UnaCloud: Opportunistic Cloud Computing Infrastructure as a Service

Eduardo Rosales, Harold Castro, Mario Villamizar
Department of Systems and Computing Engineering
Universidad de los Andes
Bogotá D.C., Colombia
{ee.rosales24, hcastro, mj.villamizar24}@uniandes.edu.co

***Abstract*—This paper presents UnaCloud: an opportunistic cloud computing Infrastructure as a Service (IaaS) model implementation, which provides at lower cost than dedicated cloud infrastructures, basic computing resources (processing, storage and networking) to run arbitrary software, including operating systems and applications. The IaaS model is provided through the opportunistic use of idle computing resources available in a university campus. UnaCloud deals with the problems associated to use commodity, non-dedicated, distributed, and heterogeneous computing resources that are part of different administrative domains. We propose an IaaS architecture based on two strategies: an opportunistic strategy that allows the use of idle computing resources in a non-intrusive manner, and a virtualization strategy to allow the on-demand deployment of customized execution environments. The proposed solution was implemented and tested through the provision of an opportunistic IaaS model, evidencing high efficiency in the deployment of virtual machines for academic and scientific projects.**

Keywords; grid computing; cloud computing; desktop grid; infrastructure as a service; unacloud; unagrid.

I. INTRODUCTION

Grid computing and cloud computing appear to be the two latest and most promising computing paradigms [1]. Grid computing is considered a paradigm in production, which surged as a vanguard technology for supporting the development of different scientific projects at a global scale [2]. In contrast, cloud computing is still an evolving paradigm. Its definitions, use cases, underlying technologies, issues, risks, and benefits will be refined in a spirited debate by the public and private sectors. These definitions, attributes, and characteristics will evolve and change over time [3]. However, cloud computing is considered the grid computing successor [4], because it represents a disruptive evolution, aimed at the customization and delivery of computing services. These services hide most of the complexities associated with the underlying infrastructure administration, can be deployed on demand and are accessed remotely via Internet [1].

There are high expectations about cloud computing paradigm for the next 1-5 years [5]. Cloud computing is attracting a lot of attention around the world [6], not only of experts in ICTs, but also academics, scientists, researchers, businessmen and common people, who are attracted by the delivery of on-demand computing services. However, there are a few cloud computing implementations, most of them exclusively based in the IaaS model, due in part to the complexity associated to the different cloud computing service delivery models (IaaS, Platform as a Service – PaaS, and Software as a Service - SaaS). Furthermore, all IaaS implementations (open source or commercial) require expensive, dedicated, robust and high performance underlying infrastructures, so they are unviable in organizations and countries with low financial resources.

Taking into account the emerging importance of cloud computing paradigm, the need of independent investigation testing of commercial providers, the financial difficulties associated with expensive underlying infrastructures, and the different cloud computing service models, in this paper we present UnaCloud, an IaaS model implementation, which provides basic computing resources through the opportunistic use of idle computing resources available in a university campus.

UnaCloud is able to deploy, manage and deliver an opportunistic IaaS model based on preexisting, non-dedicated, distributed, and heterogeneous computing resources that are part of different administrative domains. These resources are in part, conventional desktop computers, as those daily used by employees, professors or students in a university campus. These desktop computers tend to be underutilized for significant periods, resulting in plenty of idle computing resources. Due to the large amount of available computing resources on a university campus, UnaCloud represents an economically attractive solution for constructing and deploying large scale computing infrastructures, avoiding not only underutilization of non-dedicated computing resources, but also financial investments in hardware and maintenance costs associated.

UnaCloud has been initially deployed at Universidad de los Andes and, in this work, the design and details of the implementation deployed are presented along with the results obtained. The paper is organized as follows: section 2 presents the related works to IaaS model implementations and Desktop Grids and Volunteer Computing Systems (DGVCS's). Section 3 presents the UnaCloud architecture in terms of its services. Section 4 presents UnaCloud implementation. Section 5 presents the UnaCloud testing and results. Finally, Section 6 presents the conclusions and future work.

 ISBN: 978-1-61208-153-3

II. RELATED WORK

UnaCloud represents a convergence between cloud computing and DGVCS's. The service delivery model of the cloud computing paradigm is taken into account as an objective, mainly in relation to the IaaS model. The design aspects of the DGVCSs are kept into account as a mean to provide an opportunistic underlying infrastructure to support cloud computing services at lower cost. This type of convergence has been theoretically analyzed in [7], on the perspective of software engineering principles.

