Paper

The Analysis of OpenStack Cloud Computing Platform: Features and Performance

Daniel Grzonka

Institute of Computer Science, Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland

Abstract—Over the decades the rapid development of broadly defined computer technologies, both software and hardware is observed. Unfortunately, software solutions are regularly behind in comparison to the hardware. On the other hand, the modern systems are characterized by a high demand for computing resources and the need for customization for the end users. As a result, the traditional way of system construction is too expensive, inflexible and it doesn't have high resources utilization. Present article focuses on the problem of effective use of available physical and virtual resources based on the OpenStack cloud computing platform. A number of conducted experiments allowed to evaluate computing resources utility and to analyze performance depending on the allocated resources. Additionally, the paper includes structural and functional analysis of the OpenStack cloud platform.

Keywords—cloud computing, high performance computing, OpenStack, parallel environments, resource utilization analysis, virtualization.

1. Introduction

The rapid development of computer hardware caused a new issue for software developers. With the increase of computing power, the problem with efficient use of physical resources occurred. In many cases, an increase in the available resources, especially in the number of computational units, can have opposite effects and cause significant decrease in performance. Therefore, despite the fact that the development of computer hardware and programming languages has enabled the creation of a much more complex systems, the development of software techniques doesn't keep up with the technological development. This phenomenon is called software crisis [1], [2]. Over the years there has been proposed a number of solutions, i.e., new paradigms, programming languages, hardware architectures or software solutions, which are more or less, eliminate that problem. One of the latest is known as cloud computing based on abstraction of resources and ability to user's needs adoption, but the challenge is still actual [3], [4]. Therefore, in the present work, attempts are made to analyze both resource consumption and performance basing on OpenStack cloud computing platform.

The research from this paper is an extended version of preliminary results published in [1] and presented at the 29th European Conference on Modeling and Simulation (ECMS 2015) in Albena, Bulgaria.

The paper is organized as follows. In Section 2 the idea of resources abstraction is presented. Section 3 describes objectives, assumptions, and architecture of cloud computing. The structural analysis of OpenStack cloud computing platform is presented in Section 4. Finally, the Section 5 describes conducted experiments and evaluates them. The paper is summarized and concluded in Section 6.

2. Idea of Resources Abstraction

In the 1970s, when mainframe computers gain great popularity, the significant waste of computing resources caused by the single application execution at a time was observed. Due to the their high cost, to remedy this problem, the new idea of virtualization was proposed. Initially, the virtualization was defined as parallel work environments on a single computer and was considered as a method for logically dividing mainframes. This division allowed running multiple applications simultaneously [5].

In the second half of the 1990s, when personal computers obtain a sufficiently high computing power, the use of virtualization becomes a popular solution. The continuous development of idea allowed to formulate general definition of virtualization as technique for simulating the software and the hardware upon which other software runs [6]. In accordance to this definition, the term includes widely understood abstraction of various software and hardware areas like operating systems, storage, memory, networks, servers, CPUs and other. The key benefits are reduction of energy, computations, heat generation and maintenance costs, better resources utilization, space-saving for computers, the ability to adapt to the end user needs, safety increase and high flexibility in system design [1], [6]–[8].

Virtualization is based on the coexistence of host and guest machines. The host machine is the actual machine on which the virtualization takes place, and the guest machines are the virtual machines operating through the host. The tool supervising the virtual machines is hypervisor (or virtual machine monitor). When the hypervisor replaces host system we are dealing with native (or bare-metal) hypervisor. If the hypervisor operates under the operating system, we are dealing with hosted hypervisor. The most popular hypervisors are KVM, VirtualBox, VMware ESX, Xen, Oracle VM, Microsoft Hyper-V [1], [7], [9].

