Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.3, No.1, 2013-Selected from Inter national Conference on Recent Trends in Applied Sciences with Engineering Applications

Concurrency Lock Issues in Relational Cloud Computing

Shri V D Garde, Aarthi Mudaliar
NCHSE, Bhopal
Sri Sathya Sai College for Women, Bhopal
vgarde@bsnl.in
art_sanj2006@yahoo.co.in

Abstract
The widespread popularity of Cloud computing as a preferred platform for the deployment of web applications has resulted in an enormous number of applications moving to the cloud, and the huge success of cloud service providers. Due to the increasing number of web applications being hosted in the cloud, and the growing scale of data which these applications store, process, and serve – scalable data management systems form a critical part of cloud infrastructures. There are issues related to the database security while database is on cloud. The major challenging issues are multi-tenancy, scalability and the privacy. This paper focuses on the problems faced in the data security of Relational Cloud. The problems faced by various types of tenants and the type of access into the database makes a rework on the security of data, by analyzing proper locking strategies on the records accessed from the database. Data security in cloud computing addresses the type of access mode by the users (for analytical or transaction purpose) and the frequency of data access from the physical location (in shared or no-shared disk mode). Accordingly, the various data locking strategies are studied and appropriate locking mechanism will be implemented for real-time applications as in e-commerce.
Keywords: Relational Cloud, Multi-tenant, two-phase locking, concurrency control, data management.

1. Introduction

Cloud computing is a step towards the notion that all aspects of computation can be organized as a utility, and it embraces paradigms ranging from Platform as a Service (PaaS) to Software as a Service (SaaS). As industry transits from in-house data management to cloud-hosted management, cloud storage has become one of the most widely acceptable infrastructure services. It caters to deploying scalable and highly available web applications in the cloud. Scalable multitenant database management systems (DBMS) running on a cluster of commodity servers are thus critical for a cloud service provider to support a large number of small applications. Unfortunately, current cloud storage services are not adequate to support applications that require guarantees on consistency especially in the presence of data updates. Even so, questions about security and privacy keep some public sector organizations from realizing the benefits of this computing model [1].
Even though relational database technologies have been extremely successful in the traditional enterprise setting, data management in the cloud must possess additional key features which traditional relational databases follow in any client-server model. The pay-per-use model and virtually infinite scalability in the cloud have enabled the infrastructure support for new application irrespective of the access configuration at the user node. As a result, deployment of modern application ideas which were not economically feasible in the traditional enterprise setting, have become possible due to the cloud computing paradigm. This facility has posed applications often having unpredictable and varying load patterns based on their popularity. Therefore, the data management systems for these applications must adapt to these varying load patterns while being highly scalable. Furthermore, due to the large scale of cloud infrastructures, failures are common, and hence, fault-tolerance, high availability, and ease of administration are essential features of data management systems in the cloud.

2. Cloud Computing – Concept

A style of computing where massively scalable (and elastic) IT-related capabilities are provided "as a service" to external customers using Internet technologies. In addition to the traditional challenges of scalability, fault-tolerance, and high availability, database management systems (DBMS) that serve these cloud platforms face the challenge of managing small data footprints of a large number of tenants with erratic load patterns – a characteristic feature of these new applications Database multitenancy allows effective resource sharing for customer applications that have small but varying resource requirements. SaaS providers like Salesforce.com are the most common use cases for multitenancy in both the application as well as the database tier. A tenant is a customer application instance with its own set of clients and data associated with the instance. Sharing of resources amongst tenants at different levels of abstraction and distinct isolation levels, results in various multitenancy models. The three multitenancy models explored in the past are: shared machine, shared process, and shared table. [3]

3. Components of Cloud Computing [6]

Most of the current clouds are built on top of modern data centers. It incorporates Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and provides these services like utilities, so the end users are billed by how much they used. Figure **1** shows a hierarchical view of component of cloud computing. Cloud computing offers attractive economies of scale, provides on-demand services just in time, gives you an infrastructure that supports telework and continuity of operations (COOP) plans, and helps green up your data center.

