Optimal RP Based Dynamic Load Balancing in Amazon EC2 Environment

Sandipkumar Patel[1*], Ritesh Patel[2], Parth Shah[1]
[1] Department of Information Technology, C.S.P.I.T., CHARUSAT, Anand, India.
[2] Department of Computer Engineering, C.S.P.I.T., CHARUSAT, Anand, India.

* Corresponding author. Email:-sandippatel872@gmail.com
Manuscript submitted November 27, 2015; accepted April 19, 2016
doi: 10.17706/jcp.12.4.354-361

Abstract: One of the objective of cloud computing is to provide Resources as a Services to the client, in which resources are retrieved from cloud service provider efficiently and effectively over the Internet. To manage effectively the available resources of the cloud provider, resources use load balancing techniques to which has certain issues like load estimation, remote node selection, system stability, fault tolerance, performance indices, reduce cost of ownership, load level comparison, SLA violation etc. Accuracy can be implemented in form of efficient load balancing techniques. Dominant to this issue, it is necessary to develop optimal and efficient load balancing algorithms for various type of load, i.e. network load, CPU load, and Memory intensive applications. It also helps to avoiding a situation where some nodes are over loaded while others are idle or doing little work. The research carried out in this paper , the purpose of this research to improve response time by modeling behaviors of i) Progressive queue of node & ii) Request priority queue , which positively impact on optimization of resources as well as cost.

Key words: Cloud computing, datacenter, load balancing, scheduling, resource allocation, virtual machine.

1. Introduction

In current era, Cloud computing is the next big technology in the world of IT which deliver the IT services from a location other than from where it is located.

Cloud computing refers to computing on the internet, as opposed to computing on a desktop. Definition of cloud "Cloud computing defined as such structured model that defines computing services where resources as well as data are retrieved from cloud service provider via internet through some well-formed web-based tool and application" [1].

Nowadays, Up and Down time of any IT industry is unpredictable so it is require some flexible platform in which customer or client can increase capacity or add capabilities based on their resources requirement (resources can be a storage, platform, software, computation power and bandwidth) that is why cloud computing come in to picture which contains subscription-based Services, in real time over the internet, and extends existing IT's competences. Some of the most prominent cloud services providers are Amazon EC2, Google apps, Microsoft Azure, Google App Engine. On-going development and growth of cloud with its vast benefits are restricted by inconsistencies and challenges that exist in its present state like data security, energy management, load balancing etc. In cloud computing platform load balancing is applicable at two level in cloud computing [2].

VM level: Mapping done between an applications which are uploaded on the cloud to virtual machine, the

load balancer assigns the requested VM to physical computers which balance the load of numerous applications among PCs.

Host level: Mapping done between virtual machine and host resources which help to proceeds multiple incoming requests of application.

To understand it more specifically, for example: AWS cloud provider uses the Round Robin algorithm for load balancing or resource allocation. So, the adoption of virtualization technology is the promise of crafting a more dynamic and active IT infrastructure which is increased flexibility, adjust capacity on demand to better meet the needs of application workload owners, and to reduce overall costs of ownership. But in order to improve resource utility, resources must be allocated properly and load balancing must be guaranteed [3]. Therefore, Load balancing has always been a key component to building out any cloud computing architecture whose objective is to ensure that all computing resource are distributed efficiently and fairly so at good end can improves resource utility and performance.

The remainder of this paper is organized as follows: In Section 2, we cover Related Work. In Section 3, we discussed about Amazon EC2. In Section 4, Implementation architecture discussed. In Section 5, Proposed Solution.is discussed. In Section 4 Results are given and finally the conclusion of proposed work is presented.

2. Related Work

As technology growing faster, there are huge amount of user on internet to manage and fulfil their requirement, load balancer come in to picture which essentially ensure that they get spread workload equally to the all available server without any delay which help to accomplish a high user satisfaction, Maximum Throughput with minimum Response Time [4].

