International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2, Issue-4, September 2012

Scalability of Multi Tier Transactions Towards Data Confidentiality For Cloud Applications

Suvanam Sasidhar Babu, A. Chandra Sekhara Sarma, Yellepeddi Vijayalakshmi, N.V.Kalyankar

Abstract - Cloud computing provides dynamically scalable and virtualized resources as a service over the network at a nominal initial investment. Data-center works as backbone in cloud computing there a large number of servers are networked to host computing & storage needs of the users. We study and seek to improve the confidentiality of application data stored on third-party computing clouds. Scalable database services allow data query only by primary key rather than supporting secondary-key or join queries. We propose to identify and encrypt all functionally encrypt able data, sensitive data that can be encrypted without limiting the functionality of the application on the cloud. Many data intensive applications produce enormous amounts of data which travel on cloud network. As the cloud users grow, cloud architecture should accommodate movement of voluminous data to avoid data congestion in the network. Many other applications such as payment and online auction services cannot afford any data inconsistency. Cloud computing model provides benefits for private enterprise environments where a significant physical infrastructure already exists. Private cloud management platforms have been emerging in the last several years providing new opportunities for efficient management of internal infrastructures leading to high utilization.

Keywords - Cloud computing, Green computing, Data confidentiality, Program analysis, Dynamic resource allocation, Scalability, Performance, ACID, SAAS (Software as a Service).

I. INTRODUCTION

Generally in a scalable and elastic way, cloud computing infrastructures provide both cloud services over the Internet, which is called SAAS (Software as a Service), and the underlying hardware resources which is considered as IAAS (Infrastructure as a Service). The parallel machines were very expensive, a major computational change was observed as divergence in resources which categorized as distributed computing such as Network of Workstations (NOW), Cluster computing, Grid Computing etc. In present scenario, Cloud computing uses centralized resources in form of data-center. The main challenge to support transactional guarantees in a cloud computing environment is to provide the ACID properties of Atomicity, Consistency, Isolation and Durability without compromising the scalability properties of the cloud. Organizations can achieve strong data confidentiality by encrypting data before it reaches the cloud, but naively encrypting data severely restricts how data can be used. The cloud cannot perform computation on any data it cannot access in plaintext. For applications that want more than just pure storage, *e.g.*, web services that serve dynamic content, this is a significant hurdle. Cloud computing is use of scalable computing resources over Internet on a pay-as-you-go basis. It provides a cost-effective IT solution to business & scientific community. Economically the main attraction from Cloud computing are those customers only using what they need, and pay for what they actually use. Many techniques have been developed for the global optimization problem, such as expert system, control theory and stochastic model. All the methods follow the similar paradigm: i) build the application performance model and fix some model parameters by measurement or machine learning; ii) design a policy which optimizes the target utility function using the runtime input. Client devices, *e.g.* browsers, are given decryption keys by the organization to provide users with transparent data access. Of course, these devices (and users) must protect these keys from compromise. For example, a un trusted (or compromised) cloud can serve customized attack code to obtain encryption keys and decrypted data. Following the data models of Bigtable and Simple DB, transactions are allowed to access any number of data items by primary key at the granularity of the data row. The list of primary keys accessed by a transaction must be given explicitly before executing the transaction. This means for example that range queries are not supported within a transaction. Cloud supports both read-write and read-only transactions. There has been an explosion of clouds in recent years, such as Google AppEngine Amazon Web Services(AWS) and Microsoft Azure Instead of maintaining the private hardware infrastructures, web application providers can closely follow their demand curves and minimize their cost by renting cloud resources in a pay-as-you go and scalable manner. This is very appealing for mostly small and medium companies which cannot afford expensive hardware devices or try to cut down their expense. Our goal is to automatically infer the right granularity for data encryption that provides the best tradeoff between robustness and management complexity. To this end, we partition the data into subsets, where each data subset is accessed by the same group of users. We then encrypt each data subset using a different key, and distribute keys to groups of users that should have access (based on the desired access control policies). For this, we replicate data items and transaction states to multiple LTMs, and periodically checkpoint consistent data snapshots to the cloud storage service. Most important of all, this model is well scalable with multiple resource types, tiers and transaction classes. However, it's not trivial to obtain the optimized configuration scheme as a result of the infinite parameter space. The scalability of our transactional database service using a prototype implementation, the data models of Bigtable and Simple DB, transactions are allowed to access any number of data items by primary key at the granularity of the data row. The list of primary keys accessed by a transaction must be given explicitly before executing the transaction. This means for example that range queries are not supported within a transaction. Cloud supports both read-write, read-only transactions and its trends in computing history shown in figure1.