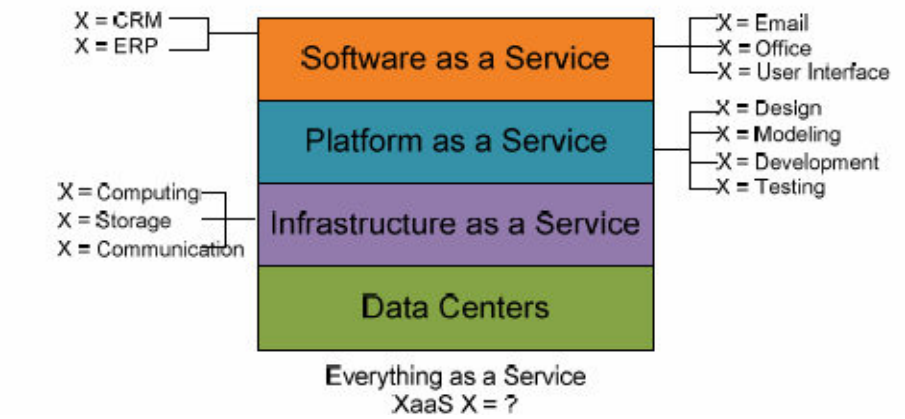

Figure 1: Components of Cloud Computing

Data Centers: This is the foundation of cloud computing which provides the hardware the clouds run on. Data centers are usually built in less populated areas with cheaper energy rate and lower probability of natural disasters. Modern data centers usually consist of thousands of inter-connected servers.

Infrastructure as a Service: Built on top of data centers layer, IaaS layer virtualizes computing power, storage and network connectivity of the data centers, and offers it as provisioned services to consumers. Users can scale up and down these computing resources on demand dynamically. Typically, multiple tenants coexist on the same infrastructure resources. Examples of this layer include Amazon EC2, Microsoft Azure Platform.

Platform as a Service: PaaS, often referred as cloudware, provides a development platform with a set of services to assist application design, development, testing, deployment, monitoring, hosting on the cloud. It usually requires no software download or installation, and supports geographically distributed teams to work on projects collaboratively. Google App Engine, Microsoft Azure, Amazon Map Reduce/Simple Storage Service are among examples of this layer.

Software as a Service: In SaaS, Software is presented to the end users as services on demand, usually in a browser. It saves the users from the troubles of software deployment and maintenance. The software is often shared by multiple tenants, automatically updated from the clouds, and no additional license needs to be purchased. Features can be requested on demand, and are rolled out more frequently. Because of its service characteristics, SaaS can often be easily integrated with other mashup applications. An example of SaaS is Google Maps, and its mashups across from the internet. Other examples include Salesforce.com and Zoho productivity and collaboration suite. Leverages the Cloud in software architecture; eliminates the need to install and run the application on the customer's own computer access type ranges from Commercial to Government level.

The dividing lines for the four layers are not distinctive. Components and features of one layer can also be considered to be in another layer. For example, data storage service can be considered to be either in as IaaS or PaaS. The figure above shows the hierarchical relationship among the different layers; however, it does not mean the upper layer has to be built on top its immediate lower layer. For example, a SaaS application can be built directly over IaaS, instead of PaaS. In the cloud computing environment, everything can be implemented and treated as a service.

4. Features of Cloud Computing

- Agility/flexibility of technology – implying a power shift
- Always on – ubiquitous
- Real time information and immediate feedback
- Provides new distribution channels, radical transparency,dynamic, ad-hoc sharing, collaboration.

5. Advantages of Cloud Computing

- Supports faster application development/deployment
- Reduces development and operating cost
- Improves overall security posture
- Increased Speed
- Increased Scalability
- Reduced Risk and Cost

6. Cloud Database

In moving a database to a cloud computing environment, there is a need to identify security requirements. In a cloud computing environment where dynamically scalable and virtualized resources are available for use over the Internet, database security is a challenge due to virtual set up and use. Security is critical due to the varied IT services that can be provided through a cloud.

Relational database management systems (DBMSs) are an integral and indispensable component in most computing environments today, and their importance is unlikely to diminish. With the advent of hosted cloud computing and storage, the opportunity to offer a DBMS as an outsourced service is gaining momentum, as witnessed by Amazon's RDS and Microsoft's SQL Azure. Such a database-as-a-service (DBaaS) is attractive for two reasons. First, *due to economies of scale*, the hardware and energy costs incurred by users are likely to be much lower when they are paying for a share of a service rather than running everything themselves. Second, the costs incurred in a well-designed DBaaS will be proportional to actual usage ("*pay-per-use*")—this applies to both software licensing and administrative costs. The latter are often a significant expense because of the specialized expertise required to extract good performance from commodity DBMSs. By centralizing and automating many database management tasks, a DBaaS can substantially reduce operational costs and perform well. [4]

The types of databases that are applied in a cloud computing environment are operational databases, end-user databases, external databases, hypermedia databases, and navigational databases. Operational databases include customer, personal, and inventory databases. End-user databases comprise of a range of data files from applications such as spreadsheets, word-processing, developed by end-users and even downloaded files. Access to a score of information from an external database is accessible for a fee from commercial online services with or without charge from many sources in the Internet. Hypermedia databases are a set of interconnected multimedia pages at a web site. It features a homepage and other hyperlinked pages of multimedia or mixed media like text, graphic, images, audio, and video. Lastly, navigational databases have objects that are found principally by following references from other objects.