In general, scheduling algorithms categorized into two key approaches based on decisions making process: Static scheduling algorithms and Dynamic scheduling algorithms. Static algorithms like Round Robin, Earliest Task First [5], Insertion Scheduling Heuristic [6], Modified Critical Path [7] and Randomize are much simpler as compared to dynamic algorithms and some of them are based on Bounded Number of Processor which are more suitable for small distributed environment. Dynamic Scheduling algorithms like Throttle load balancing algorithm, Task scheduling algorithm, DSL [8], Enhance equally distributed load balancing algorithm, Mapping Heuristic [9], Active Monitoring are intelligently manage load and some of are self-adapting and guarantee of the efficient scheduling and some of are based on Arbitrary Processor Network which more suitable for huge distributed environment.

In Randomize scheduling algorithm, load is distributed randomly to available VM by selecting one arbitrary number and forward current connection to that arbitrarily selected VM. Drawback is that it will not equally distribute load. Out of all, Simplest scheduling strategies which is used in real environment is Round Robin. This policy is straightforward and use the mechanism of time slice to divided time into small number of parts which is further allocated to VM [10], [11] .Problem with this strategy is that, it makes decision only based on current state and it will not check whether particular VM is over utilized before allocation [10]. To solve this problem up to some level, Darji [12] proposed an algorithm “Dynamic Load Balancing for Cloud Computing Using Heuristic Data and Load on Server” this algorithms continuously monitor available server and based on that it will assign the weight factor which continuously changing after monitoring completed, so its help to find out over utilize server and improve resources utilization.

In Active Monitoring Load Balancing is dynamic and it stores information about VM and currently number of request assigned to which VM. Limitation of this policy is that every time it finds least loaded VM only, it does not check whether selected VM is previously utilize or not and due to this drawback available VM is not utilize maximum time. To solve this problem Shridhar G. Damanal [13] proposed an algorithm “Optimal Load Balancing in Cloud Computing by Efficient Utilization of Virtual Machines”, which helps to improve maximum

utilization of available VM. Throttled load balancing algorithm is totally based on best fit VM in which job manager have list or make index of VM and when client request is arrived, job manager find out best suitable VM to fulfil it. This policy performs well compare to active monitoring and round robin [10]. Drawback of this algorithm is that whenever request is arrived, it searches index from top, it does not take accounting of advance parameter like processing time of request [1]. S.Damanal and G. Ram Mahana Reddy [14] Proposed an algorithm "Load Balancing in Cloud Computing Using Modified Throttled Algorithm".

Proper resource allocation is key point for cloud providers which helps to improve response time, maximum utilization of available resources and increase profit. J Bhatia, *et al*. [1] proposed algorithm "HTV Dynamic Load Balancing Algorithm for Virtual Machine Instances in Cloud" which work dynamically for optimal usage of resource utilization. K C Gouda [15] proposed algorithm "Priority based resource allocation model for cloud computing" which Minimize Resource Wastage and Provides Maximum Profit.

In cloud computing, it is necessary to design good mechanism for efficient resources allocation and increase the utility of cloud server which fulfil current requirement of big crowd of user on cloud instead of user will lead to poor performance.

3. Amazon EC2

Amazon Elastic Compute Cloud (EC2) is a web service that enables you to launch and manage server instances in Amazon's data centers using APIs or available tools and utilities [16]. AWS provides 10 category of major services and around 39 total sub services. It provide storage, cloud-based computation and other functionality that enable organizations and individuals to deploy applications and services on an on-demand basis and at affordable prices [17]. Amazon EC2 assigns public IP address to created instance for communication with internet and other AWS product like Amazon S3. It's easy to access by HTTP protocol. EC2 provides the ability to place instances in multiple locations.EC2 locations are composed of Regions and Availability Zones [17].

EC2 provides facility to use, Preconfigured templates for your instances, known as Amazon Machine Images (AMIs), that package the bits you need for your server (including the operating system and additional software) [16] and after creating instance and installing necessary software ,user also create their own modified AMI.

AWS Security Features help you to keep your data and systems secure like to log in to your instance, Amazon EC2 uses key pair for public–key cryptography to encrypt and decrypt login information. It use HTTPS for secure network access, allows to create Virtual Private Cloud (VPC) which enables enterprises to connect their existing infrastructure to a set of isolated AWS compute resources via a Virtual Private Network (VPN) connection, and to extend their existing management capabilities [17].