Manuscript Details Received on August 2012.

Dr. Suvanam Sasidhar Babu, Professor, Department. of CSE,Sree Narayana Gurukulam College of Engineering, Kolenchery, Ernakulam (Dt.), Kerala, India.

Prof. A. Chandra Sekhara Sarma, Professor, Department of CSE, Aurora Engineering College, Bhongiri, Hyderabad (A.P.), India.

Yellepeddi Vijayalakshmi, Research Student, Holy Mary Institute of Technology & Sciences, Keesara, Hyderabad (A.P.), India.

Dr. N.V.Kalyankar, Professor & Principal, Yashwant Maha Vidyalaya, Nanded, Maharastra, India.

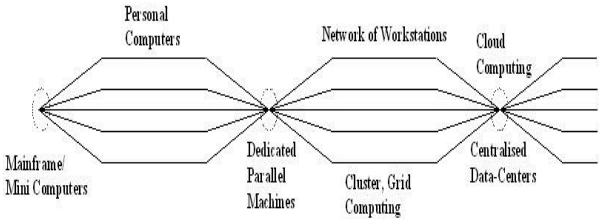

Figure1: Trends in computing history

II. RELATED WORK

Some researchers identify Cloud computing as virtualization of previously existing datacenters while some others nominate data-centers as backend resources of newly adopted Cloud computing paradigm. In some cases, some of the applications even cannot meet QoS needs when the performance gets globally optimal in the data center. However, this paper focuses on the local optimization problem for the performance of a single cloud application. Our overarching goal is to improve the confidentiality of application data stored on the cloud. We assume that the third-party computing cloud provides service availability according to service level agreements, but is otherwise un trusted. Our solution to improving data confidentiality on the cloud calls for end-to-end encryption of data by its owner (the organization) and its consumers (the users). The relational data model, typically implemented via the SQL language, provides great flexibility in accessing data. It supports sophisticated data access operations such as aggregation, range queries, joins queries, etc. An alternative approach to implement a cloud database is to run any number of database engines in the cloud, and use the cloud file system as shared storage medium. Each engine has access to the full data set and therefore can support any form of SQL queries. On the other hand, this approach cannot provide full ACID properties. A substantial time delay may be observed if radio frequency reader and cloud resources are physically located at long distance. A common trend of centralized resources at the Cloud provider's location is present in almost all existing Cloud computing architectures leading to increase in latencies. In addition, we also have some related work on the optimization problem executes a combinatorial search technique over the space of all possible configuration schemes and gets an optimized configuration scheme which makes the data center globally optimal. We present a technique that enables clients to store and use their keys safely while preventing cloud-based service from stealing the keys. Our solution works today on unmodified web browsers.

III. OVERVIEW

Our overarching goal is to improve the confidentiality of application data stored on the cloud. We assume that the third-party computing cloud provides service availability according to service level agreements, but is otherwise un trusted. a typical e-commerce application consists of three classic tiers: a front-end web server which handles all simple and static transactions, an application logic tier processing the requests that front-end server submits, and an backend database server storing and providing the relevant data about the application. We prefer to focus on achieving linear scalability specifically for Web applications, such that any increase in workload can be accommodated by provisioning more servers. Our solution to improving data confidentiality on the cloud calls for end-to-end encryption of data by its owner (the organization) and its consumers (the users). In this paper, we concern ourselves with the data persistently stored in the databases. Our techniques apply to both traditional relational databases on the cloud. Cloud provides durability for transactions by check pointing data updates into the cloud data service. Our approach might not be able to encrypt all application data. For example, the message board might want to display the average age of the users in the system. To compute this, the application must access the date-of-birth field in the database, calculate the age of each user by subtracting the DOB with today's date, and then perform a summation to calculate the average. Users store and retrieve (encrypted) data on the cloud, and they obtain their keys from the organization. Data is encrypted and decrypted locally. Therefore, protecting decrypted data and user keys is critical. Desktop applications can protect keys locally using standard techniques. There is a load balancer communicates to web servers and shared resources.