Databases also differ when it comes to private or public databases in a cloud computing environment. A private cloud database is when an organization possesses its own database applications, its servers, and no other entity outside of the organization is allowed to have an access. The servers are dedicated servers to be used by the organization only. Private databases are also an assemblage of names from external sources that are used for a particular user's exclusive prospecting purposes. A public cloud database is used by any entity worldwide and no single outside entity takes ownership of the database except the cloud service provider. The data from the database could be private or public data depend on types of data and organization that owns the data.

There are three challenges that drive the design of Relational Cloud: efficient multi-tenancy to minimize the hardware footprint required for a given (or predicted) workload, elastic scale-out to handle growing workloads, and database privacy. [1], [2]

7. Data Security

Database security assessment includes regularly auditing the database servers to help security staff identify configuration issues and policy and compliance violations. An organization's database may be designed differently from other databases in order to meet business requirements. Databases can be protected through period assessment to prevent malicious attacks. The first step to take is to set up an inventory of databases in the organization. Assessment takes the form of evaluating the database in relation to the identified threats. The results of the assessment can be used to strengthen the database.

Cryptography is one of the methods used in securing a database in a cloud computing environment. It presents a range of methods for taking comprehensible, readable data, and converting it into unreadable data for the point of secure transmission, and then using a key to change it back into readable data when it reaches its destination. The goal of cryptography reaches beyond on not only making data unreadable, it also extends into user authentication which provides the user with guarantees that the encrypted message originated from a reliable source. [5]

Data security is an approach of making certain that a data is kept out of harm's way from exploitation and that access to it is duly controlled. It also helps in protecting personal data. There are different data security technologies such as disk encryption, back-ups, data masking, and data erasure. Disk encryption is about encryption technology that encrypts data on a hard disk drive. It usually takes form in either software or hardware. File back ups are used to warrant that lost data can be recovered. Data masking is the process of masking particular data within a database table or cell to assure that data security is sustained and critical customer information is not disclosed outside of an authorized environment. And, data erasure is a software-based overwriting technique which completely destroys all electronic data stored on a hard drive or other digital media to certify that no sensitive data is leaked out.

Under the multi-tenancy access to the database, two means of access and processing occur – OLAP (Online Analytical Processing) and OLTP (Online Transaction Processing). In the OLAP access, the clients query on the data from the database; while in the OLTP access, the clients make transaction into the database. The periodicity of occurrences on OLAP is more than that on OLTP; the reason being the database maintenance phase accounts on the database integrity, consistency and maintainability issues. When multiple clients(tenants) access the data at one instance of time, some access for analytical purpose and some access for transaction purpose on the same set of data. At this point of time, there is a possibility of concurrency lock problem when a record gets locked by one user, while the same record is being tried for access by another user. This is a typical environment on client-server access of any RDBMS. To maintain the data integrity in cloud database, the concurrency lock is a must to resolve. [5]

The two primary DBMS architectures used for Cloud Database are *shared-nothing* and *shared-disk.* In a shared-disk resource model, various processes in the DBMS have access to all system resources, including the data. In the shared-nothing environment, separate DBMS resources divide up the workload, responsible for its own data, memory locations, and other resource. The main advantage of shared-nothing clustering is scalability. Figures 2 & 3 depict the architecture of shared-disk and shared-nothing models. The features of both the architectures are also tabulated in Table1. [6]

