances [12]. Finally, the cloud manager can be a part of an IaaS tool or a separate cloud layer. This layer controls users, groups, permissions, quotas and quality of service issues of the cloud environment.

Almost all cloud management tools support administrative tasks through a graphical User Interface (UI), which is named dashboard by the tool's nomenclature. This interface is available through a web browser, providing virtual infrastructure control. There is also a Command Line Interface (CLI) which is used by cloud administrators to faster and easier terminal control. The last user interactions are the APIs for controlling the tool's infrastructure and integrating them among different services. The next sections will present and discuss OpenNebula, CloudStack and OpenStack.

A. IaaS Tools in a Nutshell

OpenNebula software is designed to work with driver concept for implementing the IaaS layer [13]. Those familiar with Linux systems will feel comfortable to deploy and control OpenNebula cloud. Mostly because it does not create a proper nomenclature, integrating default Linux-based services like scripts and components. When building a cloud environment, a machine must be configured as front-end, which will host the cloud manage, and all the other machines will be slave nodes. Storage, virtual machine, data transfer, network management, and job scheduler are configured by OpenNebula services management, using a compatible technology [14].

CloudStack is conceptually built on stacks of services and internal agents. The first service is the cloud controller. The second is the storage server for hosting volumes. [15] The third service (secondary storage) is for handling ISO images, templates and snapshots. It is implemented using a distributed file system service. The last one is a network service that implements several advanced features like virtual LAN, Virtual Private Network and Generic Routing Encapsulation [16].

OpenStack builds on API stacks. These APIs are used to communicate its services using the RabbitMQ. Such messenger runs as an interface among APIs and offers infrastructure services for the cloud system. There are forty individual components available, where the essential are: orchestrator, ceilometer (metering), swift (object storage support), Neutron (networking), Keystone and hybrid support [17]. The storage may be deployed in different ways and stored locally or distributed. OpenStack is considered a modular and granular solution because enables to implement individually all infrastructure services[18]. The next sections will present and discuss more details about the tools.

B. Flexibility Evaluation

The majority of cloud tools are robust and complex technologies. In recent years, cloud applications and usage have been investigated by the scientific community [3], [11], [10]. The architecture and flexibility of IaaS cloud tools for a customizable deployment are still under research [7], [8], [10]. Therefore, we present a comparative analysis among three open source solutions to deploy an IaaS cloud. We based our flexibility evaluation on [7]'s taxonomy which is divided into seven layers: core service, support, value-added services, control, management, security, and resource abstraction. Also, we extend the analysis to cover resiliency.

Open source IaaS cloud tools present relevant and advanced options both for the cloud administrator and users. Table I shows the IaaS tools equivalency for flexibility. We surveyed all tabulated data by deploying these tools as well as looking for their official documentation. As presented in the Table I, the

TABLE I. TOOL'S FLEXIBILITY.

