International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 25 Issue 5 – APRIL 2018 (SPECIAL ISSUE).

Optimization of Bandwidth Utilization and delay reduction in Cloud using controller

#1 T.Rajalakshmi, *2 P.Priya, *3 AnithaJaganathan

1 ,2B.E (CSE), Kings Engineering College, Chennai,India

3 Assistant Professor, Kings Engineering College, Chennai, India

Abstract: Datacenter network (DCN) plays a fundamental role in cloud computing. As applications grow fastly, capacity utilization of data-center network becomes a big challenge to cloud service, especially when user's requests are unbalanced or data-center works at a peak time whentherareamountofdemandsneedtobehandled.Software-definedNetworking(SDN)isan efficient technology to manage network utilization. In this paper, we apply SDN controller for service routing to increase bandwidth utilization of DCN and meanwhile reduce delay of end- users. To the best of our knowledge, existing approaches, however, either solely focus on minimizing total cost for provider, or guaranteeing QoS for end-users. In this paper, we apply the softwaredefinednetwork(SDN)controllertoenablethecentralcontroloftheentirenetwork,and propose a joint optimization model to consider high bandwidth utilization for provider and low delay for users. We present the Genetic load balancing algorithm to model both requirements of provider's high bandwidth utilization and end-users' low delay. Specifically, we formulate the design of request allocation under those requirements as an optimization problem, which is load balancing. To solve such hard optimization problem, we develop an efficient algorithm which reduce thedelay.

Key words: Genetic load balancing algorithm, Datacenter, Controller, Request allocation

INTRODUCTION

Cloud Computing is a technology which depends on sharing of computing resources than having local servers or personal devices handletheapplications.InCloudComputing, the word “Cloud” means “The Internet”, so CloudComputingmeansatypeofcomputing in which services are delivered through the Internet. The goal of Cloud Computing is to make use of increasing computingpower toexecute millions of instructions per second. So there's a significant workload shift, in a cloud computing system.

In cloud computing elasticity in the request allocation is one of the major problem. Large-scale Internet applications, suchasvideostreaming(Netflix),websearch (Google), and social network (Facebook), provide service to hundreds of millionsofend users. The enormous and growing demand of these applications has motivated service providers to deploy geographically distributed datacenters for both reliability and performance reasons. The request allocation is able to be completed with a simple method that allocates each request to the closest datacenter. On the one hand, due to the oversubscription, datacenter may overloaded when it comes to the peak time. As a consequence some datacenters may become overloaded, while some datacenters may experience low bandwidth utilization at a particular moment. For datacenters with low bandwidth utilization, it is a waste of both investment and energy for the provider. Meanwhile, the overloading are vulnerable to failures, and can lead poor application performance. It makes user dissatisfaction. On the other hand, as business increases, there is a rising demand for end-users’ requirements for low delay. Unfortunately, with the naive localized allocation strategy, users close to the datacenters experience low delay, while users far away from the datacenters are likely to suffer high delay. Furthermore, as users are increasing to select the closest datacenters, the phenomenon of high delay will become worse when datacenters come to the peak time. Motivated by these, we focus on the request allocation in geographically distributed datacenters, and jointly optimize bandwidth for provider and delay for end-users. More precisely, some workload can be shifted from overloaded datacenters to the datacenters that have low bandwidth utilization, so as to increase the bandwidth utilization with the overall system perspective of the provider. Our request allocation algorithm can efficiently improves bandwidth utilization for provider and reduce the delay for end-users.

