Remote Job Servicing in Partitioned Public Cloud

Rohit Kumar
Department of Computer Technology
Priyadarshini Institute of Engineering and Technology(PIET)
Nagpur, Maharashtra, India, 440019
Email: torohitmail@gmail.com

Abstract— Cloud computing is a way of offering different "things" as services. Public cloud is a popular deployment model of cloud computing. With its huge infrastructure, it offers its services to customers all over the world. Managing such a large system may be quite difficult. Thus, partitioning may resolve this issue by offering you a management approach to deal with this massive infrastructure quite easily. But all the partitions also require to work in a co-operative manner to have the optimum results. This paper focuses on this co-operation issue and discusses the basic public cloud concepts, public cloud partitioning concepts along with it.

Keywords- Cloud computing; Public cloud; Data-center; Partitioning; Partition interoperability; Authentication

I. INTRODUCTION

Cloud computing is not a technology itself but it is so popular that it seems like a technology that has changed the way of using resources. Cloud computing simply refers to the use of resources on the fly, without any extra setup requirement. It allows to make computations from anywhere at any-time. Resources are provisioned on demand i.e. it is like a scenario that cloud provider is asking you- "Tell me your requirement, I will manage for you". General cloud service concept model is shown in Figure 1.

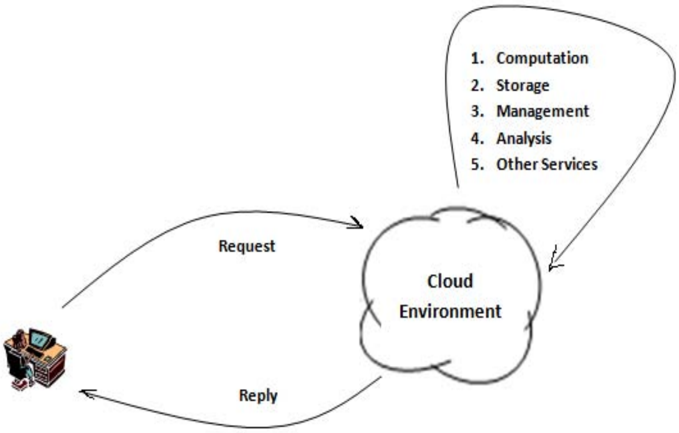


Figure 1: General Cloud Concept Model

User specifies his requirements and within seconds, the whole setup is ready. This has become possible due to the large pools of resources managed at cloud provider side. Since users have no complex setup at all, then there are no maintenance efforts and cost. User specifies his service requirement in request form. The cloud-owner handles different service requests and processes them accordingly. After that, the relevant reply or result is provided back to user. There are a number of services that are provided by a cloud provider. For example, Amazon provides its cloud services in many categories, like- Compute & Networking, Storage & Content Delivery, Database, Deployment & Management, Analytics and App Services [1]. So, it has become very easy to fulfill different requirements with no tension and on very promising prices because user is required to pay only for the fraction of resources that he uses. Thus, it can be said that computation has become a utility, like water, electricity, etc. The outer simplicity factor has made the cloud very sophisticated from inside. Cloud Service Provider (CSP) is now required to manage all the resources and related issues. Some of popular issues are shown in Figure 2.

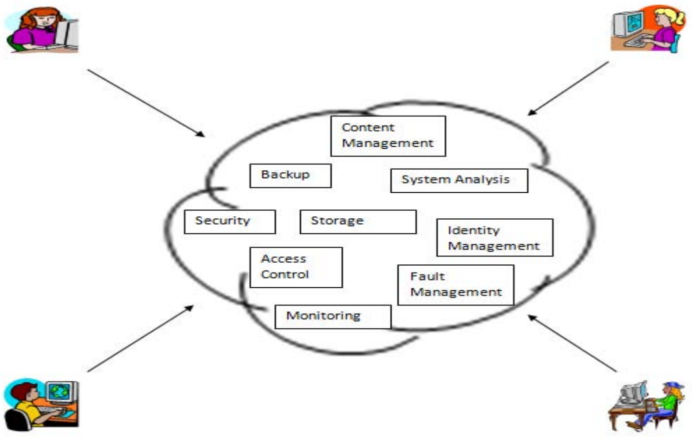


Figure 2: Cloud Management Issues

These internal cloud issues need to be maintained to cope up with Service Level Agreement (SLA), a contract between the cloud service provider and the customer (user) [2]. Cloud can be deployed in three forms:

A. *Public Cloud*

It is the most general form of cloud computing models. In this model, any user may access the services from the cloud freely or on paid basis, depending on the type of service. Most of the users go for a public cloud, when they need a cloud service. According to Gartner, a world famous research and advisory firm, end-user spending on public cloud services is expected to grow 18% in 2013 to $132 billion and almost $250 billion by 2017, including cloud advertising [3].

B. *Private Cloud*

It is also a popular form of cloud models due to security reasons. In this model, the user or organization maintains its own cloud within or outside its premises and the services offered remain dedicated for the user or organization itself. Since security is a main concern in public cloud environment, so, most of the users go for this choice due to the sensitivity of their data.

C. *Hybrid Cloud*

It is a combination of above two models. It allows some of its services open for all users, worldwide and restrict some services only for the selected users or an organization. Thus, it enjoys the benefits of both, public and private cloud models. Top three benefits that drive the companies to go for a hybrid cloud choice are [4] - The ability to leverage both public and private cloud, enhanced agility and overall cost savings.

In the following paper, Section 2 discusses about partitioning and authentication related work, Section 3 describes the proposed work. Section 4 does some analysis work and Section 5 concludes the paper and presents the future work.

II. RELATED WORK

A. *Existing Public Cloud Partitioning Model*

Gaochao Xu et al [5] gave a model that supports the concept of cloud partitioning for load balancing in public cloud. Partitions are done on geographical location basis. The general architecture is shown in Figure 3. The whole cloud is divided into some partitions. A main controller is responsible to control all these partitions. After partitioning, the process of load balancing starts. Whenever a job comes into the system, it goes to main controller. Main controller assigns this job to local partition if its load status is idle or normal i.e. there are no jobs or some jobs getting executed inside partition. If local partition is overloaded then some other partition is chosen randomly and its load status is checked and so on. A looping process dedicatedly works to find the suitable partition for the assignment of newly arrived job. After getting a suitable partition, the job is assigned to that partition balancer and the balancer assigns the job to appropriate node based on load balancing strategy, chosen by balancer. The same process goes on for next coming jobs. Main controller collects load information from all partition balancers and updates its load status table regularly.

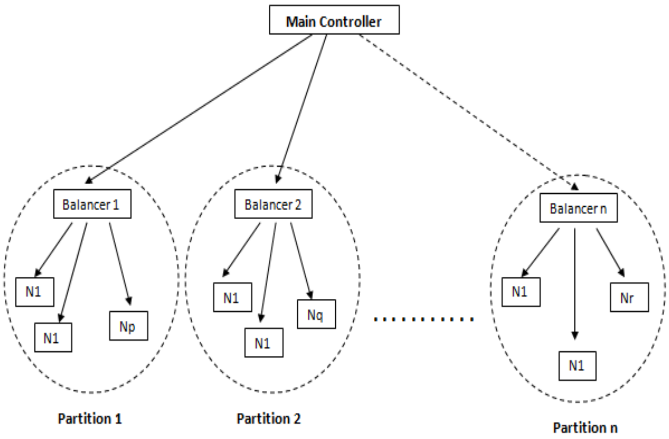


Figure 3: System Architecture

In turn, balancers gather load information from respective nodes and choose the best suited strategy for job assignment to nodes. This evaluation of each node's load status is very important. The first task is to define the load degree of each nodes. The node load degree is dependent on a number of parameters. These parameters may be static or dynamic. Static parameters are believed to remain constant with time while the dynamic parameters change their values time to time. The static parameters include the number of CPU's, the CPU processing speeds, the memory size, etc. Dynamic parameters are the memory utilization ratio, the CPU utilization ratio, the network bandwidth, etc [3]. Node load degree results are input into the Load Status Tables created by the cloud partition balancers. Each balancer has a Load Status Table and refreshes it each fixed period T. The table is then used by the balancers to calculate the partition status. Each partition status has a different load balancing solution. When a job arrives at a cloud partition, the balancer assigns the job to the nodes based on its current load strategy. This strategy is changed by the balancers as the cloud partition status changes [3].

