Secure Cloud WLAN using Dynamic Placement and Migration with a Cloud Name Resolution Protocol

Rowida Alharbi, Khaled Elleithy, Alaa Alharbi, and Wafa Elmannai
Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06604, USA
roalharb@my.bridgeport.edu, elleithy@bridgeport.edu, aalharbi@my.bridgeport.edu, welmanna@my.bridgeport.edu

Abstract— Cloud computing systems have displayed many intriguing possibilities for the end users. Several cloud computing systems are centralized, and might lead to a system’s overload. This situation influences security and leads to the blockage within the network especially when important routers and servers are malfunctioning. Over the past few years, there has been a great development in Peer-to-Peer (P2P) cloud computing. This paper introduces a transportation protocol that is a less complex in interaction between the servers in transporting the session state data between one another. The proposed methodology has been implemented using the Visual Studio platform. The platform has been extensively tested and it has demonstrated promising results.

Keywords— Cloud computing, Dynamic, Mitigation, Scale, data, security.

I. INTRODUCTION

Web computing has enabled the plug and play concept of added services in a network. The virtualization of the infrastructure paved the way to the Service Oriented Architecture (SOA). Many services’ providers benefited from the use of SOA. This benefit leads to improve the services based architecture within the cloud computing domain. The main feature of cloud computing is to use services as needed, and then the end clients can choose certain services and pay per usage. The advantages of this approach is cost reduction and scalability. The flexibility and utilization leads to higher efficiency in the overall system. It is expected that such services will be implemented in the cloud computing systems market. It is also estimated that the cloud computing market is going to reach $225 billion dollars in 2015 [1, 2 and 3]. Although the above information is promising, it raises many concerns and issues. Currently, there are no predefined protocols and regulations for the implementation of such cloud computing systems’ architecture. Thus, the system might be subject to security vulnerabilities. Business regulations, supplier management regulations, control, and other vulnerabilities are under question. Data migration is one of the top most concerns since many companies want to implement the cloud system. However, the existing standards and guidelines with regards to the cloud computing security are very limited in scope [4]. The focus of this paper is to provide a secure methodology for the migration of the cloud data through a wireless LAN network by using a hierarchy concept while inheriting all the added advantages of the cloud system.

A. Principal Cloud Computing Components:

Physical hardware, software and datacenters are not necessarily required in cloud computing at the bottom layers. It is the software that has been virtualized, and is available when needed from service providers [5].

Amazon, Goggle, Microsoft, and IBM are the major sources of cloud computing. Hadoop, Eucalyptus, Enomaly ECP, Sector and Sphere, Abiquo and MongiDB are examples of these services[6].

B. Cloud Computing’s Services

The names of the three types of services provided are:

- Structure of the service
- Bottom as a service
- Software as a service [6]

However, large datacenters provide efficient computing that requires large-scale heterogeneous components [7, 8]. Its scalability allows more efficiency and more power when it is needed by the consumer [9]. Applications are also distributed to systems. Cloud computing has established the theory of loose-oriented services which creates new opportunities in free computing services [5].

Section II of this paper provides related work. The proposed methodology is described in Section III. Section IV focuses on the conclusion.

II. RELATED WORK

A. JXTA Technology

In 2011, JXTA was introduced as a P2P platform that became available in Apache open-source by microsystems [10]. In JXTA, peers must be competent enough to find out about one another, and must be arranged into peer groups. They should look for similar networking sources and communicate with each other. In addition, peers must monitor one another. The Bottom does not need any computer language or operating system. Bottom must not use a network transmission or a topology and it also does not need authorization, security or any model of encryption.

The whole JXTA model relies on a small number of protocols. There are six protocols used in JXTA: 1) Peer Resolver Protocol (PRP); 2) Peer Discovery Protocol (PDP); 3) Peer Information Protocol (PIP); 4) Pipe Binding Protocol
(PBP); 5) Peer Endpoint Protocol (PEP) and 6) Rendezvous Protocol (RVP) [11, 12].

B. **DHT Technology**

Distribution Hash Table (DHT) is a decentralized method that provides the same functionality as a hash table. Thus, the value of any node can be retrieved by its key [13, 14]. The responsibility of preservation of the mapping using the keys is divided among the nodes. This feature allows the DHT to cover a huge number of nodes with a certain level of scalability. Complex systems such as services of a domain name, web caching, multicast, and more can be built using the DHT’s frame. Independence and distribution are emphasized by DHT where the nodes mutually come from the network without vital synchronization. Also, the reliability is provided by DHT regardless the status of the node. The scalability is also offered the hashing feature that is used in the DHT.

C. **Mapreduce**

It is a transmitting design that is utilized to process unstructured data across the clusters in the parallel computing. It was introduced on Google in 2004. The parallel computing method is abstracted by Mapreduce and has basically two tasks to do. The first task is to map the work over the nodes and reduce the collected works to be resolved into a single result, and to put it in a simple form. Its job is to decompose the assigned task and give a brief summary of the results. Results are available through Map which distinguishes the task into separate and multiple subcategories of tasks. This approach shows the results made by processing the decomposed subcategories of tasks.