Related Work: V.K. Adhikari [1] proposed the Netflix is the leading provider of on- demand Internet video streaming in the US and Canada. We find that Netflix employs a blend of data centers and Content Delivery Networks (CDNs) for content distribution. Finally, as improvements to Netflix's current CDN assignment strategy, we propose a measurement-based adaptive CDN selection strategy, multiple-CDN-based video delivery strategy, are demonstrate their potentials in increasing user's average bandwidth. B.wong and E.G.Sirer proposed [2] ClosestNode.com namely, accurate, scalable and backwards- compatible service for mapping clients to a nearby server. It provides a DNS interface by which clients can look up a service name, and get the IP address of the closest server. It is aimed at minimizing the amount of effort required for system developers to make new and existing proximity aware. H.Ballani [3] proposed the towards predictable datacenter networks in which, for extending the tenant-provider interface to explicitly account for the network. Virtual network abstractions capture the trade-off between performance guarantees offered to tenants, their costs and provider revenue. N.Laoutaris [4] proposed Interdatacenter bulk transfers with netstitcher by which, rescue unutilized bandwidth across multiple datacenters and backbone networks and it used in the non-real-time applications, (i.e.) backups, propagation of bulky updates, and migration of data. Achieving the above is non-trivial since leftover bandwidth appears at different times, for different durations, and at different places in the world. To this end, NetStitchersystemisdesigned,implementedvalidated, that employs a network of storage nodes to stitch together unutilized bandwidth, whenever and wherever it exists. It gathers information about leftover resources, uses a store-and-forward algorithm to schedule data transfers, and adapts to resource fluctuations.

A. Greenberg [5] proposed the cost of a cloud: Research problems in data center networks, namely the data centers used to create cloud services represent a significant investment in capital and ongoing costs. To encounter this problem, network agility was increased, and providing incentives to shape the resource consumption in cloud.

Issues in cloud:

In cloud computing the resource allocation is one of the main issue. And also it leads to the low bandwidth and high delay. It comes under the scalability problem. Scalability is one of the QOS factor. Due to cloud increasing nature the scalability problem is occur. Compared to other QOS factor scalability is one of the main issue to resolve. In existing approaches, the user request is allotted to the datacenter based on greedy and locality algorithm. But it does not provide the optimal solution. It may lead to the datacenter may overloaded otherwise underutilized. Therefore delay may occur to deliver the resource to the user. It was one of the main

issue in cloud. Our proposed algorithm surely provides the better result compared with greedy and locality algorithm. Cloud computing stores the data and resources in open environment. The amount of data storage increases quickly in open environment. So, load balancing is one of the main challenge in cloud environment. Load balancing is helped to distribute the dynamic workload across multiple nodes to ensure thatno single node is overloaded. It helps in proper utilization of resources .It also improve the performance of the system.

System architecture:

The architecture aims to provide the high bandwidth and low delay to the end user. In our proposed system genetic load balancing algorithm is used. The workload is shifted from the overloaded datacenter to underutilized datacenter. Their Delay doesn't contain the response time inside data centers. Genetic load balancing algorithm solve the load balancing problem effectively. The controller play a vital role, in which it controls all functions. If user send request to the provider, then provider send that request to the controller. Then controller send one key to the user. In this, controller allocate the request to the datacenter which is underutilized. Afterwards user can download that file. It provides the optimized bandwidth and low delay, because of this efficient resource allocation. At last graph is generated based on the file name and requesting time.

Module description:

File Upload: In this module, Data Provider has to upload files in a cloud server, so he/she should login first. Then only he/she can be able to do it. Data provider can directly login by giving his/her username and password. Both the username and password should match, then only the provider will be considered as a valid person to upload the file. After login, Data provider should upload the file in cloud server in an encrypted format. In our project we used drive hq as a cloud server. So all the files are stored in drive hq cloud storage. For encrypting the file, we used DES algorithm for security purposes. It makes the file more secure.

Cloud Storage:

Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the Physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. Cloud storage services may be accessed through a co-located cloud computer service, a web service application programming interface (API) or by applications that utilize the API, such as cloud desktop storage, a cloud storage gateway or Web-based content management systems. All the files uploaded by the data provider are stored in drive hq cloud server.