Support	**OpenNebula**	**CloudStack**	**OpenStack**
Resource abstraction layer			
Compute	Oned	Libcloud	Nova
Storage	Internal	Internal	Object storage (Swift)/Block Storage(Cinder)
Volume	Internal	Internal	Nova-Volume
Network	Virtual Network Manager	Internal	Neutron/Nova-network
Core service layer			
Identity service	IAM plugin	IAM API	Keystone
Scheduling	Scheduler	Internal	Nova-scheduler
Image repository	Internal	Internal	Glance
Charging and billing	Internal	CloudStack Usage	Ceilometer
Logging	Internal	Internal	Internal
Support layer			
Message bus	Internal/ RabbitMQ	internal/ RabbitMQ	RabbitMQ
Database	sqlite/MySQL	MySQL	MySQL/Galera/ MariaDB/ MongoDB
Transfer service	Internal	Internal	Nova Object store/cinder
Management layer			
Resource management	Internal	Internal	Nova
Federation management	/	/	/
Elasticity management	Auto-scaling	Elastic Load Balancing	Elastic Recheck
User/group management	Internal	Internal	Internal
SLA definition	/	/	/
Monitoring	probe/ssh/OneGate	External	External
Reporting	code reporting	/	/
Incident management	/	Internal	External
Power management	External	External	Blueprint driver
Lease management	External	External	External
Management tools			
CLI tools	OpenNebula CLI	cloudmonkey	OpenStack (CLI)
APIs	Public cloud and Plugins	Public cloud and Plugins	Public cloud and Plugins
Dashboard	Sunstone(Admin UI, User UI)	Admin UI	Horizon(Admin UI)
Orchestrator	Oneflow	Cloudstack Cookbook	Heat
Security layer			
Authentication	Basic Auth/Open Nebula Auth/x 509 Auth/LDAP	SAML/LDAP	LDAP/ Tokens(APIs)/ X.509/HTTPD
Authorization	Auth driver	SAML	Keystone
Security groups	Internal	Internal	Internal
Single sign-on	/	External	/
Security monitoring	External	External	External
Control layer			
SLA enforcement	/	/	/
SLA monitoring	/	/	/
Metering	External	Usage plugin	Ceilometer
Policy control	/	/	/
Notification service	/	Internal	/
Orchestration	Internal	Internal	Internal
Value-added services			
Availability zones	Internal	Internal	Internal
High availability	External	External	External
Hybrid support	Microsoft Azure Ama-zonEC2/IBM	Amazon EC2	HP Helion/Ama zon EC2/IBM
Live migration	Internal	Internal	Internal
Portability support	/	/	/
Image contextualization	One-context	/	/
Application support	/	/	/

IaaS solutions have different features and goals. Some features are supported inside the core solution ("internal"), others easy achieve the demand when integrating with an external and fully compatible component/framework ("external"), while a few are not supported ("/").

The resource abstraction layer is the closest to the virtualization. OpenNebula uses the Oned to provide computer resources (CPU, memory). OpenStack takes advantage of the Nova API to process compute requests, which have advanced tasks/resources like scheduling and networking. CloudStack uses the libcloud for attending the tool's services. On the storage and volume, while OpenStack uses specific components, OpenNebula and CloudStack embedded it internally. In contrast, just CloudStack handled itself the network feature. Considering that all tools are proving at least some resource abstraction, they have similar flexibility on this layer.

In the core service layer, the identity service is used on complex cloud deployments to guarantee authorization. For example, OpenStack keystone API manages all its components and users while other tools uses an external plugin or manages it internally. The scheduler prepares all the management jobs according to the resources availability and load balance, where each one have its own scheduler. The image repository provides a catalog of pre-built OS ISOs to facilitate the VMs launching. All tools enable charging and billing services. Such feature is useful for public cloud providers to control the tenants resources usage. The logging are stored internally in plain text files or database. Thereby, specific tools are necessary to audit them.

The support layer comprises additional services on a cloud system. The message bus is used to communicate among the independent services and APIs on a cloud environment. RabbitMQ is unanimous used for transferring messages among independent components. The database is used to control cloud operations and store user credentials and system logs. All tools support at least one database, where OpenStack has more compatibility options. Transfer services are an alternative to communicate or send files among management services. Each solution performs this job using its most convenient protocol.

Management layer is responsible for controlling or centralizing user operations. The monitoring feature is fundamental for checking the hosts and instance status. All three tools implement the resources functions, elasticity, groups management, and cloud virtual environment controlling. However, they presented a low flexibility concerning federation, SLA, incident and lease management. Consequently, we identified open challenges to be considered in the tools' design and their poor capabilities on this layer.

In the management tools layer are the interfaces to control a cloud infrastructure either for administrators and end users. As previously described, CLIs and APIs play on all the tools. However, each tool presents these interfaces more or less fragmented. Also, the web interface helps to manage the cloud environment by the dashboard front-end and orchestrator feature, which are useful for large cloud deployments.

Many aspects are related to the security layer. In our table, the authentication is the process for transferring credentials to control the system access. Authorization is related to the user privileges and authentication. A security group corresponds to a set of users or projects sharing affinities and privileges. The single sign-on is related to the identities sharing and authentication among distinct/independent systems. Finally, the security monitoring is a complex system that manages and works based on predefined actions to maintain elevated security levels (authorization, high availability, among others). Despite the general poor security monitoring and single sign-on (only CloudStack supports), the tools enable authentication, authorization and security groups for increase the privacy and isolation. These insights are basic security aspects which demonstrates open challenges for OpenStack and OpenNebula to integrate security monitoring and deal with single sign-on.