B. Authentication Mechanism

Authentication simply means identification plus verification [6]. In Identification, an entity claims an identity and verification is the process of checking this out. An identity may be claimed by anyone but verification becomes very important. There are three main types of authentication in distributed system:

- Message Content Authentication (verifying the message content)
- Message Origin Authentication (verifying the sender of message)
- General Identity Authentication (verifying any entity's identity)

In distributed system, message passing is a common communication form [7]. Due to communication compromise, there may be several threats [6]:

- *Passive Threats:* These threats do not cause harm to the system directly e.g. eavesdropping of messages.
- *Active Threats*: These threats do cause harm to the system directly e.g. insertion, deletion of messages, modification in data, relaying old messages, etc.

Detecting passive attacks is difficult in comparison to active attacks, since these make no changes in data actively. In an open or hostile environment, mutual authentication is a need. In mutual authentication, both parties verify each other's identity. In inter-domain authentication naming and trust issues are need to be addressed. Naming issue specifies that each communicating entity should be uniquely identifiable across domains. Trust specifies that there should be some entities across domains that can be trusted [6].

C. Cryptosystem

A cryptosystem is a pair of algorithms that take a key and convert plaintext to ciphertext and back. Plaintext is what you want to protect and the ciphertext is the encrypted or unintelligible form of plaintext. Now-a-days, encryptions are totally computerized and operate on bit level. The design and analysis of today's cryptographic algorithms is highly mathematical. The properties of a good cryptosystem should be [8]:

- There should be no way short of enumerating all possible keys to find the key from any reasonable amount of ciphertext and plaintext, nor any way to produce plaintext from ciphertext without the key.
- Enumerating all possible keys must be infeasible.
- The ciphertext must be indistinguishable from true random values.

There are two main classes of cryptosystems [6]- symmetric cryptosystem and asymmetric cryptosystem. Symmetric key cryptosystem uses a single key for both, encryption and decryption purposes. Asymmetric key cryptosystem uses a key pair. One key is used for encryption purpose and only the other one can be used for decryption purpose.

III. PROPOSED WORK

The proposed work focuses on various topics - the architecture and its core components, job assignment strategy, best partition searching, load calculation at different levels, performance analysis, partition load balancing strategies, load information updation and relevant issues, inter-partition operability and energy efficiency.

A. The Architecture

The general architecture of a partitioned public cloud is shown in Figure 4, an extended one of [3]. The core elements of the architecture are:

- *User:* It is the customer, want to access cloud services. All cloud requests originates from it. Only one user is shown in Figure 3 which is serviced by Partition 1 but in practice there will be hundreds of thousands of users in different geographical locations.
- *Partition:* A partition is an abstract logical view of many underlying components. It functions as a separate unit of it could be considered as an independent part of whole cloud environment. Partitions may be heterogeneous in nature.
- *Main Controller:* It is the most important part of a partition. It is responsible to handle local cloud requests and forwarding them to one of the balancers. It also maintains the load records of its balancers.
- *Balancer:* A balancer is a partition level entity. It gets forwarded cloud service requests from respective main controller and assigns the job to one of the underlying nodes. It is also responsible to maintain the load records of respective nodes. Number of balancers may vary from partition to partitions.