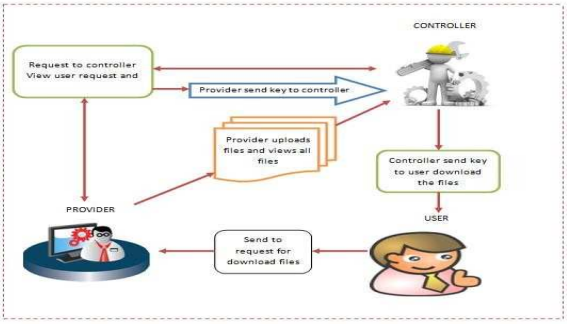
File Request:

In this module, user needs to download the files from cloud server; he/sheshould register first. For that he needs to fill the details in the registration form. These details are maintained in a database. In this module, any of the above mentioned person have to login, they should login by giving their name and password. Both the username and password should match, then only he/she will be considered as a valid user otherwise invalid user. After login data user can view all files uploaded by the data provider but user cannot download the file without key. So he/she will give key request to data provider.

Key Request Processing:

In this module, Data user's key request is processed by data provider. Data provider can accept user request or he can also reject the request. When the data provider accepts the request and then the request will be forwarded

Fig 1:- Architecture



to controller. Controller has the direct login to application by giving username and password. After Login, Controller can view user's key request for the files. Then controller will generate key to the user's mail id. Generate graph based on the overall user requests and generate graph based on the all the uploaded files.

Controller Maintenance:

In this module, we formulate the design of request allocation under those requirements as an optimization problem to avoid the delay to the user. To solve the optimization problem, we have implemented the genetic load balancing algorithm. Here we have designed three

data centers which has the request capacity as 15 for each. Controller only can decide from which data center, the user can download the file. After downloading the file, the capacity will be reduced automatically and here theload is balanced. The data center which has the maximum balance can be activated at each request.

Download:

In this module, key is generated to user's mail id by the controller. Then the user can give key for the particular file, then the file automatically retrieved from the drive hq cloud server. Then the AES algorithm performs decryption operation on the file and downloaded. User will get the original file without delay.

Algorithm:

Genetic load balancing algorithm:

Concepts of virtualization are used in cloud which leads to have a load balancing in the cloud. Virtualization means giving a logical name to physical resources and whenever this name is referred it will point towards corresponding physical resource. Multiple users will access cloud at same time and it is very necessary to serve them all with minimum response time and better service. For this reason load balancing is taken in to effect to balance the request of multiple users on virtual machines evenly. It is said that Load balancing is a NP-Complete Problem method because as the size of the problem increases the size of solution will increase too. Genetic Algorithm (GA) is much popular for solving NP-Complete problems. GA is one of technique which belongs to the class of evolutionary algorithms which generates solutions inspired by natural evolution. This algorithm implement in dynamic cloud environment and it used soft computing approach. And it provides better performance compare to RR and FCFS algorithm. The advantage of this algorithm is easily handle a vast search space, applicable to complex objective function and may avoid being trapping into local optimal solution. GA's implementation is based on three steps. Selection operator, Mutation and crossover operator.

Algorithm

if processor p is idle,

if last_message_number (p,2)=0 then last_message_number(p,1)=last_message_n umber (p,1)+1 last_message_number(p,2)=1

for i=1 to d dosend message to I direction of processor p(last_message_number(p,1),1,1)

od fi

if processor p is overloaded, min:=∞

walk:=∞

for i:=1to d do

if receive_message(p,I,4)=and receive_message(p,i,3) <walk and last_message_number(receive_message(p,i, 1),1)=receive_message(p,i,2)and last_message_number(receive_mesage(p,i,1)

,2)=1 then walk:= receive message(p,i,3) min:=i

fi od

if min≠∞

for i:= 1 to d do if i≠min then

send message to I direction of processors(receive message(receive_message(p,min,1), receive_message(p,min,2),receive_message (p,min,3)+1,1)

fi od fi

AES algorithm:

Advance Encryption Standard algorithm encrypts a 128 bit plaintext block in to a 128 bit cipher text block. The key size is either 128,192 or 256 bits. And input, output and key is represented as 4 bytes. In our proposed approach, key is generated to user's mail id by the controller. Afterwards, AES algorithm used to decrypt the text file. Then can download the requested file. In AES algorithm four steps are involved, these are byte substitution, shifting rows, mix columns, add round key.