The control layer presented a weak coverage of the tools. The poor SLA support indicates that they are most suitable for private cloud instead of public cloud. The metering service is for monitoring users' cloud utilization. CloudStack has the advantage of offering a notification service, facilitating the cloud administration. The policy control feature is a set of capabilities related to security and quotas. Through our studies was possible to find out their no support in this category. The orchestration system is for enabling easy integration of services, being internally implemented.

The last layer is the value-added, which covers additional services for a cloud system. For instance, the high availability feature is complex to achieve in the cloud. One way towards high availability is implement redundancy on every level such as disks (*e.g.*, RAID arrangement), services and database (replication) and install more than one cloud management server. The support of hybrid clouds enable to integrate a private cloud with services running on public clouds. These features and live migration are present in the surveyed tools. However, they are still lacking on virtual applications and portability, which is relevant for advanced cloud deployments. For additional service, only OpenNebula supports image contextualization to deploy VMs.

The support for flexibility of a computer system is the capability to support multiple components and applications settings [2]. It is needed in order to meet the different clients and application demands. In our approach, we used flexibility to analyze the enabling technologies and components supported by the cloud tools. Figure 2 summarizes the support of the tools' flexibility levels. Such measurement is computed using the tabulated data from Table I. As the taxonomy layers are

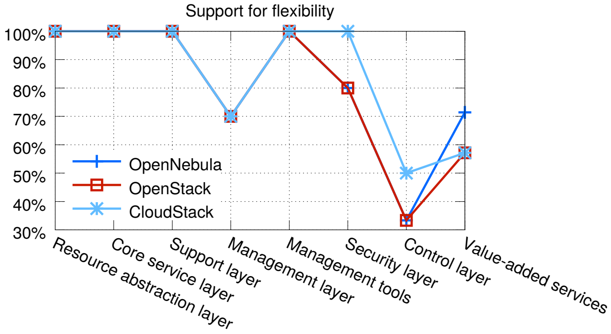

Fig. 2. Tools' flexibility levels.

shaped by items, we calculated the percentage rate in separate for each one of the layers, where an item must have at least one possible setting to be counted. Consequently, a tool achieves 100% of support for flexibility in one of the surveyed layers when it has compatibilities with all the items.

When analyzing the flexibility results, it becomes evident the tools' poor support for the control, services and management layers. The way resources abstraction technologies behave impact on cloud infrastructure services. Consequently, the resources are effecting on core service, support, control and management layers will indeed affect cloud operations and services for clients. For example, cloud security implementation will result in how reliable and confidential is the service provision. Also, the value-added services will offer more options and customization for a cloud.

Due to the relevance of flexibility for deploying a cloud environment, the tools are expected to continue increasing and improving its features. Through our comparative analysis, we can highlight that there are many studies to do for improving management and control layer in these tools (previously discussed). Such lack will impact directly to provide services for end users. Moreover, there are research challenges in security and value-added services to offer robust cloud deployments.

C. Support for Resiliency Evaluation

The computer system support for resiliency is the ability for adapting the system for constant changes or failures maintaining the availability [19]. It is also the compatibility of multiple configurations and customizations such for load balance, high performance and fault tolerance [4]. We understand that support for resiliency when referring to IaaS tools aggregate the aspects mentioned by both authors ([19] and [4]). Consequently, this is not only the ability of the system to recover in case of failures, but also the tool mechanisms and technologies that make it possible. Thus, a variety of infrastructure technologies is expected to increase the resiliency of a private cloud environment such a way that it can look for the most efficient alternative. In Table II, we surveyed these information on the tools official documentation and the deployed environments.

Table II fulfills IaaS tools support for enabling resiliency. The Full and Bare Metal virtualization refers to the virtualization for running multiple machines on the same hardware. These virtualization architectures share the hypervisor usage that emulates hardware resources to the virtual machines.