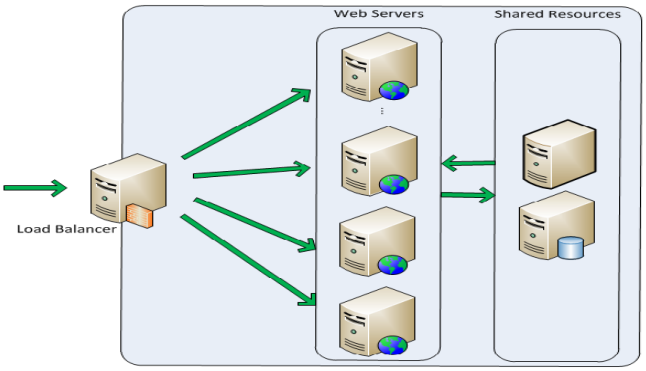

Figure: Horizontally Scaled Web Servers

IV. PROPOSED ARCHITECTURE

The database (running on the cloud) is encrypted, but keys are not revealed to the cloud. The keys are stored by the organization that "outsources" its application and user data to the cloud. To fetch data from the cloud, the user first contacts the organization to get the appropriate key(s), and then sends the query to the cloud to fetch the data. The large number of service requests to fulfill the demands of millions of users will broaden the latency problem. Cloud service provider physically may be far away from the clients, compelling data to travel from several mediums and network equipments, there by imposing a time delay in getting Cloud services. Each transaction contains one or more sub-transactions, which operate on a single data, item each. The application must provide the primary keys of all accessed data items when it issues a transaction. The application model which is used to estimate the performance metrics of all the transactions in a multi -tiers cloud application. In our study, we use layered queuing model rather than regular network model proposed in previous research. In the proposed Cloud architecture data-centers work in master-slave paradigm. Nearest data-centers form a computing zone and users may opt for treating their instances in multiple zones. The main entities involved in proposed architecture are:

1) Master /Slave Data-Center: Master data center is located at Cloud provider's administrative premises. User's accounting on pay-as-you-go basis is completed here. Slave data-center are geographically scattered to serve user's requests in minimum physical distance.
2) Users/ Brokers: Users directly communicate or via brokers submit requests which automatically reach at master data-center. Master data-center creates user instance at appropriate slave data-center considering minimum latency.

3) Service Level Agreements (SLAs): Quality of Service (QoS) and pricing negotiations are settled through SLAs. Master data-center scans SLA each time to host needs of the users.

V. SYSTEM DESIGN

The design of the cloud computing to guarantee the atomicity, consistency, isolation and durability properties. Each of the properties is discussed individually. We then discuss the membership mechanisms to guarantee the ACID properties even in case of transaction failures and network partitions. We apply our techniques by modifying the application runtime environment (for our evaluation, the PHP interpreter) to tag information associated with different database fields, and propagate them throughout the application logic. To find functionally encryptable data, one can perform static or dynamic program analysis. We use a *dynamic* approach, based on a set of training queries that exercise the application. Given a set of training queries that are representative of application-to-database queries, we modify the interface between the database and the application runtime to automatically extract meta-information as data is re-turned from the database. Note that these modifications are application independent and only need to be performed once for a particular programming environment (such as PHP, Python, or Java). We use this to build a table that maps the signatures of specific queries to fields accessed in the DB. The private cloud web server is shown in the figure 3.Data is sent from the database to the application in response to application queries. As each piece of data is retrieved from the database, it is tagged with a *field number* that corresponds to the field read. Field numbers are positive integers that uniquely identify a field in the DB. We aggregate all warnings to produce a unique list of field numbers that tagged non-encryptable data. Using the previously produced table (which maps field numbers to field details in the DB), we produce a list of all database fields whose values must be exposed in plaintext for the application to function properly.