Each and every round consists these four rounds. Similarly decryption performs the inverse operation of above four.

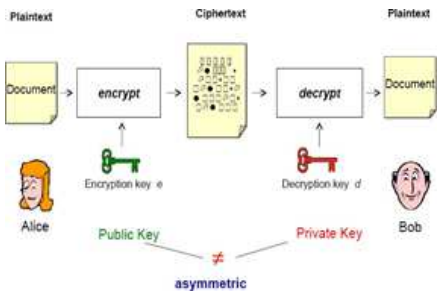


fig 2:- AES algorithm

Algorithm –AES Encryption

Cipher (byte in [16], byte out [16], key array

round key [Nr+1]) begin

byte state [16];

state = in;

AddRoundKey (state, round key [0]); for i = 1 to Nr-1 stepsize 1 do SubBytes (state);

ShiftRows (state); MixColumns (state);

AddRoundKey (state, round key[i]); end for

SubBytes (state); ShiftRows (state);

AddRoundKey (state, round key [Nr]); end

Algorithm –AES Decryption

Cipher (byte in [16], byte out [16], key array round key [Nr+1])

begin

byte state [16]; state = in;

AddRoundKey (state, round key [Nr]); for i = Nr-1 to 1 stepsize 1 do

Inverse SubBytes (state); Inverse ShiftRows (state);

AddRoundKey (state, round key[i]); Inverse MixColumns (state);

end for

Inverse ShiftRows (state); Inverse SubBytes (state);

AddRoundKey (state, round key [0]); end

Algorithm-AES key expansion

{

word temp

for(i = 0; i < 4; i + +)

w[i] = (key[4*i], key[4*i + 1], key[4*i + 2], key[4*i + 3]);

for(i = 4; i < 44; i + +)

{

temp = w[i – 1]; if (i mod 4 = 0)

temp = Sub Word(Rot Word(temp)) ⊕

Rcon[i/4];

w[i] = w[i-4] ⊕ temp

}

}

CONCLUSION

Nowadays Cloud computing is the trending and emerging technology. The one of the main issue in cloud computing is balancing problem. Cloud computing stores the data and resources in open environment. The amount of data storage increases quickly in open environment. So, load balancing is one of the main challenge in cloud environment. Load balancing is helped to distribute the dynamic workload across multiple nodes to ensure that no single node is overloaded. It helps in proper utilization of resources .It also improve the performance of the system. A number of techniques have been proposed by researchers for resource allocation. However There are many gaps to be filled by making these techniques more effective. More work is required in the area of cloud computing to make it acceptable by the cloud service consumers. This paper presents optimization of bandwidth and delay reduction in cloud environment using controller.

REFERENCES

[1]Amazon web services [Online], 2014.

[2]V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, “Unreeling netflix: Understanding and improving multi-CDN movie delivery,”in Proc. IEEE Conf. Comput. Commun., 2012, pp. 1620–1628.

[3]A. Cockcroft. (2011). Netflix in the cloud 17785.

[4]B. Wong and E. G. Sirer, “Closestnode.com: An open access, scalable, shared geocast service for distributed systems,” Operating Syst. Rev., vol. 40, no. 1, pp. 62–64, 2006.

[5]H. Ballani, P. Costa, T. Karagiannis, and

A. Rowstron, “Towards predictable datacenter networks,” in Proc. ACM SIGCOMM Conf., Toronto, ON, Canada, 2011, pp. 242–253.

[6]N. Laoutaris, M. Sirivianos, X. Yang, and

P. Rodriguez, “Interdatacenter bulk transfers with netstitcher,” in Proc. ACM SIGCOMM Conf., Toronto, ON, Canada, 2011, pp. 74– 85.

[7]A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: Research problems in data center networks,” ACM SIGCOMM Comput. Commun.Rev., vol. 39, no. 1, pp. 68–73, 2008.

[8]A. Singh, M. Korupolu, and D. Mohapatra, “Server-storage virtualization: Integration and load balancing in data centers,” in Proc. ACM/IEEEConf.Supercomput.,2008,p.53.