Each node in the network is expected to submit its results along with the status of an assigned task. Synchronization takes place in the network. The task will be reassigned in case of late submission by any node. It is a significant function because it is a free source library. Developers do not need to worry about programming to monitor the task, conduct the failure node, and the communication of intra-cluster.

The above three technologies are extremely valuable in cloud computing. However, their restrictions raise security concerns. Furthermore, they do not offer a reasonable data integration as they should.

III. **PROPOSED SOLUTION**

In this paper, we provide a methodology for deploying secure cloud architecture. The global cloud servers on a network securely communicate and pass session state data among each other as needed, which is shown below in Figure 1. This design improves scalability because Local Cloud servers can share multiple global cloud servers which eliminate a single point of failure. Furthermore, if a load balancer erroneously or intentionally redirects a user to a different cloud server that is attached to a separate global cloud server, the user’s session state will be requested from the Global Cloud server that served the user’s previous request. Security is also improved as local clouds can be configured to encrypt session data while sharing session state.

![Figure 1: The Model of Secure Communication between Global and Local Cloud servers](image)

Data transfers between the global clouds and the local clouds remain unencrypted, but eavesdropping attacks can be eliminated by keeping global cloud servers and linked local cloud servers in trusted networks or on the same computer. IPv6 protocol is used to add scope to an address within a port, as well as a registered resource. Scope can be added to further identify the services/resources that are listening on a port. A group of resources connected in the network using the same scope is known as a Cloud. Clouds are similar to IPv6 scopes but there are subtle differences. In a cloud, we can use the same resources in between all the servers and machines, where only in IPv6, we can have registered servers or machines that use the services provided to them. The registered machines can only communicate in between each other to make it more secure within a cloud network. Let’s consider the network is divided into a hierarchy where multitude of machines are registered into a common cloud known as a global cloud. The individual machines registered in the global cloud can be called their own local clouds. That means each individual machine is registered and connected to the network has a Local Cloud. The Global Cloud has the most administrative rights and allows applications to communicate over the network. Local clouds allow applications behind a firewall or connected through a common subnet to communicate.

Thus, we can solve the problem of data security as well as take care of the problem of redundancy. The overall network load might overwhelm the network. By using data migration and dynamic placement, we can have a control and create a methodology for load balancing on the network. Mimicking P2P network can help in implementing a load balanced cloud network. Implementation of peer-to-peer design using IP6 protocol can be easily incorporated in the proposed framework.
The implementation of the data migration between multiple machines that have their own local and global clouds, requires us to design a methodology to call on all sockets’ synchronously to look up and count the results of our search terms. We use the following functional calls:

- WSALookupServiceBegin to begin the count and return a handles.
- WSALookupServiceNext to retrieve a set of clouds matching the scope. Call this function until the application has retrieved all the clouds.
- WSALookupServiceEnd to finish the count.

Figure 3 and Figure 4 provides snapshots of the implemented framework.

To show the readily available clouds in the network, we will have to generate a mechanism to show the scope of the methodology while we are searching in these clouds. This way we have complete access and control over the whole system. The implemented framework is based on the three search terms provided in Table 1.

### TABLE 1: THE MAIN TERMS OF THE WLAN COMMUNICATION

<table>
<thead>
<tr>
<th>Any</th>
<th>This will help in showing clouds that are connected.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Global</td>
<td>This will present the computer’s connection to global cloud.</td>
</tr>
<tr>
<td>Local</td>
<td>This will exhibit computer’s connection to any of the local cloud.</td>
</tr>
</tbody>
</table>

![Figure 2: Control Flow Diagram of the Proposed Methodology](image1)

![Figure 3: Server Startup](image2)

![Figure 4: Application to name various nodes in the cloud](image3)
We can encapsulate this functionality by creating a common collector class that implements two basic functions like the IEnumerable with another embedded class that implements IEnumerator. This allows a loop to count the clouds. We also will have some other functionalities that are designed into the application for passing the various data structures that will be built. For example, the Reset function will pass a complicated set of data structures into the WSALookupServiceBegin function. The LUP RETURN_ALL parameter tells the subsequent call to WSALookupServiceNext to fully populate the PNRPCLOUDINFO data structure. MoveNext uses the WSALookupServiceNext to retrieve the results. In this case, true or false is returned depending on whether a result has been returned.

IV. CONCLUSION

This paper provides a methodology for migration of the cloud data on a wireless LAN network using a hierarchy concept while inheriting all the added advantages of the cloud system and keeping it secure. A class hierarchy is implemented to improve the functionality of the cloud system. Extensive test results show that the implementation is highly secure since no entity has a complete copy of the data required via a node. In case that any of the nodes is compromised, a section of the global cloud is disconnected from the network while the majority of the network continues to be fully connected and highly secure.

REFERENCES