Chapter 4

Optimizing the Cost through Content Configuration Model in Cloud Computing Environment

4.1 INTRODUCTION

One of the cloud service models is Infrastructure-as-a-service. In that model, the actual infrastructure such as servers, hardware, disks etc., are shared via Internet for number of users over the cloud. This chapter deals about the clustering of the contents of every incoming task and to route it to the appropriate virtual machine instances to solve the content configuration issue. This chapter also deals with the cost optimization across various cloud data center heads.

Most cloud level services are meant to do the jobs according to service level agreement (SLA) in the cloud. Many policies have been discussed and decided during the SLA negotiation between the cloud user and the cloud service providers. After that, mutually both parties may agree to share the resources between the various cloud servers across the globe. The cloud users start sending the requests to the private cloud such as Amazon EC2 or Azure cloud platform. Subsequently, the cloud users have to wait and get the resources within the particular time frame fixed by the service providers.

Due to low bandwidth ratio in some segments across the Internet, the contents will be delivered to the users in a delayed manner. In order to reduce this kind of delay, all the contents that are transmitted over the Internet via cloud servers should be segregated according to the type of the content. The tasks that have been assigned to the cloud server may be clustered according to their contents such as video streaming content, audio content and other files content.

Many content configuration load balancing scenarios have been proposed for cloud based environments. Cloud load balancing works with an efficient module called cloud load balancer, a hardware unit which routes the traffic according to all the incoming HTTP and HTTPS traffic. This router works only for few protocols. Some of the algorithms that have been designed particularly for content based load
balancing are used for routing the traffic according to the content of the incoming tasks. Then clustering of the tasks has been done according to their file type.

The proposed task clustering algorithm is very much useful for performing this kind of load balancing.

Figure 4.1 clearly represents the load balancing for all incoming user requests by cloud load balancer. It distributes all the incoming network traffic across multiple virtual machine instances. Cloud load balancers manage online traffic by distributing workloads across multiple servers and resources, automatically or on demand. They maximize the workload performance and help prevent overload.

4.1.1 BENEFITS OF CONTENT CONFIGURATION LOAD BALANCING OF TASKS

The cloud load balancing in the cloud computing environment provides the following advantages:
- Application scaling
- Heavy traffic analysis and minimization
- Balance loads across multiple cloud servers
- Route traffic to the closest virtual machine

Fig 4.2. Load balancing scenario for content configuration

The work in this chapter, content configuration load balancing, uses cloud load balancer with HTTP load balancing to distribute traffic to different VM instances based on the incoming requests is depicted in figure 4.2. The video content files are distributed to server and further sent to the respective VMs i.e., VM1, VM2...etc., as the same way other content files are also have their own distribution among the respective VMs.
4.2. PROPOSED WORK

4.2.1 NOTATIONS USED FOR TASK CLUSTERING ALGORITHM FOR LOAD BALANCING OF TASKS

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DHT_n$</td>
<td>Distributed Hash Table</td>
</tr>
<tr>
<td>V.DHT$_n$</td>
<td>Video streaming task’s index</td>
</tr>
<tr>
<td>CSn</td>
<td>Cloud Servers</td>
</tr>
<tr>
<td>CU$n$</td>
<td>Cloud Users</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
<tr>
<td>RT</td>
<td>Response time</td>
</tr>
<tr>
<td>BWu</td>
<td>Bandwidth Utilization</td>
</tr>
</tbody>
</table>

Table 4.1: Notations used for task clustering algorithm for load balancing of tasks

From the above table 4.1, it is easy to understand the various standard notations used in this proposed model. Content configuration load balancing of tasks is one of the key research areas in the multimedia cloud computing model. By use of the DHT (Distributed Hash Table) based concept as introduced by Fabian and Feldhaus (2014). DHT model has been developed for the load balancing among multiple video based files in the cloud server as suggested by Chen et al (2014). The detail understanding of the proposed methods are explained clearly in this chapter.
Cloud user has several needs for storage, compute and networking in cloud environment. The contents may differ according to the cloud server availability. The tasks submitted by the cloud user through the cloud domain are processed in the task clustering server.

