Chapter 5

A Solution for Geographic Regions Load Balancing in Cloud Computing Environment

5.1 INTRODUCTION

Cloud computing is one of the most interesting way of distributing the data as well as to get the prominent services through the Internet connection. Although there are various interesting services available in cloud computing, scientists and researchers consider IaaS (Infrastructure as a Service) as a core valued service in cloud platform. Cloud service level agreement (SLA) plays an essential role in all the cloud based services, which has to be agreed by cloud service providers and clients.

The load balancing scenario in cloud is useful in such a way that, the traffic in every server will be reduced and has a smooth flow of HTTP (Hypertext Transfer Protocol) packets. The design of the traffic routing algorithm will play a crucial role in routing the packets that contain different query models. All the routing processes contain many routing tables that has index of all the incoming requests.

Barr et al (2011) stated that amazon web services uses different elastic load balancing techniques for managing the zones based load balancing of tasks. By having an idea over the concept, it is possible to do the optimal region level load balancing of tasks in the following ways.

The routing process has some scheduling function in which the queries that are coming into the server in different timelines will be added in that routing table as tiny consignments. Then it is scheduled to the respective VMs (Virtual Machines) for the further processing. Actually, some precedence should be maintained for the workflow in processing the queries. They are,

- Always, the loads in every cloud servers should get balanced (adjusted) according to the incoming requests.
- The computational cost for scheduling of tasks should always remain cheap when compared to the other processing systems in the servers.
The architecture of hybrid region load balancing scenario in cloud systems is represented in Figure 5.1. Here the load balancing for the entire incoming user requests is done and solved by proper scheduling of incoming tasks through round robin based scheduling policies. The cloud load balancer module has been introduced to distribute all the incoming network traffic across multiple regions. Assume if there are some failures in the servers in particular region and there are some heavy loaded tasks such as video streaming applications are requested by the public cloud users, then the query has to be rerouted to the nearest region for delivering the exact need to the cloud users on time.
5.2 BENEFITS OF HYBRID REGION LOAD BALANCING OF TASKS

The hybrid region load balancing in cloud computing environment offers the subsequent benefits to its users:

- Request-response in a ‘on-time’ delivery method
- Dense traffic investigation and traffic flow elimination
- Balance loads through various servers in nearest region matching
- Re-Routing the HTTP traffic to the nearest region

5.2.1 NOTATIONS USED FOR HYBRID REGION LOAD BALANCING ALGORITHM FOR LOAD BALANCING OF TASKS

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DHT_{ri}$</td>
<td>Distributed Hash Table</td>
</tr>
<tr>
<td>$R_{i}$</td>
<td>Region</td>
</tr>
<tr>
<td>CSn</td>
<td>Cloud Servers</td>
</tr>
<tr>
<td>CUn</td>
<td>Cloud Users</td>
</tr>
<tr>
<td>VM</td>
<td>Virtual Machine</td>
</tr>
<tr>
<td>K</td>
<td>key value</td>
</tr>
</tbody>
</table>

Table 5.1: Notations used for hybrid region load balancing algorithm for load balancing of tasks
5.3 PROPOSED WORK

Fig 5.2. Distribution of loads through Region ReRoutingLoad balancing (RRRL) algorithm

Generally, in the cloud based systems, the usual scenario for geographical region load balancing of tasks can be done in a three way hierarchical fashion. Similarly, in the proposed method, the cloud users request the tasks from the public cloud systems. Then those requests are routed to the cloud load balancer for routing to the further cloud servers in various regions across the globe.

The proposed algorithm named Region ReRoutingLoad balancing (RRRL) algorithm has been designed in such a way that if there are faults in some cloud servers in any of the regions, then automatically the tasks are rerouted to the nearest cloud server in the nearer region that has been represented in figure 5.2. This kind of rerouting reduces time in responding to the users tasks as well as increasing the overall throughput in the region based load balancing of tasks.
**Algorithm 1: Region ReRoutingLoad (RRRL) procedure**

**Procedure** ReRoutingLoadToNearestCloudServers(Regions, Keyvalue, Stats)

**Begin**

\[
\text{While } R_i \text{ is in Region}_x \text{ in regionList} \quad \quad // R \text{ is the set of regions}
\]