The virtualization method, due to the way of implementation, can be divided on two approaches, namely paravirtualization and full virtualization [10]. Full virtualization might be based on both hosted and native hypervisor that provide virtual hardware for virtual guest operating system. Guest operating system has the impression that it operates on a real, physical hardware. In fact, the virtualized OS instructions, which would interfere with the activities of other virtualized environments or host operating system, are captured and executed by virtualization layer. Performance in this solution is, unfortunately, less than in paravirtualization, but in return it does not impose any restrictions or special requirements for guest machine [6], [10].

Paravirtualization bases on native hypervisor, that operates directly on hardware and provides it to virtual machines and guest operating systems. This solution requires modifying the guest kernel to translate nonvirtualizable instructions to hypercalls that communicate directly with the native hypervisor. The mechanism provides very good performance that is very close to the system native operation. Very good performance for the virtual machines is huge paravirtualization advantage. The disadvantage is the need to modify the systems kernel or the use of dedicated systems [6], [10].

3. Cloud Computing Technology

Linking the virtualization with distributed systems and with the concept of services resulted in the new idea called cloud computing. Cloud computing is defined as "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" [11]–[13].

Cloud computing paradigm is based on flexible idea of services providing, where the virtual machines are the base for delivering any cloud services. A service can be defined by a number of virtualized functionalities provided by servers including hardware, software, storage, computing capacities and infrastructure with server rooms. Services are usually priced on a pay-per-use business model or subscription fee, and the access is realized via Internet with thin-client, e.g., mobile device, tablet or computer with Web browser. Additionally, this approach allows for effective resource utilization, flexible developing of complex systems, customization of services, elastic system management, on-demand resource provision and, as a result, reduction of energy consumption and maintain costs [9], [12]. The services are referred as:

- Infrastructure as a Service (IaaS),
- Platform as a Service (PaaS),
- Software as a Service (SaaS).

IaaS is a model that relies on providing client IT infrastructure, i.e., hardware like machines, network, storage, software and support. In this model the client have the remote access to infrastructure. This level is responsible for managing and virtualization of the resources.

PaaS provides ready, often tailored to the needs of the user, application set. This model does not require to purchasing hardware or software installation – all the necessary programs are located on servers provider. The customer on his side has access to the interface via Internet and a thin client.

In SaaS customer receives a needed functionality and he doesn't have to worry about the infrastructure and the work environment. This model provides access to specific, functional tools. Programs are running on the provided server. The customer is not obliged to purchase software licenses – only is paying for every use, and access to them is obtained upon request [4], [13], [14].

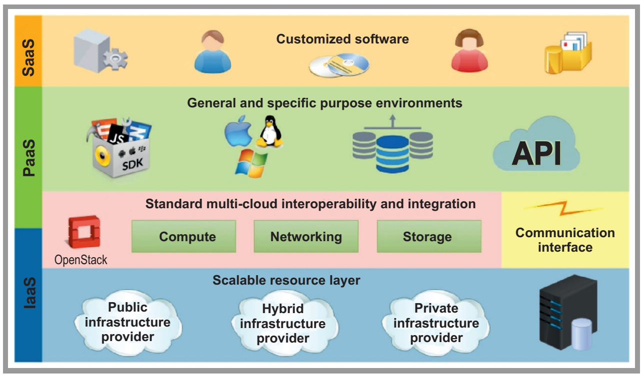

Fig. 1. Cloud computing fundamental models.

Figure 1 presents general model of cloud computing consisting of three levels: IaaS, PaaS and SaaS. In terms of internal policies, the following part can be distinguished [4]:

- Private cloud, which is part of the organization or autonomous provider of services;
- Public cloud, which is offer by external, publicly accessible provider (e.g., Amazon, Google, Microsoft);
- Hybrid cloud, which is a combination of private and public clouds. A part of the applications and infrastructure is placed in the private cloud, and the rest in the public cloud.

