© August 2024| IJIRT | Volume 11 Issue 3 | ISSN: 2349-6002

Optimized Load Balancing Techniques for Cloud Computing

Mahendra[1], Deepshikha Joshi[2],
[1]*Research Scholar, Department of Computer Science and Engineering, BM College of Technology, Indore, India*
[2]*Assistant Professor, Department of Computer Science and Engineering, BM College of Technology, Indore, India*

Abstract - Cloud registration facilitates the flow of information and provides users with valuable resources, billing customers only for resource usage. Cloud computing saves data and ensures accessibility, with increased transparency leading to higher information hoarding tendencies. Stack adjustment tests are specifically designed for cloudy conditions. Load adjustment evenly distributes dynamic workloads across hubs to avoid overloading, legalize assets, and enhance system performance. Most current algorithms enable stack modification and improved asset utilization, using memory, CPU, and system stacks. The load adjustment system identifies overloaded hubs and redistributes the load to less burdened hubs. Load balancing ensures workloads are evenly distributed across cloud data centers, preventing any from being overwhelmed or underutilized. This study proposes a hybrid approach combining Honey Bee (HB) with Particle Swarm Optimisation (PSO) for an optimal load balancing strategy, evaluated using the CloudSim simulator. The hybrid method outperforms the individual approaches of HB and PSO in terms of response time, request processing, data center utilization, and virtual machine costs, demonstrating enhanced system responsiveness and efficiency.

Keywords - Cloud Registration , Load Balancing , Hybrid Approach, Honey Bee (HB) Algorithm , Particle Swarm Optimization (PSO).

I. INTRODUCTION

Cloud computing (CC) is yet another example of Cutting-edge cloud computing technology provides customers with access to their online assets and storage space at an affordable price. Customers have continuous web-based access to their stored assets and are billed only for the portion they use. In cloud computing, the provider outsources all assets to the client company. However, cloud computing faces several challenges that need to be addressed, with stack adjustment being a primary issue. The load adjustment process distributes workloads across various hubs in the system, ensuring effective and fair asset distribution. This helps mitigate framework bottlenecks caused by load imbalance, leading to high customer satisfaction. Though relatively new, the load adjustment approach offers excellent asset utilization and improved response times. Customers benefit significantly from using cloud processing in various ways.

Cloud Computing Components:

1. On-Demand Services: Cloud registration offers clients on-demand benefits.
2. Accessibility: Users can access services whenever needed.
3. Broad Network Access: Cloud computing capabilities can be accessed through the system.
4. Multiple Access Points: Capabilities can be accessed through various means.
5. Instant Elasticity: The number of assets can be increased anytime based on customer requirements.

Challenges in Cloud Computing:

1. Security: Ensuring the safekeeping of data.
2. Load Management: Efficiently managing the workload.
3. Performance Monitoring: Keeping track of execution.
4. Reliable Services: Providing reliable and robust service options.
5. Resource Scheduling: Efficiently scheduling assets.

6. Scalability and Quality of Service: Managing scale and quality.
7. High-Speed Internet: Requiring a fast and high-bandwidth internet connection.

II. CLOUD COMPUTING MODEL

A. Services Provided by Cloud Computing Administration

Cloud computing administration refers to the numerous services offered through a network of servers in the cloud. These services provide customers with a wide range of functionalities [1].

1. Software as a Service (SaaS): SaaS allows buyers to access applications provided by the seller. These applications operate on a cloud-hosted framework and can be accessed through various interfaces, such as an internet browser. Customers do not control the underlying infrastructure or the ability to intake new items [2].

Customers who cannot design their own software but need custom applications can benefit from using a Software as a Service (SaaS) platform. Examples of SaaS applications include:

Customer Resource Management (CRM)
Video Conferencing
IT Benefits Administration
Financial Reporting
Web Research
Web Content Administration

2. Platform as a Service (PaaS)

PaaS provides customers with all the resources necessary to construct apps. It offers all possible administration services found online, eliminating the need for users to download and install software [3]. Customers upload their applications to the cloud-computing platform and have access to various tools and programming languages for app development. While users have control over the applications they submit, they do not manage the arrangement of servers, operating systems, or storage capacity [4].