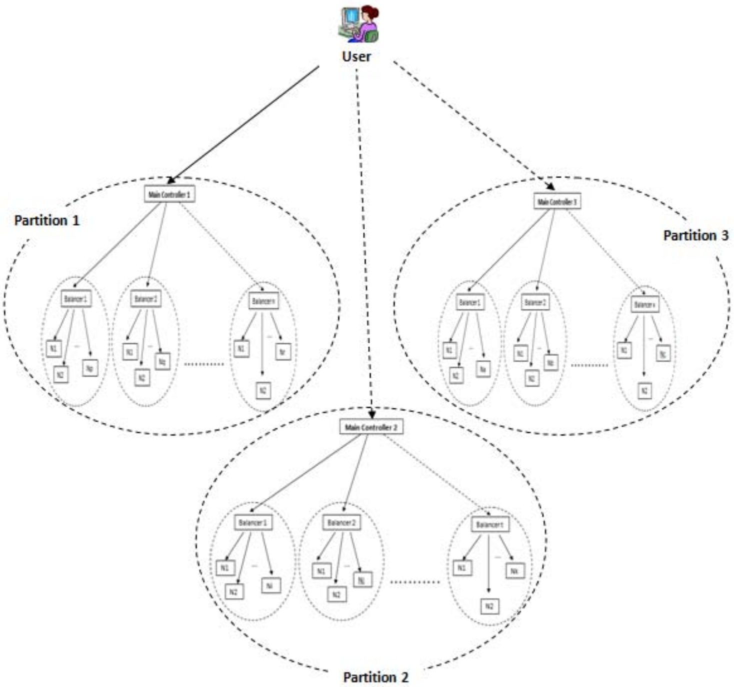


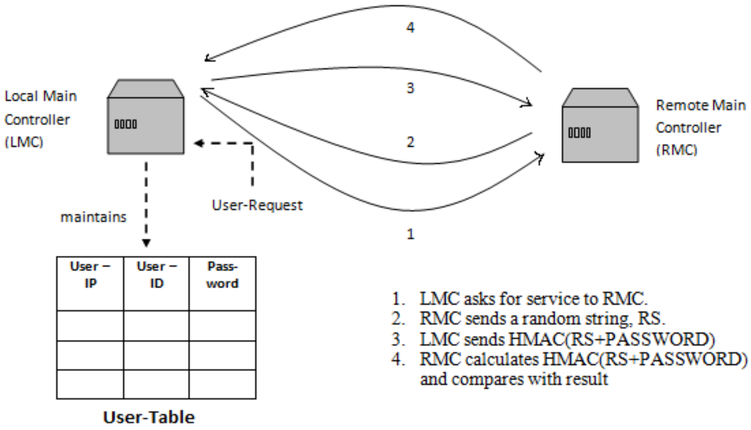
Figure 4: Partitioned Cloud Model

- *Node:* It is the lowest level entity in the proposed architecture. It is responsible for the actual processing and computation. There may be different number of nodes under different balancers.

Cloud users request for their cloud services and get responses from the local partition. If local partition is too busy to handle the request then, request is transmitted to another partition with the help of local partition. Request is handled firstly by local partition's main controller. If local partition main controller status (partition status) is not overloaded then the job is transferred to one of local balancers depending upon their load statuses. Main controller keeps track of their balancer's utilization i.e. load information. Different balancers update their status utilization information time to time at main controller by sending load messages. A load message consists of a Balancer-ID and the current Utilization Percentage. Subsequently, each balancer maintains its local nodes load statuses. Local nodes also maintain their current load information by sending load packets, containing Node-ID, Optimal Load and Optimal Utilization Percentage.