Benefits of cloud computing include [4], [15]:

- universal access – the user can use the service from anywhere in the world, using devices with Internet access. On the other hand, cloud computing may cause excessive network traffic;
- saving money – user does not need to maintain the entire infrastructure and is exempt from the costs of purchase and repair servers, maintenance of rooms and charges for electricity;

- performance and scalability – at any time the user may obtain the required hardware resources;
- reliability – thanks to the distribution of resources, and the systems flexibility in case of failure, the virtual environment is recreated on the new infrastructure;
- popularity – services in cloud computing are becoming more widely available, even for small businesses;
- security – use of cloud computing reduces the risk of errors related to the human factor. Simultaneously, the virtualization technology greatly increases the operations security level in cloud computing environments.

4. The OpenStack Project

OpenStack is a free and open source platform under the terms of the Apache license that possesses a set of tools for the creation and management of private, public and hybrid cloud computing. The software is developed for a control of wide range of processing, storage and networking resources throughout a data centre. It can be treated as an Infrastructure as a Service model strongly connected with Platform as a Service model (see Fig. 1). OpenStack manage the IT infrastructure, provide communication interface, virtualizes resources and prepare environment. It provides a modular architecture that gives the flexibility in the clouds design, including integration with existing systems and third-party technologies, e.g., Amazon EC2, GoGrid, Rackspace [1], [16].

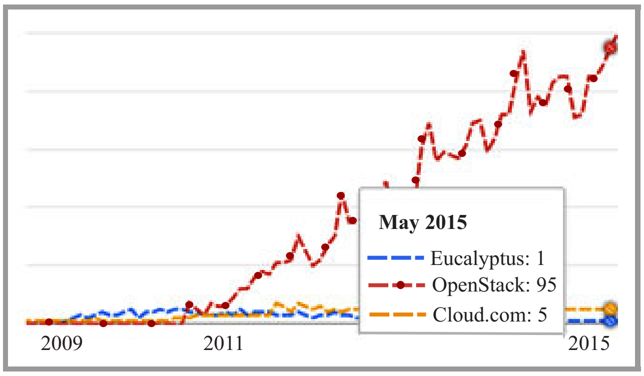

Fig. 2. Interest rate of most popular cloud platforms.

The OpenStack Cloud Computing Platform is probably the most popular open source software for creating and managing private, public, and hybrid clouds. Referring to the Google Trends Explore [17], [18], the OpenStack is the most popular platform among three the most known cloud platforms, i.e., Eucalyptus, OpenStack and Cloud.com. Figure 2 presents interest rate basing on the number of searches have been done for a particular term in relation to the total number of searches done on Google over time (a higher ratio means higher interest).

OpenStack was started in 2010 as a common project of Rackspace (a large US hosting company) and NASA (the US space agency). Currently, it is managed by the OpenStack Foundation, a non-profit corporate entity established in 2012. At the moment, the foundation has over 18,000 members from over 150 countries around the world. The purpose of the foundation is to promote the OpenStack software and its community, which develop project. The Foundation is composed of three committees: Board of Directors, Technical Committee and User Committee [1], [16].

The software is built modular and consists of many services working together through the open APIs. The newest version of the OpenStack platform – Kilo (2015.1.0) released on 30 April 2015 – may use the following modules (some of which are optional): Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove, Sahara, Ironic, Zaqar, Manila, Designate, Barbican [1], [16]. The most important and fundamental are the first three components – they are implementing major features [19]:

- Compute (Nova service) – module for arranging, managing and providing virtual machines. It is "designed to provide power massively scalable, on demand, self service access to compute resources" [16];
- Object Store (Swift service) – module for creation and managing object storage system;
- Image Service (Glance service) – module which provide a service for uploading and discovering data assets. It retrieves and process data about virtual machine images.

The installation can be carried out in several ways, e.g. [1], [16], [19], [20]:

- directly from GitHub repository,
- Ubuntu Juju (based on installation Metal as a service layer),
- Vagrant OpenStack Provider (popular tool for creation and configuration of virtual environments),
- DevStack (tool for the installation of the platform from source with the set of initial configuration).