If K is in Keyvalue, then   // K is an active region

Send L to K  \quad // sending load to region K

Else

\[
R_i \text{ communicates } K_i \text{ to CLB} \quad // \text{ Cloud Load balancer}
\]

If K is in DHT\(_k\) then  // indexing in Distributed hash table

\[
K_i \text{ resolved to } R_i
\]

Increment KeyValue of R

Make K as Region\(_d\)  // searching next nearest region as K is overloaded

Decrement Rank of R\(_i\) in regionList

getNextregion()

Make I as Nearest region

**End if**

**End if**

**End while**

**End**

The algorithm1 - Region ReRoutingLoad balancing (RRRL) portrays the region based requests rerouting mechanisms in the cloud computing environment. The
algorithm uses three different parameters, the first one is about various kinds of regions, the second one denotes key values assigned for every region in the DHT indexes and the other is about stats in the DHT (Distributed Hash Table) mapping with all the incoming requests such as ACTIVE, BUSY or IDLE. This stated algorithm having repeated procedural steps for regionList and then, over getNextRegion() method that is used for getting next nearest region.

![Region based load balancer for routing user requests](image)

Fig 5.3 Region based Load balancer for routing user requests

From the figure 5.3, it is easy to understand about the region query transferring process that has been given in the algorithm 2. The actual process method is same like the regular scenario in the cloud computing environment. Always, the cloud users are submitting some requests to the cloud systems.

The region based load balancer is used to monitor all the incoming requests and to route all the requests to the appropriate regions based on the availability of the zones in it. And inside every zone there are many cloud data centers that support the virtualization and virtual machine instances are placed over it to gratify the user requests according to the web based application.
Algorithm 2: Region-based load balancer in CloudServers

```plaintext
Procedure Regionbased load balancer in CloudServers(Regions, Zones, Field-Id)
Begin
"LoadBalancer Definitions":
[
{
...
Declarations essentials:
{
"DNS Name": "load balancer-region based", "Security Groups": ["Field-Id"],
"Policies": {"Load balancer zone enabling Policies": [],
"Zonal Application enforcement Policies": [],
"Default Policies": []
}
"Load Balancer Name": "Name",
{
Create Load Balancer module with VMs
"Time of creation": "Region based Time",
"Availability regions": ["Region ‘N’"],
"Purpose": "Region based load balancer with relevant Field-Id",
...
}
]
End
```

In order to perform the load balancing of tasks among multiple regions, firstly the system has to analyze and calculate the number of incoming user requests (loads) to the cloud systems. The algorithm 2 describes clearly about the regions, zones, number of virtual machines instances and field identity (Field_Id). The declaration begins with the initialization of all the load balancer definitions for all the regions. Then the system has to find and declare the essential parameters like Domain Name System (DNS), Policies for naming regions, zonal based loads application, and load balancer name.
After declaration, the load balancer module performs the tasks for region based query transferring by looking into the field id and zones with geographic times and it starts to reroute the requests to the various zones that contains different virtual machine instances. Sometimes zonal server failure may happen. In order to avoid that, load balancer module has to reroute the user queries for failover strategies as represented in figure 6.4. Failover is defined as rerouting all the cloud user requests from the failed cloud data center to the next data center. Usually failover happens in a cloud datacenter due to hardware failure, storage server failure, or scheduled maintenance of cloud server etc., So always in a cloud systems, there should be more than two cloud data centers to serve the user requests in case of the server failure at a point of time.

5.4 ANALYTICAL MODEL

In order to justify this work, an analytical model has been designed for the above mentioned systems. The regions are represented in cloud as R and each region are
assigned with a key value of K in the Distributed hash table (DHT) in their respective routing table’s index. Also, they are assigned with a status value like, ACTIVE, BUSY or IDLE according to the cloud server’s current status update in the routing table. If there are more requests in a particular server then it’s denoted as BUSY stats.

Then the scheduler has to schedule the incoming tasks to the next nearest ACTIVE cloud servers. The proposed model is very useful for distribution of loads across geographical regions for optimal load balancing using key value mapping cloud load balancer for hybrid region mapping. The proposed algorithm which works for HTTP load balancing to distribute traffic to different cloud servers in various geographical regions based on the incoming requests is given in the figure 5.5.

