International Journal of Information Technology and Knowledge Management
July-December 2012, Volume 5, No. 2, pp. 233-237

A FRAMEWORK FOR ENHANCING SECURITY AND PERFORMANCE IN MULTI-TENANT APPLICATIONS

Simarpreet Singh Gulati[1] and Sakshi Gupta[2]

ABSTRACT: The ability to scale a web application or website is tied directly to understanding where the resource constraints lie and what impact 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 fix any performance problem and security issues. When you start adding new hardware/update existing hardware in a web cloud, the complexity starts increasing which affects performance and hence security.

While priced cloud computing services save pains to maintain the computational environment, there are several drawbacks such as overhead of virtual machines, possibility to share one physical machine with several virtual machines, and indeterminacy of topological allocation of their own virtual machines. Multi-tenancy is one of key characteristics of the service oriented computing especially for Software as a Service (SaaS) to leverage economy of scale to drive 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 enhance its performance. This paper also identifies the potential performance bottlenecks, summarizes corresponding optimization approaches and best implementation practices for different multi-tenant business usage models.

Keywords: Cloud Computing, Cloud Platform, Cloud Security, Performance Evaluation, Multitenant

1. INTRODUCTION

Recent progress of engineering has cut down costs of computers and network, and this change gave a huge impact on high performance computing environment. Grid computing and cloud computing, which are computer environments consisted of commodity computers and commodity network devices, are grabbing people's attention rapidly. Grid computing and cloud computing are now recognized as a convenient source that allows users to bring out computational power as much as they need, whenever they want. Cloud computing service such as Amazon EC2 seems to bring a gigantic supercomputer by our side, however, is it really reasonable to utilize the paid service as research environment for everyday activities. In case the priced cloud computing service replaces supercomputers, what could be obstacles 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 quite common to build a cloud computing environment as a flock 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 bothered 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 allocation of their own virtual machines.

Companies of various 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 acquire 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 characteristics of the SaaS application is Multi-tenancy. By leveraging Multitenancy, SaaS providers can significantly ease operations and reduce delivery cost for a big number of tenants. As illustrated in Figure 1, in a multi-tenant enabled service environment, user requests from different organizations and companies (tenants) are served concurrently by one or more hosted application instances and databases based on a scalable, shared hardware and software infrastructure.

[1] Assistant Professor, Department of Computer Science and Applications, Guru Nanak Khalsa Institute of Technology & Management Studies, Haryana, India
E-mail: sgulati21@gmail.com

[2] M. Tech Scholar, Ambala College of Engineering and Applied Research, Haryana, India, *E-mail: sakshigupta553@gmail.com*

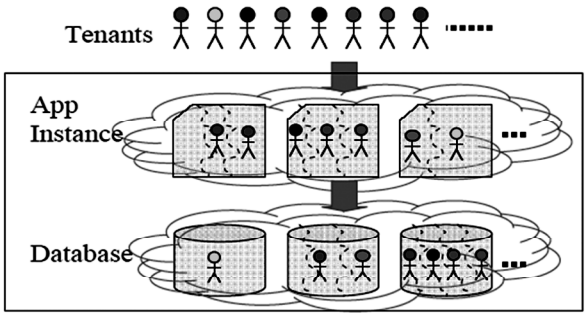

Figure 1: A Multi-Tenant Enabled Service Environment

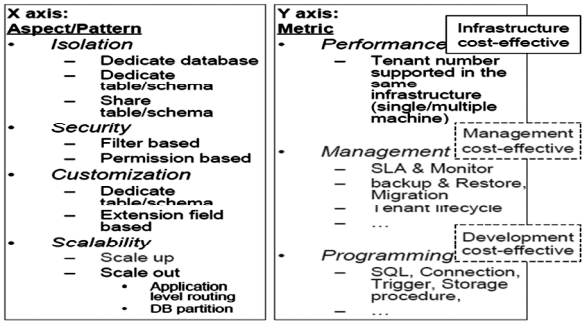

Figure 2: Overview of Multi-tenant Date Tier Work

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

1. **Resource Isolation**: Separate the resources allocation and usage among tenants;
2. **Security**: Prevent invalid 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 support 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. Small and Medium Business (SMB), the average cost of the service for each tenant should be kept as low as possible. There are mainly three kinds of service cost:

1. **Infrastructure Cost**
2. **Management Cost**
3. **Application Development Cost**

Although a typical SaaS application is composed of application instance (e.g. user interface, business logic, process etc.) and database, this paper mainly focuses on data tier multi-tenancy study. As illustrated in Figure 2, we first explore all kinds of potential implementation patterns of data tier multi-tenancy from the aspects of isolation, security, customization and scalability etc. Generally, 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 focuses on the performance evaluation via a set of simulations, and identifies potential performance bottlenecks, corresponding optimization approaches and best implementation practices for different multitenant business usage models.