Elastic Load Balancing (ELB) is also one of the well-known services of EC2 which automatically distributes incoming application traffic across multiple Amazon EC2 instances in specific cloud region which help to achieve greater levels of fault tolerance in your applications and higher availability. Currently Amazon ELB only supports Round Robin (RR) and Sticky session Algorithms.

AWS provides one of the most useful tools for cloud user which is CloudWatch. Amazon CloudWatch monitors your AWS resources and the applications you run on AWS in real-time and collect statistics raw data then convert this raw information into readable form like generate metrics and graph for different parameter like CPU Utilization, latency, sum request etc.

4. Implementation Architecture

Proposed architecture is selected to monitor research parameters given in proposed algorithm. As shown in Fig. 1 Basic components of architecture are Cloud Controller, Node controller, Virtual Machine, Agent, User and Load balancer.

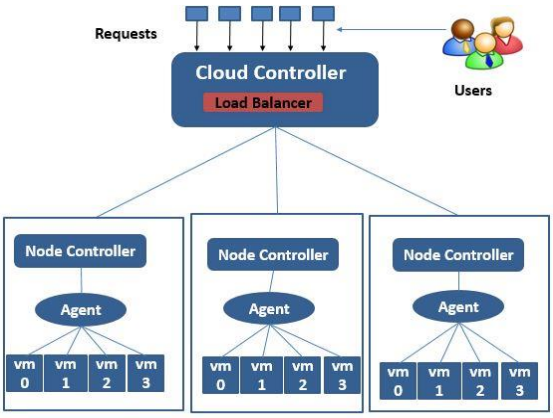

Fig. 1. Architecture [18].

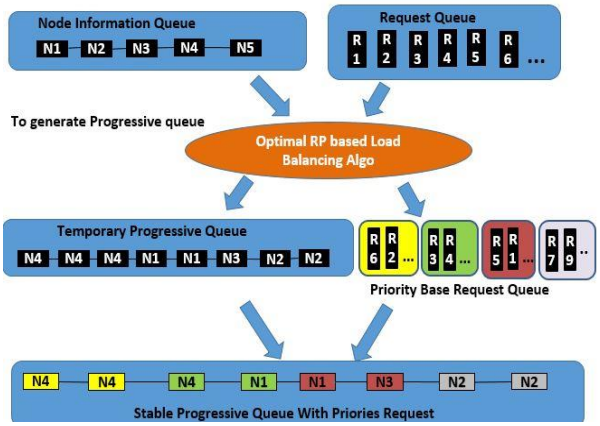

Fig. 2. Working model.

Cloud Controller: It is first step and root of whole architecture in cloud computing. It provides web interface to user and also authenticate request coming from the users and pass to node controller.

Node controller: It is next step after cloud controller where requests are processed. VMs running on NC are called instances in cloud. NC interact with OS and hypervisor.

Agent: Agent are the services which monitors and audit VM.

User: Users who access the services of cloud.

VM: Separate instance is created for every user on demand services.

First of all users request for the services offered by cloud provider ,request can be send by users to Cloud Controller and after that request can be given to the respective node controller where all the VMs has been stored with individual VM ids. Load balancer algorithm would be implemented on cloud controller which balance the load on node. Initially load balancer distribute load to node in round robin fashion. Cloud Controller will allocate the resources to particular VM as per the Request made from the user. Then VM processing the request and response to the node controller.

5. Proposed Algorithm

In our proposed algorithm, we proposed efficient and optimal dynamic load balancing algorithms which continuously monitoring the resources on VM for knowing current status and based on that it will design queue of VM by assign weight factor and also create request priority queue. So high priority request is forward to less utilize instance which helps to reduce overload rejection and also improves response time, processing time and also reduce cost up to some level.

In above Fig. 2 help to understand how proposed work is implemented. Here proposed Optimal RP based dynamic load balancing algorithm generate two queue by using node information queue and request queue.