In the context of IaaS models, Amazon Web Services (AWS) [8] is considered a precursor because it is in productive use since 2006, offering basic processing and storage capabilities via Internet. Amazon Elastic Compute Cloud (Amazon EC2) [9] and Amazon Simple Storage Service (Amazon S3) [10] popularized a commercial IaaS model, based on a pay-per-use contract and the provision of resizable compute capacity in the cloud. The OpenNebula [11] project was the first open source tool that extended the benefits of cloud computing technologies to data centers and clusters, transforming physical infrastructures in virtual infrastructures of high flexibility and performance. Eucalyptus [12] is the first research-oriented open source software implementation that utilizes compute clusters in order to foster community research exploration of cloud computing systems. Nimbus [13] is an open source toolkit that allows transforming clusters into an IaaS model able to interoperate with grid computing conventional tools, including: Globus Toolkit, Sun Grid Engine (SGE) or PBS.

On the other hand, in the context of DGVCS's, the Worm [14] and Condor [15] projects are pioneers in the opportunistic use of homogeneous computing resources connected by LAN infrastructures. The GIMPS [16] and SETI@home [17] projects are characterized by their unique purpose, Internet scalability and the ability to leverage non-dedicated, distributed and heterogeneous computing resources (at the hardware, system and administrative domain level).

The Distributed.net [18] and BOINC [19] projects are characterized by an approach not limited to a unique purpose, being able to support multiple distributed scientific research projects. The last four projects described, are based on lightweight, portable and easy to install agents/clients that are continuously running as a background process in low priority, leveraging idle computing resources in a non-intrusive manner. Finally, projects like Bayanihan Computing. NET [20], OurGrid [21], Integrade [22] and UnaGrid [23], offer specialized support to cluster and grid computing initiatives with large processing demands, assuming the deployment of middleware and workload management systems to process multiple jobs. In the Nebulas project [24], different requirements and possible solutions to build customizable clouds (called Nebulas) using distributed voluntary resources are proposed; however, they are neither implemented, nor evaluated.

Unlike the commercial and academic IaaS model implementations, UnaCloud does not require large financial investments to purchase and maintain cluster architectures composed by multiple nodes exclusively dedicated to the provision of the virtual machines resources. In contrast, UnaCloud uses a commodity and non-dedicated underlying infrastructure, implementing opportunistic design concepts broadly studied in the context of DGVCS's.

UnaGrid is the first on-demand opportunistic Desktop Grid [23]. It uses virtualization technologies to deploy Customized Virtual Clusters (CVC) based on an opportunistic underlying infrastructure. Due to the above, the UnaGrid infrastructure is able to support cloud computing service models, even though UnaGrid functionalities are currently focused on cluster and grid computing technologies.

To the best of our knowledge, our work is the first to analyze the prospect and performance of using an opportunistic underlying infrastructure to support an IaaS model.

III. UNACLOUD ARCHITECTURE

UnaCloud began as a research effort to explore and obtain the innovative features and advantages of cloud computing paradigm. This effort is aimed at the provision of computing infrastructures for the development of e-Science projects and to support computing related activities. To achieve this, one of our most important limitations is the funds to purchase the dedicated computing resources required by all IaaS model implementations (even open source IaaS model implementations).

Therefore, the UnaCloud objectives require the extension of DGVCS's design concepts to provide an opportunistic underlying infrastructure able to support an experimental IaaS model at lower cost. In spite of the multiple problems related to use a non-dedicated infrastructure, functionalities included in UnaCloud are supposed to be similar to those available in conventional IaaS models. However the availability of the computing resources is dependent on the behavioral pattern of their currently owners, so it is normally not effective to ensure any type of QoS or SLA. Thus, UnaCloud works on a best-effort basis.

UnaCloud architecture is based on the integration of an information system with an underlying computing infrastructure, that is, a Web portal capable of coordinating information and communications on opportunistic infrastructures to provide basic computing services, operating systems and applications through a cloud computing IaaS model. The UnaCloud architecture overview is illustrated in Fig. 1.

As shown in Fig. 1, there are four types of UnaCloud users. An IaaS user demands the IaaS model without specifying the deployment location. IaaS users access the UnaCloud Web interface to customize and/or deploy virtual machines with general-purpose configurations (e.g., virtual machines used to support academic activities). Grid users demand IaaS model specifying the deployment location on specific underlying infrastructure computers. Grid users access the UnaCloud Web interface to customize and/or deploy suitable execution environments for e-Science applications (e.g., cluster, grid or cloud computing environments). IaaS-Grid users can take any of the above roles. Additionally, Administrators access to all available Web interfaces, with all privileges and get exclusive access to administration services.

