Open Source SaaS Cloud Platforms: Concepts, Virtualization Overhead and Deployment Issues

Nikolina Frid, Vlado Sruk, Hrvoje Mlinarić
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
{nikolina.frid, vlado.sruk, hrvoje.mlinaric}@fer.hr

***Abstract**. This paper explores modern open-source cloud platforms giving insight into the theoretical concepts they are based on, along with the analysis of challenges they face when deployed in business environment. In the first part of the paper, theoretical concepts like cloud computing, virtualization and software as a service paradigm, which form the basis for this new concept – the SaaS cloud platform, are discussed. Afterwards, the analysis of performance and deployment issues of the two most prominent open-source platforms Ulteo and LTSP Cluster is presented. We also discuss proposed solution to the issue of user application profile persistence between sessions.*

Keywords. SaaS Cloud, Virtualization, Open Source

1 Introduction

Recent advances in deployment and reliability of fast wide-area networks combined with high performance inexpensive server computers and virtualization technologies available for commodity hardware are key enabling technologies for successful implementation of cloud computing [3]. Important factor for its acceptance presents pressure to reduce IT costs and increase scalability of the software solutions. Today cloud computing is becoming pervasive, and serve as primary source of computing power in a board range of applications from personal and enterprise computing applications to high performance computing.

The SaaS Cloud platform emerges as a unification of cloud computing and “software as a service” paradigm. The necessity for replacing traditional software products in different application areas from education, business to government is present and a lot of effort is engaged into devising adequate SaaS Cloud platform solutions to replace the existing traditional software solutions.

While proprietary cloud solutions are striving to be as much aligned with customer needs as possible by providing almost any type of service needed, they tend to keep the customer in a vendor lock-in position. On the other hand, open-source solutions maintain a broader perspective by producing more generic solutions diverse enough to satisfy the majority of potential users while maintaining its main stronghold – the reduced Total Cost of Ownership (TCO). What is more, open-source sets high standards when it comes to portability and cross-platform compatibility with both other open-source and commercial software.

In the following chapters we discuss foundations of SaaS Cloud platforms, existing open source solutions and their shortcomings concerning performance and deployment.

2 SaaS Cloud Platforms

Cloud computing is a way of unifying various computer resources (hardware and software) in order to deliver them as services to customers. According to [1] the exact definition would be:

> *A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.*

It is also important to emphasise that the allocation of resources should be dynamically handled without any major intervention from the service provider.

There are several concepts Cloud Computing draws heavily on and a distributed computing paradigm is the most important one. Clouds are massively parallel, encapsulated as an abstract entity from an outside perspective and the services they provide can be dynamically configured and delivered on demand. There is also a strong liaison with Grid computing – Clouds can be perceived as the next step in evolution of Grid. Finally, the business model of Cloud Computing is manifestation of Utility Computing since the billing model “pay per use” is no different form the one for the basic utilities.

According to the level of services delivered, three main types of Clouds, which client can access, are distinguished. Each type represents one level, and each level is the upgrade of the previous one. The basic level delivers computer infrastructure (virtual

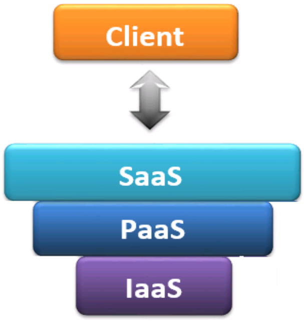

Figure 1. Cloud types

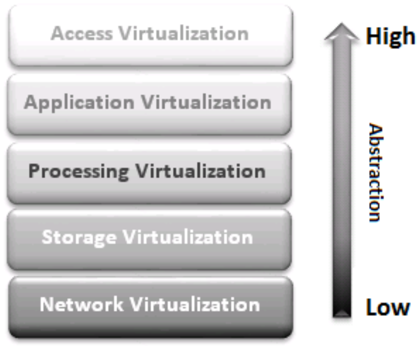

Figure 2. Levels of Virtualization

machines, data storage, network resources) on demand and it corresponds to IaaS Cloud type. Next level, the PaaS Cloud type, delivers a computing platform typically including operating system, programming language execution environment, database, and web server required mostly by application developers. On the top there is the SaaS Cloud which provides all types of business applications making the system below transparent to the end user, Figure 1.