Temporary Progressive Queue (TPQ) which contain information like free memory, performance and load on specific node given by Optimal RP based dynamic algorithm and also contain list of node information for future allocation in next iteration.

Request Priority Queue (RPQ) which put the incoming request in priority bucket for better allocation. Priority is decide according to the specific need.

Using above two queue **Stable Progressive Queue (SPQ)** is generated by assigning high priority request to less utilize node in TPQ. SPQ will swap by TPQ for next iteration of Optimal RP based dynamic load balancing algorithm. Queue is updated every time after completion of instance monitoring. If new request of client arrives then current pointer of queue is moved ahead if particular node capacity is over.

In Below Fig. 3 and Fig. 4 which helps to understand proposed solution. In Fig. 3 processing request without priority which overloaded the node. In Fig. 4, first of all different type of request is sorting in

descending order. Then queue of node is generated according to weight factor. In below figure node-4 has highest weight factor means it is capable to handle more number of request for example 4.5 means capable to handle 4 request. Here highest priority assign to video request because it's required more resources so start assigning the request form first node of queue in descending order. Using this proposed work, number of over utilize nodes are reduce, its means maximum utilize resources existing node.

Algorithm

Algorithm: - Optimal RP Based Dynamic VM Load Balancing
Input: -Node Basic information (free memory and processor), Performance, User Requests
Output: - Increase the performance (Response time) and efficient utilization of available resources.

1. Users send request for different services to cloud controller.
2. ORPDLB retrieve node information and request details form cloud controller.
3. ORPDLB Generate Two queue and efficiently utilize VM.

a. Progressive Queue of node

Two parameters are consider to calculate progressive queue
- Load On the server
- Current Preformation of server

L_A :- Load Factor
P_B :- Performance Factor
$P_{B(avg)}$:- Average of current response time.
Q_t :- WeightFactor
q_t :- Progressive queue parameter

Step:-1
- Find the load factor (L_A) of each available node
- L_A = Total Resources- Used Resources.
- Where A is free memory which in terms of %.

Step:-2
- Find the Performance (P_B) of node.
  - Response time = Finish time -Arrival time + Transmission delay.
- $P_B = P_{B(avg)}$ – (previously calculated $P_{B(avg)}$)
- $P_B = P_B / (P_{B(avg)})*100$// For P_B in terms of %.

Step:-3
- Calculate Q_t factor by subtracting Performance from Load.
- $Q_t = L_A - P_B$.
- If Q_t is less than zero than Q_t =0.

Step:-4
- Identify the Minimum value of Q_t from all available node.
- Node having value of Q_t =0 is not consider for in calculation.
  - Min_ Q_t = min (all Q_t's)
  - Min_factor = Min_ Q_t

Step:-5
- $q_t = Q_t$ / min_factor
- Generate Progressive queue based on value of q_t.

b. Request Priority

1. Read client request data from HTTP header field.
2. Retrieve Content type from http request header.
3. It can be video, audio, text, image and application.

```
IF (Content Type = Video
{            Assign Priority=First
} else if (Content Type = Application)
   {    Assign priority = Second
    } else IF (Content Type = Audio)
    {    Assign priority = Third
     } else IF (Content Type = Text)
     {    Assign priority = Fourth
     }
```

4. Check For Efficient utilization of VM
   - Create table /index of currently allocated request to each VMs
   - Find least loaded VM from table
   - Check least loaded VM is used in previous iteration.
5. Assign Request to node in descending order based on request priority.
6. Still Request is in queue?
   Yes →update node information for further allocation.
   No→ Exit

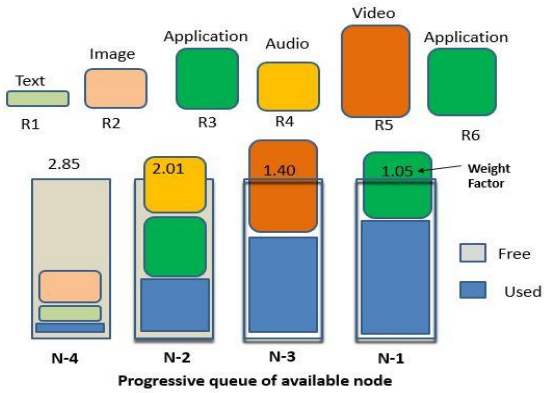

Fig. 3. Logical scenario without priority of request.