Figure 4.3 representing the architectural framework for content configuration load balancing in cloud. The hierarchy followed from the cloud user to cloud server via task clustering server, which has been clustering the task according to the incoming content from the cloud users. The proposed task clustering algorithm clarifies the task clustering method that has been given in the next section. Also, according to the content configured in the cloud data center, the cost can be minimized in it by reducing the flow of heavy contents among various cloud data centers.
4.2.2 CONTENT CONFIGURATION TASK CLUSTERING USING CONTENT FILTERING CLOUD SERVER

Figure 4.4 represents the life cycle of multimedia cloud systems. It is composed of various stages of processes such as procurement, storing in Multimedia cloud, processing, distribution and presentation. Always the multimedia content starts their life cycle from procurement such as raw data of any video or audio content. Then the particular media content will be sent for processing such as video processing, image processing or audio synchronization processing and all other processing things. After the processing stage, it will be sent to the cloud data center for the storage of the multimedia content files. Then the file may be broadcasted to all the cloud data centers for the application purpose. In some video streaming websites, the users might experience the delay in getting the buffered video content. For example the requested video through www.youtube.com has been streamed from the cloud data center located at some remote location.

Cloud user’s tasks are clustered and classified into groups and processed in the respective VMs. Each VM has unique server configuration based on the type of task it is clustered. Allocation of tasks has been done at the clustering server side and tasks have been scheduled to the VMs. Figure 4.5 depicts the flow of work in accessing the physical resources in multimedia cloud systems such as cloud data center by public
with the delivery of cloud services for multimedia such as video streaming service and multimedia storage services. After allotting a particular service to the cloud user, then the content configuration load balancer module will check and filter the contents and the files will be sent to the cloud data center according to the availability of update in the cloud status table. Through proper virtual machine instances concept the user tasks have been executed appropriately.

Figure 4.6 represents the cloud based content filtering model. It has some remote clustered VMs and their remote addresses are stored in the index of the clustering server. Each task has different content in it and assigned to the individual VMs in order to optimize the load. By content configuration load balancing, the overall cloud server throughput becomes high and hence the performance might be increased due to the task clustering process.

The content filtering cloud server is also called as Dominant cloud server. The Dominant cloud server takes active role in segregating the contents after vibrant
analysis of the incoming clustered file such as clustered video file embedded with hash values through hash function H(x). Similarly the other contents are also clustered using content filtering cloud server to give original content to the cloud user.

Fig 4.6. Content filtering cloud server for load balancing among the different contents

4.3 ANALYTICAL MODEL

The content of every tasks have been clustered using task clustering server. Every task and every content in it have been assigned with an index, i.e., for referring in the server, each content should have its own unique identity for its lookup service. For that, the DHT (Distributed hash table) concept is being introduced, to map every content with the respective server.

The role of DHT is very useful in assigning a key value to the contents. DHT usages are huge as like increased bandwidth utilization, low level latency in searching the content and increased hard disk capacity to provide a better file-sharing service among nearest cloud servers. Assume there are several contents in a task clustering cloud server.

Then, a proper DHT mechanism should be there to organize all the tasks in a linear way. Here the assumption is, if there are ‘n’ number of DHT is assigned for
every video based incoming contents into the task clustering server, then DHT$_1$ to DHT$_m$ belongs to a common index called DHT$_n$, that is given in the equation 1.

$$\sum DHT_n \in \{DHT_1 + DHT_2 + \cdots + DHT_m\}$$  \hspace{1cm} (1)

Also, some DHT formation has to be designed more secure against some malicious online attacks. And the DHT must be assigned with some strong key pair to avoid the third party attacks. Already DHTs are dealt with many distributed systems’ applications such as load balancing, data integrity, and performance related QoS issues.

---

**Fig 4.7. Process map for clustering of tasks in cloud**

Figure 4.7 represents the process map flow of the proposed work. There are some steps to be followed in this work.