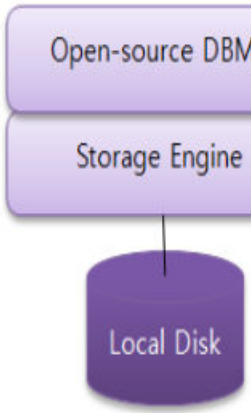

Figure 2: Shared-Nothing

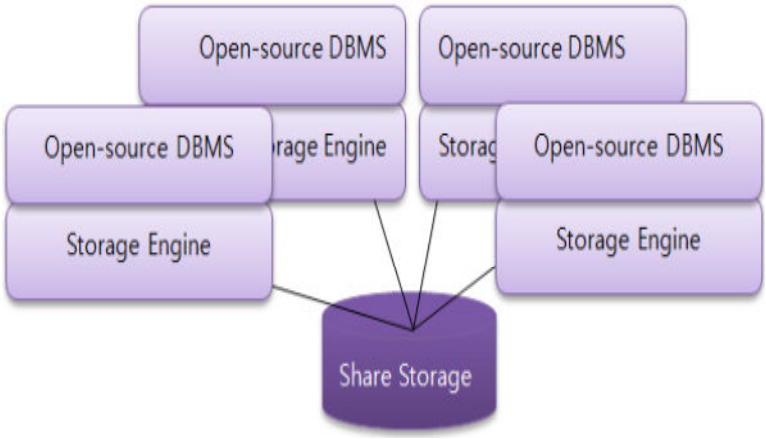

Figure 3: Shared-Disk

Table 1: Summary of features of Shared-Disk and Shared-Nothing Architecture

Shared-Disk	Shared-Nothing
Quick adaptability to changing workloads	Can exploit simpler, cheaper hardware
High availability	Works well in a high-volume, read-write environment
Dynamic load balancing	Fixed load balancing based upon the partitioning scheme
Performs best in a heavy read environment	Almost unlimited scalability
Data need not be partitioned	Data is partitioned across the cluster
Messaging overhead limits total number of nodes	Depends on partitioning, data shipping can kill scalability

In both the architectures, the concurrency locking issue has to be implemented, which requires the basic semantics of database transactions, ACID. ACID is an acronym for *Atomicity:* All parts of transaction complete or none complete, *Consistency:* Only valid data written to database, *Isolation:* Parallel transactions do not impact each other's execution and *Durability:* Once transaction committed, it remains. [3], [5]

At par with the traditional relational databases, the Relational Cloud needs to conform to the ACID semantics, as the latter is the essential feature for any relational database, across all the data in the database. To maintain the relational data concept in the cloud, concurrency control mechanism has to be implemented. Concurrency Control is the method of maintaining the consistency and integrity of relational data when transaction occurs on the same data consecutively by different users at the same time, that results in *Deadlock* situation. Under such environment, there is a possibility of the data being locked infinitely and not being accessed or used by other users at the later point of time. To resolve such situation, there are three methods that are practiced – *Time Stamp commit, Two-Phase Locking and Optimistic methods*. Depending on the type of data access, these methods are implemented. Time stamp method is used where the pattern of data access is mostly consistent and applicable for transaction purposes (OLTP). Two-phase locking method is used where the load pattern is changes randomly and applicable for both analytical and transaction purposes. The Optimistic approach is not suitable for a varying load pattern of data access.

According to the **two-phase locking (2PL)** protocol a transaction handles its locks in two distinct, consecutive phases during the transaction's execution: [5]

1. **Expanding phase** (Growing phase): locks are acquired and no locks are released (the number of locks can only increase).
2. **Shrinking phase**: locks are released and no locks are acquired.

The serializability property is guaranteed for a schedule with transactions that obey the protocol. The 2PL *schedule class* is defined as the class of all the schedules comprising transactions with data access orders that could be generated by the 2PL protocol (or in other words, all the schedules that the 2PL protocol can generate).Hence, for this study, two-phase locking method is suitable for large scale data access either doe analytical(OLAP) or transaction(OLTP) processing.

Conclusion

This is the ongoing study on the data security issues in the DbaaS (Database-as-a-service) component of Cloud computation. The data security in cloud database is required mostly on the real-time applications like e-commerce, where trading takes place online with varying workload on the data. Data locking strategy is very important aspect in dealing with real-time data more frequently. From the above discussed issue, it is argued that no particular method can satisfy all the pros and cons of the data security issues. It can be justified that the Concurrency Locking strategies have adverse effects on the Shared-Disk technique, as compared to the Shared-Nothing effect.

References

[1]. Curino, C., Jones, E., Popa, R., Malviya, N., Wu, E., Madden, S., Balakrishnan, H., Zeldovich, N.: Relational Cloud: A Database Service for the Cloud. In: CIDR. pp. 235–240 (2011)

[2]. Carlo Curino, Evan Jones, Yang Zhang, Eugene Wu and Samuel Madden, Relational Cloud: The Case for a Database Service, MIT-CSAIL-TR-2010-014, March 2010, Available in http://www-users.cselabs.umn.edu/classes/Fall-2012/csci8980-2/papers/relational.pdf

[3]. Das, S., Nishimura, S., Agrawal, D., El Abbadi, A.: Live Database Migration for Elasticity in a Multitenant Database for Cloud Platforms. Tech. Rep. 2010-09, CS, UCSB (2010)

[4]. Divyakant Agrawal, Amr El Abbadi, Sudipto Das, and Aaron J. Elmore, Database Scalability, Elasticity, and Autonomy in the Cloud, http://www.cs.ucsb.edu/~aelmore/papers/dasfaa.pdf

[5]. Gregory Chanan, Apache HBase Internals: Locking and Multiversion Concurrency Control, The Apache Software Foundation.

[6]. Sunguk Lee, Shared-Nothing vs. Shared-Disk Cloud Database Architecture, International Journal of Energy, Information and Communications Vol. 2, Issue 4, November, 2011.