International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 07 | July 2018 www.irjet.net p-ISSN: 2395-0072

Approaches to Minimize Resource Utilization in Cloud Services

Pratheeksha R[1], Yasha Ravindra[2], Prema T.H[3], Pavithra Rani[4], Kavita V. Horadi[5]

[1,2,3,4]*Students, Dept. of Computer Science, B.N.M Institute of Technology, Karnataka, India*
[5]*Assistant Professor, Dept. of Computer Science, B.N.M Institute of Technology, Karnataka, India*

Abstract - *Facilitating load balancing and improving quality of service in data centers eases the usage of cloud computing applications. This can be done by classifying the servers based on the number of requests being processed. The switching decisions are made based on the environment of the servers, so as to achieve good quality of service. Round Robin Algorithm is used when the server is in idle state, Opportunistic Routing Algorithm for normal state wherein it measures the distance between servers using their latitude, longitude and the IP address to direct the requests to the nearest server. For overloaded state, Skewness Algorithm is used, by considering the least CPU utilization. Thereby, it strives to work even when the servers are overloaded and is accomplished by utilizing the existing resources.*

Keywords: Load balancing Algorithms, Server Switch, Quality of Service, Efficient Resource Utilization, Cloud service, Request redirection

1. INTRODUCTION

Cloud is a metaphor for the internet and cloud computing is the delivery of computing services such as servers, storage, databases, networking, software, analytics and more-over the cloud. It delivers the on–demand computing devices from applications to data centers on a pay-per-use basis. Companies offering such services are called cloud providers. There are three service models of cloud computing, namely:

a. **IaaS**: It refers to online services that provide high-level APIs used to dereference various low-level details of underlying network infrastructures.

b. **PaaS**: The capability provided to the consumer is to deploy on to the cloud structure consumer-created or acquired applications created using programming languages.

c. **SaaS**: The capability provided to the consumer is to use the provider's application running on a cloud infrastructure.

There are three deployment models, namely:

i. **Public**: A cloud is called a "public cloud" when the services are rendered over a network that is open for public use. Public cloud services may be free.

ii. **Private**: It is cloud infrastructure operated solely operated for a single organization, whether managed internally or by a third-party, and hosted either internally or externally.

iii. **Hybrid**: It is a composition of two or more clouds (private or public) that remain distinct entities but are bound together, offering the benefits of multiple deployment models. This work incorporates infrastructure-as-a-service categorized under cloud services.

2. RELATED WORK

"Jeffrey S. Chase[1]" has explored the interrelated factors that shape the role of server switching and request distribution for server clusters. These factors include the emergence of content delivery networks, the increasing prevalence of dynamic content, persistent connections in HTTP 1.1 standard, the changing nature of web server clusters and the opportunities to apply server switching techniques to service protocols other than HTTP. Following these approaches in a fully general way requires addressing the challenge of independently routing multiple requests arriving on the same transport connection.

"Gil Sang Yu and Seong Gon Choi[2]", have proposed a video quality improvement system (VQIS) that operates in multimedia servers and utilizes the server switching technology to increase Quality of Service of the User Equipment. VQIS provides the server switching function to improve QoS of the User Equipment, if QoS degradation occurs in User Equipment. It is verified by constructing testbed that the proposed method can provide guaranteed video quality in User Equipment. It only describes the theoretical matters but the actual implementation is not presented.

"Wathit Chaloemwat and Sukumal Kitisin[3]" simulated a threshold based auto-scaling program with and without skewness algorithm. The program also had process migration. It was observed that skewness algorithm and process migration can help smooth out the fluctuation of the number of virtual machines being spawned and then very soon be deleted resulting in the reduction of the overhead of such process.

"Muhammad Sohaib Shakir and Abdul Razzaque[4]" compared various load balancing algorithms – Round Robin, Equally Spread and Throttled Algorithm. On running the simulation, it was found that there is no difference between the costs but there was a slight difference between overall response time of all the user bases, and round robin is found to be the best.

"Gary K. W. Wong and Xiaohua Jia[5]" introduced the social relations and profiles among nodes in the network as the key metrics and proposed the Social Relation Opportunistic Routing (SROR) distributed protocol to solve the routing issue in mobile adhoc networks (MANETs). Proposed algorithm is evaluated using ns-2 simulator and by comparing it with benchmark algorithms and found that SROR out performs current available methods.

"Sam Safavi and Usman A Khan[6]" developed a distributed algorithm to localize a network of robots moving arbitrarily in a bounded region. In case of mobile networks when robots are not able to find nearby robots to implement distributed algorithm, Opportunistic Algorithm is used that implements a location update when there are nearby robots and does not update otherwise.

