iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 Issue: V Month of publication: May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62952

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Implementation of Improved Scheduling Load Balancing Algorithm in Cloud Environment

Ganesh V. Gujar[1], Satish R. Devane[2], Ratnadeep R. Deshmukh[3]
[1]Research Scholar, [2]Principal, [3]Head CS/IT
CSIT, Dr. BAM University, Aurangabad, India

***Abstract:** The advantages of cloud computing can accordingly be seen in that; There is low price when the traditional computing systems are compared, there is a wide scope for expansion that is always needed by the business and fast availability of the resources. Since service users are a huge group controllable through a system and performance of the service is done according to traffic, user experience is as result incomparable and necessary service availability factors are catered for. It is that demand such as huge amounts of traffic and extensive utilization of huge cloud has become so common in the case of this before. Load balancing is the most important component of the cloud computing architecture, making it possible to provide the necessary uptime, capacity, reliability, and performance of web applications that are implemented by exploiting the load balancing technique. The specified train of thought addresses the implementation of as well as dealing with the procedures or types of cloud computing as well as working with the load balancing system. This ratio will be used as a benchmark to highlight different algorithm implementations that include (1) Location-Distance-Cost Policy, (2) reconfigurable, and (3) load balancing such as round-robin, throttled, location diversification, and weighted among others. Besides the previous one, we see the other function that is to remove this curtain of the dense fog using cloud computing load balancing technologies.*

***Index Terms:** Cloud Computing, Load Balancing, Load Balancing Algorithms, Round Robin, Throttled, Equally Spread, Cloud Analyst, Best Performing VM algorithm.*

I. INTRODUCTION

A. Cloud Computing

Cloud computing is the best way of supporting shared resources, software, information technology frameworks like software, platform, infrastructure, and many others. It reduced and required minimal administrative work. Users will not use their own computer foundation but take advantage of IT service providers where they will host your applications and store your data and receive payment based on consumption. This allows an aggregation and de-aggregation work arrangement where sprawling can vary from year to year without spending anything on capital planting on infrastructure software. [1].

1) Cloud Computing Services

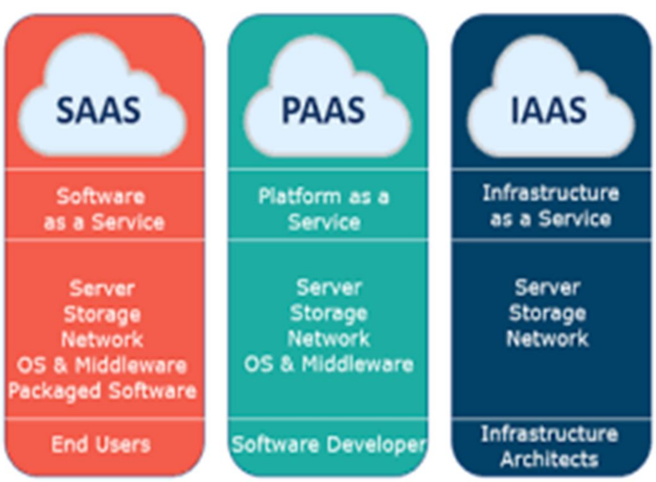


Figure 1.1 Cloud Services

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

a) SAAS

SaaS, or software as a service, works in that a third-party provider hosts software applications and makes them available to customers online. Instead of purchasing and installing the software, users pay a subscription fee or based on usage to access the software programs and data through a web browser or mobile app.

This means that users do not need special hardware or software to access the applications. SaaS presents several business opportunities compared to traditional software delivery models [2]. SaaS providers install, maintain, and upgrade the software, making it easier for users and freeing up IT computer resources.

Furthermore, SaaS software promulgates pricing that accommodates changes granted the solution is scale-based. Another benefit of this model is the immense variety of business functions the SaaS application can perform. Anywhere from customer relationship management to human resources to bookkeeping financial management and project management are accomplished via SaaS [3]. A few prominent providers include Salesforce when it comes to CRM, Workday for HR, QuickBooks for accounting, and Microsoft Office 365 which covers productivity and collaboration. Users may pay for just the resources they consume and have no costs or further obligation.

b) PaaS

Platform as a Service is another cloud computing model where a third-party provider offers consumers access to a cloud-based platform to develop, run, and manage applications. The consumer does not need to build or manage the underlying infrastructure and typically only pays a subscription fee to the provider. A Paas provider offers hardware environments, the operating system, runtime, middleware, and development tools.