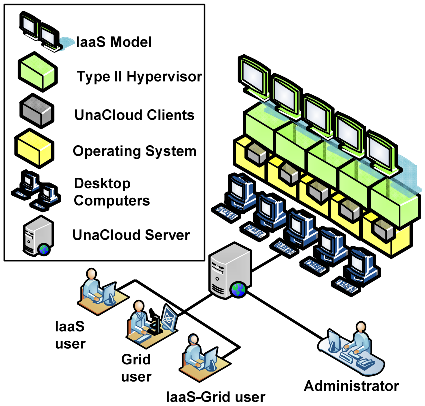

Figure 1. UnaCloud architecture overview.

UnaCloud architecture is divided into two main components: UnaCloud Server and UnaCloud Client. These components are implemented using open source and loose coupling information and communication technologies, which promote the UnaCloud interoperability and extensibility, and are appropriated to the special conditions of a commodity opportunistic underlying infrastructure.

A. *UnaCloud Server Architecture*

UnaCloud Server is a Web application whose main function is to provide an entry to all UnaCloud services, including the provision of customization, deployment, access, management and monitoring interfaces. As shown in Fig. 2, UnaCloud Server is composed of three layers:

- *Interface layer*: is a Web portal that supports a user Web Interface (WI), which supports the presentation for accessing and consuming all available UnaCloud services. This interface provides an IaaS model based on self-service. This layer is also responsible for managing the user information, including secure access through authentication and authorization mechanisms. The Web portal is available via Internet and so, can be accessed using any Web browser.
- *Core layer*: is responsible for processing all user requirements and deliver solutions in the form of UnaCloud services. The first service supported is the Customized Environment Manager (CEM), which processes and prepares orders related to all of customization settings (made through WI), including availability verifications of the computing resources. The availability verifications are performed through a virtual and physical resources information database that is managed by a service named Persistence Manager (PM). PM is also responsible for managing the operations used to provide basic IaaS traceability reports. The next core layer service is Virtual Machine Manager (VMM), which works in conjunction with PM to manage the virtual machine information. VMM is also responsible for preparing hypervisor orders to operate all the IaaS virtual machines, including: start, stop, restart and monitor operations. Finally, Physical Infrastructure Manager (PIM) service, works in conjunction with PM to manage the physical machines information. PIM is also responsible for preparing operating system orders to operate the entire underlying infrastructure, including basic operations such as: turn off, restart, logout and monitoring.

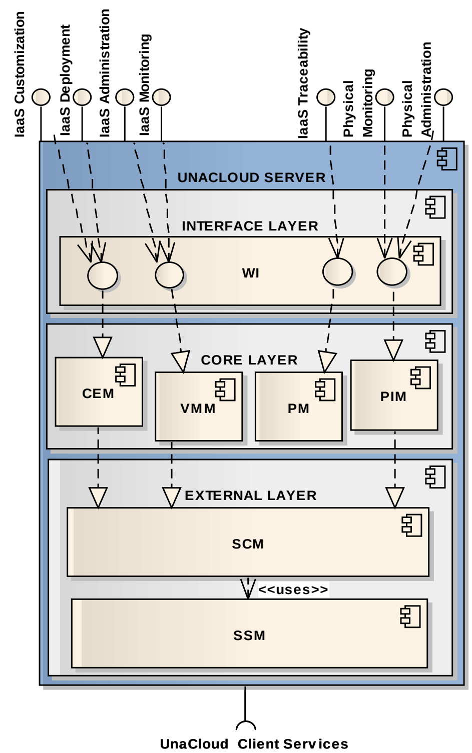

Figure 2. UnaCloud server architecture.

- *External layer*: is responsible for managing the communication services on the server side to deliver all the UnaCloud Server orders to the UnaCloud Clients. The first service supported is Server Communication Manager (SCM), which supports the connection, disconnection and message passing between UnaCloud Server and UnaCloud Client. SCM works in conjunction with Server Security Manager (SSM) service, which is responsible for managing the security schema in communications, including confidentiality and non-repudiation mechanisms.