2.1. Virtualization

In computing, virtualization refers to the ability to create virtual (logical) instead of real (physical) versions of hardware, data storage, network or operating systems and installed applications where the framework partitions the real (physical) resources.

As in Cloud Computing, the levelling approach can be applied in virtualization likewise [5]. Figure 2 shows the different levels of virtualization corresponding to resource virtualized.

Network-level Virtualization refers to making real network resources transparent to end user thus adjusting virtual network topology to current needs [7].

Hiding real characteristics of the storage system, dividing it into parts which appear to end user as a whole is the task of Storage Virtualization [8].

The ability to virtually divide one physical system into multiple parts and vice versa - unify multiple physical systems into one is a characteristic of Process Virtualization – the backbone of almost every Cloud.

The top two levels: Application and Access Virtualization are the most abstract and most important for building SaaS Cloud.

Application Virtualization enables application execution on client computer without installation. The server, which has the application installed, creates a virtual package which encompasses environment for application execution on client computer. The server delivers this package on demand, and the application runs on client computer completely oblivious of the fact that it is not installed on the client. Despite this isolation, the application, with the help from its container package, can still communicate with the client host OS and access other applications and hardware on the host.

The alternative for accessing an application that is not installed locally is through Access Virtualization (sometimes referred to as Presentation Virtualization), which abstracts application execution from its presentation. While the application runs on server, presentation data is transferred through the network to client that establishes a session with the server. This way multiple users can run the same application in parallel. This technology is based on X Window system, which uses SSH protocol for network transfer, and is used on Unix-based systems. Microsoft has developed its own Remote Desktop Protocol (RDP) with server and client solutions and is currently the only way presentation of Microsoft applications can be virtualized.

2.2. Software as a Service

The final building block for SaaS Cloud is “software as a service” paradigm which implies having applications installed on a server computer and delivering them over network to clients as services on demand. The main idea of this model is to separate owning software from it use [9]. Application is provided as a service and delivered to clients through application or presentation virtualization.

SaaS, in its core, relies on the basic principles of SOA whose definition according to W3C [10] is:

> *“A set of components that can be invoked and whose interface descriptions can be published and discovered.”*

SOA is not limited by a single technology although the SOAP protocol and web services are by far the most common representatives of this architecture. They have a public interface in XML format that other parties can discover and use for communication over internet protocols – HTTP above all [10, 11].

SaaS, as used in the clouds, automatically discovers available software services, communicates over public interfaces with the service provider and

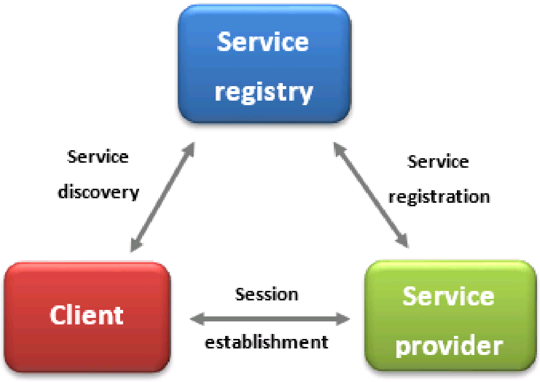

Figure 3. SaaS service discovery

sets up a session. Available software services are automatically discovered by client through a "service registry" followed by setting up a session between the client and the application server. The discovery of the services and connection negotiation can be conducted directly or through an intermediate (session manager / connection broker) which is much more often the case. Service registry is maintained by cloud provider. This process is shown on Figure 3.

In SaaS Cloud, user operating environment is built dynamically, each time by a combination of different services according to user needs.