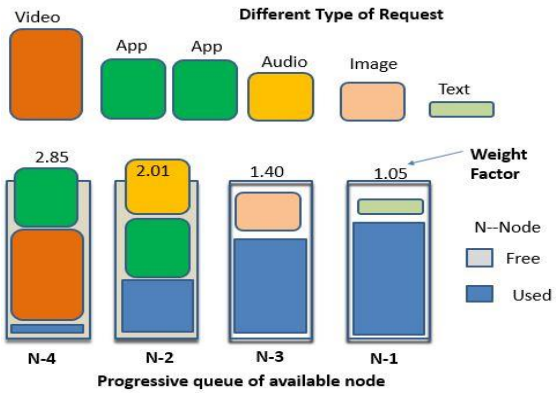

Fig. 4. Logical scenario with priority of request.

6. Results

AWS platform is used to execute proposed work on real environment. For proposed work experiment here two instances is created and this two instances is used in proposed optimal RP based load balancer to serve the incoming request. Here, Experiments done with different number request send to same configuration of VM in single datacenter. Compare the results with default load balancer results. After that one can conclude that the proposed Optimal RP based load balancing policy is how much good compare to existing one.

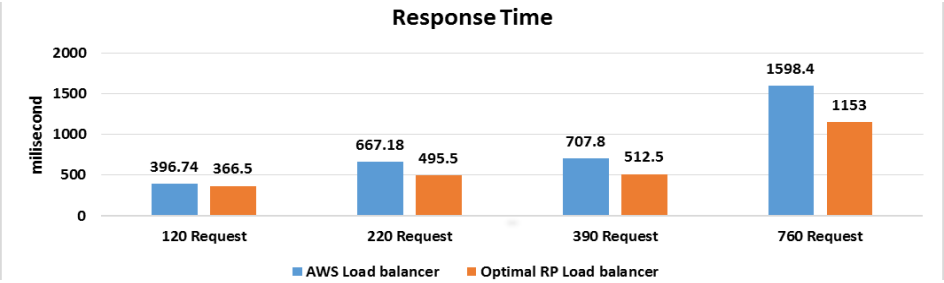

Fig. 5. Results.

Now here graphs for response time are shown in Fig. 5 for each configuration. So it can be concluded that proposed Optimal RP based load balancing gives the good result compare to default AWS load balancer in terms of response.

7. Conclusion

After carrying out the above research work, there are various static and dynamic load balancing algorithms. Each one has some drawback. In some algorithms user request is randomly assigns or assigned in sequence. So proposed load balancing algorithm mainly focus on generating progressive queue of node based on current load on that node and performance of that node. Second, it generate priority queue base on type of content in request. Third, find least loaded VM to efficient utilization of available VM which directly impact on response time. Major issue of load balancing algorithms are performance, response time of node and processing time of node. Using above work the issue can be resolved and it will be helpful to improve response time and processing time. This approach will be helpful for dynamically distribute the load based on user request priority to improve response time.

References

[1] Bhatia, J., Patel, T., Trivedi, H., & Majmudar, V. (2012, December). HTV dynamic load balancing

algorithm for virtual machine instances in cloud. *Proceedings of 2012 International Symposium on InCloud and Services Computing* (pp. 15-20). IEEE.

[2] Tai, J. Z., Zhang, J. M. *et al*. A R A: Adaptive resource allocation for cloud computing environments under bursty workloads. 978-1-4673-0012-4/11 ©2011 IEEE.

[3] Cherkasova, L., Gupta, D., & Vahdat, A. (February 2007). *When Virtual Is Harder than Real: Resource Allocation Challenges in Virtual Machine Based on Environments.* (Technical Report HPL-2007-25).