Steps involved in the process map:

- Step-1: Initially, the pool of users submits their tasks to the cloud server.
- Step-2: Then there is a cloud server replica that takes every backup of files that is stored in the cloud server.
Step-3: According to the incoming tasks, contents are distributed to the respective servers by generating and assigning the DHTs to it.

Step-4: If there are video stream files, they are sent to the video cloud server for further process such as video streaming into the website.

Step-5: Similarly for other contents, the same procedure will be continuing.

4.4. TASK CLUSTERING ALGORITHM FOR CONTENT CONFIGURATION LOAD BALANCING OF TASKS

The proposed task clustering algorithm is used to minimize the latency in the cloud server. Figure 4.8 clearly represents its flow of process. Firstly, there are many users who need immediate response from the cloud server sending their queries continuously from any of the public cloud model such as google cloud bus.

Fig 4.8. Assigning DHT for incoming contents in cloud server

By the upload of tasks into the task clustering server, the tasks are actually split according to the content in it. DHTs are assigned to each task. For example, if it is a video streaming task, then V.DHT$_1$,… to V.DHT$_n$ will be assigned according to the incoming tasks. Similarly this method will be continuing for other related contents
also. The proposed algorithm explains clearly about this process using java based methods.

Algorithm: Task clustering procedure

```
Procedure AssignContentsToVMs(contentList)

Begin

    DHT = 0
    vmIndex = 0

    while (ContentList.size() > 0)

        Content = ContentList.getNextContent()
        vm = getVMsList(vmIndex)
        vm.scheduleContentToVM(Content)
        DHT.assignContenttoDHT(DHTList)
        totalVMs = getVMsList().size()
        vmIndex = Mod(vmIndex + 1, totalVMs)
        DHT = Mod(DHTList + 1, totalDHTs)
        ContentList.remove(Content)

    end while

End

```

This algorithm depicts the overall content assignment to the respective virtual machines (VMs) in the cloud computing environment. The algorithm uses two different lists, the first one is about various kinds of requests (denoted as contents) from the cloud user and the other is about DHT (Distributed Hash Table) mapping with all the incoming contents. This specified algorithm having repeated procedural steps for ContentList and then, over getNextContent() method that is used for getting response from Task clustering Server in a (FirstInFirstOut) FIFO manner.
And every content is sent to respective content VMs in a Round-robin method. The method getVMsList(vmIndex) is used for task completion in the VM. These steps are repeated until the ContentList is denoted for NULL value and all tasks have been submitted for completion in the Task clustering Server VM.

4.4.1 CONTENT CONFIGURATION PROBLEM TO REDUCE COST IN CLOUD DATA CENTERS

Content like video files can be streamed across various countries from different cloud data centers. Content configuration problem may occur in cloud data centers. This problem can be solved only by analyzing the cost per data center for the entire content deliveries. Assume if there are 100 cloud data centers located across the country, then there might be at least 1000 times greater ratio of the users who dependent on the particular cloud data centers in the respective region. The cost factor is an important and essential parameter, which has to be calculated in order to analyze the resource mapping for the users. In order to reduce the cost among many clusters in the cloud data center, firstly the initial cost has to be calculated.

Initial cost \( (C_I) \) is expressed by adding data center ownership cost \( (C_O) \) and existing cluster cost \( (C_C) \) that is given in equation 2.

\[
C_I = C_O + C_C
\]

After calculating the \( (C_I) \), it is easy to analyze the data centers’ overall costs \( (C_{DOV}) \) by adding the initial cost \( C_I \), cost for energy consumption of independent nodes \( C_E \), cost for task migration among cloud data centers virtual machines \( C_{TM} \) and cost for workload calculation \( C_{WC} \) given in equation 3.

\[
C_{DOV} = C_I + C_E + C_{TM} + C_{WC}
\]

4.4.2 EXPERIMENTAL MODEL FOR ANALYZING COST REDUCTION

In Cloud computing environments, the cost analysis has been done by developing comprehensive model for actual cost reduction in the cloud data centers. Assume there are four data center heads located in four different locations and they have costs such as initial cost and energy consumption of independent nodes \( C_E \). We are
introducing an even analysis model table and graph to understand this scenario. From table 4.2 we could clearly understand about the given scenario.

