ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Sp.Iss 7 , 2022

Analysis and Design of Load Balancing Techniques in Cloud

Naresh Vurukonda[1]

Department of Computer Science and Engineering,Koneru Lakshmaiah Education Foundation (KLEF), Deemed to be University, Vaddeswaram, Green fields, Guntur, Andhra Pradesh, India-522302.

naresh.vurukonda@kluniversity.in

V.Vidyasagar[2]

School of Technology Management and Engineering, SVKM'sNarseeMonjee Institute of Management Studies Deemed to be University,Hyderabad Campus, Mahbubnagar, Telangana,509301, India.

Email: vidyasagar24@gmail.com

DOI : 10.48047/IJFANS/11/S7/006

Abstract: Our objective is to develop an efficient load balancing algorithm like throughput, latency, Fault tolerance for example for the clouds of different sizes like virtual topology counting on the appliance requirement. Cloud computing involves virtualization, distributed system and some web services. A cloud consists of several elements like clients, data centre and distributed servers. Central to those issues lies the establishment of an efficient load balancing algorithm. The load is often CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among the various servers of a system to enhance both resource utilization and job reaction time while avoiding a situation where many of the servers are heavily loaded while other servers are idle or doing little or no work. Load balancing gives each one within the system or every server within the network does approximately the equal amount of labour at any instant of the given time. These techniques are often sender initiatives, receiver initiatives or symmetric type like combination of sender initiated and receiver initiated types.

Keywords: Cloud Computing; Load Balancing; Honey Bee Foraging Algorithm; Ant Colony Optimiza-tion Technique.

1. Introduction

In recent years, cloud computing has gained immense popularity due to its ability to provide flexible and cost-effective access to computing resources. It offers users the convenience of accessing shared services and resources over the internet, making it a highly attractive option for organizations. Despite its advantages, organizations considering a move to public cloud environments have concerns about the impact on their existing systems and networks. Cloud computing does pose challenges, particularly in terms of security, fault tolerance, and load balancing.

Cloud computing's appeal lies in its ability to offer on-demand services, easy maintenance, device and location independence, and cost-effectiveness. Major industry players like Google, Microsoft, and Amazon provide public cloud services, which are widely used due to their efficient computing and storage technologies.

However, managing cloud resources efficiently is a critical task. Cloud servers, equipped with various services, need effective resource scheduling and allocation. Proper utilization of resources is essential to avoid wastage and unnecessary costs. Efficient resource management

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Sp.Iss 7 , 2022**

involves determining how tasks and applications are distributed across servers, optimizing computing power, and ensuring effective storage allocation.

Cloud computing architecture comprises a front end, providing client interfaces for accessing cloud platforms, and a back end, consisting of servers, storage systems, and databases. The front end facilitates user interaction, while the back end manages and delivers the required resources and services.

2. Design/Methods/Modeling

Researchers and industries are actively exploring techniques to enhance resource management, improve security measures, and optimize cloud computing performance. Cloud computing continues to evolve, promising better services and solutions for users and organizations alike.

2.1 Algorithm:

1. It is straightforward, straightforward to implement, and starvation-free as all processes get justifiable share of central processor.
2. One of the foremost normally used technique in central processor planning as a core.
3. It is preventative as processes square measure allotted central processor just for a hard and fast slice of your time at the most.
4. The disadvantage of it is additional overhead of context switch.

2.2 Algorithmic rule for load balancing:

1. Creates same size of Cloudlets.
2. Cloud Coordinator divides the assigned Cloud task into same size of cloudlets
3. Produce Broker and User assigns the task to Cloud Coordinator(CC)
4. Cloud Coordinator sends cloudlets to Virtual Machine Manager and Virtual Machine Manager sends the list of the required resources to the Datacenters.
5. Request for the execution of the Cloudlet is shipped to the Virtual Machine by Virtual Machine Manager from the Host.
6. Cloudlet programing is completed in Virtual Machine in line with Shortest Job First program-ming policy.
7. Sends the dead job as Cloudlets in a very wrap file to the Virtual Machine Manager.
8. Virtual Machine Manager any passes the dead Cloudlets as wrapped file format to Cloud Coor-dinator.
9. Cloud Coordinator mixes all dead Cloudlets in wrapped file type combine to create the com-plete task.
10. CC sends the dead task in etch file format to the user/client.
11. Print the Result.

2.3 Round-Robin Load Balancing Algorithm::

Round-Robin Scheduling Algorithm is the simplest load balancing algorithm which selects the first server and assigns tasks in a round robin manner. In that case, though, it is only ap-plicable to cloud computing because some nodes may be heavily loaded and some may not be. The main advantage of this algorithm is that if the system fails it will not affect the en-tire

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Sp.Iss 7 , 2022**

system or the other servers but it will affect only the performance of the system.It is straightforward, straightforward to implement, and starvation-free as all processes get justifiable share of central processor. One of the foremost normally used technique in central processor planning as a core. It is preventative as processes square measure allotted central processor just for a hard and fast slice of your time at the most. The disadvantage of it is additional overhead of context switch.Here this round robin load balancer sends the request from the client to the servers in a round robin way means it sends all the client requests to the servers at a time if the sever is busy then this load balancer starts sending the requests from the starting server again and it can be implemented very easily. But, in some cases it cannot be divide the work that much effectively and it can provide work process efficiently.

2.4 SHORTEST JOB FIRST:

The Shortest Job First (SJF) scheduling algorithm operates on a priority basis and follows a non-preemptive approach, meaning once a processor is assigned to a process, it cannot be interrupted until the task is completed. SJF gives preference to processes with the smallest size, aiming to minimize execution time and enhance system efficiency by reducing average waiting time.

Despite its benefits, SJF encounters challenges, particularly in predicting a process's burst time before CPU execution, a task often difficult due to its unpredictable nature. If a process doesn't complete within the anticipated time frame, questions arise regarding the next course of action. To address this issue, experiments are conducted, and ongoing research explores the Dynamic Time Quantum SJF schedule to improve load balancing performance further.

3. Results and Discussion

```
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_1
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_1
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #3, Host #0
0.1: Broker: VM #3 has been created in Datacenter #3, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #2
0.1: Broker: Sending cloudlet 2 to VM #1
0.1: Broker: Sending cloudlet 3 to VM #3
0.1: Broker: Sending cloudlet 4 to VM #0
1.1: Broker: Cloudlet 0 received
1.1: Broker: Cloudlet 1 received
1.129126213592233: Broker: Cloudlet 2 received
1.129126213592233: Broker: Cloudlet 3 received
2.18: Broker: Cloudlet 4 received
2.18: Broker: All Cloudlets executed. Finishing...
2.18: Broker: Destroying VM #0
2.18: Broker: Destroying VM #2
2.18: Broker: Destroying VM #1
2.18: Broker: Destroying VM #3
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
Datacenter_0 is shutting down...
Datacenter_1 is shutting down...
Broker is shutting down...
Simulation completed.
Simulation completed.
```

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Sp.Iss 7 , 2022**

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	1	0.1	1.1
1	SUCCESS	2	2	1	0.1	1.1
2	SUCCESS	3	1	1.03	0.1	1.13
3	SUCCESS	3	3	1.03	0.1	1.13
4	SUCCESS	2	0	1.08	1.1	2.18

```
Starting CloudSimExampleRR...
Initialising...
Starting CloudSim version 3.0
Datacenter_0 is starting...
Datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_1
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.0: Broker: Trying to Create VM #3 in Datacenter_1
0.0: Broker: Trying to Create VM #4 in Datacenter_0
0.0: Broker: Trying to Create VM #5 in Datacenter_1
0.0: Broker: Trying to Create VM #6 in Datacenter_0
0.0: Broker: Trying to Create VM #7 in Datacenter_1
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: VM #4 has been created in Datacenter #2, Host #0
0.1: Broker: VM #6 has been created in Datacenter #2, Host #1
0.1: Broker: VM #1 has been created in Datacenter #3, Host #0
0.1: Broker: VM #3 has been created in Datacenter #3, Host #0
0.1: Broker: VM #5 has been created in Datacenter #3, Host #0
0.1: Broker: VM #7 has been created in Datacenter #3, Host #1
```

```
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #2
0.1: Broker: Sending cloudlet 2 to VM #4
0.1: Broker: Sending cloudlet 3 to VM #6
0.1: Broker: Sending cloudlet 4 to VM #1
0.1: Broker: Sending cloudlet 5 to VM #3
0.1: Broker: Sending cloudlet 6 to VM #5
0.1: Broker: Sending cloudlet 7 to VM #7
0.1: Broker: Sending cloudlet 8 to VM #0
0.1: Broker: Sending cloudlet 9 to VM #2
1.1: Broker: Cloudlet 0 received
1.1: Broker: Cloudlet 1 received
1.1: Broker: Cloudlet 2 received
1.1: Broker: Cloudlet 3 received
1.1654205607476635: Broker: Cloudlet 7 received
1.2754205607476636: Broker: Cloudlet 4 received
1.2754205607476636: Broker: Cloudlet 5 received
1.2754205607476636: Broker: Cloudlet 6 received
2.256862745098039: Broker: Cloudlet 9 received
2.366862745098039: Broker: Cloudlet 8 received
2.366862745098039: Broker: All Cloudlets executed. Finishing...
2.366862745098039: Broker: Destroying VM #0
2.366862745098039: Broker: Destroying VM #2
2.366862745098039: Broker: Destroying VM #4
2.366862745098039: Broker: Destroying VM #6
2.366862745098039: Broker: Destroying VM #1
2.366862745098039: Broker: Destroying VM #3
2.366862745098039: Broker: Destroying VM #5
2.366862745098039: Broker: Destroying VM #7
Broker is shutting down...
Simulation: No more future events
CloudInformationService: Notify all CloudSim entities for shutting down.
```

Cloudlet ID	STATUS	Data center ID	VM ID	Time	Start Time	Finish Time
0	SUCCESS	2	0	1	0.1	1.1
1	SUCCESS	2	2	1	0.1	1.1
2	SUCCESS	2	4	1	0.1	1.1
3	SUCCESS	2	6	1	0.1	1.1
7	SUCCESS	3	7	1.07	0.1	1.17
4	SUCCESS	3	1	1.18	0.1	1.28
5	SUCCESS	3	3	1.18	0.1	1.28
6	SUCCESS	3	5	1.18	0.1	1.28
9	SUCCESS	2	2	1.16	1.1	2.26
8	SUCCESS	2	0	1.27	1.1	2.37

4. Conclusions

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, **Volume 11, Sp.Iss 7 , 2022**

In cloud storage, load equalization throughout knowledge access plays a key role within the performance of a data acquisition system. Only load balancing is done good, can we fill use of system memory and computing resources to cut back the latency of the distributed operation. This paper deals with the balancing process to storage and calculation connected load brought by the original page write system, and puts forward a spread of load-balancing ways, and any measures the load balance impact of every theme through experiments. Final-ly, we came to realize that the most optimum load partition strategy is the channel storage date calculation responsive partition. It allocates without any problems access to several access units, avoiding the formation of access hotspots, and provides an effective and clear thanks to expanding the information measure of network instrumentation and services, in-crease throughput, enhance network processing capabilities, and improve network flexibil-ity and convenience.

Acknowledgements

Provide acknowledgements accordingly. List here those individuals or institutions who gaves help, assistance during the research

References

[1] Shalini joshi, Uma Kumari (2017), Modi university of science and technology. “An com-prehensive analysis on load balancing algorithms in cloud computing” published on 5th No-vember2017. https://www.researchgate.net/publication/332696891A_Comprehensive_Analysis_of_Load_Balancing_Algorithms_in_Cloud_Computing.
[2] Muhammad asim shahid, Noman Islam, Muhammad Mansoor Alam, Shahrulniza Musa (2020), Malaysian institution of Information Technology. “Comprehensive study of ap-proaches of load balancing” published on 14th july 2020.https://ieeexplore.ieee.org/abstract/document/9139971/.
[3] Abhishek Kumar Tiwari, Sreeja Nair (2016), OIST, Bhopal. “Review of load balancing using swarm technology”.
http://ijcsit.com/docs/Volume%207/vol7issue2/ijcsit2016070264.pdf
[4] Lalit Mukati, Arvind Upadhyay (2019). “Topic on static and dynamic Load balanc-ing”.https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3365568.
[5] Pawan Kumar, Rakesh Kumar (2019). “Issues and Challenges on load balancing Algo-rithms”.
https://www.researchgate.net/publication/330893507_Issues_and_Challenges_of_Load_Balancing_Techniques_in_Cloud_Computing_A_Survey.
[6] Basetty Mallikarjuna, Arun Kumar Reddy Doddi (2019). “Role of Load Balancing in next Generation”.
https://www.researchgate.net/publication/335652844_The_Role_of_Load_Balancing_Algorithms_in_Next_Generation_of_Cloud_Computing.
[7] R. Swathy, B. Vinayagasundaram, Mohamed Abouhawwash, Mohamed Abu Elsoud (2020). “Theoretical Approach for load balancing”.
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0231708.
[8] Amy Babay, John Schultz, Thomas Tantillo, and Yair Amir. 2018. Toward an intrusion-tolerant power grid: Challenges and opportunities. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS’18). IEEE, Los Alamitos, CA, 1321–1326.
[9] Christiana Aristidou and Evdokia Marcou. 2019. Blockchain standards and government applications. Journal of ICT Standardization 7, 3 (2019), 287–312.
[10] Allen Clement, Edmund L. Wong, Lorenzo Alvisi, Michael Dahlin, and Mirco Marchetti. 2009. Making Byzantine fault tolerant systems tolerate Byzantine faults. In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI’09), Vol. 9. 153–168.
[11] Guy Zyskind, Oz Nathan, and Alex Pentland. 2015. Enigma: Decentralized computation platform with guaranteed privacy. arXiv:1506.03471