B. *Inter-Partition Operability*

- *Need of Inter-Partition operability:* Cloud computing is an internet enabled concept. So, internet traffic characteristics get applied to it. Sometimes, internet traffic surprisingly gets high and needs special attention to handle the situation. Such traffic sparks may cause any system or cluster of systems to get overloaded. In case of partitioned public cloud, it may be the case that local cloud partition becomes unable to handle any further load, then, to handle further local requests, it has to ask for help to other partitions. Local main controller will send request-handling request to the some other partition based on topology, load and cost. But, since each user is registered only with its local partition, then, how the remote partition will get to know that whether a service request is authentic one or not? To get out of this problem, some kind of authentication mechanism should be there, to discriminate valid requests
- *Need of Authentication:* Load Balancing focuses on the concept to share the load to get the work done fast. So, whenever a local partition gets overloaded, it makes sure to provide the request servicing from a remote partition. To get serviced from a remote partition, local requests have to pass through some kind of authentication test. But, since users are registered with only local partition, thus, they can't prove their authenticity with remote partition. The other way remains is to make an authentication with the help of local main controller. It is supposed that all partitions main controllers are authenticated to each other topologically and each directly connected main controller pair share a common secret key. So, we are required to make some inter-partition authentication communication to let the request serviced.
- *Inter-Partition Authentication process:* Process looks like, as shown in Figure 5:



1. LMC asks for service to RMC.
2. RMC sends a random string, RS.
3. LMC sends HMAC(RS+PASSWORD)
4. RMC calculates HMAC(RS+PASSWORD) and compares with result

Figure 5. Inter-Partition Authentication

In Figure 5, Local Main Controller (LMC) authentication process is shown. Each Partition maintains a table containing the relevant details of topologically connected partitions. When a request comes to Local Main Controller (LMC), it verifies the request by making some authentications steps. Each partition main controller maintains a table, containing the cloud user's IP address, User-ID and Password. IP helps in identifying user's location. IP may get changed, each time user makes a log-in. User-ID and Password provides base for user-authentication. If partition is not overloaded, request is handled locally. Otherwise, a session with suitable Remote Main Controller (RMC) is established, depending upon RMC's utilization and transfer cost, to get the request satisfied. The detailed process is given below:

```
1.  User makes a request with username and password.
2.  Credentials are checked and request is considered for service by LMC.
3.  if (NOT_OVERLOADED)
                handle request locally.
    else
                select  a RMC based on topology, load and  transmission cost.
                LMC asks for service to RMC.
                RMC sends a random string, RS.
                LMC sends HMAC (RS+PASSWORD)
                RMC calculates HMAC (RS+PASSWORD) and compares with result.
                if (MATCHED)
                    Grant remote service permission.
```

```
            else
                Service refused and abort.
    4. LMC connects user request to RMC.
```

To set-up a remote connection, we have to authenticate the remote request. This can be done, as shown in Fig. 5, The Local Main Controller (LMC) sends a remote service request to Remote Main Controller (RMC). In response, RMC sends a random string (RS) to LMC. This random string will help to avoid the replay attacks. Upon receipt of random string, LMC creates the hash-digest with the help of random string and the mutual password, shared between LMC and RMC, and sends it to RMC. The hash-digest of a message can be calculated using A *KEY_VALUE* and a *DIGEST_ALGORITHM*. RMC also calculates the hash-digest using random string and mutual password and compares the result with received hash-digest. On a successful match, the remote service request is accepted and now the user may get services from RMC via LMC, otherwise, remote servicing request is rejected.

IV. ANALYSIS

The benefit of choosing HMAC is that, it is one way process and reversely irrecoverable. This property makes it suitable to be used on an open channel. Although it makes use of a common key, shared between communicating parties i.e. symmetric key [9], but it is more secure than symmetric cryptography because symmetric cryptography is vulnerable to known ciphertext attack i.e. with some set of ciphertexts only, attacker may try to get the key. The process of generating a hash-digest using HMAC is shown below, with the help of a flow chart.

There are many HMAC calculators are available online. Just input your plain-text message, the key-value and the message-digest-algorithm. Message-digest-algorithm is an important choice from security point of view, since each algorithm has different level of security. Some algorithms have confronted some practical as well as theoretical attacks also. That's why MD-5 and SHA-1are considered as obsolete, now. Different algorithm produce different length HMAC. It's not always preferable to choose the longest HMAC producing algorithm. For example SHA-512 produce much longer HMAC than SHA-256. But even then, SHA-256 is considered a good choice, since no known attack is still in picture for it. A practical example of calculating HMAC, using different digest-algorithms is given below:

Message: *Hello RMC1, I have a job for you.*

Key_value: *1346783908864537852673874690*

HMAC-SHA1: *f8d83d061c4ba3ec784a9609c05450da1196ab99*

HMAC-MD5: *e8d872ac73a01d4ad095f9e22d5cdabe*

HMAC-SHA224: *89a6c855426c8b3a52bb66e5e6a2aa193e8561f58b01b299e2273e4d*

HMAC-SHA256: *050c2aa0a866dfd62a547352d21855f0a85fa35a71e8c4d5c5e6351cff1d3314*

HMAC-SHA384:
2f9ee56fb83aa523cd67cfc592058790e7bd1471915e8a834c56d4f1f8ee5de993d68c818b96cd7b03de94896cc06114

HMAC-SHA512:
9ef9c5c1ac329d0e4f9c0b327159869a84aa9e45b157d25789e59f97e07f52cb31034aa7ebbb1802c22994432d53e1cb96d9cee2c940820f888a43068d8b38ad

HMAC-SHA256 maintains a good trade-off between time complexity and security. This scheme will not pose any significant additional burden on LMC. Obviously servicing-time will get increased somewhat, but it will be tolerable since LMC will work like a gate-pass only.

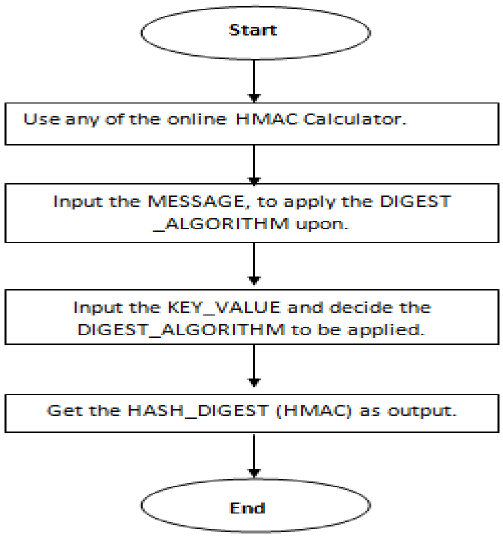

Start
Use any of the online HMAC Calculator.
Input the MESSAGE, to apply the DIGEST _ALGORITHM upon.
Input the KEY_VALUE and decide the DIGEST_ALGORITHM to be applied.
Get the HASH_DIGEST (HMAC) as output.
End

Figure. 6. Online HMAC calculation

V. Conclusion and Future Work

Performance is based on efficiency. So, efficient job handling is required. A public cloud is a very large infrastructure. It can be divided into some partitions. If a partition gets overloaded then it may ask to another partition for help. To get this concept practical, some sort of authentication is needed. HMAC can be used for this purpose but while implementing this model in real world, several new challenges may occur, that need to be addressed.

References

[1] Amazon Web Services http://aws.amazon.com/
[2] Service Level Agreement http://searchitchannel.techtarget.com/definition/service-level-agreement
[3] Gartner Public Cloud Service Forecast https://www.gartner.com/doc/2642020/forecast-public-cloud-services-worldwide
[4] Cloud Velocity http://www.cloudvelocity.com/the-top-three-benefits-of-hybrid-cloud-deployment/
[5] Gaochao Xu, Junjie Pang, and Xiaodong Fu , “A Load Balancing Model Based on Cloud Partitioning for the Public Cloud”, TSINGHUA SCIENCE AND TECHNOLOGY , ISSN ll1007 - 0214 ll04 /12ll pp 34-3 9 , Volume 18, Number 1, February 2013
[6] Thomas Y. C. Woo, Simon S. Lam “Authentication for Distributed Systems” ACM Press and Addison-Wesley, 1997.
[7] Distributed System http://www.cs.york.ac.uk/rts/books/RTSbookFourthEdition/distributedSystems.pdf
[8] Cryptography https://www.cs.columbia.edu/~smb/classes/f06/l03.pdf
[9] RFC 2104 http://www.ietf.org/rfc/rfc2104.txt