B. *UnaCloud Client Architecture*

UnaCloud Client is a lightweight, highly portable and easy to install client which is installed and run directly on the underlying opportunistic infrastructure. This Client is based on the design concepts of agents/clients implemented on

Copyright (c) IARIA, 2011. ISBN: 978-1-61208-153-3

DGVCS's such as: GIMPS, Distributed.net and SETI@home (studied in Section 2). These design concepts proposed the execution of background and low priority processes to use idle computing resources in a non-intrusive manner. UnaCloud Client incorporates these concepts, but apply them to the virtual machine execution processes, facilitating not only the deployment of an opportunistic IaaS model, but also the continuous and optimized utilization of the underlying infrastructure in a time-sharing hardware model.

UnaCloud Client is responsible for receiving and processing all of UnaCloud Server orders to provide a dynamic and on-demand IaaS model. To achieve this, as shown in Fig. 3, UnaCloud Client is composed of two layers:

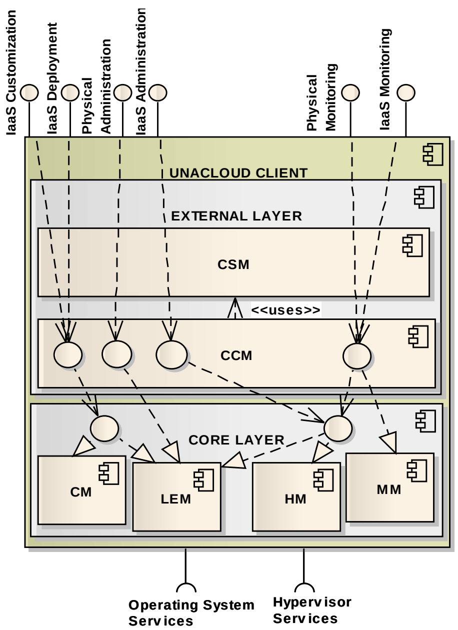

Figure 3. UnaCloud client architecture.

- *External layer*: is responsible for managing the communication services on the client side. The first service supported is Client Communication Manager (CCM), which supports the connection, disconnection and message passing between UnaCloud Client and UnaCloud Server. CCM works in conjunction with the Client Security Manager (CSM) service, which supports confidentiality and non-repudiation mechanisms for secure message passing.
- *Core layer*: is responsible for attending and meeting UnaCloud Server orders through local operating system and hypervisor invocations. The first service supported is Context Manager (CM), which is the counterpart in the client of CEM, and is responsible for adapting the virtual machine execution context to all of customization settings required by an end-user through WI. The next service is Local Executor Manager (LEM), which executes multiple commands using invocations to the local operating system services. LEM executes all commands required to meet the orders sent by VMM and PIM from the UnaCloud Server side. The next service is Hypervisor Manager (HM), which executes multiple commands using invocations to the local hypervisor. HM executes all commands required to meet the orders sent by VMM from the UnaCloud Server side. Finally, Monitoring Manager (MM) service is responsible for monitoring the state of CPU, RAM and SWAP memory, hard disk, network and operating system variables on the physical machine where UnaCloud Client is running.

UnaCloud Client can be installed on any desktop computer or server using Windows, Linux or Mac operating systems. It supposes a horizontal scaling model, based on the easy aggregation of single desktop computers or entire computer laboratories.

IV. UNACLOUD IMPLEMENTATION

To meet UnaCloud objectives, its implementation of an opportunistic IaaS model is able to provide the following services:

- *IaaS customization*: UnaCloud allows the customization of execution environments through five settings: software, hardware, quantity, location (optional) and time. Software settings allow customizing the type of operating system, its version and all applications installed on it, after the deployment new applications can be installed on the VMs. Hardware settings allow customizing hard disk and RAM memory sizes, and the CPU cores number. Quantity setting allows choosing the instances number to deploy. Location setting allows choosing the IaaS model deployment location on specific underlying infrastructure computers. The last setting only applies to Grid users who desire to optimize and document the use of the opportunistic infrastructure. Finally, time setting allows configuring the IaaS execution time. For users who want to skip the full IaaS customization process, settings only involve the selection of the IaaS deployable image name, the instances number to deploy, the location (only for Grid users) and the execution time of the deployment.
- *IaaS deployment*: UnaCloud allows the on-demand deployment of the execution environments, customized in the previous service. The IaaS deployment includes the provision of necessary data for its remote access, using standard mechanisms such as: Remote Desktop, VNC or SSH. The remote access data provided includes: the virtual machine name and IP address, the remote access mechanism name and port and, the guest operating system root user and password (UnaCloud deliver virtual machines with root privileges).
- *IaaS administration*: UnaCloud allows operating virtual machines, including basic operations such as start, stop, restart, change execution time and monitoring.