![Diagram showing geographical regions based Load Balancing using key values mapping](image)

**Fig 5.5.** Geographical regions based Load Balancing using key values mapping

The users in the private or public clouds may send enormous number of requests such as HTTP traffic generation requests in the cloud servers, which are
maintained by cloud load balancer module for hybrid region mapping through the proposed Region ReRoutingLoad balancing (RRRL) algorithm. Then according to the regions, key values will be assigned and the statistics will be displayed in each server’s DHT. Cloud servers in DHTs maintain an index for O(log n) cloud user request, where n is the total number of requests in the regional servers DHT index. The steps in the DHT process are;

✓ Generating cloud server indexes
✓ Creation of key values
✓ Assigning of stats value to the regions.

By the analytical modeling, assume always the number of incoming tasks referred as requests or loads into the cloud servers. Actually the load depends on the number of VMs running inside every cloud server for that time span of ‘T’. Assume \( \sum_{i} VM_i \) is the total VMs running in the set of cloud server \( \sum_{CS_i} \).

The performance of the individual cloud server has been derived by analyzing the time span

\[
T (\sum CS_i) = \{CS_1t_1 + CS_2t_2 + \ldots + CS_nt_n\} \tag{1}
\]

Under region based load balancing the time span can be calculated by analyzing the total load variation within the particular region and the average load taken by the previous region, then the load of the cloud server in a region is calculated by,

\[
CS (i,T) = \sum_{k}^{i} (VM(k,T)) \tag{2}
\]

where, k is the key value in the cloud servers’ DHT and T is the time elongation for rerouting the request. If the cloud server in region r is overloaded, then the requests should get rerouted to the nearest region by the equation (1) and (2).

5.5 EXPERIMENTAL SETUP

There are various regions in the geographical cloud systems, in which the load is equally examined and the nearest region has to be mapped in the cloud server. The throughput and latency are calculated using CloudSim simulation.
From Table 5.2, it is clearly understandable about the comparison between the two algorithms for finding minimal latency by total number of tasks scheduled for RR Algorithm: Round-robin algorithm and RRRL Algorithm: Region ReRouting Load balancing algorithm.

<table>
<thead>
<tr>
<th>No. of tasks</th>
<th>Latency in RR Algorithm (ms)</th>
<th>Latency in RRRL Algorithm (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>122</td>
<td>89</td>
</tr>
<tr>
<td>200</td>
<td>275</td>
<td>111</td>
</tr>
<tr>
<td>300</td>
<td>356</td>
<td>210</td>
</tr>
<tr>
<td>400</td>
<td>566</td>
<td>430</td>
</tr>
</tbody>
</table>

RR Algorithm: Round-robin algorithm

RRRL Algorithm: Region ReRouting Load balancing algorithm

Table 5.2. Latency comparison with various load balancing scheduling algorithms

The number of tasks that are sent to the cloud server is the most important factor to be considered for the simulation scenario. The cloud users requests are based upon the regions which have been assigned to the roles for the DHT in the zone wise. Also the latency issues that are rising continuously in the region wise load balancing paves way for the upcoming research scenarios.

This chapter is fully revealing the importance about the geographic regions based load balancing if task is in the cloud computing environment. As there are some issues between various load balancing models in many different zones, there is a central authority which regulates all these issues through proper load balancing mechanism.
Fig. 5.6. Comparison of Latency (ms) with other algorithms

Fig 5.7. Number of cloud servers vs. throughput (requests/sec)
From figure 5.6, it is easy to come for a conclusion that the proposed RRRL algorithm works better than round-robin algorithm. Also, from figure 5.7, when considering the throughput (requests/sec) for the rerouting of tasks with various regions, the proposed Region ReRouting Load balancing algorithm has higher throughput ratio amongst various number of cloud servers.

5.6 SUMMARY AND PATH FORWARD

The region based rerouting of tasks in cloud computing for optimal query rerouting across many geographical regions have been deeply investigated for efficient algorithmic method. This approach helps in balancing the load in hybrid region based cloud servers by inspecting QoS (Quality of service) metrics such as throughput (requests/sec) and latency.

In the proposed Region ReRouting Load balancing algorithm, the load balancing is done optimally to decrease the latency in rerouting the requests to various cloud servers in several regions. This research works outcome shows decent precision in high throughput achievement among cloud servers. This algorithm could be extended for cost computational methods in each cloud servers across various geographical regions.