[4] Kaur, R., & Pawan, L. (2012). Load balancing in cloud computing. *Proceedings of International Conference on Recent Trends in Information, Telecommunication and Computing.*

[5] Hwang, J. J., Chow, Y. C., & Anger, F. D. (1989). Scheduling precedence graphics insystems with inter-processor communication times. *SIAM J Comput,* 244-257.

[6] Rewinin, H. E., Lewis, T. G., & Ali, H. H. (1994). *Task Scheduling in parallel and Distributed System Englewood Cliffs,* 401-403. New Jersey: Prentice Hall.

[7] Wu, M., & Gajski, D. (1990). Hypertool: A programming aid for message passing system. *IEEE Trans Parallel DistribSyst,* 330-343.

[8] Sih, G. C., & Lee, E. A. (1993). A compile-time scheduling heuristic for Interconnection-constraint heterogeneous processor architectures. *IEEE Trans Parallel DistribSyst,* 175-187.

[9] Rewinin, H. E., & Lewis, T. G. (1990). Scheduling parallel programs onto arbitrary target machines. *J Parallel DistribComput,* 138-153.

[10] Nusrat, P., Amit, A., & Ravi, R. (May 2014). Round robin approach for VM load balancing algorithm in cloud computing environment. *International Journal of Advanced Research in Computer Science and Software Engineering, 4(5),* 34-39.

[11] Shah, M. R., *et al.* (2013). Allocation of virtual machines in cloud computing using load balancing algorithm. *International Journal of Computer Science and Information Technology & Security.*

[12] Darji, V., Jayna, S., & Rutvik, M. (Jul.–Aug. 2014). Dynamic Load Balancing For Cloud Computing Using Heuristic Data and Load on Server." *IOSR Journal of Computer Engineering, 16(4),* 59-69.

[13] Domanal, S. G., & Reddy, G. R. M. (2014). Optimal load balancing in cloud computing by efficient utilization of virtual machines. *COMSNETS,* 1-4.

[14] Domanal, S. G., & Reddy, G. R. M. (2013, October). Load balancing in cloud computingusing modified throttled algorithm. *Proceedings of 2013 IEEE International Conference on Cloud Computing in Emerging Markets* (pp. 1-5). IEEE.

[15] Gouda, K. C., Radhika, T. V., & Akshatha, M. Priority based resource allocation model for cloud computing. *International Journal of Science, Engineering and Technology Research*, 2278-7798.

[16] Amazon Elastic Computing Cloud. From www. aws.amazon.com/ec2

[17] Qi, Z., Lu, C., & Raouf, B. (2010). Cloud computing: State-of-the-art and research challenges. *J Internet ServAppl, 1,* 7–18, Springer.

[18] Krimit, S., Harshal, T., & Parth, S. (2012). Architecture for securing virtual instance in cloud. *International Journal of Computer Science and Information Technologies, 3(3),* 4279–4282.

Sandip Patel obtained his bachelor's degree in information technology from CSPIT Changa, Gujarat Technological University Ahmedabad in 2012. He had completed the M.Tech in computer engineering from CSPIT, Changa, CHARUSAT University in 2015.Currently he is working as Assistant Professor at the Department of Information & Technology, CHARUSAT, Changa, Gujarat. His research interest include cloud computing, networking and documentation in latex.

Ritesh Patel obtained his bachelor's degree in computer engineering from Ganapat Univesity, Mehsana, Gujarat in 2002; master degree in computer engineering from DDU in 2004, Nadiad, Gujarat and pursuing PhD in area of cloud computing from CHARUSAT, Changa, Gujarat. Currently he is working as Associate Professor at U & P U. Patel Department of Computer Engineering, CHARUSAT, Changa, Gujarat. His research interest include parallel computing, next generation networks, advanced computer architecture and cloud computing.

Parth Shah is Professor at the Department of Information Technology at Charotar University Science & Technology, Changa, Anand, Gujarat. He graduated with a bachelor of engineering in computer engineering and post graduated in computer engineering and now is pursuing a PhD. His research interest includes information & network security, cloud computing and computer organization & architecture.