Details of the installation process are described in a previous publication [1].

The platform is open source and can be modified and adapted as needed. Services can be managed from Horizon dashboard (single Web-based on user interface) or by custom developed software. The access and communication is possible after authentication based on security certificates, which are the responsibility of Keystone Identity Service [1], [16].

5. Experimental Results

In this section the results of conducted experiments are presented. The study include:

- in-depth analysis of resources (CPU and memory) utility (both at full load and without) and consumption,
- the performance measurement of tasks execution depending on selected OpenStack set of virtualized resources,
- evaluation of effectiveness of the applied technologies.

In this research, 5 sets of virtual resources named flavors were considered. These were, namely: tiny, small, medium, large and extra large sets presented in the Table 1. Default environment configuration allowed to create up to 10 instances and assigning among others 20 VCPUs, 50 GB of memory and 1 TB of storage.

Table 1
Available sets of the virtual resources

Flavor	VCPUs	Disk [GB]	RAM [MB]
m1.tiny	1	1	512
m1.small	1	20	2048
m1.medium	2	40	4096
m1.large	4	80	8192
m1.xlarge	8	160	16384

OpenStack provides two virtual operating systems images by default [16]:

- CirrOS (9.3 MB) – a minimal Linux distribution that was designed for use as a test image on clouds,
- Trusty Tahr (244.4 MB) – distribution of Ubuntu 14.04.

According to OpenStack minimum requirements for proof-of-concept environment with several instances of CirrOS the following resources are needed [16]:

- Controller node: 1 processor, 2 GB memory, and 5 GB storage,
- Network node: 1 processor, 512 MB memory, and 5 GB storage,
- Compute node: 1 processor, 2 GB memory, and 10 GB storage.

The initial part of the research was based on CirrOS Linux distribution. The experimental server (specification is presented in Table 2) allowed to create 2 virtual compute nodes with maximum 3 instance per node [1].

Firstly, the impact of the existence of running instances in environment was measured. Analyzed parameters were CPU and RAM usage. For this purpose the 3 non-load instances had been deploying simultaneously every time.

Table 2
Specification of Pinokio experimental server

Processor	Intel Core i7-4710HQ CPU @ 2.50 GHz
Cores	4
Threads	8
Architecture	x86 64 bit
Virtualization	Intel VT-x
Memory	16 GB SODIMM DDR3 Synchronous 1600 MHz (0.6 ns)
Disk	256 GB ADATA SP600
Operating system	Ubuntu Server 14.10 (x64)

The results are presented in Table 3. With increasing number of running instances, the growing CPU usage was observed. Each started instance irreclaimable get and use about 2% of computing power (about 5–7% per 3 running instances). The operating memory usage was constant [1].

Table 3
Resources consumption caused by running instances

Instances	CPU usage [%]	RAM usage [%]
0	5	97.1
3	12	97.1
6	18	97.1

Next, resources usage was analyzed in the full load scenario. In this study, the matrix multiplication was performed as the CPU intensive task. It is one of the most popular benchmark tasks [21]. The size of used matrices was 1024×1024. The results of measurements are presented in Table 4. It should be noted, that instances number 1, 3, and 5 were automatically assigned to first compute node. As it can be seen, the full use of available resources was not reached. This is due to the amount of virtual computing machines limited by server hardware specification. But the obtained results lead to the conclusion that the OpenStack platform allows to fully use available computing resources [1].