Table 4.2. Cost calculation in data center heads considering various cost parameters

<table>
<thead>
<tr>
<th>Cloud Data center heads</th>
<th>Cost calculation in data center heads</th>
<th>Various cost parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$C_{DOV}$ ($ per kWh)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$C_C$</td>
</tr>
<tr>
<td>R</td>
<td>0.08</td>
<td>0.04</td>
</tr>
<tr>
<td>S</td>
<td>0.09</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>U</td>
<td>1.2</td>
<td>0.9</td>
</tr>
</tbody>
</table>

\[ C_E (kW) = C_C + C_{ID} \]  

\[ C_E (kW) \] is calculated by the equation 4, where \( C_{ID} \) is the energy consumption cost in the idle machines and \( C_C \) existing cluster cost. Figure 4.9 depicts the overall \( C_{DOV} \) cost in kilowatt per hour for all the four cloud data center heads. In that, the data center U, having more energy spent due to many modules such as energy consumption by individual cloud data center as well as idle machines in the cloud data centers represented as \( C_E \) (kW) and the workload calculation for every cloud data center \( C_{WC} \) (kW) is more when compared to the other cloud data center heads.
4.5. EXPERIMENTAL SETUP

There are various contents of tasks coming into the cloud server, in which the load has been equally analyzed. Then the content configuration filtering has been done in the task clustering cloud server for finding the performance of upload bandwidth utilization, task migration time and latency in streaming the video contents.

Fig 4.10. Number of Virtual Machines vs Video Latency time (ms)
From table 4.3, it is clearly understandable view about upload video bandwidth utilization by different cloud servers. The video upload bandwidth utilization of the cloud servers for various workload distributions have been analyzed by creating 200 cloud servers and each server is assumed in single core having up to 4000mips (millions instructions per second) with 8GB RAM and 1TB storage. Figure 4.10 depicts the number of Virtual Machines placed and calculating the Video Latency time (ms). The result graph shows that there is less latency if the number of VMs increased. This indicates that the work is shared between various numbers of VMs in the cloud servers.

The users submit the tasks to the provided 200 cloud server virtual machines. Every VM is assumed to have an equal and distributed load in the cloud server. On netbeans IDE with CloudSim, this setup is tested to find the optimal standard deviation values. Hence, it is clearly understood that the proposed algorithm is considered to be utilizing less bandwidth among multiple cloud servers by using the good content configuration load balancing technique. Fig.4.11 represents the video upload bandwidth utilization (mbps) graph, which consists of three axes. In the graph X-axis represents the number of tasks submitted by the users and the Y-axis represents the number of VMs and Z-axis represents bandwidth utilization by the servers in mbps (megabits per second) unit.

<table>
<thead>
<tr>
<th>No. of tasks</th>
<th>No. of VMs</th>
<th>Upload bandwidth utilization (mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>1.3</td>
</tr>
<tr>
<td>20</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>15</td>
<td>5.6</td>
</tr>
<tr>
<td>40</td>
<td>20</td>
<td>8</td>
</tr>
<tr>
<td>50</td>
<td>25</td>
<td>12.48</td>
</tr>
</tbody>
</table>

Table 4.3. Video upload bandwidth utilization calculation using task clustering algorithm for video tasks
Fig 4.11. Video upload bandwidth utilization (mbps) graph

4.6 SUMMARY AND FUTURE WORK

In this second phase of this research work, the content configuration load balancing of tasks in cloud computing environment is evidently analyzed for the video streaming purpose and to reduce the cost per cloud data centers.

This approach helps in balancing the load in multiple cloud servers by investigating QoS (Quality of service) metrics such as video tasks upload bandwidth utilization and latency. In the proposed task clustering algorithm, the cloud users submitting the tasks to the cloud server and in response to that, the cloud servers are optimally balancing the load to decrease the latency in the streaming of videos through Internet.

The result shows good improvement in video tasks upload bandwidth utilization among cloud servers. This algorithm could be extended for private cloud storage mechanisms to balance the tasks based on the contents stored and to decrease the buffering speed of the video further more in near future.