Compared to traditional models of application development and hosting, PaaS has several benefits. Instead of infrastructure management, it enables developers to concentrate on application development and innovation. Scalable and flexible environments that can handle varying workloads are provided by PaaS providers along with built-in tools and services for application testing, deployment, and management.

Developers have the chance to go into the market more quickly with their applications as the time taken to develop an application is reduced since no hardware or infrastructures are used using PaaS [4]

c) IaaS

Infrastructure as a Service (IaaS) is a cloud computing service that can be accessed through the internet. IaaS offers customers virtual machines, storage, and networking infrastructures which they can deploy to run and manage their applications and workloads. Generally, IaaS users pay a subscription fee or usage-based charges to the provider based on how many resources they consume [5].

IaaS has several benefits over traditional infrastructure deployments. It enables customers to allocate and then scale back resources in line with demand without any investment in physical hardware or infrastructure set-up or maintenance. Additionally, IaaS users have access to various types of computational resources like VM instances, storages, databases, and networks that can be employed for developing their own applications and services. Lastly, IaaS grants its clients immense power over their environments through its ability to self-configure or self-administer [6].

A selection of services is offered by IaaS providers including virtual machines, object storage, block storage load balancers as well as databases which businesses may utilize in building up their own apps and features. Among the most popular ones are Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP) and IBM Cloud [7].

II. LETURATURE SURVEY

A. Static Environment [13]

These algorithms do not depend upon the current state of the system and have prior knowledge regarding systemresources and details of all tasks in an application like, capacity of nodes, processing power, memory.

B. Dynamic Environment [14]

These algorithms take decisions concerning load balancing based upon the current state of the system and cannot rely on the prior knowledge of different resources.

Table 2.1: Existing and proposed load balancing algorithms

Sr No	Problem statement	Proposed Scheme	Finding	Algorithm compare	Parameters Used	Reference
1	Size Data of cloud user's request is same or different	Hybrid algorithm	Minimize response Time, Data center Request and Data Center Processing Time	Round Robin, Throttled algorithm and Equally Spread Current Execution algorithm.	1. Response Time 2. Data Center Processing Time 3. Data Center Request Serving Time	[9]
2	How to Use Cloud Resourses Effective and Efficiently for Cloud Computing	Hybrid algorithm	Identify Least Loaded VM Identify Busy/Ideal VM, Assign Load to All VM	Round Robin, Throttled load Balancing, Active Monitoring Algorithm	1: No of Request/VM 2: State of VM 3: Load/VM	[10]
3	Low Performance-Failed Server Due to Overload on Vm	Failure Detection algorithm, Resource Allocation Algorithm, Monitoring	Detect Overloaded VM, Migrate traffic to under loaded VM	General Technique, Time Monitoring Stsyem	1: Health Check 2: Load on VM	[11]
4	To Solve Mult objective Optimization Problem	Hybrid algorithm for Mult objective algorithm	Minimize Response Time, Cost, Bandwidth	Priority Scheduling algorithm, Single Objective scheduling algorithm	1: Response Time 2: Cost 3: Bandwidth	[12]
5	Resource allocation in Hybrid Cloud is Difficult	Adaptive Workflow Management System	Dynamic resource allocation in heterogeneous Cloud	Heterogeneous Earliest Finish Time (HEFT), DAG	1: Cost 2: Deadline 3: Makespan	[13]
7	Previous Proposed algorithm works on single SLA parameter	Novel Scheduling heuristic based on SLA	Achieve load balancing, Resource Utilization	RR, FCFS,	1: Resource Utilization 2: Load balancing	[14]
8	Dynamically Provisioning VM for higher profit	CA(Combinational Auction provision)	It performs better in case of resource and profit	CA Greedy method	1: Cost 2: Resource Utilization	[14]
9	Users get slow Service when they are directed to Overloaded Data Center or Overloaded Link	Dynamically Direct User request to less busy or High Performing Data Center	Enhance the User Experience by always Directing Queries to Best Performing Data Center		QOS parameters.	[12]

C. Advanced Load Balancer

1) Hardware Based Load Balancer (Application Delivery Controller)

Application delivery controller (ADC) essentially functions as a load balancing optimizing end-user performance, reliability, data center resource use and security for enterprise applications. The basic problems with load balancer were originally created, by producing high availability, scalability, and security [15]. ADC can obscure the original goals by providing fundamental role played by load balancer in ADCs. ADC overcome these problems which are created by load balancing were [16].

a) *High Availability:* Availability is the easiest ADC attribute to tie back to the original load balancer as it relates to all the basic load balancer attributes like scalability, high availability, and predictability. However, ADCs take this even further than the load balancer did. Availability for ADCs also represents advanced concepts like application dependency and dynamic provisioning. ADCs can understand that in today's world, applications rarely operate in a self-contained manner; they often rely on other applications to fulfill their design. This knowledge increases the ADC's capability to provide application availability by taking these other processes into account as well. The most intelligent ADCs on the market also provide programmatic interfaces that allow them to dynamically change the way they provide services based on external input. These interfaces enable such things as dynamic provisioning and the addition and/or subtraction of available servers based on utilization and need in peak period. i.e. 24x7 availability [17]
b) *Scalability:* Scalability means increasing or decreasing capacity. It is the ability to meet demand and maintain performance of an application. It includes scaling down as well as up, particularly in cloud computing environments where elasticity is a means to contain costs and make efficient use of computing resources. Depending on an organization's current and future cloud initiatives, it may lead to advantageous to include the ADC in emerging cloud computing frameworks. Scalability is the capability of dynamically, or easily, adapting to increased load without impacting existing performance. Service virtualization presented an interesting opportunity for scalability; if the service, or the point of user contact, was separated from the actual servers, scaling of the application would simply mean adding more servers which would not be visible to the end user [18].
c) *Performance:* Performance enhancement was another obvious extension to the load balancing concept. Load balancers improved the performance of applications by ensuring that connections were not only distributed to services that were available, but also to the services with the least number of connections and/or processor utilization. This made sure that each new connection was being serviced by the system that was best able to handle it. Later, as SSL/TLS offload became common for load balancing, it reduced the amount of computational overhead of encrypted traffic as well as the load on back-end servers improving their performance as well. Performance measured by

- Connection Capacity (Provide max. number of Connections)-e.g. F5 –BIG-IP (LTM) local traffic manager handles 192millions connections with 320 Gbps throughput.
- Transaction per Seconds-Rather to measure the ability to pass packets or open and close the connections compare with decisions per second.
- SSL (Secure Socket Layer)-Use to secure entire web application. Also measure Transaction per second (TPS) and bulk encrypted rates that secure the data can be exchanged and what rate.

2) Software Based Load Balancer (Application Delivery Controller)

Because hardware-based ADCs or load balancers pose many of the same challenges as other hardware components. They're limited and monolithic, and not designed for cloud-based or virtual environments. They also typically fall under the same infrastructure team that handles networking, servers, and so on. As such, they are acquired as part of the infrastructure budget, and they are costly to purchase, maintain, and scale. That expense, along with complexity of configuration, means that each device is often tasked with providing ADC services for a very large number of applications examples is multi tenancy. It's not commercially practical to devote a single hardware ADC to each application, but there's a tradeoff in ADC configurations that require shared resources. So, it loses performance and flexibility. Limitations of hardware-based ADCs are High costs, Takes Time to get going, Requires specific expertise, Rigid in Nature [19]. Software-based ADCs, on the other hand, can tackle these challenges head-on and allow users cost effective applications to create business value through innovative, engaging, and high-performing applications. Software based ADCs provides,

a) Rapid deployment and flexibility
b) Achieve more power and better performance
c) Able to run anywhere
d) Cost saving

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

III. PROPOSED ALGORITHM

The "BestPerformingVM load balancing algorithm" aims to select the most suitable virtual machine (VM) for load assignment based on performance metrics. Here's a simplified explanation of how this algorithm typically works,

1) *Initialization:* The algorithm starts by initializing variables such as bestVmId (the ID of the best-performing VM found so far) and bestPerformance (the performance score of the best-performing VM).
2) *Performance Evaluation:* For each VM in the data center, the algorithm evaluates its performance based on various factors such as remaining resources (CPU, RAM, bandwidth).
3) *Scoring:* The algorithm computes a performance score for each VM based on the evaluation results. This score typically represents the VM's suitability for handling additional load, with higher scores indicating better performance.
4) *Selection:* The algorithm selects the VM with the highest performance score as the best performing VM. If multiple VMs have the same highest score, the algorithm employs Round Robin algorithm to make the selection.
5) *Load Assignment:* Once the best performing VM is identified, the algorithm assigns the incoming load or task to this VM, considering factors such as its current workload and capacity.
6) *Update:* After load assignment, the algorithm may update relevant data structures or state variables to reflect the changes in VM allocation and performance. Performance score will get re-calculated.
7) *Return:* Finally, the algorithm returns the ID of the selected VM, which can then be used by the system to route incoming requests or tasks accordingly.

Overall, the BestPerformingVM load balancing algorithm aims to optimize resource utilization, enhance system performance, and ensure efficient handling of workload by selecting VMs with the most available resources and capacity to handle additional load effectively.

A. Pseudo code for BestPerformingVM algorithm (BP-VM)

Input: List of VMs VM_List, Map of VM states vmStatesList, Map of VM allocation counts vmAllocationCounts
Output: VMid selected VM for load assignment

```
1. Initialize variables:
  - bestVmId = -1
  - bestPerformance = Double.MIN_VALUE

2. For each VM in VM_List:
  a. Get VM performance metrics:
    - remainingBandwidth = calculateRemainingBandwidth
      (VM)
    - remainingCpu = calculateRemainingCpu (VM)
    - remainingRam = calculateRemainingRam (VM)
  b. Calculate a performance score for the VM based on metrics:
    - performanceScore = calculatePerformanceScore
      (remainingBandwidth,
      remainingCpu, remainingRam)
  c. If performanceScore is higher than bestPerformance:
    - Update bestPerformance to performanceScore
    - Update bestVmId to the current VM's ID

3. Update VM allocation counts and states:
  a. Increment the allocation count for bestVmId in
    vmAllocationCounts.
  b. Update vmStatesList to mark bestVmId as BUSY.
```

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

4. Return bestVmId as the selected VM for load assignment
Functions:
- calculateRemainingBandwidth (VM):
  - Simulate remaining bandwidth calculation based on the number of cloudlets
  - Example: return Math.max (0, VM. getTotalBandwidth () - VM. getAllocatedBandwidth ())
- calculateRemainingCpu (VM):
  - Simulate remaining CPU calculation based on the number of cloudlets
  - Example: return Math.max (0, VM. getTotalCpu () - VM. getAllocatedCpu ())
- calculateRemainingRam (VM):
  - Simulate remaining RAM calculation based on the number of cloudlets
  - Example: return Math.max (0, VM. getTotalRam () - VM. getAllocatedRam ())
- calculatePerformanceScore (remainingBandwidth, remainingCpu, remainingRam):
  - Combine the factors into a comprehensive performance score
  - return 0.5 * remainingBandwidth + 0.4 * remainingCpu + 0.1 * remainingRam

IV. PERFORMANCE COMPARISON

Configuring user bases, broker policies, data center configurations, and internet characteristics among the components of the cloud analyst tool to analyze different load balancing algorithms with different scenarios. Performance analysis of BestPerformingVM Load Balancing algorithm is compared with RR, Throttled, Honeybees, Threshold and Equally Spread algorithms in closest distance and Optimized Response Time service broker policies in different scenarios.

1) Scenario1: Performance comparison of 4 different Data Centers, all data centers are in region 0 and 5 userbases in region 2. VM -RAM-512MB, CPU-10000MIPS, Bandwidth-1000MB, Number of VM in each DC-5, Policy-Closet data center and Optimized response time.

Table 4.1- Result-Closest Data Center

Algorithm	Response time(ms)	Processing time(ms)
BP-VM	425.76	125.44
RR	425.76	125.43
TH	425.76	125.44
ES	425.76	125.44
HB	425.76	125.44
THR	426.48	126.20

Table 4.2- Result-Optimized Response Time

Algorithm	Response time(ms)	Processing time(ms)
BP-VM	424.67	125.39
RR	424.68	125.41
TH	424.67	125.39
ES	424.67	125.4
HB	424.67	125.4
THR	425.29	126

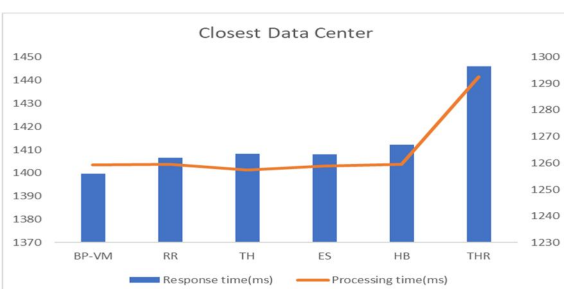


Figure 4.1 Closest data center

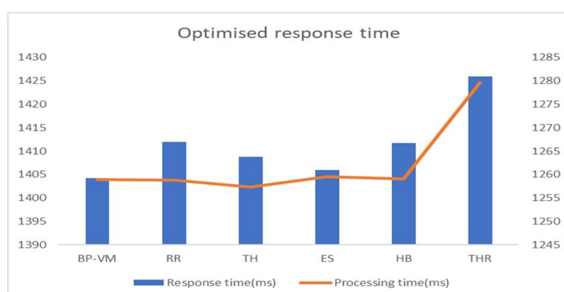


Figure 4.2 Optimized Response Time

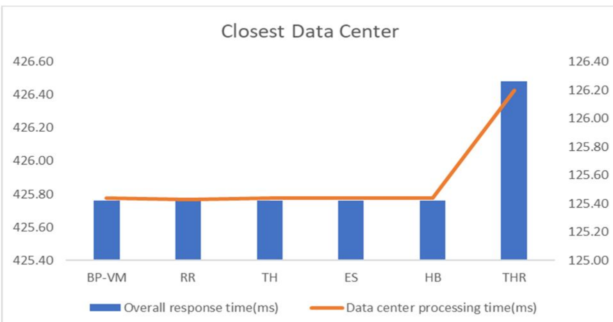
2) Scenario2: Performance comparison of 5 Data Centers is in region 0,1,2,3, and 4, Userbases in region 1, 2,3,4, and 5. VM - RAM-1000 MB, CPU-10000MIPS, Bandwidth-100 00MB, Number of VM in each DC-5, Policy-Closet data center and Optimized response time.

Table 4.3 Result-Closest Data Center

Algorithm	Response time(ms)	Processing time(ms)
BP-VM	1404.23	1258.98
RR	1411.97	1258.76
TH	1408.83	1257.36
ES	1405.95	1259.56
HB	1411.7	1259.06
THR	1425.91	1279.68

Table 4.4 Result-Optimized Response Time

Algorithm	Response time(ms)	Processing time(ms)
BP-VM	1404.23	1258.98
RR	1411.97	1258.76
TH	1408.83	1257.36
ES	1405.95	1259.56
HB	1411.7	1259.06
THR	1425.91	1279.68



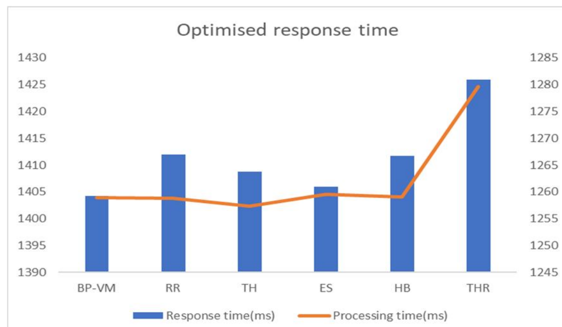


Figure 4.3 Closest data center

Figure 4.4 Optimized Data center

V. CONCLUSION AND FUTURE SCOPE

Several load balancing algorithms have been simulated to process user requests in a cloud environment. The scheduling criteria for each algorithm, such as average response time and data center service time of various data centers and VMs are found after analysis. Here, we compared the algorithmic performance using VMs across different regions, User Bases, and different Data Centers with their different configuration. The results are displayed in the graph. When compared to other algorithms, the Best Performing VM load balancing algorithm performs well overall in all scenarios. Our future scope is to test the performance of Best Performing VM load balancing algorithm in heterogeneous cloud environments.

REFERENCES

[1] Nimisha Gupta, Bharti Chauhan et al, 2014. Cloud Computing: Comparison with Previous Technique and Research Challenges. International Journal of Computer Applications, Volume 85 – No 8, 43-47.
[2] Sinha, G., & Sinha, D. 2020. Enhanced Weighted Round Robin Algorithm to Balance the Load for Effective Utilization of Resource in Cloud Environment. EAI Endorsed Transactions on Cloud Systems, 6(18). https://doi.org/10.4108/eai.7-9-2020.166284.
[3] Jonathan Stuart Ward, and Adam Barker. 2014. Observing the clouds: a survey and taxonomy of cloud monitoring. Ward and Barker Journal of Cloud Computing: Advances, Systems and Applications, 1-30, DOI 10.1186/s13677-014-0024-2.
[4] Tharam Dillon, Chen Wu and Elizabeth Chang. 2010. Cloud Computing: Issues and Challenges. International Conference on Advanced Information Networking and Applications, 27-33.
[5] Majid Mehmood, Kinza Sattar. Load Balancing Approach in Cloud Computing. Information Technology & Software Engineering, Volume 5, Issue 3 - 1000153, http://dx.doi.org/10.4172/2165-7866.1000153
[6] Kapil Kumar, Abhinav Hans. 2015. Differences and Problems Task Scheduling Algorithm -A Survey. International Journal of Hybrid Information Technology, 145-152, Vol.8, No.6, http://dx.doi.org/10.14257/ijhit.2015.8.6.15
[7] Abhijit Aditya, Uddalak Chatterjee, Snehasis Gupta. 2015. A Comparative Study of Different Static and Dynamic Load Balancing Algorithm in Cloud Computing with Special Emphasis on Time Factor. International Journal of Current Engineering and Technology, 1898-1907, Vol.5, No.3.
[8] Geethu Gopinath P P, Shriram K Vasudevan. 2015. An in-depth analysis and study of Load balancing techniques in the cloud computing environment. International Symposium on Big Data and Cloud Computing, 427 – 432.
[9] Mayanka Katyal, Atul Mishra. 2013. A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. International Journal of Distributed and Cloud Computing, pp 5-14Volume 1 Issue 2.
[10] Haozheng Ren, Chao Yin. 2015. The Load Balancing Algorithm in Cloud Computing Environment. International Conference on Computer Science and Network Technology, 925-928.

[11] Dina Farouk Altayeb, Fatima Abdelghani Mustafa. 2019. Analysis on Load Balancing Algorithms Implementation On Cloud Computing Environment. International Journal of Innovative Research in Advanced Engineering, 32-36 Issue 02, Volume 6, doi://10.26562/IJIRAE.2019.FBAE10081.

[12] Chukwu eke, Chiamaka Ijeoma et al. 2019. Review of Hybrid Load Balancing Algorithms in Cloud Computing Environment. International Journal of Trend in Research and Development, 31-37, Volume 6(6).

[13] Mayanka Katyal, Atul Mishra. 2013. A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. International Journal of Distributed and Cloud Computing, 5-14Volume 1 Issue 2.

[14] Geethu Gopinath P P, Shriram K Vasudevan. 2015. An in-depth analysis and study of Load balancing techniques in the cloud computing environment. International Symposium on Big Data and Cloud Computing, 427 – 432.

[15] https://www.citrix.com/products/netscaler-adc/resources/what-is-an-adc.html. What is ADC.

[16] Consideration when choosing an ADC- White paper. https://f5.com/resources/white-papers/top-considerations -when-choosing-an-adc.

[17] Reasons to choose ADC -White paper. https://www.nginx.com/wp-content/uploads/2014/07/NGINX_Ebook_5Reasons_02_0728b.pdf.

[18] F5 load balancing methods- http://www.dasblinkenlichten.com/f5-ltm-ve-load-balancing-methods.

[19] Barracuda load balancing ADC https://www.barracuda.com/assets/docs /Datasheets/Barracuda_Load_Balancer_ADC_DS_US.pdf