- *IaaS traceability*: UnaCloud allows checking the IaaS model traceability at user level. UnaCloud delivers a basic report that includes information associated with deployed virtual machines, the underlying infrastructure used on the deployment, IaaS customization settings chosen by the UnaCloud user and the execution period selected.
- *Physical infrastructure administration*: UnaCloud allows operating physical machines that compose the underlying infrastructure, including basic operations such as: turn off, restart, logout and monitoring. This functionality is only available for administrators.

V. UNACLOUD TESTING AND RESULTS

UnaCloud Client has been deployed in three computer laboratories (Waira I, Waira II and Alan Turing) at Universidad de los Andes. Each laboratory has 35 computers with Intel Core 2 Duo (1.86GHz) processors, 4GB of RAM and Windows XP as their main operating system. In addition, UnaCloud Server was deployed on a virtual machine running on a server, which is located in the data center (for availability reasons) of the Department of Systems and Computing Engineering. As illustrate in Fig. 4, the networking infrastructure is based on three switches and a multilayer switch interconnected via a GigE LAN.

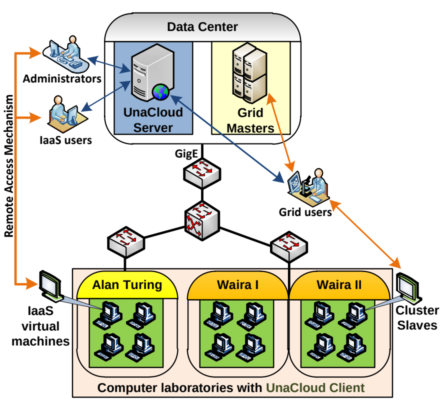

Figure 4. UnaCloud deployment.

UnaCloud Client runs only one virtual machine per desktop computer, mainly to avoid resource competition between virtual machines. Due to the fact that the opportunistic underlying infrastructure is not capable of type I hypervisors, the virtualization operations request the assistance of type II hypervisors suitable for desktop computer based on x86 architectures. Due to the above, each desktop computer has installed the VMware Workstation type II hypervisor, which assists the virtual machines operations to deploy the opportunistic IaaS model. All hypervisor services are accessed through the VMware platform with VIX libraries.

A. *Cloud evaluation*

As show in Fig. 4, in order to test the UnaCloud services for Grid users, some grid computing components were used, including a master node that has assigned two CPU cores and 2GB of RAM memory, and 35 slave nodes that have assigned two CPU cores and 1GB of RAM memory. This virtual infrastructure supports the e-Science experimentation of the Department of Biological Sciences, which is developing projects that analyze the coffee, cassava and potatoes genome, to improve production affected by biological organisms [25], [26] and [27].

Figure 5. UnaCloud IaaS located deployment.

As illustrate in Fig. 5, the virtual infrastructure deployment was assisted by UnaCloud following an IaaS located deployment. Grid users deployed the 35 grid slave nodes in about 7 seconds. The average time that each virtual machine took in parallel to load the guest operating system (Debian 4) and to enabling network services (to be accessed via SSH) was approximately 4 minutes, time in which a slave is ready to receive jobs from its cluster master.

As show in Fig. 4, in order to test the UnaCloud services for IaaS Users, some IaaS virtual machines were used, including software development and data mining, customized execution environments. This virtual infrastructure supports the academic activities of students of the Department of Systems and Computing Engineering. As illustrate in Fig. 6, the virtual infrastructure deployment was assisted by UnaCloud following a non-located IaaS deployment. IaaS users deployed 70 virtual machines in about 13 seconds. The average time that each virtual machine took in parallel to load the guest operating system (Windows XP) and to enabling network services (to be accessed via Remote Desktop) was approximately 5 minutes.

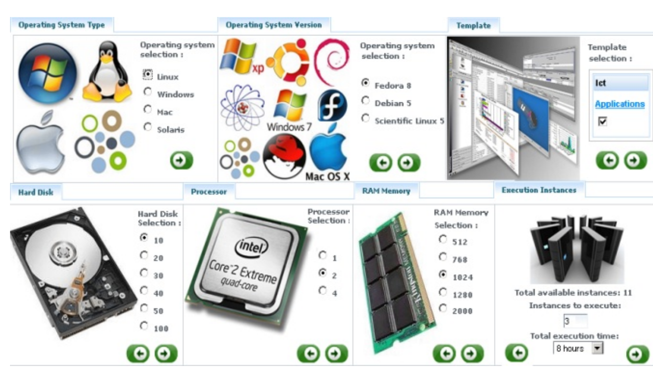

Figure 6. UnaCloud IaaS non-located deployment.