3 Overview of existing solutions

While there are many proprietary solutions for building SaaS Cloud, the offer of open-source alternatives is rather modest. What is more, when comparing the features of one of the most prominent proprietary solutions – the XenApp by Citrix with available open-source alternatives, most solutions have many serious shortcomings [6].

Available open source solutions were compared against each other and XenApp as a commercial opponent in six categories representing most important features of SaaS cloud solution [11]: the method of application delivery (application or presentation virtualization), support for Windows and Linux application servers, protocols supporting, ability to load balance, ability to integrate with other services (file servers, printers, active directories...) and finally how well the solution was documented. Only two among them have proven to be alternatives worth considering: Ulteo OVD and LTSP – Cluster.

The two best open-source SaaS Cloud platforms provide effective solution for building a cloud. In fact, in some ways they are quite ahead of their proprietary opponents, especially when it comes to support for Linux-based applications which are completely neglected by most proprietary solutions. Other than that, they give a considerable amount of control to the cloud administrators who can, with reasonable effort input, fine tune the system to suite the given requirements. However, there are some noticeable performance and deployment issues, discussed in the next two chapters, which should be resolved in future development.

4 Performance issues

The basic SaaS cloud platform is characterised by flexibility and capability to scale on user demand. These characteristics establish significant requirements on cloud resources such as CPU, memory, cache, disk storage, network traffic as well as process speed, system-user-response time, task speed, transaction speed and latency [12].

Utilising SaaS in private (or public) cloud environment, where multiple SaaS instances and cloud-based applications are being run simultaneously by multiple users, load balancing is considered as critical performance issue affecting scalability and availability.

4.1. Load balancing

Load balancing is a process of reassigning the jobs to the individual computing nodes in the cloud. This helps maximising the throughput of the system thus improving the response time for each job. Besides this performance endorsement, load balancers serve to promote availability of cloud resources.

The goal of running applications in Cloud is to have a server farm with high redundancy of installed software – single application is installed on multiple application servers. This configuration helps to improve reliability because it is considered that there always exists a spare server to transfer the user to if

Table 1. SaaS Cloud solutions comparison

	Virtualization method	OS support	Visualization method	Loadbalancing	Integration with other services (LDAP, NFS, SAMBA...)	Well documented
Ulteo	presentation v.	Linux, Windows	X, RDP	yes	yes	yes
LTSP Cluster	presentation v.	Linux, Windows	X, RDP	yes	yes	yes
LTSP	presentation v.	Linux*	X	no	poor***	no
Cameyo	application v.	Windows	N/A**	N/A**	no	no
Citrix XenApp	pres. & app. v.	Windows	RDP	yes	yes	yes

* Can support windows through rdesktop but only as a connection to predefined computer
** Not documented
*** Most services are either not supported or not documented

one fails. In case when multiple users access the same application simultaneously they should be dispatched to different servers thus achieving balanced performance and usage-level.

On tested platforms dispatching policy is a bit altered. If user has already an established session with application server and tries to run another application that is also available on that server, the application will be scheduled on that server even if other servers, which provide the same service, are exists and could accept request. Currently, load balancing takes into account new users only, which are dispatched correctly across the server farm.

Although there is a rationale behind this way of implementation, since it minimizes the need for duplication of user data across multiple servers it can lead to overload of one server while others remain underused.

4.2. Availability

Most prominent solutions like Ulteo and LTSP Cluster are based on presentation virtualization instead of application virtualization. This leads to high dependency on network connection between client and service provider because the application runs on the server side and only I/O traffic is streamed between the client and the application server. The availability and performance of this type of system strongly correlates with the level of network service in-between. Two major problems are present, the first one the availability of network connection and the second one the throughput of the network when working with large data streams.

In situation when application server, which has established sessions with one or more clients, crashes, all sessions are instantly terminated. However, users can immediately log back into the system and will be redirected to other available servers. The availability of the system as a whole is preserved, although without transparent session transfer from the failing server to any other available server. This should definitely be taken into consideration in future development.

Implementation of transparent session transfer could also resolve the issue of previously discussed load balancing.

4.3. Virtualization overhead

As mentioned earlier, SaaS principle can be realised by either of the two levels of virtualization – application and presentation virtualization.

Both Ulteo and LTSP Cluster rely on presentation virtualization thus creating more virtualization overhead than application virtualization.

The most prominent manifestation of this overhead is created network traffic since the I/O data constantly needs to be transferred through network. Standard benchmarks do not give a straightforward indication of performance that is relevant for interactive applications in SaaS environment where one of the most important parameter is system responsiveness perceived by the user. However, there are several tools to monitor system state. Some of them are comprehensive suites that include variety of utilities and functions to monitor bear system performances such as Zabbix.

Several task-oriented benchmark tests were conducted along with the examination of user experience compared to desktop applications.

The test environment consisted of two Windows and two Linux application servers along with the central hub, which acted both as session manager and a web server to which clients connected through their browsers. All servers had the same basic hardware features: Intel Xeon dual-core CPU, 4 GB of RAM and 50 GB hard drive space with operating systems - Windows 2008 R2 and Debian Squeeze. Nineteen users in total were testing the services (applications) provided by this Cloud and expressed their level of satisfaction afterwards in questionnaire. These results, given in Table 2, show that user experience with the use of applications, provided by cloud, is at satisfactory level – very close to desktop environment. The only standout is noticeable for multimedia applications whose performance was dissatisfactory for 26 percent of users (5).

The dissatisfaction of part of users with performance of multimedia applications is justified by detailed log examination that shows the correlation between processing activities of application servers tested and network traffic they created. This is most notable when working with high-quality multimedia files. In this case low network throughput results in serious decrease of sound/image quality and unacceptable glitches during play occur.

Figure 4 presents processor load during one test and corresponding number of processes activated. It is noticeable that number of processes is accumulated during work - even though all users were not active all the time, which is shown by the level of processor load, the number of processes remained almost the same after reaching a certain level – the point where all the users logged in.

On the corresponding network traffic load it is

Table 2. User evaluation of service quality

Satisfaction level / Type of application	Not satisfactory	Satisfactory	Excellent
Office applications		✓	
Multimedia applications	✓(5)	✓(14)	
Small desktop spplications			✓

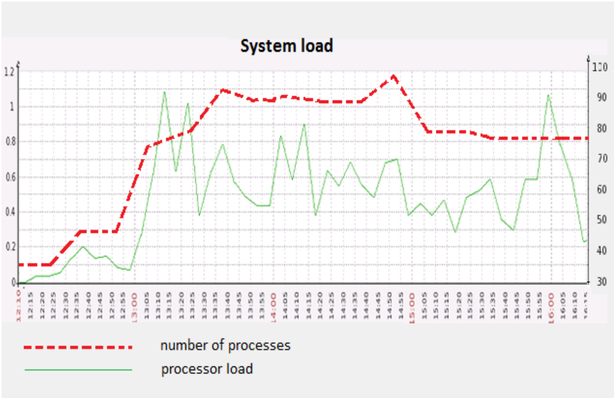

Figure 4. System load

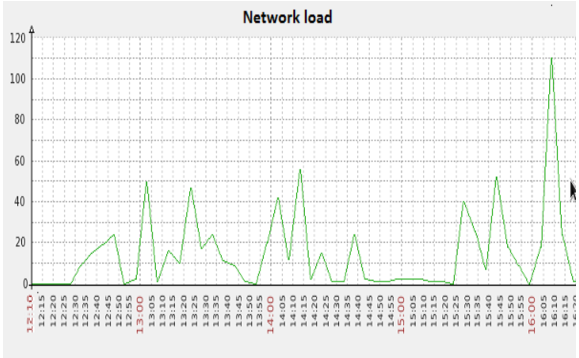

Figure 5. Network traffic load

interesting to notice a great peak just after 4 pm, Figure 5. It was the end of working hours at the company and all users were simultaneously logging out of the system, which created significant network load.

Since all applications are accessed through special Java/NX containers, which emulate terminals in seamless mode, some overhead appears there too. However, it is not perceivable during regular operation of client machines.

5 Deployment issues

Although, on-site software systems could offer significant degree of information security, the main stronghold of SaaS solutions lies in the fact that the software could be accessed remotely through a web browser which considerably decreases software installation, maintenance cost and contributes to mobility and portability.

Relying on open source solutions to build SaaS Cloud decreases software cost but requires additional effort to achieve certain functionality available in proprietary products. The following section describes our implementation of a missing functionality – persistence of user data between sessions but also discusses a possible obstacle encountered when using proprietary software on SaaS cloud platforms.

5.1. User application profile management

User application profile management ensures that the user's personal settings are applied to the applications used, regardless of the location and client device.

Saving user data between sessions, such as application preferences, is one of the main deployment issues. With each new session a temporary profile is created on the application server(s) connected. After the end of the session the profile is deleted along with all user data stored on that particular server.

In the environment where company is already maintaining distributed directory information services (such as Microsoft Active Directory), roaming user profile can be created and data can be stored in the directory structure. The main drawback of this solution is that each time all data has to be pulled from network, which creates unnecessary network load.

Another solution lies in the fact that user data, containing application preferences and history, is either stored in a set of files in home folder on Linux system or, even simpler, in a registry hive on Windows. The main idea is to save user settings to a non-volatile location on session end and to restore the data on the next session start.

To be able to do this, a close examination of the way user application settings were stored and retrieved on both platforms was performed. On Windows, all user preferences are stored in registry hive `HKEY_USERS` under user's SID (security identifier) that can easily be obtained through user's login name, and on Linux systems - in hidden folders and files in user's home folder. Also, each OS has a set of login and logout shell scripts that manipulate user data and restore user preferences. These scripts can be modified by adding hooks - a set of shell scripts devised to modify system behaviour according to need.

By putting to use the results of examination explained above, we have modified system login and logout scripts to accept special hooks that, on logout

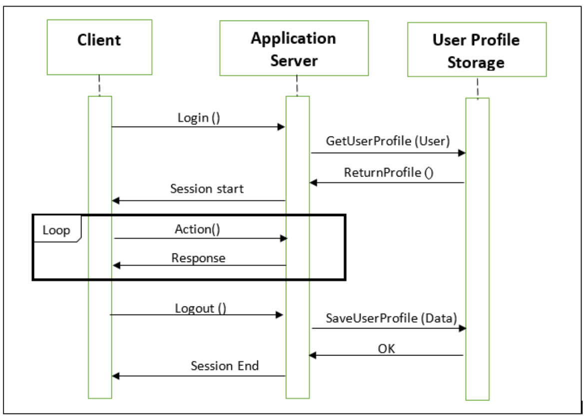

Figure 6. Application Profile Handling Sequence Diagram

copy user data to specialised user profile storage and, on the next login retrieve the data and copy it over newly created registry hive/home folder. Although each time user logs into system, a completely new profile is created, these scripts restore data from previous session from user profile storage. The whole process is transparent for the user and imposes minimal overhead. Sequence diagram shown on Figure 6 depicts the implemented protocol.

We believe that this tweak can be a good starting point for future development to resolve this issue.

5.2. Proprietary software

Many large software companies provide a cloud version of their software and encourage users, who do not want to maintain a local installation, to subscribe to these services rather than to set up their own cloud. Others are simply not ready for the cloud market yet, hence they have a licensing model that SaaS cloud provider does not profit from [2].

Application servers running Microsoft Windows operating system can be accessed only through Microsoft Remote Desktop Services which use RDP - a proprietary protocol. To ensure legal use of this proprietary software, according to the current situation, each user who has remote access has to have an RDS license and a special roaming licence for Microsoft application (Office, Visio etc.). The situation is similar with other proprietary software and should be carefully taken into consideration at planning stage.

Depending on the number of users and amount of proprietary software the cost of setting up this type of cloud could even bring the cost effectiveness of this venture into question.

6 Conclusion

After a careful consideration it can be said, without any hesitation, the best two open-source SaaS Cloud platform solutions are worthy opponents to their proprietary counterparts. They provide various functionalities that satisfy most companies' needs in area of SaaS Cloud computing.

In contrast to proprietary solutions, they provide full support for heterogeneous integration with Linux application servers and other open-source software. What is more, integration with other services such as file servers, printers, remote directories etc. is fully supported and constantly in development.

Our test environment confirmed negligible virtualization overhead mostly concerning network performance. Also, a deployment issue concerning persistence of user data between sessions was addressed and a possible solution to the problem was presented.

Having in mind that this area of software development has just begun its development, we believe that present issues will be resolved in future to come and open-source SaaS Cloud platforms have a good perspective.

7 Acknowledgments

We would like to thank the University Computer Centre (SRCE) of University of Zagreb, Croatia and especially to Dobriša Dobrenić and Damir Glavač for their contributions to the project and their assistance in system realization.

References

[1] Foster, I.; Zhao, J.; Raicu, I.; Lu, S. "Cloud Computing and Grid Computing 360-Degree Compared". Grid Computing Environments Workshop, 2008. GCE '08, Texas, USA, 2008.

[2] Armburst, M.; Fox, A.; Griffith, R. et al. "A View of Cloud Computing". Communications of the ACM. 53(4), 2010.

[3] Vouk, M.A. "Cloud Computing – Issues, Research and Implementations". Journal of Computing and Information Technology. 16(4). pp. 235-246. 2008

[4] Sahoo, J.; Mohapatra, S.; Lath, R. "Virtualization: A Survey on Concepts, Taxonomy and Associated Security Issues". Second International Conference on Computer and Network Technology, Thailand, 2010.

[5] Kusnetzky, D. "Virtualization is More than Virtual Machine Software", 2007, www.kusnetzky.net/publications/ImpactPapers/20070829_Virtualization_is_more_than_VM.pdf, April 2013.

[6] Fallenbeck, N.; Picht, H.; Smith, M.; Freisleben, B. "Xen and the Art of Cluster Scheduling". 2nd International Workshop on Virtualization Technology in Distributed Computing - ICAC, USA, 2006.

[7] Moshafar, N.M.; Chowdhury, K.; Boutaba, R. "A survey of network virtualization". Computer Networks. 54(4), 2010.

[8] Singh, A.; Korupolu, M.; Mohapatra, D. "Server-storage virtualization: integration and load balancing in data centres". 2008 ACM/IEEE conference on Supercomputing, USA, 2008.

[9] Turner, M.; Budgen, D.; Brereton, P. "Turning software into a service". Computer. 36(10). pp. 38-44. 2003.

[10] Mohan, A.; Knight, C.; Munro, M. "Understanding service-oriented software", IEEE Software. 21(3), 2004.

[11] Cusumano, M. "Cloud computing and SaaS as new computing platforms". Communications of the ACM. 53(4), 2010.

[12] Lee, J.Y.; Lee, J.W.; Cheun, D.W., Kim, S.D. "A Quality Model for Evaluating Software-as-a-Service in Cloud Computing". Seventh ACIS International Conference on Software Engineering Research, Management and Applications, Italy, 2009.

[13] Voras, I.; Mihaljević, B.; Orlić, M.; Pletikosa, M.; Žagar, M. Et al. "Evaluating Open-Source Cloud

Computing Solutions". 34th International Convention for Information and Communication Technology, Electronics and Microelectronics (MIPRO), Croatia, 2011.

[14] Popović, K.; Hocenski, Ž. "Cloud computing security issues and challenges". 33th International Convention for Information and Communication Technology, Electronics and Microelectronics (MIPRO), Croatia, 2010.