The cloud storage supports many technologies. Table II refers to used solutions on each cloud tool. For example, it is fundamental to fit the applications performance needs because storage technologies and local/distributed deployment have performance contrasts. Also, the compatibility vary among hypervisors and workloads for disk formats. They may impact on performance [20] due to the different supported hypervisor and I/O scheduler algorithms. Therefore, the most appropriate disk format vary when heterogeneous workloads are considered.

Depending on the hypervisor and virtualization technologies, the network compatibility may impact on the application's performance. The OS support refers to the number of host OS coverage on full virtualization deployments. A combination of OS and hypervisor also implies on different performance results, resources consumption, and system resiliency. For example, the containers' virtualization are built inside a native OS to simplify the virtualization layer and reduce the overhead. Finally, the Object storage is a robust and new way for offering data storage, dealing with disks as objects [21].

Figure 3 presents the tools' support for resiliency. The results are computed based on Table II data, measuring the amount of supported technologies by the cloud tools for each item. Using such quantitative results of the survey, each item' percentage is calculated considering only the number of available technologies supported for the three cloud tools. The total number of technologies (among the tools) supported by each item is considered 100% and each tool supports a specific number of technologies that reflects in a percentage value.

We identified that OpenStack presents the best percentage of resiliency, followed by CloudStack. These two solutions received the best averages because of the large support for storage, host OS, and hypervisors. All tools have Linux Containers (LXC) compatibility, enabling the deployment of OS level virtualization. As object storage is recommended for advanced storage systems to treat huge amounts of data (*e.g.*, data mining, big data). Consequently, we can highlight challenges to increase it for better resiliency. Our insights are demonstrating challenges for the network, OS, and disk format.

TABLE II. IAAS TOOLS SUPPORT FOR RESILIENCY.

Support	**OpenNebula**	**OpenStack**	**CloudStack**
Full and Bare Metal virtualization	Xen, KVM, Vmware	Hyper-V, VMware, Xen, KVM, VirtualBox	Hyper-V, Xen, KVM, VMware, VirtualBox
Storage technology	NFS, SSH(transfer), Ceph	LVM, Ceph, Gluster, NFS, ZFS, Sheepdog	NFS, SMB, SolidFire, NetApp, Ceph, LVM
Disk Formats	qcow2, vmfs, ceph, lvm, fslvm, raw, dev	LVM, qcow2, raw, vhd, vmdk, vdi	LVM, VMDK, VHD, qcow2
Network	dummy, ebtables, VLAN, OVS, vmware	Neutron, and B.Switch, Brocade, OVS, NSX, PLUMgrid	bridge, VLAN, DHCP, DNS, NVP, BigSwitch, OVS
OS	Ubuntu, Debian, RedHat, SUSE, CentOS	Debian, Ubuntu, RHEL, CentOS, Fedora, Suse	Debian, Ubuntu, RHEL, CentOS
Container Virtualization	LXC	LXC	LXC
Object Storage	Ceph	Cinder, Ceph	Cinder, Ceph

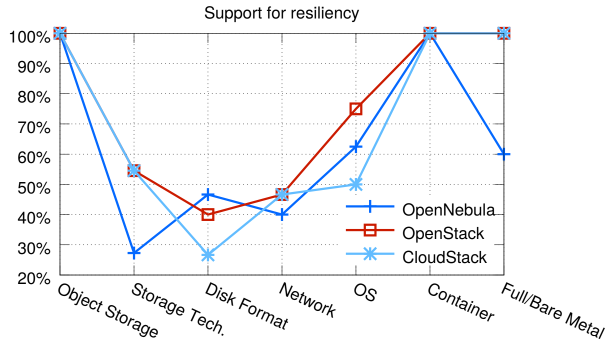

Fig. 3. Tools' support for resiliency levels.

D. Remarks

Our analysis clarified the robustness of cloud tools, detailing with their support as well as an estimation of their support for flexibility and resiliency. While flexibility may affect end users, the resiliency may impact the cloud system performance. As there are several solutions available, sometimes with poor documentation about the capabilities, it also becomes difficult for decision-making and requires a deep learning.