Both case studies demonstrate how UnaCloud provides an opportunistic IaaS model and validates all of its services. The validation process shows that UnaCloud incorporates relevant features in the cloud computing context. These features are summarized in Table 1. As mentioned before, concepts like SLA or QoS are not part of the UnaCloud initial scope.

TABLE I. UNACLOUD CLOUD COMPUTING FEATURES

Feature	*Description*
Usability	UnaCloud provides Web interfaces, whose operation is almost intuitive, requiring basic IT knowledge.
Self-service	UnaCloud users can unilaterally consume basic computing resources on a self-service model.
Broad network access	UnaCloud provides basic computing services that are available over the network and are consumed through standard secure remote access mechanisms.
On-demand services customization	UnaCloud provides services to customize execution environments required on demand by the end-user. This customization is able to meet large scale computational requirements.
Time-sharing hardware	UnaCloud incorporates an opportunistic strategy that allows the use of idle computing resources in a non-intrusive manner. This strategy allows the simultaneous opportunistic use of the underlying infrastructure by multiple users.
Virtualization	UnaCloud uses a virtualization strategy to allow the on-demand deployment and assign of customized execution environments.
Scalability	UnaCloud uses an opportunistic commodity horizontal scaling infrastructure and is based on a private cloud deployment model.
Interoperability and loose coupling	UnaCloud is based in loose coupling and interoperability services that can operate over highly heterogeneous, distributed and non-dedicated infrastructures.
Extensibility	UnaCloud is based in open source tools, broadly diffused in order to facilitate its extensibility.
Delegated administration	UnaCloud hides the underlying infrastructure complexity to end-users and provides services to support common administration tasks.
Security	UnaCloud uses authentication, authorization, confidentiality and non-repudiation mechanisms to secure the IaaS model deployment.
Measured service	UnaCloud records and reports the IaaS model traceability at user level.

B. *Performance degradation perceived by the owner user*

In order to analyze the performance impact perceived by resource owners due to the simultaneous execution of the virtual machine as a background and low priority process, three tests were performed. In the first tests we evaluated the performance when a virtual machine executes intensive processing applications. To achieve this, the execution of a CPU intensive application was performed by the resource owner, using three different environments: without executing the virtual machine in background and executing the virtual machine (making intensive use of processing) having one core and two cores assigned, respectively; the results of the tests are shown in Table 2. The results show that the execution of the processing virtual machine in background affected the performance perceived by resource owners by less than 1%.

TABLE II. CPU PERFORMANCE IMPACT

Environment/Test	**Task Completion Time (seconds)**			
	Test 1	***Test 2***	***Test 3***	***Test 4***
Without VM	53,94	81,01	108,05	134,99
With a VM (1 Core)	54,16	81,42	108,39	135,58
With a VM (2 Cores)	54,21	81,46	108,58	135,60

In the second set of tests, the performance impact on the resource owners was evaluated when they execute storage intensive applications (I/O). To achieve this, file compression operations of different sizes were executed by resource owners. These tests were executed within the same environments as the first tests and the results are shown in Table 3. The results evidence a low impact, less than 3%, in the performance perceived by resource owners. It is justified in the operating systems default mechanisms to manage the local processes priority. These mechanisms ensure computing resources to higher priority processes, while dynamically reducing the computing resources allocated to lower priority processes. The third tests confirm this fact.

TABLE III. I/O PERFORMANCE IMPACT

Environment/Test	**Task Completion Time (seconds)**			
	Test 1 200 MB	***Test 2 500 MB***	***Test 3 1 GB***	***Test 4 2 GB***
Without VM	104,10	259,85	521,16	1041,42
With a VM (1 Core)	105,66	262,43	526,63	1060,75
With a VM (2 Cores)	106,02	263,03	527,06	1063,07

A third set of tests were executed in order to monitoring the processor usage from both the resource owner processes, and the background and low priority virtual machine processes, which had two CPU cores assigned. Intensive processing tasks were executed within both environments. The results are shown in Fig. 7.