2. 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 [1][5][6] 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], 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. From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection can not be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. However, this dynamic feature also makes traditional integrity insurance techniques futile and entails new solutions. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in

multiple physical locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world.

Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works [3]-[7]. These techniques, while can be useful to ensure the storage correctness without having users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on single server scenario and most of them do not consider dynamic data operations. As an complementary approach, researchers have also proposed distributed protocols [2]-[8] for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes is aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

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 broader scope of performance.

3. DESIGN PATTERNS OF DATA TIER MULTI-TENANCY

3.1 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 illustrated in Figure 3:

(*a*) Totally isolated (Dedicate database pattern): eachtenant 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

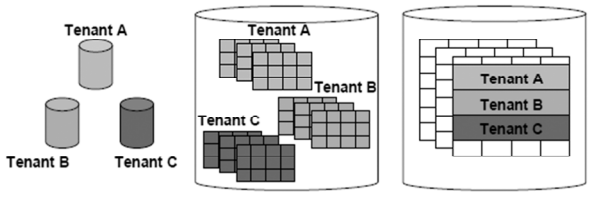

Figure 3: Isolated vs. Shared Data Environment

To be noted, in the 3rd pattern, records of all tenants are stored in a single shared table sets mixed in any order, in which a tenant ID column is inserted in each table to associate the data records with the corresponding tenants.

3.2 Security Patterns

This section focuses on the data security isolation among tenants, which is also described as "preventing a user from getting the privileges to access data belonging to other tenants". It aims to safeguard 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 illustrated in Figure 4.

(*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 ensured to 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'. Although 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 privileges 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.

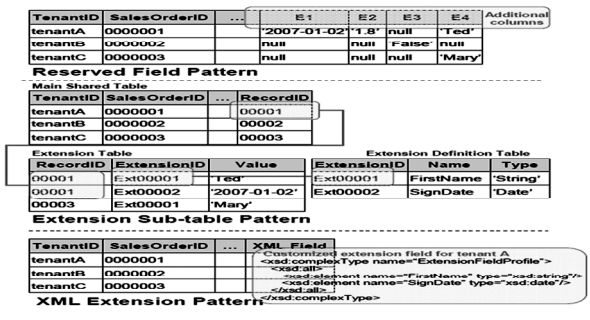

Figure 4: Filter vs. Permission based Security Date Access

3.3 Scalability Patterns

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:

Scale Up: (Vertical scaling) through adding more resources (such as CPU, memory, 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.

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 inevitably 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 focus mainly on two scale-out approaches:

Security and Performance in Cloud Computing: 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 #1 challenge seen related to Cloud Computing according to our architecture.

- The main security concerns include performance, reliability compliance, privacy in interoperability 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 logging 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.

4. 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 appropriately. 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 several-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.

5. PERFORMANCE ISSUES

Everybody seems 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.

6. FUTURE WORK

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 [7]. We will start from the open source database server (like mySQL [8], Derby [9] 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 need to marshal resources in such a way that a program continues running smoothly even as the number of users grows.
2. It's not just that servers must respond to hundreds or thousands of requests per second.
3. The system must also coordinate information coming from multiple sources fast, not all of which are under the control, of the same organization.

With these equations there is a possibility that the security can be breached, 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] Draft NIST Working Definition of Cloud Computing v15, http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc

[2] Foley M.J., 2008. "Microsoft 2.0: How Microsoft Plans to Stay Relevant in the Post-Gates Era. Indianapolis: Wiley".

[3] Whittaker Z., 2008. "Egnyte: Using and Sustaining Enterprise 2.0 | Enterprise Alley | ZDNet". [Online]. Available at: http://blogs.zdnet.com/enterprisealley/?p=289 [Accessed 6th November 2008].

[4] Weiss A., 2007. "Computing in the Clouds, netWorker, **11(4)**, pp. 16-25.

[5] Togio J.W., 2002. "Disaster Recovery Planning: Preparing for the Unthinkable". 3rd ed. New York: Prentice Hall.

[6] Beard H., 2008. "Cloud Computing Best Practices for Managing and Measuring Processes for On-Demand Computing, Applications and Data Centers in the Cloud with SLA's". Amazon.com: Emereo.

[7] [Mills09] Elinor Mills, "Cloud Computing Security Forecast: Clear Skies", 2009 http://news.zdnet.com/2100-9595_22-264312.html

[8] http://coolwebdeveloper.com/2009/03/is-cloud-computing-reliable-enough-how-to-monitor-downtime-or-poor-performance-of-the-cloud/

[9] IsecT Ltd., 2004. "Notice Board Technical Briefing: Securing Physical Access and Environmental Services for Datacenters. [E-book] Available at:http://www.noticebored.com/NB_tech_briefing_on_datacenter_security_SAMPLE.pdf [Accessed 9th November 2008]