To discuss performance system aspects, the next section (IV) presents a performance overview on private IaaS cloud deployments. These tools can reduce the overall performance when the recommended deployment is not followed, or tools that do not support application needs. Therefore, robustness is important to enable a scalable and customizable environment for clients running their application in a cloud virtual pool.

IV. PERFORMANCE ON IAAS PRIVATE CLOUDS

This section evaluates cloud system's performance using intensive micro-benchmark and scientific applications. As presented on Section III-C, the IaaS cloud tools support several infrastructure technologies. Our goal is to discover performance insights when cloud tools are similarly deployed. Consequently, our test scenario is described on methodology section (IV-B). Therefore, we concentrate in the performance differences and variations on our three private clouds. Such results will give an overview of the performance on cloud virtualized systems compared to the native environment.

A. Cloud Deployment

We deployed the cloud pools following tools' official documentation. Our environment consist of two elements: cloud controller (also called as front-end or cloud management) and node. The controller configures an IaaS tool and distributed storage. All nodes are slaves of the controller for allocating instances, dealing with the virtualization layer (KVM, Libvirt, QEMU). The nodes communicate among each other and with the controller by a network switch.

Concerning the network, we installed a flat implementation using Linux bridges. Such configuration is justified by the small number of hosts on each cloud pool, which does not requires advanced features (*eg.,*OpenStack Neutron). Additionally, the distributed storage was chosen due to the reliability of the network and full compatibility with the tools. The hosts communicate and transfer volumes using the SSH protocol or via Remote Procedure Call (RPC).

B. Methodology

Our methodology is based on those used in related work's performance analysis. We chosen well known micro-benchmarks and applications as the baseline. The average performance is computed over 40 samples for each one of the tests. We intentionally deploy identical clouds, using a mutual hypervisor as done by [22]. The cloud tools were the only differences. Such environments are built in order to compare the fairest possible.

1) Intensive workloads: are run to evaluate individual intensive computations. The following benchmarks will stress CPU, memory, storage and networking:

- **LINPACK**: performs CPU-intensive computations for solving linear equations [23]. In the experiments, we set up a matrix with dimensions of 8000x8000 running on a single CPU core.
- **STREAM**: measures the memory bandwidth for four synthetic vector kernels (Add, Copy, Scale and Triad) by using larger data set than the system cache size. By default it uses three arrays, each one with 610.4 MiB, requiring a total of 1.8 GiB. [24].
- **IOzone**: stresses I/O performance by Write, Read, Re-Read and Rewrite disk operations for measuring the throughput with a 5GB files [25].
- **IPerf**: evaluates network bandwidth [26]. We used a default TCP window size of 85.0 KByte, transferring in an interval of 10 seconds.

2) Scientific Applications: are evaluated in the three deployed clouds by the following NAS [27] kernels:

- **Embarrassingly Parallel (EP)**: is an application that does not have dependency between tasks.
- **Multi Grid (MG)**: stresses memory and communicates [27] by mesh operation sequences.
- **Integer Sort (IS)**: performs random memory access by sorting operations.
- **Fourier Transform (FT)**: communicates one to all for fast Fourier transformation.

3) Environment of Tests: For deploying our three cloud tools, we used 4 identical Supermicro blades. Each one has 24 GB of RAM over a speed of 1333 MHz, a CPU socket Intel Xeon X5560 (quad-core 2.80GHz), 3 physical disks (SATA II 7200 RPM) organized on a RAID5 and interconnected over a gigabit (10/1000) network. The virtual environment was configured using QEMU/KVM 2.0.0 hypervisor to offer virtual resources. Our IaaS cloud deployed were the CloudStack 4.5.2, OpenNebula 4.12 and Openstack Kilo. As native OS environment for both deploying cloud tools and running experiments, we used Ubuntu Server 14.04 with 3.19.0 kernel. Each physical server hosted one large virtual instance, which had available the full machine resources and used the same OS version.

C. Intensive Workloads Performance

Figure 4 presents the micro-benchmarks' performance. We ran experiments in the four scenarios: Native, OpenNebula, OpenStack, and CloudStack to compare their differences.