In the test, after measuring CPU usage with no virtual machine running, we initiate (time 3) a virtual machine using nearly 50% of the CPU, and at 5, we increase its computational requirements to nearly 100%. We then modify the CPU need from the resource owner. Between 7 and 8 the resource owner demands a 50% of the CPU and the virtual machine load automatically decreases to 50%. Between 9 and 10 the resource owner increases the consumption to about 100% and the virtual machine automatically reduces their consumption to a minimum. Between 11 and 12 the resource owner goes back to a 50% demand, and after 12, the resource owner leaves the physical machine.

The results show that the virtual machine only consumes idle processor cycles, or all cycles in the case of a fully available resource (not temporarily used or a dedicated resource). This fact guarantees a very low impact on the performance perceived by the resource owners. Based on the tests results, we conclude that virtualization and opportunistic strategies incorporated in UnaCloud represent a non-intrusive solution for deploying large scale virtual

 ISBN: 978-1-61208-153-3

infrastructures, encouraging the use of idle computing resources and providing an efficient solution to preexisting resources underutilization problem.

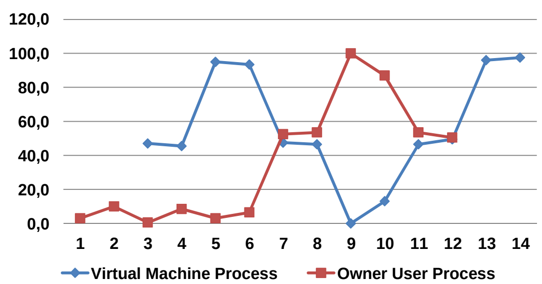

Figure 7. CPU usage for virtual machine and resource owner.

VI. CONCLUSIONS AND FUTURE WORK

UnaCloud is an opportunistic cloud computing IaaS model implementation, which provides at lower cost, basic computing resources (processing, storage and networking) to run arbitrary software, which include operating systems and applications. UnaCloud deals with the problems associated to the use of commodity, non-dedicated, distributed, and heterogeneous computing resources that are part of different administrative domains. To achieve this, we proposed an IaaS architecture based on two main strategies: a virtualization strategy to allow the on-demand deployment of customized execution environments and, an opportunistic strategy based on the validation and extension of DGVCS's design concepts to provide a commodity, non-dedicated, and heterogeneous underlying infrastructure.

Our IaaS architecture supposed a convergence between cloud computing paradigm and DGVCS's. The results not only demonstrate the convergence viability, but also offer promising opportunities to meet customized computational requirements thought the use of open source, low cost, extensible, interoperable, efficient, scalable and opportunistic IaaS model. In addition, UnaCloud represents an economically attractive solution for constructing and deploying large scale computing infrastructures, avoiding not only, underutilization of non-dedicated computational resources, but also financial investments in hardware and costs associated with physical space, temperature-controlled environment, maintenance process, etc.

UnaCloud cloud computing features are promising to reduce the development cycle and the generation of results time of any activity or project depending on the agile provision of computing resources, including academic, scientific and even commercial initiatives.

New challenges will have to be faced in order to improve the IaaS model offered: a requirement is to analyze how to guarantee statistic QoS, improving the best-effort scheme currently in use. Future work also includes UnaCloud extension to provide networking on-demand customization, creation of an API that allows that new services or applications can be incorporated to UnaCloud, compatibility with other type II hypervisors, PaaS and SaaS service models, and public, community and hybrid cloud computing deployment models. We also are preparing the UnaCloud solution as an Open Source project that will be released on 2012.

REFERENCES

[1] R. Buyya, Y. Shin, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," Proc. 10th IEEE International Conference on High Performance Computing and Communications, IEEE Press, 2008, pp. 5-13, doi:10.1109/HPCC.2008.172.

[2] Carl Kesselman and Ian Foster, The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, Nov. 1998.

[3] P. Mell and T. Grance, "NIST Definition of Cloud Computing". National Institute of Standards and Technology (NIST), 2009.

[4] M. Vouk, "Cloud computing - Issues, research and implementations," Proc. 30th International Conference on Information Technology Interfaces, IEEE Press, 2008, pp. 31-40, doi:10.1109/ITI.2008.4588381.

[5] R. Buyya, S. Pandey, and C. Vecchiola, "Cloudbus toolkit for market-oriented cloud computing," Lecture Notes in Computer Science, vol. 5931, 2009, pp. 24-44, doi:10.1007/978-3-642-10665-1_4.