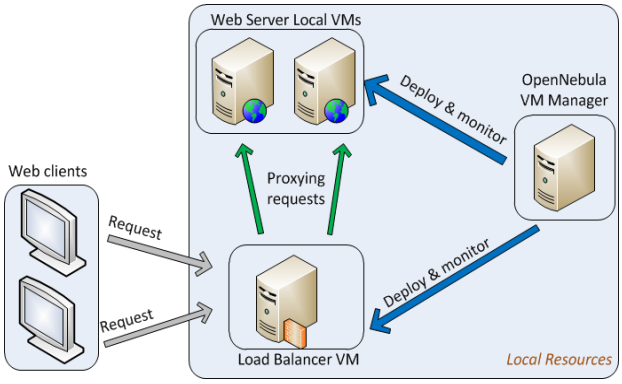

Figure3: Private Cloud Web Server

The increasing popularity of the cloud computing approach is driven by the advancement of the virtualization technology. It allows different logical machines to share the same hardware but to run isolated from each other. As a result physical hosts can be utilized better and the computing resources can be allocated easily in a very flexible manner. These isolated running operating systems (OS) are called virtual machines (VMs). A number of virtualization software solutions are available. They usually include a hypervisor also called virtual machine manager (VMM). The hypervisor assign resources to the VMs and lets them operate as if they were running on different machines independently. Virtualization software abstracts the hardware and provides flexible and agile way to control it, it is not a cloud itself. A layer that controls it has to be deployed so it can access the entire infrastructure within a data centre and to manage all the resources. This layer is called cloud management platform.

VI. CONCLUSION

The architectural design of Cloud computing is still in its infancy and needs exploration towards the efficient utilization of large scale IT infrastructure. Deploying and managing cloud effectively when good performance is required is hard when the cloud administrator does not have full control over the underlying hardware components. The cloud management platform provides centralized point for managing hosts with enabled virtualization. Many Web applications need strong data consistency for their correct execution. However, although the high scalability and availability properties of the cloud make it a good platform to host Web content, scalable cloud database services only provide relatively weak consistency properties. Data confidentiality is one of the key concerns that prevent organizations from widely adopting third-party computing clouds. Encrypted data on the cloud prevents privacy leakage to compromised or malicious clouds, while users can easily access data by decrypting data locally with keys from a trusted organization.

VII. FUTURE WORK

The database tier is typically difficult to replicate on the fly and may suffer from the problem of data synchronization and consistency. And application providers even just manipulate the database service which clouds provide instead of building their own private database. If the impacts are really great and non-negligible, we have need to devise our model to adapt these features. An intuitive reasoning for such a classification is more helpful for the developers in later implementing encryption and decryption functionality in the applications. In this way it will be able to react very quickly to any increase in the load and manage all the requests in a reasonable time. Different approaches could be followed for such a provisioning - this could either be based on the CPU/memory usage of each VM based on the problem sizes of the requests handled by the workers. Recovering from a failure only causes a temporary drop in throughput and a few aborted transactions. Recovering from a network partition, however, may possibly cause temporary unavailability of Cloud, as we explicitly choose to maintain strong consistency over high availability.

VIII. ACKNOWLEDGEMENT

The authors would like thank to our management of Sree Narayana Gurukulam College of Engineering, Kadayiruppu Executive Director T.A Vijayan, Director Dr. C.E. Krishnan and Principal P.N.Joshi and all colleagues for their precious collaboration for providing facilities and system implementation.

REFERENCES

[1] Chetna Dabas and J.P Gupta, "A Cloud Computing Architecture Framework for Scalable RFID," *Proceeding of the International Multi* Conference of Engineers and Computer Scientists (IMECS 2010,) Hong Kong, Vol 1, March 2010, pp 217-220.

[2] Google AppEngine: http://code.google.com/appengine/.

[3] Amazon Web Service: http://aws.amazon.com/.

[4] Eucalyptus: http://www.eucalyptus.com/.

[5] Windows Azure: http://www.microsoft.com/windowsazure/. CLAUSE, J., AND ORSO, A. Penumbra: automatically identifying failure-relevant inputs using dynamic tainting. In *Proc. of ISSTA* (2009).

[6] MOLNAR, D., AND SCHECHTER, S. Self Hosting vs. Cloud Hosting: Accounting for the security impact of hosting in the cloud. In *Proceedings of the Ninth Workshop on the Economics of Information Security* (2010).

[7] M. Armbrust, A. Fox and R.Griffith etc. *Above the Clouds: A Berkeley View of Cloud Computing*. Technical Report No. UCB/EECS-2009-28. University of California at Berkley, USA, Feb. 10, 2009.

[8] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, Volume 41, Number 1, Pages: 23-50, New York, USA, January, 2011.