The results demonstrates small overheads between the native and virtualized environments. LINPACK results demonstrated a similar performance between the cloud instances and close to Native. By other hand, IOzone results had more variation. Such performance is related to the workload's behavior. When comparing among the tools, the native outperform them using IOzone. On Read and ReRead operations, the cloud instances had similar performance to Native. Otherwise, it is

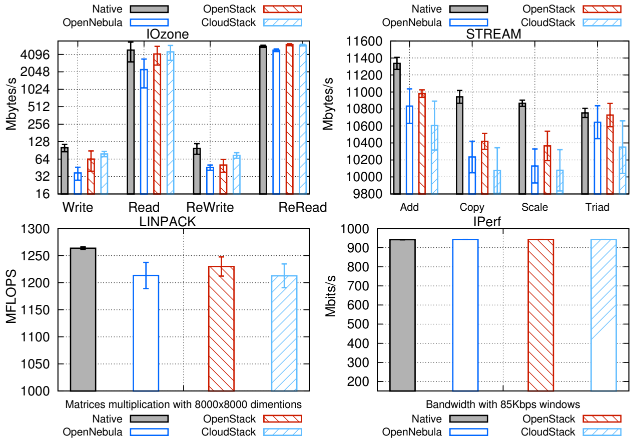

Fig. 4. Intensive workloads' performance.

possible to realize that OpenNebula had poor disks throughput, mainly on Write and ReWrite operations.

The memory experiments presented the OpenStack instances slightly better than the OpenNebula and Cloudstack because of the less variation. For the network results, the instances and native environment had a very similar throughput.

Several researches analyzed the performance variation on public cloud systems such [28] and [5]. In our experiments, we also found high variation on some workloads either on native and cloud environments. In general, OpenStack instances presented less variation in respect to other tools. For example, the overall disks performance presented high standard deviation (of a total of 40 tests). Even the native environment, which is supposed to have less variation. We observed an unfavorable deviation on disks workloads of 14.5% on write, 20.9% on ReWrite and 37.4% on disk read operations. Also, it trends to become worst on virtualized environments (*eg.,* cloud instances). On the other hand, the workloads deviation on CPU, memory and network resources presented less variation and are more stable. Our experiments contribute with new insights on these micro-benchmarks. The next sections presents scientific applications performance analysis (Section IV-D).

D. Scientific Applications Performance

Figure 5 demonstrates the NAS-OMP kernels performance results on: OpenNebula, OpenStack, CloudStack and Native. For this kernel suite, we only ran the experiments on a single machine and instance (OpenMP), setting up all the available resources. We can highlight the similar performance results of the deployed cloud environment when comparing to the native. Also, among the tools there were no significant contrasts and performance variations.

In the NAS-MPI kernels, the results were different. Figure 6 shows the NAS-MPI kernels results. The Message-Passing Interface (MPI) had performance variation and contrasts. The native environment presented the best results on scientific applications, followed by the cloud instances. While Openstack cloud shown the highest variations, CloudStack and OpenNebula clouds presented better performances. However, the differences among the deployed environment are not big. Our experiments demonstrated that when running scientific applications on private clouds, the performance does not impact for shared memory. However, sometimes the applications running on distributed fashion will have performance degradation.

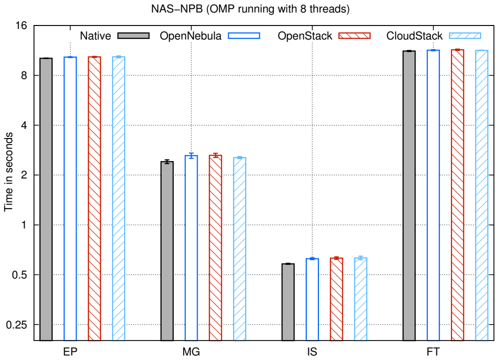

Fig. 5. NAS-OMP performance.

