

Improved Performance of Multi Tenant Application Under TaaS

Damanpreet kaur¹, Deepak Aggarwal²

¹Baba Banda Singh Bahadur Engineering College, Fategarh Sahib, India

²Baba Banda Singh Bahadur Engineering College, Fategarh Sahib, India

Abstract

Multi-Tenancy refers to a software architecture in which a single instance of software runs on a server and serves multiple tenants. A tenant is a group of users who share a common access with specific privileges to the software instance. With a multitenant architecture, a software application is designed to provide every tenant a dedicated share of the instance including its data, configuration, user management, tenant individual functionality and non-functional properties. Multi-tenancy contrasts with multi-instance architectures, where separate software instances operate on behalf of different tenants. In this case performance factor is very important. So we improved performance under TaaS where TaaS is testing as service. Before using any multi-tenant application in real world we test its performance under TaaS.

Keywords: Cloud Computing, Cloud computing model, Multi-tenancy.

1. Introduction

Cloud computing is a comprehensive solution that delivers IT as a service. It is an Internet-based computing solution. The flexibility of cloud computing is a function of the allocation of resources on demand. This facilitates the use of the system's cumulative resources, negating the need to assign specific hardware to a task. For Example: One way to think of cloud computing is to consider our experience with email. Our email client, if it is Yahoo!, Gmail, Hotmail, and so on, takes care of housing all of the hardware and software necessary to support our personal email account. When we want to access our email we open our web browser, go to the email client, and log in. The most important part of the equation is having internet access. Our email is not housed on our physical computer we access it through an internet connection, and we can access it anywhere. If we are on a trip, at work, or down the

street getting coffee, we can check our email as long as we have access to the internet.

2. Model

Cloud computing has two type of model i.e Service Model and deployment model.

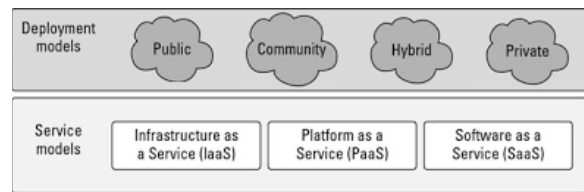


Fig 1: Cloud Computing Model

Service Model

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

Deployment Model

- Private Cloud
- Public Cloud
- Hybrid Cloud
- Community Cloud

2.1 Service Model

Cloud computing providers offer their services according to three fundamental models:

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

These three model is also known as SPI which means software as a service, platform as a service, Infrastrure as a service.

IaaS is used for build, PaaS is used for deploy, SaaS is used for buy.

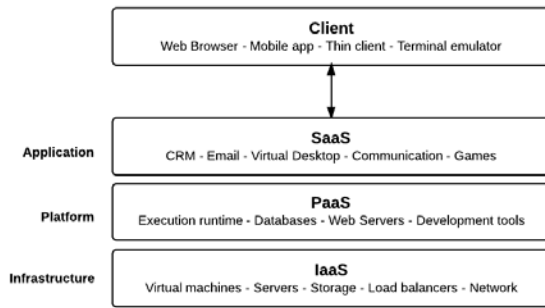


Fig: 2: Service Model

2.1.1 IaaS

Infrastructure as a Service (IaaS) is a form of cloud computing that provides virtualized computing resources over the Internet. Infrastructure as a Service is a way of delivering Cloud Computing infrastructure – servers, storage, network and operating systems – as an on-demand service. Rather than purchasing servers, software, datacenter space or network equipment, clients instead buy those resources as a fully outsourced service on demand. Example of IaaS is Amazon Web Services.

2.1.2 PaaS

Platform as a Service (PaaS) can be defined as a computing platform that allows the creation of web applications quickly, easily and without the complexity of buying and maintaining the software and infrastructure underneath it. PaaS is analogous to SaaS except that, rather than being software delivered over the web, it is a platform for the creation of software, delivered over the web. PaaS provides a method for programming languages to interact with services like databases, web servers, and file storage, without having to deal with lower level requirements like how much space a database needs. Example of PaaS is Google App Engine.

2.1.3 SaaS

Software as a service (SaaS) is a way of delivering applications over the Internet—as a service. Instead of installing and maintaining software, we simply access it via the Internet, freeing ourselves from complex software and hardware management. SaaS applications are sometimes called Web-based software, on-demand software, or hosted software. SaaS applications run on a SaaS provider’s servers.

It’s the idea that instead of buying licenses for software and then installing them inside our own infrastructure that instead these licenses can be installed in the cloud by a service provider. This allows us to scale up or scale down our needs instantly without having to buy new hardware, etc. Example of SaaS is servers, storage and networking.

2.2 Deployment model

A cloud deployment model represents a specific type of cloud environment, primarily distinguished by ownership, size, and access. Cloud computing has four types of deployment models.

- Private cloud
- Public cloud
- Hybrid cloud
- Community cloud

2.2.1 Private Cloud

Private Cloud is also known as internal cloud. A private cloud is owned by a single organization. Private clouds enable an organization to use cloud computing technology as a means of centralizing access to IT resources by different parts, locations, or departments of the organization.

2.2.2 Public Cloud

Public Cloud is a type of cloud hosting in which the cloud services are delivered over a network which is open for public usage. This model is a true representation of cloud hosting; in this the service provider renders services and infrastructure to various clients. The IT resources on public clouds are usually provisioned via the previously described cloud delivery models and are generally offered to cloud consumers at a cost or are commercialized.

2.2.3 Hybrid Cloud

Hybrid Cloud is a type of cloud computing, which is integrated. It can be an arrangement of two or more cloud servers, i.e. private, public or community cloud that is bound together but remain individual entities. A hybrid cloud is a cloud environment comprised of two or more different cloud deployment models. For example, a cloud consumer may choose to deploy cloud services processing sensitive data to a private

cloud and other, less sensitive cloud services to a public cloud.

2.2.4 Community Cloud

A community cloud is similar to a public cloud except that its access is limited to a specific community of cloud consumers. The community cloud may be jointly owned by the community members or by a third-party cloud provider that provisions a public cloud with limited access. The member cloud consumers of the community typically share the responsibility for defining and evolving the community cloud. Community Cloud is a type of cloud hosting in which the setup is mutually shared between many organisations that belong to a particular community, i.e. banks and trading firms. It is a multi-tenant setup that is shared among several organisations that belong to a specific group which has similar computing apprehensions.

3. Tenant

The idea of “Tenant” in context of Cloud Computing is not as simple as it might first appear. In Cloud Computing environment, each user is called tenant. A tenant has provided the ability to customize some parts of the application such as color of the User Interface (UI) or business terms but he is not enabled to customize the application’s code.

So here we can define the Tenancy as: “*Tenancy is a term used in cloud computing that represents each customer who has slightly given the ability to customize parts of application rather than fully customization.*”

3.1 Classification of Tenant

On the basis application usage, tenant can be classified into two types i.e Single Tenant and multitenant.

3.1.1 Single Tenant

Single-tenancy refers to a situation in which a single user or set of related users make use of dedicated resource. Some of the benefits of Single-tenant Applications are as follows:

- Enables the customer to customize the app as per requirement.

- Because application executes on dedicated machine so no one can easily access the data that the application maintains
- Provide control related to application incorporating scheduling, maintaining and backups etc.

3.1.2 Multitenant

It is an architecture in which a single instance of a software application serves multiple customers. Each customer is called a tenant. Tenants may be given the ability to customize some parts of the application. Multi-tenancy can be economical because software development and maintenance costs are shared. It can be contrasted with single-tenancy, an architecture in which each customer has their own software instance and may be given access to code. With multi-tenancy architecture, the provider only has to make updates once. With single-tenancy architecture, the provider has to touch multiple instances of the software in order to make updates. Multitenancy is a situation in which many users make use of shared resources. In other words, it is an architecture in which a single instance of a software application serves multiple users. Many modern, commercial, on-demand applications are Multi-tenant Applications.

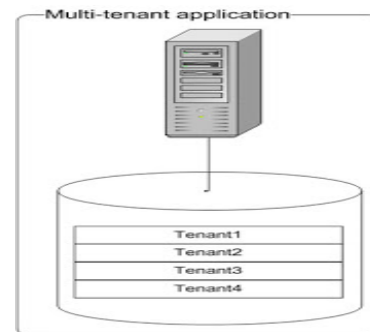


Fig 3: .Multi-tenant

4. Related Work

Chang Jie Guo, Wei Sun, Ying Huang, Zhi Hu Wang, Bo Gao et al. explains that the development of a native multi-tenant application poses many new research challenges brought by heterogeneous requirements. The design and implementation principles to instruct the native multi-tenant offering development and operation management are

proposed and a programming model and framework to simplify and speed up the application development.

Lijun Mei, W.K. Chan, T.H. Tse et al. found that cloud computing is an emerging computing paradigm. It aims to share data, calculations, and services transparently among users of a massive grid. Although the industry has started selling cloud computing products, research challenges in various areas, such as UI design, task decomposition, task distribution, and task coordination, are still unclear. Therefore, to study the methods to reason and model cloud computing as a step toward identifying fundamental research questions in this paradigm.

Sudip Chahal, Jay Hahn-Steichen, Das Kamhout, Rick Kraemer et al. in an Enterprise Private Cloud Architecture and Implementation Roadmap states that the private cloud is a shared multi-tenant environment built on a highly efficient, automated, and virtualized infrastructure. The private cloud is a shared multi-tenant environment built on a highly efficient, automated, and virtualized infrastructure.

Hong Cai, Ning Wang, Ming Jun Zhou et al. describes the core SaaS multi-tenancy models consisting of tenant interceptor, tenant context, tenant map, tenant propagation, remote resources such as database server.

5. Research Proposal

When resource get increase for the cloud then performance get decrease and we find out what was the reason and how to improve it Attempting to weave multi-tenancy throughout the fabric of an application's core logic and its underlying infrastructure is a complex undertaking. To build metadata-driven, multitenant applications from scratch without any prior experience is destined to be a time consuming and error-prone effort.

Ultimately, many would be SaaS providers struggle to succeed in building multitenant applications and end up wasting valuable time that could have been spent targeted on the innovation of core application functionality and features. One problem is that

traditional application development frameworks and platforms are not equipped to handle the special needs of modern Internet applications. As a result, new types of platforms are emerging to help simplify the development and deployment of multitenant applications. Force.com is the first and most mature general purpose, multitenant, Internet application development platform available today.

Multi-tenancy is basically a new software architecture principle in the realm of the Software as a Service (SaaS) business model. It permit to make full use of the economy of scale, as multiple customers – “tenants” – share the same application and database instance.

We have exploited distinct algorithm to create a data centre with varying number of host and run cloudlet on it. The algorithm shows how to create a data centre with ‘n’ number of hosts and run on different cloudlet on it. The cloudlet list show the Virtual Machine list creates to implements a cloud for Multi-Tenant's Applications. Algorithm also shows how to create a datacenter with one host and run two or more than two cloudlets on it. The cloudlets run under Virtual Machines i.e. VMs with the same MIPS (Millions of Instructions per Seconds) needs and take the same time to complete the execution. Algorithm also shows how to create a datacenter with two or more hosts and run two cloudlets on it. The cloudlets run in VMs with distinct MIPS requirements. The cloudlets will take distinct time to complete the execution depending on the requested VM performance.

To Implement these algorithm we use Java Language for CloudSim Tool and whatever the Result produce by CloudSim Simulator.

6. Overview of Simulator

6.1 Cloud Sim

CloudSim is basically a library for simulation of cloud computing scenarios. CloudSim provides basic classes for describing virtual machines, data centers, applications, users, computational resources and policies.

CloudSim supports VM scheduling at two levels: at Host level and VM level. At *Host Level* – It is possible to specify how much of the overall processing power of each core in a host will be assigned at each VM. At *VM Level* – In this the VMs assign specific amount of the available processing power to the individual task units that are hosted within its execution engine.

The fundamental classes of CloudSim which constitutes the basic building blocks of simulator are described as follows:

- **Data Center** – This class of CloudSim models the core infrastructure level services offered by resource providers in a cloud computing environment.
- **DatacenterBroker** – This class models a broker that is responsible for mediating between users and service providers depending on users QoS needs and deploys service tasks across clouds.
- **Cloudlet** – This class models the cloud based application services that are commonly deployed in the data centers.
- **VirtualMachine** – This class models an instance of a VM whose management is the responsibility of host component during its life cycle.

7. Experimental Result

Create Datacenters: These are the resource providers in Cloud Sim. We need at least one of them to run a Cloud Sim simulation. The various snapshots show the results after running the cloud sim scripts are:

Script1: shows how to create a datacenter with one host and run one cloudlet on it.

The output shows the status, Virtual machine ID, Start Time and Finish Time.

Table 1: Create a datacenter with one host and one cloudlet

Script 1			
Sr. No.	Vm ram in MB	Host ram in MB	Debt value in %
1.	128	128	22.8
2.	256,1024	128	aborted

Script 2			
Sr. No.	Vm ram in MB	Host ram in MB	Debt value in %
1.	128	256	32.8
3.	256	128	Aborted

Description: For measuring performance we define the value of host ram and virtual ram. The output shows the status, Virtual machine ID, host ram, vm ram, Start Time and Finish Time. When the value of host ram is greater than vm ram debt is increase otherwise simulation is aborted.

Script2: Shows how to create a datacenter with one host and run two cloudlets on it.

Table 2: Datacenter with one host and run two cloudlets.

Script 2			
Sr. No.	Vm ram in MB	Host ram in MB	Debt value in %
1.	128	256	32.8
3.	256	128	Aborted

Description: The cloudlets run in VMs with the same MIPS requirements. The cloudlets will take the same time to complete the execution. The total debt increase as the numbers of cloudlets increase and the value of host ram is greater than vm ram otherwise simulation is aborted. Hence we can compare the performance when the numbers of cloudlets increase.

8. Conclusion & Future Work

Recently lots of efforts have been made to design and develop cloud technologies focusing on defining new policies, methods and mechanisms for efficiently managing Cloud Infrastructures.

Researchers require tools to test the newly developed methods and policies to evaluate the hypothesis prior to real deployment in an environment where one can reproduce tests. Simulation approach assists the one in evaluating cloud computing systems and application behaviors offer significant benefits as they permit cloud developers (1) to evaluate the

performance of their provisioning and service delivery policies in a repeatable controllable environment free of cost.

In this paper, we have discuss simulator based on high performance computing network that are Cloud Sim for cloud computing. We have performed small simulation as well with some parameter to understand the cost of using cloud computing in real world.

References

- [1] Andy Zaidman, Slinger Jansen, Cor-Paul Bezemer, Jaap Kabbedijk, October 16, 2014. “Deining Multi-Tenancy: A Systematic Mapping Study on the Academic and the Industrial Perspective”, Journal of Software and Systems.
- [2] Burkhard Neidecker, Keith Jeffery , 2010. “The Future of cloud computing”.
- [3] Cor-Paul BEZEMER , 2014. “Performance Optimization of Multi-Tenant Software Systems” , The Netherlands by Gildeprint Drukkerijen. ISBN: 978-90-79982-16-5.
- [4] Darrell M. West, Allan A. Friedman October 2010.” Privacy and Security in Cloud Computing” Vol. 3.
- [5] Dr.S.Mehfuz, Dr.G.Sahoo, Rashmi, August 2013.” Securing Software as a Service Model of Cloud Computing: Issues and Solutions”. International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol.3, No.4.
- [6] Dr. P Jayarekha, Pallavi G B, July-2014.” Multitenancy in SaaS: A comprehensive Survey”. International Journal of Scientific & Engineering Research, Volume 5, Issue 7, ISSN 2229-5518.
- [7] Dr.T.Bhuvanewari, M.Saraswathi, November 2013.” International Journal of Advanced Research in Computer Science and Software Engineering” Volume 3, Issue 11, ISSN: 2277 128X.
- [8] Jack Brass, 2010 “Physical Layer Network Isolation in Multi-tenant Clouds” International Conference on Distributed Computing Systems Workshops.
- [9] Jeffrey Shafer, March 13, 2010. “I/O Virtualization Bottlenecks in Cloud Computing Today” Pittsburgh, PA, USA.
- [10] Makan Pourzandi, Nelson Gonzalez, 2012. “A quantitative analysis of current security concerns and solutions for cloud computing” Gonzalez et al. Journal of Cloud Computing: Advances, Systems and Applications.
- [11] Mohammad Ashraf Hossain, Monjur Ahmed ,January 2014.” CLOUD COMPUTING AND SECURITY ISSUES IN THE CLOUD” International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.1.
- [12] Mohamad Abou El-Nasr, Hany Soliman , Mohamed Magdy Mosbah, October 2013 “CURRENT SERVICES IN CLOUD COMPUTING: A SURVEY” International Journal of Computer Science, Engineering and Information Technology (IJCEIT), Vol.3, No.5.
- [13] Prof. Mohamed Abu Rizkaa, 2010. “A Dynamic Cloud and Virtualization Benchmarking Suite” Arab Academy for Science, Technology and Maritime Transport Workshops on Enabling Technologies: Infrastructure for Collaborative EnterpriseCloudGauge.
- [14] Parul Thakur & Neha Kamdar, Abhishek Pipliya, Om Parkash Karada, 2013” Analytical Survey Model on Consumption of Cloud service Models”. ISSN: 2278-8948, volume-2, issue-4.