"Brian Guenter, et. al[7]" presented an automated provisioning system that aims to meet workload demand while minimizing energy consumption in data centers and reliability costs in hosting clusters. The Centralized server provisioning systems considered typically scale to few hundreds of nodes and are limited to single administrative domain. Secondly, the coordinated management between compute, storage and network resources is lacking which is much needed for global energy management.

"Simon Kiertscher and Bettina Schnor[8]", proposed an Energy-aware resource management strategies which are based on thresholds. The basic idea is to turn off currently unused nodes, and turn on again when the load increases. The availability of enough computing power in an unexpected peak load situation is investigated using trace driven simulation for different scenarios like Suspend-toRAM capable nodes, over-provisioning, load forecasting and anti-flapping strategies. Heterogeneous clusters are not addressed in these strategies.

"Simon Kiertscher and Bettina Schnor[9]", designed an energy saving mechanism which extends the capabilities of traditional load balancers. The energy saving daemon called Cherub running on the front node of the cluster turns nodes on and off depending on the current load situation. Different load determination and forecasting methods are evaluated to detect the utilization of back end servers in testbed using requests from a web server log file. This mechanism lacks focus on heterogenous server load balancing environments and to deal with different thresholds for every class of machines.

3. SYSTEM DESIGN

This work considers two modules – a shopping application named Exclusive Buy and an administrator module, monitoring the servers.

The shopping application consists of products from various categories that the user can purchase.

The administrator application contains provision to add or delete a new server, view the server configurations, view active connections, monitor connections and see the available bandwidth.

This work is hosted on a cloud and when the end user sends request in the form of clicking on the shopping application, their IP address is used to identify their location and the request is intended to go to the server in that location. If the desired server is busy, the main controller looks for other servers in the cloud, which can be in one of the following states at any given time:

1. Idle

2. Normal

3. Busy

The above-mentioned states are determined based on the request the servers are processing. Figure 1.1 shows the system design and Table 1.1 shows the states of server based on the requests.

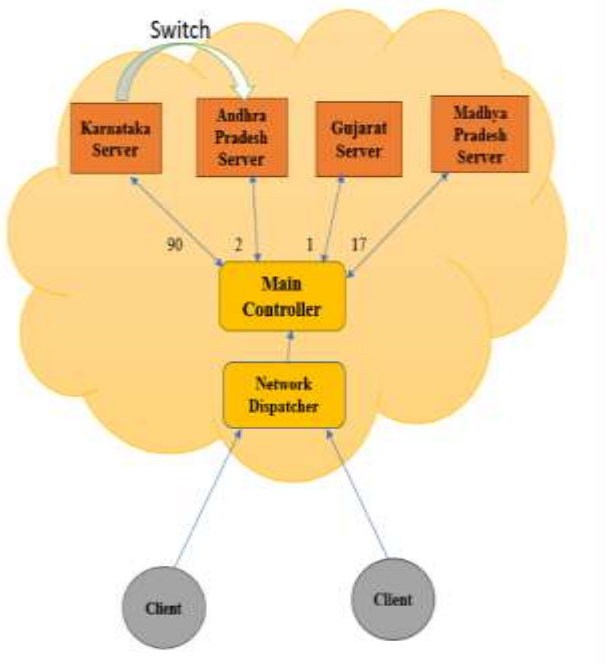

Fig – 1: Proposed Model

The table assumes the threshold is 100, however, in simulation, the threshold considered is 3.

If servers are idle, Round Robin Algorithm is used to send the request. If the servers are in Normal state, Opportunistic Routing Algorithm is used, else Skewness Algorithm is used.

Table – 1: Server Status

Sl. No.	No. of Requests	Server Status
1.	100	Threshold Value
2.	90+	Overload
3.	11 - 90	Normal
4.	0 - 10	Idle

4. ALGORITHMS

The project is implemented in Java and the pseudocode of the algorithms is mentioned below:

A. Round Robin Algorithm:

The algorithm directs the request to any idle server in the cloud.

ALGORITHM RoundRobin(Request for an overloaded server)

//Algorithm to direct a request to an idle server when the requested server is overloaded
//**Input**: Incoming request
//**Output**: Idle server allotted

1. **if** new connection comes for overloaded server **then**
2. **if** idle servers are available **then**
3. Connect to the server in idle state
4. **end if**
5. **end if**

B. Opportunistic Routing Algorithm:

Once the servers are in normal state, the request can be sent to any server. Doing so, might be ambiguous. To eliminate the ambiguity, the request is sent to the server in close proximity from the intended server. The distance is calculated using Euclidian formula.