Table 4
Resources consumption caused by running instances

Instances	CPU usage [%]	RAM usage [%]
0	5	97.1
1	19	97.2
2	32	97.3
3	41	97.3
4	51	97.4
5	55	97.5
6	56	97.5

The extended part of the research was based on Ubuntu 14.04 Trusty Tahr Linux distribution. The experiments were performed on external cloud provider servers equipped with AMD Opteron 6274 @ 2.2 GHz processors. The aim of the study was to analyze computing performance depending on selected OpenStack flavor (set of available virtual resources). As in previous experiment, for this purpose the matrix multiplication with matrices of 2048×2048 size were used. Multiplication was parallelized using OpenMP API with the static scheduling clause. The results of the measurements are presented in Table 5. It should be noted that only parallelizable part of the task (multiplication) was measured. The results seem to be astonishing, because the top performance for sequential scenario (one thread) was received for medium flavor. In this case, the characteristics of the environment should be take into account, namely its distribution and virtualization. The experiment shows that the increase of available virtual resources may reduce the performance. It can be caused by communication between distributed resources and as well as by the observed switching between physical resources (threads and fragmented memory).

Table 5
Time of sequential (one thread) and parallel matrix multiplication for each flavor

Flavor	Time [s]				
m1.tiny	231.94	232.80	233.90	234.54	234.90
m1.small	54.79	55.02	55.97	56.39	57.42
m1.medium	48.91	35.74	37.01	37.83	38.92
m1.large	51.93	37.07	19.95	22.04	23.43
Number of threads:	1	2	4	8	16

Table 5 also presents scenarios where the software takes full advantage of available resources – to those achieved the best time results. This study shows how important is the selection of appropriate virtual resources to the task, and how much impact on the performance has distribution and virtualization of resources.

Table 6
Speedup of parallel matrix multiplication depending on the selected flavor

Flavor	Speedup [s]				
m1.tiny	1	0.996	0.992	0.989	0.987
m1.small	1	0.996	0.979	0.972	0.954
m1.medium	1	1.368	1.322	1.293	1.257
m1.large	1	1.401	2.603	2.356	2.216
Ideal speedup	1	2	4	8	16
Number of threads:	1	2	4	8	16

The last Table 6 presents achieved speedup in relation to set of virtual resources. The results were determined on the basis of the Amdahl's law that is a model for the expected speedup and the relationship between parallelized application [22]. The highest speedups were gained in the cases where the number of VCPUs is equal the number of threads. However, comparing the results in Table 5, the speedup often does not result in faster task execution.

6. Summary and Conclusions

Presented article is an extension of previous studies on OpenStack cloud computing platform analysis [1]. The main topic of the paper is a problem with effective use of available physical and virtual resources based on the OpenStack cloud computing platform. Conducted research has shown that OpenStack well manages and utilizes resources, but the platform takes significant part of resources. The process of creating virtual machines, consumes very large amounts of memory. Even in the idle time, the virtual servers are constantly maintained and incessantly keep resources. In this experiment, the cost of platform implementation was greater than the benefits.

The second part of the research was based on the extended infrastructure. The performance analysis of tasks execution depending on selected OpenStack flavor has shown that increase of allocated resources may reduce the performance – especially in the case of distributed environments. Selection of appropriate resources is a very important and non-trivial task. An interesting issue in the future would be a development of a trade-off model as a solution of this problem.

References

[1] D. Grzonka, M. Szczygieł, A. Bernasiewicz, A. Wilczyński, and M. Liszka, "Short analysis of implementation and resource utilization for the OpenStack cloud computing platform", in *Proc. 29th Eur. Conf. Modell. Simul. ECMS 2015*, Albena (Varna), Bulgaria, 2015, pp. 608–614.

[2] E. W. Dijkstra, "The humble programmer", *Commun. ACM*, vol. 15, no. 10, 1972.

[3] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", *Future Gener. Comp. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.

[4] M. Armbrust *et al.*, "A view of cloud computing", *Commun. ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[5] C. D. Graziano, "A performance analysis of Xen and KVM hypervisors for hosting the Xen Worlds Project", Graduate Theses and Dissertations, Iowa State University, paper 12215, 2011.

[6] K. Scarfone, M. Souppaya, and P. Hoffman, "Guide to Security for Full Virtualization Technologies", Recommendations of the National Institute of Standards and Technology, Special Publication 800-125, USA, 2011.

