Configuring Auto Scaling Servers with Elastic Load Balancer on Real Time Applications

[1]Dr. Kalli Srinivasa Nageswara Prasad*, [2]K.Viswa Prasad, [3]S.V.V.D.Venu Gopal

Sasi Institute of Technology & Engineering, Andhra Pradesh, India
[1]ksnprasad@sasi.ac.in, [2]viswa@sasi.ac.in, [3]venugopal@sasi.ac.in

Abstract

In the current age of digitalization, cloud technology has inflated apace. It was a relatively new technology that relied on the web and centralized servers to store and operate several apps. The newest version of utility computing that replaces its space at numerous information centres permits the user for economical computing by consolidative storage, memory and process. Because the use of the web grows, businesses square measure transferring their operations from ancient computing to cloud computing, leading to a rise within the range of cloud users and hypertext transfer protocol request stress. As a result, a load leveling configuring with Auto scaling answer is needed to supply purchasers with congestion free and dependable on demand service. To handle the demand, the load balancer determines once to begin and stop virtual machines within the cloud and with Auto scaling service; anyone will simply increase or decrease the capability of the backend as traffic fluctuates.

Keywords: *Cloud Computing, Load Balancer, Auto Scaling, AWS EC2, Round Robin, Resource Provisioning*

1. Introduction

As a result of its ability to offer world-class services to all of its customers, cloud computing has found success in the business sector. Cloud computing is the most current technological shift away from traditional systems. Cloud computing is gaining popularity among businesses because it requires less capital outlay and requires less upkeep. In the cloud, virtualization, grid computing, autonomic, and utility computing are all components. Hosted services are referred to as "hosted" in this context, although the term encompasses much more. In this model, consumers only pay for the resources they utilise. Access to Internet-based information technology capabilities and services is unfettered for customers of cloud computing services. The IT industry's operations have altered substantially as a consequence of cloud computing if this can be achieved using load balancers. Because it needs less infrastructure and upkeep, it offers additional benefits to the IT business. It is better to utilise virtual machines than physical ones since the former do not end up in landfills as electronic garbage. Cloud computing is the newest and most exciting technology in the world of computer science, drawing the attention of all the experts. Using the cloud as a computer platform is a hybrid approach to computing (services). As a general term, it means using the internet to access computer resources and applications.

2. Infrastructure as a service

IT infrastructure services are those that can be provided on demand for a fee. [7, 8]. Processing, storing, and networking are all made easier. There are a number of services that begin with hosting, including web servers, storage, computing hardware, operating systems, virtual instances, load balancing, internet connectivity, and bandwidth provisioning. IAAS has the ability to distribute resources and scale dynamically. IAAS providers may use Load Balancing as a technique to specify conditions for scaling up and down of applications using an automatic scaling facility. Internet bandwidth, low latency, and lowcost communication are required for this service. Because of the lower resource requirements and ongoing maintenance costs, the cloud is becoming increasingly popular among businesses. The pay-as-you-go policy of the cloud makes it an honest option for organizations and provides Quality of service, making it an attractive option for businesses. [15]

Load balancing mechanisms are critical in cloud environments, and an auto scaling mechanism must even be provided to keep the system from overloading and crashing in order to provide reliable service to clients. The load balancing mechanism distributes the load among one or more nodes of a cloud system, allowing for an efficient service model with an auto scaling feature. The platform was dynamically scaled up and down based on the amount of traffic it received from clients, which saved money and physical resources. In cloud computing, latency-based routing is a new concept that provides global clients with load balancing supported DNS latency through the assorted hosted zones .[15]

The following are the Load Balancing parameters:

- **Dynamicity:** LBs and servers may be added or removed according on the current work load.

- **Per-connection-consistency(PCC):** All packets that belong to the same connection are sent to the same server.
- **Uniform load distribution**: using sophisticated techniques for load balancing to make effective use of the servers.
- **Efficient packet processing:** The LB should have little effect on communication delay.
- **Resilience**: it should be impossible for a client to "clog" the LB and likewise the servers with fake traffic.

3. Objective of Load Balancing

Since the incoming load is unpredictable, and the variable also depends on many other factors, the importance of load balancing in a cloud system cannot be overstated. The following criteria must be met by an honest load balancer.

- Significantly enhance operational efficiency.
- In case of system failure caused by excessive load, it must be fault-tolerant and provide a fallback route.
- Most essential, it must maintain the integrity of the system and carry out regular, uninterrupted activities

4. Literature Survey/Related works

It has been shown that the CP approach with various switching mechanisms can optimize resource utilization and service provisioning, as has been investigated by Junaid et al. Different strategies, such as IDLE, NORMAL, and OVERLOADED, can be implemented in a CP-based LB model. To make well informed decisions, we rely on the partition's current status. During an IDLE state, no work is being done by a partition. Similarly, when a partition is being used for processing but its load is normal, it is referred to as the "NORMAL" status. A partition that has exhausted its processing capacity and all of its resources are being used is known as an OVER-LOADED state, on the other hand. Load balancing is helped by these states. However, the refresh time has not yet been calculated in order to have the best means of refreshing and further optimization. This thesis fills in that void.

For load balancing, Tiwari et al. [13] studied cloud partitioning techniques. In their model, cloud resources are partitioned and a load balancer is defined for each partition. In order to optimize cloud computing performance, a load balancer analyses the load on each partition and makes strategic decisions regarding job scheduling.

Different approaches to load distribution were examined by Mesbahi and Rahmani [17] and other researchers. They divided load balancing methods into two categories: static and dynamic approaches. There are two types of dynamic LB algorithms: those that are distributed and those that are not distributed. Probabilistic and deterministic algorithms are the two types of static algorithms that can be found. They also looked at various approaches to balancing the load.

It was found that Randles et al. [2] looked into distributed load balancing algorithms as the cloud works in distributed environments. They looked at active clustering, biassed random sampling, and HoneyBee foraging behaviour as three different distributed load

balancing algorithms. The system's throughput is tested on a variety of resources that are readily available for experiments.

The Hybrid Job Scheduling (HJS) algorithm, described by Javanmardi et al. [12], aims to improve LB in cloud computing. Decisions were made based on job length and virtual machine capacity. For example, the degree of imbalance (DI), execution cost, and execution time are used to evaluate its performance. For load balancing, it concentrated more on CPU utilisation. Additionally, a fuzzy inference engine was used to improve the solution's performance.

In addition to improving cloud computing performance, K. Shyamala et al. [16] presented efficient resource allocation mechanisms in the Cloud. Different algorithms for allocating resources are examined. Green cloud computing, cloud resource allocation, user priority resource allocation, multi-dimensional resource allocation, and optimal joint multiple resource allocation are a few examples.

PA-LBIMM scheduling was developed by Chen et al. [6] as a way to ensure that the load-balance of a system is maintained. Improved-Min-Min mechanism was found to be superior to its predecessor (LBIMM). Both of these mechanisms' lifespans, or "makespan," are taken into account when estimating their performance. PA-LBIMM was able to reduce the time it takes to complete a task, but it was unable to prioritise the needs of the users.

Biogeography-based Optimization (BBO) was proposed by Kim et al. [10]. It's a job scheduling method that makes use of cloud load balancing. Genetic Algorithm (GA) was compared to this adaptive process and found to be superior

When it comes to LB, Tripathy and Patra [11] sought to reduce switching time and increase resource utilisation. The server's efficiency and throughput are improved by optimising the server's job scheduling to overcome the limitations of the protocols that are currently in use. In order to demonstrate its efficiency, the algorithm's time complexity has been limited to Autory out switching and job completion time.

When it came to task planning, Raja Manish Singh [14] had some thoughts to share. Particle Swarm Optimization (PSO), multi-objective techniques, and independent task scheduling and workflow-based task scheduling are some of the topics covered in this book. Cost, time, performance, execution time, and QoS (Quality of Service) were all taken into account.

IaaS was the focus of Schwiegelshohn and Tchernykh's [5] work on scheduling and long-term planning. The research takes into account a variety of SLAs and deadlines. They also looked at the benefits of parallel computing and how it affects scheduling. Parallel processing was compared to the performance of a single machine scenario. They tried out several scheduling techniques to see whether the theory was correct. Experiments were conducted to demonstrate that when using parallel computing, tasks may be completed on time.

Honey Bee Behavior Inspired Load Balancing was created by Babu and Krishna [9]. (HBB-LB). Priorities connected with separate tasks on multiple machines may also be supported. Comparatively, it is proven to be more efficient than other approaches, such as First in First out (FIFO). When making judgments, the HBB-LB considers a factor known as degree of imbalance. In addition, it utilises honey bees' behaviour to improve

resource usage by 27 percent. Reduces the frequency of migrations and maximises the use of virtual machines (VMs) In this context, it is restricted to balancing the workload of jobs that are not preemptively reliant

5. Implementation

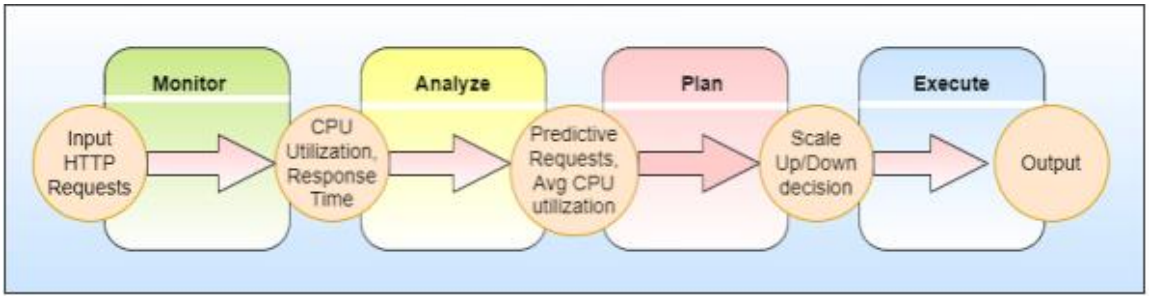
Auto scaling is used to perform scaling up and scaling down automatically. Load balancing will distribute incoming traffic to multiple targets. Auto scaling of Amazon EC2 servers based on traffic demand makes the process of launching and terminating automated. You can use Auto Scaling in order to ensure that you have enough EC2 instances available to handle the application load. Scalability and elasticity are included in the service An Auto Scaling group is a collection of EC2 instances that you create (ASG). AWS auto scaling allows you to set a minimum and maximum number of instances in your ASG, so that the number of instances doesn't exceed the size. Scaling policies can be used to determine how the ASG grows and shrinks (on-demand/dynamic scaling, cyclic/scheduled scaling). Scaling Plans can define the triggers and instances that are provisioned and reprovisioned. Create new EC2 instances in the cloud by using an AMI, a key pair, an instance type and a security group with the launch configuration template.

6. Amazon Elastic Load Balancing (ELB)

Amazon EC2 instances, containers, and IP addresses can be automatically distributed by ELB to handle incoming application traffic. ELB can handle a single Availability Zone or a number of Availability Zones. ELB has features such as automatic scaling, robust security, and high availability. It is possible to use a Load Balancer to perform basic load balancing on both layers 4 and 7 at the same time. When messages are delivered via Layer 4, load balancing takes place at the intermediate transport layer. Transmission Control Protocol (TCP) is an Internet Layer 4 protocol for Hypertext Transfer Protocol HTTP/TCP. Network packets are simply forwarded to and from the upstream server by Layer 4 load balancers, which do not examine their contents. At the application level, where the data in each message is handled, Layer 7 load balancing occurs.

HTTP is the most common Layer 7 protocol for website traffic on the Internet. While Layer 4 load balancers are capable of routing HTTP traffic, Layer 7 load balancers are far more capable of doing so, making them a better choice for TCP-based traffic. The message is read by a Layer 7 load balancer after the network traffic has been terminated. Based on the message's content, it can decide on load balancing.

7. MAPE (Monitor, Analyze, Plan, Execute)



Monitoring: The monitoring system collects data via http requests from a cloud environment. Measuring web application performance is used to determine the best auto-scaling methods. CPU usage and request response time are taken into account.

Analysis: Analysis involves a more in-depth look at the data that has been acquired thus far. Data from measurements and current system usage is combined with forecasts of future workloads to provide a complete picture of the present situation. Because there is always a delay between the establishment of resources for scaling choices, reactive is a complex method. From 120 to 180 seconds, the VM starting time varies.

Plan: Based on the predictive requests and average cpu utilization in analysis phase, the plan phase decides to scale up or scale down the servers.

Execution: In the planning phase, the implementation phase has already been chosen.Execution might take place after scaling up or down. APIs provided by cloud service providers are in charge of carrying out planned activities. It takes some time for VMs to set up, thus they are only accessible for a short amount of time.

8. Parameters Taken

- CPU Utilization (Percentage)
- Network in (Bytes)
- Network out (Bytes)
- CPU Credit Usage (Count)

9. Target Groups

Each load balancer will be allocated a target group. The load balancer uses Target Groups to choose which server to send traffic to. Because the load balancer examines the target group first, then the registered servers in it, and it can only distribute the load to those servers, every server should be registered to the target groups. It is impossible to spread the load equally when a server is not registered. If one of the servers goes down, the load balancer will automatically redirect traffic to a healthy server. When a Target Group Listener is formed, it is immediately linked to a target group. Listeners assume the role of narrator in this scenario. A port and an IP address may be used by developers to design more complex routing. Another benefit of load balancing is the ability to add and remove servers from target groups without affecting with the performance of other servers. Target Group defaults to utilizing the Round Robin method to distribute traffic across its targets (servers).

Round Robin Algorithm: The round robin technique distributes duties to the next VM in line regardless of how busy the current VM is. Neither resource capabilities nor work duration are taken into consideration by the Round Robin policy. As a consequence, lengthier tasks and more priority result in longer response times.

Weighted Round Robin Algorithm: The better-capable VMs get more jobs in the weighted round robin since it takes into consideration the resources of each VM. However, while picking the proper virtual machine, it didn't take into account the time required to complete the tasks.

Improved Weighted Round Robin Algorithm: The Using the improved weighted round robin method, the most appropriate VMs are selected based on their processing capacity, load, and the duration and priority of the tasks they have been assigned. As a result of using a static scheduling approach, VMs can be allocated according to their processing power, number of incoming tasks, and size. To assign a job, this algorithm takes into account both the VM's current workload and information from the previous section to make a decision. An additional cycle (like a loop) may be required to complete a task at run time, which may result in a longer completion time than was originally anticipated.

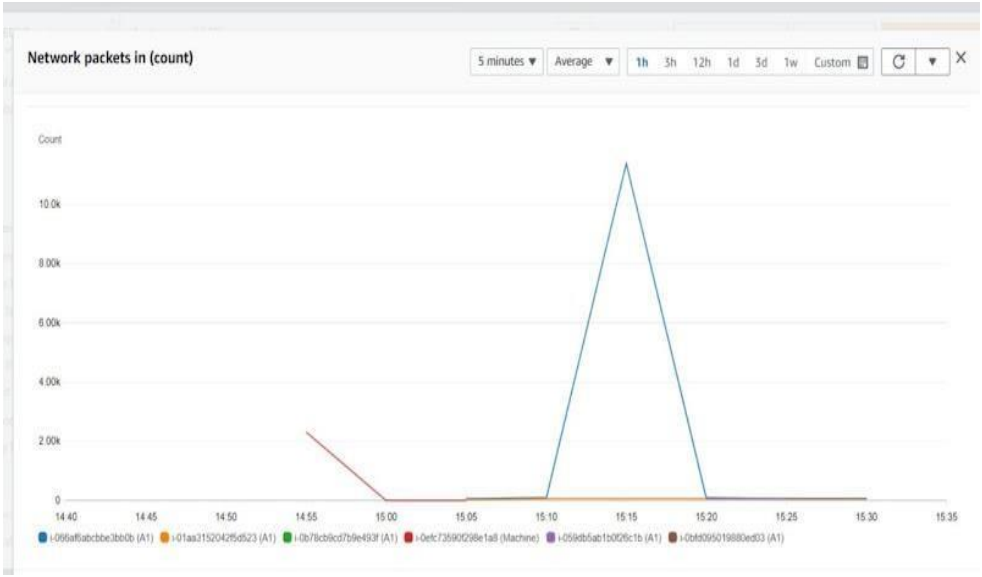
Rearrangement is done by moving jobs from overloaded VMs to free slots in other underutilized/underutilized machines, thus rescuing the scheduling controller. Every time one of the VMs performs a task, the load balancer uses resource prober to determine which VMs are under- or over utilized. If no unused VMs exist, the load balancer will not distribute jobs among them. Overloaded VMs will be shifted to underutilized/underutilized VMs as soon as they are discovered. The load balancer can only assess the load on the resources (VMs) once all of the tasks on each VM have been completed. At no point does it take a look at the resource's (VM) load as a whole in order to reduce the overhead on the virtual machines The number of task migrations and resource probe executions in the VMs will be reduced as a result.

10. Resource Monitoring

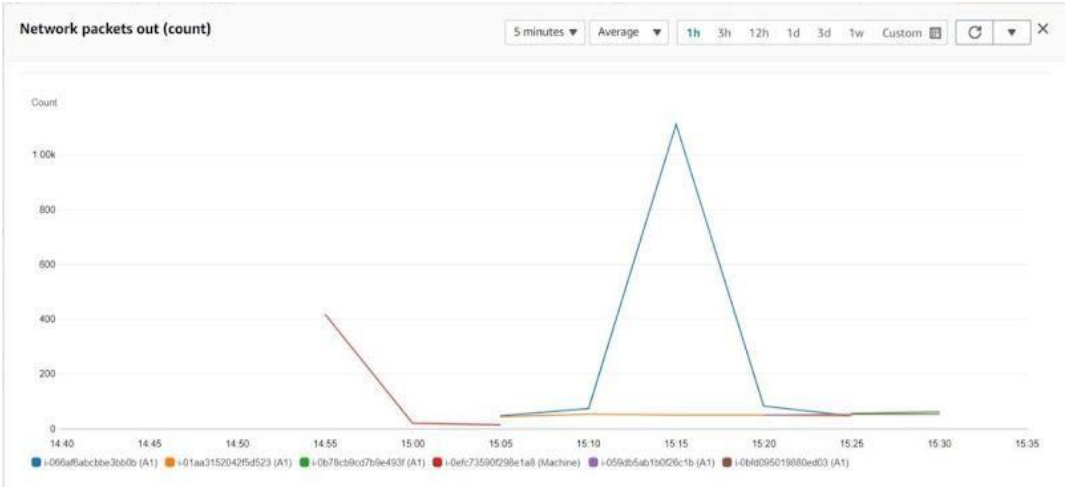
Using the scheduler's methods, the static scheduler locates the best VM and assigns tasks to it (basic round robin, weighted round robin, and improved weighted round robin). Run-time arrival jobs can be assigned based on which virtual machine is least utilised at the time of arrival. Every time it detects an idle or least-loaded VM, the load balancer/scheduler controller uses resource monitor data to decide whether or not to move the job from a heavily loaded VM. When jobs are assigned to VMs, the resource monitor communicates with the resource probers on each VM to gather information such as the capabilities of the VMs, the current load on each VM, and the number of jobs in the execution and waiting queues on each VM. For operational decisions, the task requirement estimator sends the predicted results to the load balancer for use in determining task completion time.

11. Results

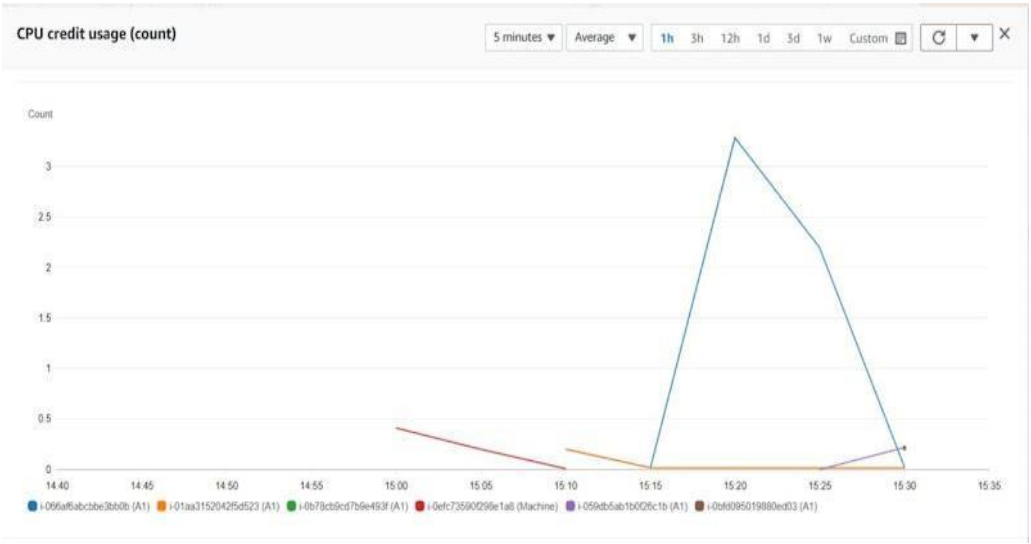
Network Packets In (count)



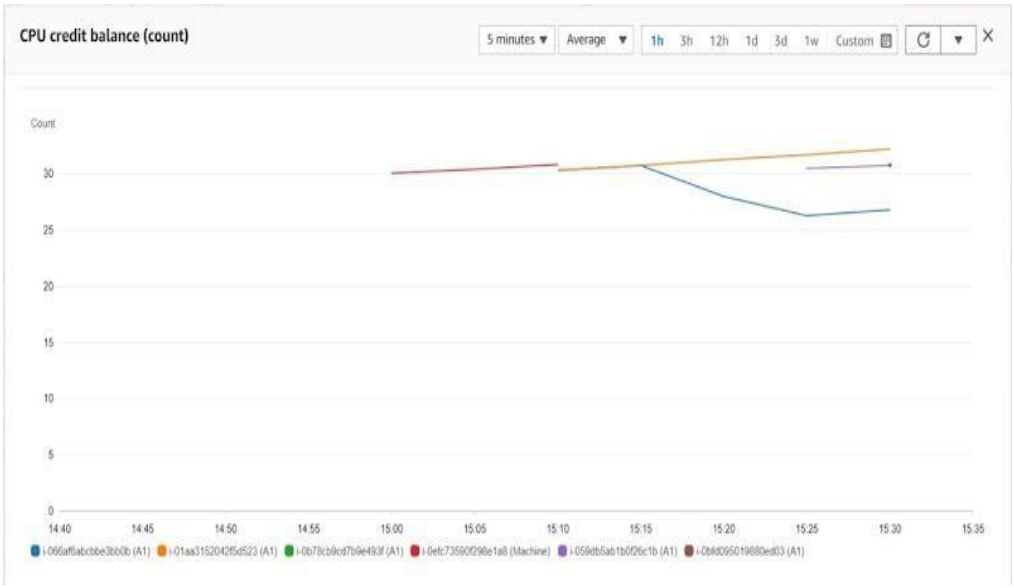
Network Packets Out (count)



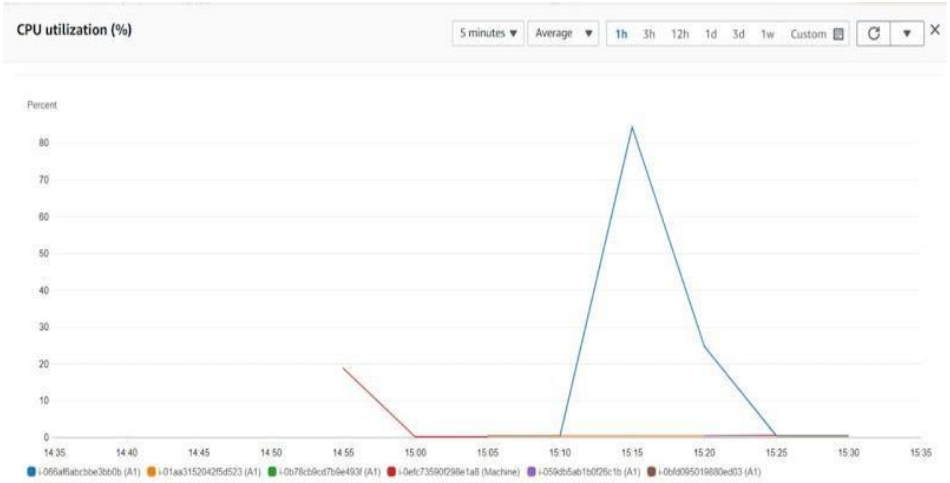
CPU Credit usage (count)



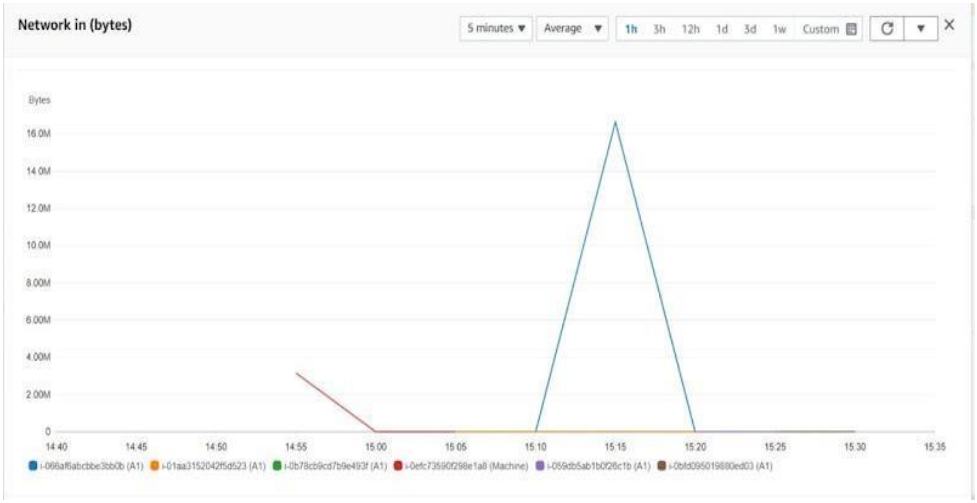
CPU credit balance (count)



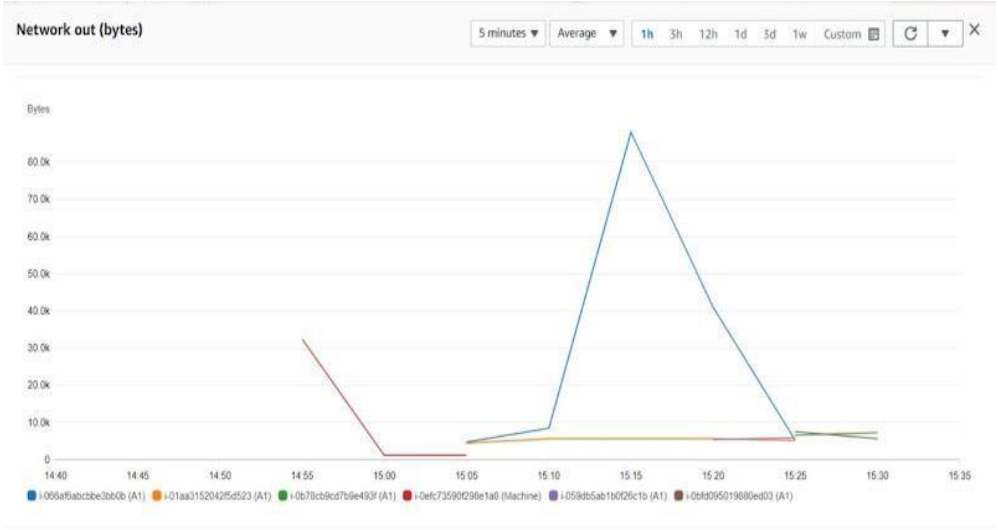
CPU utilization (%)



Network in (bytes)



Network out (bytes)



12. Conclusions & Future Scope

Cloud computing data centre load balancing has been a major challenge and an active research area for the past few years. For cloud computing environments, we've done a survey of the current load balancing techniques and solutions. In order to provide Continuous Services and multiple clouds in one environment with full privileges, a full multi cloud simulation must simulate the pool of requirements that satisfies all user requests. It is possible, for example, to link multiple clouds together while maintaining administrative control and the ability to implement various administrative, access, and security policies on each cloud. In addition, it includes the simulation of numerous users from various clouds simultaneously accessing the resources. We chose CPU utilization, N/W Input, and N/W Output & Fault Tolerance Mechanism as a result of parameter tuning. Configuration automation can be achieved by deploying instances using Ansible's Play-Books, while Automatic Incremental Updating of jobs created in Jenkins can lead to future integration with Ansible & Jenkins. We can also consider additional cloud load balancing solutions based on our three-level classification and conduct a trend survey for load balancing solutions.

References

[1] Ali M. Alakeel, A Guide to Dynamic Load Balancing in Distributed Computer Systems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.6, June 2010.

[2] Martin Randles, David Lamb and A. Taleb-Bendiab, A Comparative Study Into Distributed Load Balancing Algorithms For Cloud Computing, International Conference On Advanced Information Networking And Applications Workshops, pp. 551-556, 2010.

[3] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanazadeh, and Christopher, IPCSIT Vol-14, IACSIT Press Singapore 2011.

[4] N. Ajith Singh, M. Hemalatha, “An approach on semi distributed load balancing algorithm for cloud computing systems” International Journal of Computer Applications Vol-56 No.12 2012.

[5] Uwe Schwiegelshohn, Andrei Tchernykh, Online Scheduling For Cloud Computing And Different Service Level, IEEE, pp. 1061-1068, 2012.
[6] Huankai Chen, Professor Frank Wang, Dr. Na Helian and Gbola Akanmu, User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing In Cloud Computing, IEEE, 2013.
[7] Dan C. Marinescu, Cloud Computing Theory and Practice, Morgan Kaufmann, USA, Elsevier, 2013.
[8] Amazon web services cloud watch Website, November 2013.
[9] Dhinesh Babu L.D. And P. Venkata Krishna, Honey Bee Behavior Inspired Load Balancing Of Tasks In Cloud Computing Environments, IEEE, (13): 2292–2303, 2013.
[10] Sung-SooKim, Ji-Hwan Byeon, Hong Yu, Hongbo Liu D. Biogeography Based Optimization For Optimal Job Scheduling In Cloud Computing, Applied Mathematics And Computation, (247):P266-280, 2014.
[11] Lipsa Tripathy and Rasmi Ranjan Patra, Scheduling In Cloud Computing, International Journal On Cloud Computing: Services And Architecture, 4 (5):21-7, 2014.
[12] Saeed Javanmardi, Mohammad Shojafar, Danilo Amendola, Nicola Cordeschi, Hongbo Liu and Ajith Abraham, Hybrid Job Scheduling Algorithm For Cloud Computing Environment, IEEE, pp. 43-52, 2014.
[13] Mangal Nath Tiwari, Kamalendra Kumar Gautam, Dr. Rakesh Kumar Katare, Analysis Of Public Cloud Load Balancing Using Partitioning Method And Game Theory, International Journal Of Advanced Research In Computer Science And Software Engineering, 4 (2):807-812, 2014.
[14] Raja Manish Singh, Sanchita Paul and Abhishek Kumar, Task Scheduling In Cloud Computing: Review, International Journal Of Computer Science And Information Technologies, 5 (6):7940-7944, 2014.
[15] Ashalatha R et.al “Evaluation of Auto Scaling and Load Balancing Features in Cloud” in International Journal of Computer Applications (0975 – 8887) Volume 117 – No. 6, May 2015.
[16] K. Shyamala and T. Sunitha Rani, an Analysis on Efficient Resource Allocation Mechanisms in Cloud Computing, Indian Journal Of Science And Technology, 8 (9):814–821, 2015.
[17] Mohammadreza Mesbahi, Amir Masood Rahmani, Load Balancing In Cloud Computing A State Of The Art Survey, I.J. Modern Education And Computer Science, 3:64-78, 2016.
[18] Junaid, M., Sohail, A., Rais, R.N.B., Ahmed, A., Khalid, O., Khan, I.A., Hussain, S.S. and Ejaz, N., 2020. Modelling an Optimised Approach for Load Balancing in the Cloud. IEEE Access.
[19] Yoann Desmouceaux et.al “Joint Monitorless Load-Balancing and Auto scaling for Zero-Wait-Time in Data Centres” in IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 18, NO. 1, MARCH 2021.