ALGORITHM OpportunisticRoutingAlgorithm(Request for an overloaded server)

//Algorithm to direct a request to a server in normal state when the requested server is overloaded

//**Input**: Incoming request

//**Output**: Normal server, near to the requested server is allotted

1. **if** new connection comes for overloaded server **then**
2. **if** servers are in normal state **then**
3. Dist1 = ComputeDistance(lat1,long1)
4. Dist2 = ComputeDistance(lat2,long2)
5. **end if**
6. **if** Dist1 > Dist2 **then**
7. Send request to server located in lat1, long1
8. **else**
9. Send request to server located in lat2,long2
10. **end if**
11. **end if**

C. Skewness Algorithm:

When the servers are busy, the CPU utilization is considered to switch the request with least CPU utilization.

ALGORITHM SkewnessAlgorithm(IncomingRequest)

//Algorithm to direct a request to a server in overloaded state with low CPU utilization

//**Input**: Incoming request

//**Output**: Overloaded server, with low CPU utilization is allotted.

1. **if** new connection comes for overloaded server **then**
2. **if** servers are in overloaded state **then**
3. CPUPerc [ ] cpus = new CPUPerc [ ]
4. **for** i=0 to cpus.length **do**
5. check for the server with low CPU utilization
6. **end for**

7. Connect to the server with low CPU utilization

8. **end if**

9. **end if**

5. WORK ANALYSIS

The system incorporates the various states that a server could be in processing the request. These states are identified based on the number of requests sent to the server. Also, the system utilizes the active servers efficiently i.e., the requests are handled by the servers until the threshold is reached unlike the existing system design where the requests are redirected to other servers well before it touches the threshold. The core idea is to ensure that the multiple requests are distributed equally among the existing servers, which is made possible by considering the CPU utilization of each server. Furthermore, when the servers are in the normal state, the longitude, latitude, along with the IP address of the data centers are considered to choose the nearest server to redirect the requests which reduces ambiguity in server selection. In addition, the system is processor independent and works smoothly in different processor cores like dual core, quad core, octa core etc. The system can also be accessed from a mobile application. Thus, the system overcomes certain limitations of the existing system in minimizing the utilization of resources and consequently improves the quality of service.

6. CONCLUSION

The requests sent by the end user is sent to the server based on the local IP address. This mapping is done using the details in the IP Lookup table. Depending on the load of the server, appropriate algorithm among Round Robin, Skewness and Opportunistic Routing Algorithm is chosen. For a system of multiple servers, the problem of job distribution is solved using concept of load balancing. The administrator can monitor the server status, active connections and the available bandwidth.

REFERENCES

[1] J.S Chase, "Server switching: yesterday and tomorrow", The Second IEEE workshop on Internet Applications, WIAPP 2001.

[2] Gil Sang Yu, Seong Gon Choi, "Video quality improvement system by using server switching technology", 2013 15th International Conference on Advanced Communications Technology (ICACT).

[3] Wathit Chaloemwat and Sukumal Kitisin, "Horizontal Auto-Scaling and Process Migration Mechanism for Cloud Services with Skewness Algorithm", 2016 13th International joint Conference on Computer Science and Software Engineering (JCSSE).

[4] Muhammad Sohaib Shakir, Engr. Abdul Razzaque, "Performance Comparison of Load Balancing Algorithm using cloud analyst in cloud computing", IEEE 2017.

[5] Gary K.W. Wong, Xiaohua Jia, "A novel socially-aware opportunistic routing algorithm in mobile social network", 2013 International Conference on Computing, Networking and Communications (ICNC).

[6] Sam Safavi, Usman A Khan, "An Opportunistic Linear Convex Algorithm for Localization in Mobile Robot Networks", IEEE Transactions on Robotics 2017.

[7] Brian Guenter, Navendu Jain, Charles Williams, "Managing Cost, performance and reliability tradeoffs for energy-aware server provisioning", 2011 Proceedings IEEE INFOCOM.

[8] Simon Kiertscher, Bettina Schnor, "Optimizing Energy Efficiency and Quality of Service in Large Scale Web Server Environments", 2016 IEEE International Conference on Internet of Things(iThings) and IEEE Green Computing and Communications(GreenCom) and IEEE Cyber, Physical and Social Computing(CPSCom) and IEEE Smart Data(SmartData).

[9] Simon Kiertscher, Bettina Schnor, "Energy Aware Resource Management for Clusters of Web Servers" , 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing.

[10] Information on Computing from www.computer.org/Publications/dlib .