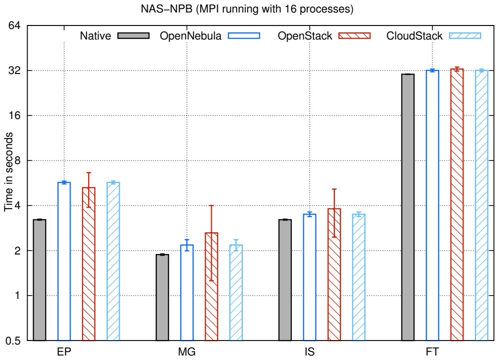

Fig. 6. NAS-MPI performance.

E. Remarks

The previous subsections (IV-C, IV-D) presented a performance overview of OpenStack, OpenNebula and CloudStack tools. With such analysis, we find out that a implementation of distributed file system does not increases the performance degradation in the cloud. Such event is by the combination of a fair network throughput (Figure 4) and the para-virtualized VirtIO driver. In STREAM benchmark, the results for memory bandwidth showed that performance increases along with the native environment. Similar event occurred with LINPACK due to the KVM hypervisor improvements. In general, the cloud tools presented similar results.

In our work, we found some performance contrasts among the tools. The results may vary on different installations approaches or experiments. For instance, CloudStack requires at least three system VMs (secondary storage, console proxy and virtual router) to work. These VMs are used for robustness and reliability. Consequently, the VMs system reduce the resources available for run instances. In our environment, these VMs were not hosted on the compute nodes to allow full resource accesses. Another example is the high number of OpenStack components, which may overload the controller node when running instances. In this approach, we maintained the controller dedicated for the cloud management and monitoring, and the openstack compute nodes just had installed the Nova component. In contrast, OpenNebula has a centralized deployment and does not has such limitations because its core is fine-grained.

V. CONCLUSIONS

This paper researched and analyzed IaaS cloud solutions for deploying private clouds. Such study contribute for new insights concerning flexibility and resiliency. We demonstrated their distinct capabilities through a enhanced methodology that can be used for future studies as well as update the information of the surveyed tools. Also, we point out challenges for improving the tools' robustness levels such as SLA, management, advanced security options, and virtualization.

We also introduced performance experiments for OpenNebula, OpenStack, CloudStack and native environments. The goal was to evaluate performance by using intensive workloads and scientific applications. Our work contributed for empirical performance insights on three different IaaS private clouds and well known test scenarios. Through analysis, we evidenced high performance variations on IOzone workload and poor performance on specific resources on OpenNebula instances. The workloads running on OpenStack instances were the most stable when comparing with the other cloud instances (OpenNebula and CloudStack). In addition, in almost all the cases at least one cloud deployment presented results close to native. We noticed a small virtualization overhead.

Moreover, all deployed tools achieved similar performance among them and native environment concerning the scientific applications. We concluded in our analysis that private clouds are a good alternative for intensive workloads and scientific applications. These outcomes where not found on public cloud providers when looking for the performance analysis on related works (Section II). As future work, we intend to: (I) evaluate more applications and benchmarks for VM scheduling, deployment, and image transfer; (II) customize the deployed clouds for testing different network and storage options; (III) continue deploying and analyzing private IaaS cloud tools using our methodology.

ACKNOWLEDGMENT

The authors would like to thank for the partial support in this research from HiPerfCloud project[1] and its supporters, computer network department of SETREM[2], and CNPq - (MCTI/CNPQ/Universal 14/2014 - Process 457684/2014-3).

REFERENCES

[1] ISO JTC1/SC38 Technical Report, "ISO/IEC 17789:2014. Information Technology - Cloud Computing - Reference Architecture." 2014, last access in Sept, 2015. [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=60545

[2] R. Buyya, C. Vecchiola, and T. Selvi, *Mastering Cloud Computing*. San Francisco, CA, USA: McGraw Hill, 2013.

[3] A. Iosup, S. Ostermann, M. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance Analysis of Cloud Computing Services for Many-Tasks Scientific Computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 931–945, June 2011.

[4] ISO JTC1/SC38 Technical Report, "ISO/IEC 17788:2014 Information Technology - Cloud Computing - Overview and Vocabulary," 2014, last access in Sept, 2015. [Online]. Available: http://www.iso.org/iso/catalogue_detail?csnumber=60544

[5] D. Jayasinghe, S. Malkowski, J. Li, Q. Wang, Z. Wang, and C. Pu, "Variations in Performance and Scalability: An Experimental Study in IaaS Clouds Using Multi-Tier Workloads," *Services Computing, IEEE Transactions on*, vol. 7, no. 2, pp. 293–306, April 2014.

[6] B. Furht and A. Escalante, *Handbook of Cloud Computing*, ser. Computer Science. Springer, USA, 2010.

[7] R. Dukaric and M. B. Juric, "Towards a Unified Taxonomy and Architecture of Cloud Frameworks," *Future Gener. Comput. Syst.*, vol. 29, no. 5, pp. 1196–1210, Jul. 2013.

[8] R. Prodan and S. Ostermann, "A Survey and Taxonomy of Infrastructure as a Service and Web Hosting Cloud Providers," in *Grid Computing, 10th IEEE/ACM International Conference on*. IEEE, Oct 2009, pp. 17–25.

[9] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing," vol. 34, pp. 115–131, 2010.

[10] Z. Zhang, C. Wu, and D. W. Cheung, "A Survey on Cloud Interoperability: Taxonomies, Standards, and Practice," *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 4, pp. 13–22, Apr. 2013.

[11] E. Roloff, M. Diener, A. Carissimi, and P. O. A. Navaux, "High Performance Computing in the Cloud: Deployment, Performance and Cost Efficiency." *CloudCom*, vol. 2012, pp. 371–378, 2012.

[12] F. Xu, F. Liu, H. Jin, and A. V. Vasilakos, "Managing Performance Overhead of Virtual Machines in Cloud Computing: A Survey, State of the Art, and Future Directions," *Proceedings of the IEEE*, vol. 102, no. 1, pp. 11–31, 2014.

[13] OpenNebula, "OpenNebula," 2015, last access Aug, 2015. [Online]. Available: http://opennebula.org/

[14] G. Toraldo, *OpenNebula 3 Cloud Computing*.

[15] N. Sabharwal, *Apache CloudStack Cloud Computing*.

[16] Cloudstack, "Cloudstack," 2015, last access Aug, 2015. [Online]. Available: https://cloudstack.apache.org/

[17] O. Khedher, *Mastering OpenStack*.

[18] OpenStack, "OpenStack Roadmap," 2015, last access Aug, 2015. [Online]. Available: http://openStack.org/software/roadmap/

[19] V. Castano and I. Schagaev, *Resilient Computer System Design*. Springer International Publishing, 2015.

[20] A.-C. Bujor and R. Dobre, "KVM IO Profiling," in *Networking in Education and Research, 2013 RoEduNet International Conference 12th Edition*, Constanta, Romania, Sept 2013, pp. 1–6.

[21] L. Coyne, S. Gopalakrishnan, J. Sing, and I. Redbooks, *IBM Private, Public, and Hybrid Cloud Storage Solutions*. IBM Redbooks, 2014.

[22] A. Paradowski, L. Liu, and B. Yuan, "Benchmarking the Performance of OpenStack and CloudStack," in *17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*. Virginia St Reno, NV, USA: IEEE, June 2014, pp. 405–412.

[23] J. J. Dongarra, P. Luszczek, and A. Petitet, "The LINPACK Benchmark: Past, Present, and Future," *Concurrency and Computation: Practice and Experience*, vol. 15, p. 2003, 2003.

[24] J. Mccalpin, "The Stream Benchmark," 1996, last access in Sept, 2015. [Online]. Available: http://www.cs.virginia.edu/~mccalpin/papers/bandwidth/node1.html

[25] Iozone, "Iozone File System Benchmark (Official Page)," 2006, last access in Sept, 2015. [Online]. Available: http://iozone.org

[26] Iperf, "What is Iperf?" 2015, last access in Sept, 2015. [Online]. Available: http://iperf.fr/

[27] D. H. Bailey *et al.*, "The NAS parallel benchmarks," *International Journal of High Performance Computing Applications*, vol. 5, no. 3, pp. 63–73, 1991.

[28] M. Armbrust *et al.*, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

---

[1] http://hiperfcloud.setrem.com.br/

[2] http://setrem.com.br/