[7] M. Serafin, *Wirtualizacja w praktyce* (*Virtualization in practice*). Gliwice, Poland: Wydawnictwo Helion, 2012 (in Polish).

[8] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures", *Commun. ACM*, vol. 17, no. 7, pp. 412–421, 1974.

[9] M. Tajvidi, R. Ranjan, J. Kołodziej, and L. Wang, "Fuzzy cloud service selection framework", in *Proc. IEEE 3rd Int. Conf. Cloud Networking CloudNet 2014*, Luxembourg, 2014, pp. 443–448.

[10] J. Sahoo, S. Mohapatra, and R. Lath, "Virtualization: A survey on concepts, taxonomy and associated security issues", in *Proc. 2nd Int. Conf. Comp. Netw. Technol. ICCNT 2010*, Bangkok, Thailand, 2010, pp. 222–226.

[11] P. Mell and T. Grance, "The NIST Definition of Cloud Computing", Recommendations of the National Institute of Standards and Technology, Special Publication 800-145, USA, 2011.

[12] Y. Mhedheb, F. Jrad, J. Tao, J. Zhao, J. Kołodziej, and A. Streit, "Load and Thermal-Aware VM Scheduling on the Cloud", in *Algorithms and Architectures for Parallel Processing*, *LNCS*, vol. 8285, pp. 101–114. Springer, 2013.

[13] M. Marks and E. Niewiadomska-Szynkiewicz, "Hybrid CPU/GPU platform for high performance computing", in *Proc. 28th Eur. Conf. Model. Simul. ECMS 2014*, Brescia, Italy, 2014, pp. 508–514.

[14] K. Smelcerz, "Recent developments in mobile cloud scheduling: State-of-the-art, challenges and perspectives", *J. Telecommun. Inform. Technol.*, no. 4, pp. 51–57, 2013.

[15] F. Gens, "Defining Cloud Services and 'Cloud Computing", IDE, 2008 [Online]. Available: http://blogs.idc.com/ie/?p=190

[16] OpenStack Website and Documentation [Online]. Available: http://www.openstack.org/

[17] Google Trends [Online]. Available: http://www.google.com/trends/

[18] O. Sefraoui, M. Aissaoui and M. Eleuldj, "OpenStack: Toward an Open-Source Solution for Cloud Computing", *Int. J. Comp. Appl.*, vol. 55, no. 3, pp. 38–42, 2012.

[19] K. Pepple, *Deploying Openstack*, 2nd ed. O'Reilly Media, USA, 2011.

[20] K. Jackson and C. Bunch, *OpenStack Cloud Computing Cookbook*, 2nd ed. Packt Publishing, UK, 2013.

[21] J. M. Cecilia, J. M. García, and M. Ujaldón, "The GPU on the Matrix-Matrix Multiply: Performance Study and Contributions", in *Parallel Computing: From Multicores and GPU's to Petascale*, B. Chapman *et al.*, Eds. *Advances in Parallel Computing*, vol. 19, pp. 331–340, 2010.

[22] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities", in *Proc. Spring Joint Comp. Conf. AFIPS '67 (Spring)*, Atlantic City, New Jersey, USA, 1967, pp. 483–485.

Daniel Grzonka received his B.Sc. and M.Sc. degrees with distinctions in Computer Science at Cracow University of Technology, Poland, in 2012 and 2013, respectively. Currently, he is Research and Teaching Assistant at Cracow University of Technology and Ph.D. student at Jagiellonian University in cooperation with Polish Academy of Sciences. He is also a member of Polish Information Processing Society and IPC member of several international conferences. The main topics of his research are grid and cloud computing, multi-agent systems and high-performance computing.

E-mail: grzonka.daniel@gmail.com
Institute of Computer Science
Faculty of Physics, Mathematics and Computer Science
Tadeusz Kościuszko Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland