International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-4, Issue-6)

Research Article

June 2015

An Review for Ornamental Security and Performance for Cloud Applications

Bhawna Sehgal
Mtech Student (C.S.E)
H.E.C Jagadhari
Kurukshetra University
Haryana, India

Er. Jasbeer Narwal
Assistant Prof. (Mtech)
H.E.C Jagadhari
Kurukshetra University
Haryana, India

Abstract:

The capacity to measure a web application or website is tangled directly to understanding where the resource constraints lie and what effect the addition of various resources has on the application. Unfortunately, architects more often than not assume that simply adding another server into the mix can solve any performance problem and security issues. When you start adding new hardware/update existing hardware in a web cloud, the difficulty starts increasing which disturbs performance and hence security. While valued cloud computing services save troubles to maintain the computational environment, there are various shortcomings such as overhead of virtual machines, possibility to share one physical machine with several virtual machines, and indeterminacy of topological portion of their own virtual machines. Multi-tenancy is one of key features of the service oriented computing especially for Software as a Service (SaaS) to influence economy of scale to bring down total cost of ownership for both service consumer and provider. This paper aims to study the technologies to build a cost-effective, secure and scalable multi-tenant infrastructure and how to improve the security and to increase its performance. This paper also identifies the potential performance blockages, summarizes corresponding optimization approaches and best implementation practices for different multi-tenant business usage models.

Keywords: Cloud Security, Cloud Computing, Performance evaluation, Cloud Platform, Cloud Application

I. INTRODUCTION

Latest progress of engineering has cut down costs of computers and network, and this change gave a vast effect on high performance computing environment. Grid computing and cloud computing, which are computer environments consisted of commodity computers and commodity network devices, are clutching people's attention quickly. Grid computing and cloud computing are now recognized as a suitable source that allows users to reveal computational power as much as they need, whenever they want. Cloud computing service such as Amazon EC2 seems to bring a colossal supercomputer by our side, however, is it really sensible to apply the paid service as research environment for everyday activities. In case the priced cloud computing service replaces supercomputers, what could be problems for transition? First question would be which is more cost-effective to purchase a supercomputer and use it for a couple of years, or to rent computational nodes as you go. Second question would be how fast and secure their applications run on the commercial computational cloud. Virtualization technology has been developed, and it is relatively common to build a cloud computing environment as a group of virtual machines. This methodology has pros and cons. One of pros for users is that computational environment looks homogeneous; therefore, users will never be concerned with heterogeneous hardware or software environment. Cons for users are, for example, overhead of virtual machines, possibility to share one physical machine with several virtual machines, and indeterminacy of topological portion of their own virtual machines.

The Companies of several sizes have outsourced their business applications to third party service providers through Software as a Service (SaaS)[4][8] deals supported by service oriented computing architecture. Such outsourcing deals span a fairly wide range of applications to support business operations. The typical ones include payroll, call center, procurement, finance and accounting, human resource management etc. SaaS providers usually develop or obtain SaaS applications and host them as services to serve specific needs of their clients by leveraging service oriented computing technologies[2][3]. One of the key feature of the SaaS application is Multi-tenancy. By leveraging Multitenancy, SaaS providers can ease operations and lessen delivery cost for a big number of tenants. In a multi-tenant enabled service environment, user requests from different organizations and companies (tenants) are served simultaneously by one or more hosted application instances and databases based on a scalable, shared hardware and software infrastructure.

Multi-tenant infrastructure should take care the following key aspects:

1. Resource Separation: Separate the resources allocation and usage among tenants;
2. Security: Prevent illegal resources access and potential malicious attack;
3. Customization: Support tenant-specific features or Service Level Agreement(SLA) through configurations;
4. Scalability: Scale the SaaS application's delivery infrastructure to livelihood growing number of tenants with well managed cost increase, performance and availability guarantee;

To make the service offerings more profitable and more attractive to those clients with very limited IT investment budget, e.g. the average cost of the service for each tenant should be kept as low as possible, Small and Medium Business (SMB),. There are mainly three kinds of service cost:

1. Application development cost: To satisfy each customer, additional development might be involved to address its unique requirements. There is always a balance between customer satisfaction improvement and development cost management.
2. Management cost: The tenant related operational management processes and activities, e.g. lifecycle management, monitoring, data backup etc..
3. Infrastructure cost: It includes the hardware, software and utilization costs. Generally, for a given system, the total throughput can be used to measure the maximal tenant number the system can support with an acceptable SLA.

Although a typical SaaS application is consist of application instance (e.g. user interface, business logic, process etc.) and database, this paper mainly emphases on data tier multi-tenancy study. firstly, we explore all kinds of possible implementation patterns of data tier multi-tenancy from the sides of isolation, security, customization and scalability etc. Mostly, the cost of these patterns should be studied from the infrastructure, management and development aspects by using different kinds of measurement metrics. This paper only emphases on the performance evaluation via a set of simulations, and recognizes potential performance blockages, corresponding optimization approaches and best implementation practices for different multitenant business usage models

II. CLOUD COMPUTING

Cloud computing is a model for assisting convenient, on-demand network access to a shared pool of configurable computing resources (e.g. Storage, Applications, Networks, Servers, and Services) that can be quickly provisioned and released with insignificant management effort or service provider interaction. This cloud model encourages availability and is consist of four deployment models and three service models. The "cloud" is defined as the Internet surrounding every part of our daily lives, similar to the clouds in the sky. However many new enterprise related axioms have evolved from the original "computing in the cloud" concept; "Software + Services", "Software-as-a-Service", which has evolved as a more Microsoft related term, and "social-media" which is a basis in social networking and development. While a common delusion for cloud computing is storage space on the Internet, the cloud offers many services, infrastructure benefits and scalability which may not be possible within ordinary local-area enterprise networks. When cloud storage is used as the primary location of files and documents, a certain trust is left in the hands of the storage provider to ensure certain steps are taken to prevent data loss and maintain the integrity of the file system; aiding maximum uptime, reducing downtime and sustain the highest levels of physical protection and data security.

When something disturbs cloud storage, things can go wrong for many end users. While data which is stored in the cloud isn't actually stored in the cloud; rather a Data Center housing hundreds of servers and thousands of networking cables, physical disasters are one of the greater threats to the cloud.

As physical disasters go, some will affect the whole cloud, or whole datacenter if you think geologically or physically, and some will affect portions or individual sections. Natural disasters are a great concern to those who run and use cloud computing services. As many natural disasters are unpredictable, from volcanoes and tsunamis, floods to earth tremors, recovering from these disasters are often impossible. Preventing disasters from affecting the cloud itself is the only genuine thing the staff, management and planners can predict. Nobody would build a datacenter; let alone any Government building,School or Hospital, Business Venture, or any building or structure of importance in a geographic location where an active or inactive volcano lies, e.g. In case of cloud downtime or event which causes the cloud to fail, a backup solution is often used in an alternate location. This ensures a constant stream of data being backed up to an alternate datacenter, away from any potential natural disaster, but keeping data secure and maximizing authorized accessibility.

III. DESIGN PATTERNS OF DATA TIER MULTI-TENANCY

1. Resource Isolation Patterns: In the data tier, there are varying degrees of data isolation for a multi-tenant application that ranges from an isolated environment to a totally shared environment. Implementation patterns along this spectrum include three models as given below:

a) *Totally isolated (Dedicate database pattern):* each tenant owns a separate database

b) *Partially shared (Dedicate table/schema pattern):* multiple tenants share a database, but each tenant owns a separate tables/schema

c) *Totally shared (Share table/schema pattern):* multiple tenants share same database, and share same tables/schema.

2. Security Patterns: It focuses on the data security isolation among tenants, which is also described as "preventing a user from getting the rights to access data belonging to other tenants". It aims to protect the security of each tenant at comparable security levels as those of the traditional single-tenant system. In general, there are two patterns to realize the data security mechanisms as given below:-

a*) Filter-based pattern in application level*: Through adding the application level filter into each user request of tenant, a tenant's data can be accessed only by the tenant its self. For dedicate database or dedicate table/schema isolation

patterns, the filter is based on database name or schema name to access associated database or schema associated with the corresponding tenant. While for share table/schema isolation pattern, the filter is based on the tenant ID column in every table to access records associated with the appropriate tenant, e.g. modifying a SQL statement with where clause 'tenantID =XXX'. While easy to implement, this approach has potential security risks. Since all the tenants share a single platform level DB account and connection, a malicious tenant user may access other tenants' data via SQL injection attack. For example, a hacker can modify the above SQL statement's where clause as 'tenantID =XXX or 1==1' to access data of all tenants.

b) *Permission-based pattern in DBMS level:* Each tenant is assigned a dedicated DB access account and connection which only has rights to access its own resources (e.g. the dedicated database or tables/schema in 1st & 2nd isolation patterns). While for 3rd isolation pattern, we can leverage the row level access control mechanism provided by DBMS, e.g. the label based access control (LBAC) feature. In this way, we can completely prevent potential SQL injection attack.

3. Customization Patterns: In data customization feature, there are also various flexibility degrees for a multi-tenant application that ranges from complex schema customization to simple field extension. Obviously, for the dedicated database or table/schema isolation patterns, this is not an issue since the tenants have their separated schemas. The changes of the data model of one tenant can be made directly to its specific database/tables without impact to other tenants. However, for the share table/schema isolation pattern, because of the sharing of schema, it can only support data field extension, which flexibility degree is usually measured by the maximal number of extension fields.

a) *Reservation field pattern*: Pre-define a fixed number of additional data columns in each table with a common column type (e.g. Varchar). This pattern is very easy to implement but has some limitations. Since the extended columns exist for all rows in the table no matter if it needs or not, some storage space will be wasted. Furthermore, this mechanism can't support those tenants requiring more extension fields than the predefined number.

b) *Extension sub-table pattern*: A sub table, which associates with the main table via the record id column, is built to store all the extended fields of the records in main table. This approach is very flexible. It does not have limitation in the maximum number of the extended fields. However, it may suffer from poor performance resulted by join-search.

c) *XML extension field pattern*: This approach leverages the new XML features provided by some DBMS. For each record, all of its extended field data are stored in a single XML field. It can provide better programming interfaces than the dynamic extended sub-table approach, but will also produce significant performance overhead.

4. Scalability Patterns: The Cost effective scalability is very important for multitenant system. In an ideal situation, the maximum number of tenants supported by the multi-tenant system should increase in direct proportion to the increase of resources, while still keeping the performance metrics of each tenant in a predefined and acceptable level. Generally, there are two kinds of patterns for scaling:

a) *Scale up*: (Vertical scaling) through adding more resources (such as memory, CPU, and disks I/O) to the existing machines. This is an easy-to-use and manageable approach. However, it may not provide linear scalability. As you add resources, overhead comes out in resource management that limits the scalability of single systems.

b) *Scale out*: (Horizontal scaling) through adding additional machines to the existing system. Compared with scale up, this approach provides a more cost-effective and smooth scalability, since it can incrementally extend the system by adding more resources to a low-cost hardware set established initially. Although scale out may certainly increase the management complexity, it can also improve the reliability and availability of the system, in some cases because of redundancy. In this paper, for scalability aspect, we emphasis mainly on two scale-out approaches:

i) *Application level routing*: Each machine of the hardware set has a standalone database server instance. User requests from different tenants are intercepted and dynamically routed to corresponding database through an application level, tenant aware dispatcher. Theoretically, as the number of machines added increase, the total throughput of system can almost lineally scale.

ii) *DB partitioning*: All machines of the hardware set share a single set of database server instance via clustering technology. Tenant users' requests are automatically routed to pre-configured DB partitions in an implicit way. Comparing with application level routing, this approach provides a uniform view to maintain machines and decrease the complexity of management. However, to provide a set of complicate common clustering oriented management features, the DB partitioning technology will certainly introduce more performance overhead, and also face some challenges in scalability and availability etc.

IV. PERFORMANCE AND SECURITY IN CLOUD COMPUTING

The Cloud architecture extends to the client, where web browsers and/or software applications access cloud applications.Cloud storage architecture is loosely coupled, where metadata operations are centralized enabling the data nodes to scale into the hundreds, each independently delivering data to applications or users.

Security is the challenge seen related to Cloud Computing according to our architecture.

• The main security concerns include privacy in interoperability, performance, reliability compliance and visibility under virtualization.

• The Good News: Since Security is seen as such a major issue, it is getting much attention. This attention is resulting in Security-related benefits such as greater segmentation and better categorization and performance is another issue if we change the level of security in the Cloud.

With increasing Business complexity, organizations are seeking innovative business models and specialized technologies to cater to customer demands. Cloud computing technologies can provide organizations competitive advantage in the market, cost reductions, higher margins, simplified maintenance and management of applications across the enterprise, greatly extended scalability, agility, high availability, automation, large data storages and reliable backup mechanisms.
By using Cloud Computing environments, organizations can focus on their core business as opposed to concerning themselves about infrastructure scalability. Organizations may explore use of cloud computing initially for better performance through peak demand periods but eventually adoption could spread to other areas.

V. SERVICE MODELS

Cloud Software as a Service (SaaS). The ability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.
Cloud Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations.
Cloud Infrastructure as a Service (IaaS). The capability provided to the consumer is to providE processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls).

VI. DEPLOYMENT MODELS

Community Cloud: The Cloud infrastructure is shared by various organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be managed by the organizations or a third party and may exist on premise or off premise.
Public Cloud: The Cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.
Private Cloud: The Cloud infrastructure is operated only for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.
Hybrid Cloud: The Cloud infrastructure is a composition of two or more clouds (private, community, or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds).

VII. CURRENT VIEW

Critics argue that Cloud Computing is not secure enough because data leaves companies' local area networks.
It is up to the clients to decide the vendors, depending on how willing they are to implement secure policies and be subject to 3rd party verifications. Salesforce, Amazon and Google are currently providing such services, charging clients using an on-demand policy. Statistics suggest that one third of breaches are due to laptops falling in the wrong hands and about 16% due to stolen items by employees. Storing the data in the cloud can prevent these issues altogether. Moreover, vendors can update application/OS/middleware security patches faster because of higher availability of staff and resources.
According to cloud vendors, most thefts occur when users with authorized access do not handle data correctly. Upon a logout from the cloud session, the browser may be configured to delete data automatically and log files on the vendor side indicate which user accessed what data. This approach may be deemed safer that storing data on the client side. There are some applications for which Cloud Computing is the best option. One example is the New York Times using Amazon's cloud service to generate PDF documents of various-decade old articles. The estimated time for doing the task on the Times' servers was 14 years, whereas the cloud provided the answer in one day for a couple hundred dollars.
However, the profile of the companies that currently use the Cloud Technology includes Web 2.0 start-ups that want to minimize material cost, application developers that want to enable their software as a service or enterprises that are exploring the cloud with trivial applications. The fact that Cloud Computing is not used for all of its potential is due to a variety of concerns. The following surveys the market in terms of continuous innovation, academia and industry research efforts and Cloud Computing challenges.

VIII. PERFORMANCE ISSUES

Everybody appears to be talking loud about Cloud Computing nowadays. But the recently reported outages at Salesforce, Amazon and Google has made us think otherwise and wonder if the cloud is really ready to meet all the hype and attention its getting. No doubt, there are cost savings related to licensing, maintenance and application / server management. But does this ensure that your end users are getting the online experience you want them to have?

Many Cloud Computing providers provide custom built management consoles or control panels for managing server resources. These consoles provide customers with availability statistics and status messages in the event of significant outages that impact end users.

IX. RELATED WORKS

In the hosted applications of the early 90s [1][5], companies only moved their hardware and applications from their premises to the data centers, and paid a premium to have their applications hosted. This was a typical single-tenancy scenario without any hardware or software sharing across customers of the service provider. To achieve more benefits from improving the sharing efficiency, some hosting service providers gradually started to leverage virtualization technologies [14][15][16] on machine, operation system etc levels, but each tenant still owns dedicate application instance and database in these hosting models. In recent years, a native multi-tenant model, as exemplified in SaaS [6][7][8][9][10], achieves great successes. In this model, a single instance of application or a single database can both serve multiple tenants. For the multi-tenant data tier, Fred & Gianpaolo studied the similar topics [11]. They evaluate patterns on aspect of multi-tenant data customization, and provide a performance report based on SQL Server. Our work differentiates in at least two ways. First, this paper touches more perspectives and corresponding design patterns of multi-tenant data model, such as the isolation, security and scalability patterns. Secondly, this paper conducted a wider scope of performance.

X. CONCLUSION

In this paper, we explore many kinds of typical multi-tenant data tier implementation patterns on aspects of isolation, security, customization and scalability. We also evaluate performance of these patterns through a series of experiments, and summarize a set of valuable conclusion and best practices on how to design an effective multi-tenant data model. This work can help the service provider and multi-tenancy application developer. We have already applied parts of the study results into the design and implementation of a real multitenant application. The hands-on experiences will help us to touch more research topics on performance optimization and scalability aspects in data tier, such as tenant behavior awareness load balancing in distributed database cluster environment. Another goal of our research is to explore technologies to transform traditional DBMS to be more suitable for multi-tenant environments [17]. We will start from the open source database server (like mySQL [18], Derby [19] etc), and refine its engine, query optimizer, data model organization structure etc. We believe that a new kind of DBMS with native multi-tenancy design will emerge to support both SaaS applications developers and service providers.

For those deploying software out in the Cloud, scalability is a major issue.

1. The system must also coordinate information coming from multiple sources fast, not all of which are under the control, of the same organization
2. The need to marshal resources in such a way that a program continues running smoothly even as the number of users grows.
3. It's not just that servers must respond to hundreds or thousands of requests per second.

With these equations there is a possibility that the security can be broke, but the performance will be increased according to our scenario when the number of users are increased. In future we want to design a protocol which will be more secure and the performance of the cloud will increase.

REFERENCES

[1] Lijun Mei,W.K. Chan,T.H. Tse, "A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues", 2008 IEEE Asia-Pacific Services Computing Conference.

[2] Hong Cai, Ning Wang, Ming Jun Zhou, "A Transparent Approach of Enabling SaaS Multi-tenancy in the Cloud '', 2010 IEEE 6th World Congress on Services.

[3] Chang Jie Guo, Wei Sun, Ying Huang, Zhi Hu Wang, Bo Gao , "A Framework for Native Multi-Tenancy Application Development and Management''2007 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services.

[4] Zeeshan Pervez, Sungyoung Lee, Young-Koo Lee," Multi-Tenant, Secure, Load Disseminated SaaS Architecture'' Department of Computer Engineering, Kyung Hee University, South Korea.

[5] Jack Brass, "Physical Layer Network Isolation in Multi-tenant Clouds" International Conference on Distributed Computing Systems Workshops 2010.

[6] Pankaj Goyal, "Policy-based Event-driven Services-oriented Architecture for Cloud Services Operation & Management" 2009 IEEE International Conference on Cloud Computing.

[7] Guoling Liu. "Research on Independent SaaS Platform" School of Information Science and Technology Shandong Institute of Light Industry Jinan, China.

[8] Qiang Li, Qinfen Hao, Limin Xiao, Zhoujun Li, "Adaptive Management of Virtualized Resources in cloud Computing Using Feedback Control" The 1st International Conference on Information Science and Engineering (ICISE2009)

[9] MingXue Wang, Kosala Yapa Bandara and Claus Pahl, "Process as a Service - Distributed Multi-tenant Policy-based Process Runtime Governance", 2010 IEEE International Conference on Services Computing.

[10] Enrique Jiménez Domingo, Javier Torres Niño, Angel Lagares Lemos, Miguel Lagares Lemos, Ricardo Colomo Palacios, Juan Miguel Gómez Berbís, **"**CLOUDIO: A Cloud Computing-oriented Multi-Tenant Architecture for Business Information Systems",2010 IEEE 3rd International Conference on Cloud Computing.

[11] Wei-Tek Tsai, Qihong Shao, Jay Elston, "Prioritizing Service Requests on Cloud with Multi-tenancy", IEEE International Conference on E-Business Engineering 2010.

[12] Afkham Azeez, Srinath Perera, Dimuthu Gamage, Ruwan Linton, Prabath Siriwardana, Dimuthu Leelaratne, Sanjiva Weerawarana, Paul Fremantle, "Multi-Tenant SOA Middleware for Cloud Computing" WSO2 Inc. Mountain View, CA, USA.

[13] Franclin S. Foping, Ioannis M. Dokas, John Feehan, Syed Imran. ,"A New Hybrid Schema-Sharing Technique for Multitenant Applications", Cork Constraint Computation Centre, University College Cork, Ireland.

[14] "Application Tenancy" Progress Software Corporation, 14 Oak Park, Bedford,MA 01730 USA.

[15] Sivaram Shunmugam, "Cloud computing with red hat solution", Red Hat Asia Pacific Pte Ltd.