[6] Google Trends Labs, "Cloud Computing", [Online], http://www.google.com/trends?q=cloud+computing&ctab=0

[7] V. Cunsolo, S. Distefano, A. Puliafito, and M. Scarpa, "Applying Software Engineering Principles for Designing Cloud@Home," Proc. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), May 2010, pp. 618-624, doi:10.1109/CCGRID.2010.76.

[8] Amazon Web Services, LLC, "Amazon Elastic Compute Cloud (Amazon EC2)", [Online], http://aws.amazon.com/, Aug. 10, 2011.

[9] Amazon Web Services, LLC, "Amazon Elastic Compute Cloud (Amazon EC2) ", [Online], http://aws.amazon.com/ec2, Aug. 10, 2011.

[10] Amazon Web Services, LLC, "Amazon Simple Storage Service (Amazon S3) ", [Online], http://aws.amazon.com/s3, Aug. 10, 2011.

[11] OpenNebula.org, [Online], http://opennebula.org, Aug. 10, 2011.

[12] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus open-source cloud-computing system," Proc. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID), IEEE Press, May 2009, pp. 124-131, doi:10.1109/CCGRID.2009.93.

[13] Nimbus, [Online], http://workspace.globus.org/, Aug. 10, 2011.

[14] J. Shoch and J. Hupp, "The "worm" programs-early experience with a distributed computation," Communications of the ACM, vol. 25, March 1982, pp. 172-180, doi: 10.1145/358453.358455.

[15] M. Litzkow, M. Livny, and M. Mutka, "Condor - A Hunter of Idle Workstations," Proc. 8th International Conference of Distributed Computing Systems, IEEE Press, June 1988, pp. 104-111, doi:10.1109/DCS.1988.12507.

[16] Mersenne Research, Inc, "GIMPS: Great Internet Mersenne Prime Search", [Online], http://www.mersenne.org, Aug. 10, 2011.

[17] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home An Experiment in Public-Resource Computing," Communications of the ACM, vol. 45, Nov. 2002, pp. 56-61, doi:10.1145/581571.581573.

[18] Distributed.Net, [Online], http://www.distributed.net, Aug. 10, 2011.

[19] D. Anderson, "BOINC: A System for Public-Resource Computing and Storage," Proc. 5th IEEE/ACM International Workshop on Grid, IEEE Press, Nov. 2004, pp. 4-10, doi:10.1109/GRID.2004.14.

[20] L. Sarmenta, S. Chua, P. Echevarria, J. Mendoza, R. Santos, S. Tan, and R. Lozada, "Bayanihan Computing .NET: Grid Computing with XML Web Services," Proc. 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, IEEE Press, 2002. pp. 434-435, doi:10.1109/CCGRID.2002.1017182.

[21] B. Francisco and M. Rodrigo, “The OurGrid Approach for Opportunistic Grid Computing,” Proc. First EELA-2 Conference, Feb. 2009, pp. 11-19.

[22] A. Goldchlegery, F. Kon, A. Goldman, M. Finger, and G. Bezerra, "InteGrade: object-oriented Grid middleware leveraging the idle computing power of desktop machines," Concurrency and Computation: Practice and Experience, vol. 16, 2004, pp. 449-459, doi:10.1002/cpe.824.

[23] H. Castro, E. Rosales, M. Villamizar, and A. Miller, "UnaGrid - On Demand Opportunistic Desktop Grid," Proc. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, IEEE Press, June 2010, pp. 661-666, doi:10.1109/CCGRID.2010.79.

[24] A. Chandra and J. Weissman, "Nebulas: Using Distributed Voluntary Resources to Build Clouds," Proc. 9th Workshop on Hot Topics in Cloud Computing (HotCloud09), USENIX, June 2009.

[25] A. González, H. Castro, M. Villamizar, N. Cuervo, G. Lozano, S. Orduz, and S. Restrepo, "Mesoscale Modeling of the Bacillus thuringiensis Sporulation Network Based on Stochastic Kinetics and Its Application for in Silico Scale-down", Proc. HIBI '09. International Workshop on High Performance Computational Systems Biology, IEEE Press, Oct. 2009, pp. 3-12, doi=10.1109/HiBi.2009.17.

[26] S. Restrepo et al., “Computational Biology in Colombia,” PLOS Computational Biology, vol. 5, Oct. 2009, doi:10.1371/journal.pcbi.1000535.

[27] A. Vargas et al., “Characterization of Phytophthora infestans Populations in Colombia: First Report of the A2 Mating Type,” Phytopathology, Sep. 2009, pp. 82-88, doi:10.1094/PHYTO-99-1-0082.