3. Infrastructure as a Service (IaaS)

IaaS frees clients from managing or controlling the core cloud infrastructure. As administrative clients, they manage the operating systems, storage, and applications provided to them, with limited control over networking components. Foundation Providers handle storage and processing capacities, using virtualization to allocate and resize resources to meet individual customer needs [5]. Clients are responsible for submitting the software stacks necessary for their services, with providers arranging the requested resources upon request. This model helps avoid the costs of acquiring, housing, and managing equipment and software infrastructure, allowing for quick scalability to meet demand [6].

B. The Multiple Layers of Service

Each administration consists of several different levels, with oversight provided by the customers.

Cloud Deployment Models:

1. Public Cloud:
   - Controlled by an organization and made accessible to the general public or a large segment of an industry.
   - Resources are open to anyone without restrictions.
2. Private Cloud:
   - Used exclusively by a single organization.
   - Administered by the organization itself or a third party separate from the organization.
   - Not widely available to the general public.
3. Community Cloud:
   - Shared by several organizations with common concerns such as security needs, strategic concerns, and consistency issues.
   - Monitored by either an independent third party or the associations themselves.
4. Hybrid Cloud:
   - Formed by combining at least two different kinds of clouds (public, private, or community).
   - Maintained by standardized technology that enables information and application mobility.
   - Examples include using cloud bursting to shift workloads between clouds.

III. VIRTUALISATION

The term “virtualization” refers to things that do not In the real world, virtualization provides an experience identical to the real [7]. Virtualization refers to using a computer to simulate the tasks of an entire system. This aspect is crucial in the cloud environment because it allows users to access numerous apps and services provided by the cloud. Various forms of virtualization are utilized in cloud computing.

Types of Virtualization:

1. Full Virtualization:
   - Involves installing a complete computer system on another machine.
   - The virtual machine provides access to all the capabilities of the primary machine.
   - Used in offices when the client's actual computer is unavailable.
2. Paravirtualization:
   - Allows several operating systems to coexist on a single piece of hardware.
   - Enables efficient use of system resources such as memory and CPU.

VI. LITERATURE REVIEW

In the first part of this study, we explore what cloud computing (CC) is and its various services, highlighting current concerns and challenges. We then identify several security threats based on the quality of service provided by CC and discuss long-term ramifications from the perspective of Cloud Computer Discovery. This book also provides an overview of current cloud platforms for research and development [8].

We propose a method called "load allocation," closely related to load balancing, focusing on the Liquid Galaxy project, which aims to emulate Google Earth using several virtual computers. Our simulation results, compared to previously suggested cloud load balancing approaches, show that jobs are dynamically divided across various virtual machines in different data centers, resulting in superior response times and makespan times [9].

To address scheduling issues and enhance throughput and resource usage without negatively impacting the CC platform's performance, we used CS-SS load balancing and grasshopper optimization with MapReduce. The proposed methodology, investigating the MakeSpan parameters, employs a Particle Swarm algorithm based on mutation to distribute workloads equally across data centers. This approach aims to improve cloud computing's fitness function by reducing performance indicators like MakeSpan time and implementing an efficient load balancing strategy [10].

We evaluate existing cloud load balancing solutions, comparing them to state load-balancing algorithms to determine the best result. This evaluation includes aspects such as reliability, reaction time, adaptability, performance, resource utilization, and fault tolerance, demonstrating that system performance is enhanced by these adjustments [11].

The research also examines benchmark load balancing techniques and presents methods for opportunistic load balancing (OLB) and load balance min-min (LBMM) scheduling. Analytical model findings and CloudSim simulation results are analyzed for validity, showing that HEC-Clustering Balance outperforms other techniques in reducing HEC server processing time by 19% and 73% in two experiments [12].

We explore three load balancing approaches (Round Robin, Throttled, and Active Monitoring) using a cloud analyst simulator and evaluate service broker techniques used in cloud data centers. Additionally, a mechanism for distributing load among small base stations (sBSs) is provided, utilizing electrocardiogram-based encryption and the AES cryptographic technique for enhanced security. Combining load balancing with carbon offset (CO) saves time and money, with results indicating system utilization savings between 68% and 72.4% [13].

Our approach uses the Bat algorithm for encrypting load-balancing features, demonstrating how cloud computing and load balancing enable businesses to manage network traffic and workloads efficiently by distributing the load logically among virtual servers. We compare First Come First Serve, Shortest Job First, and Least Connection First approaches to load balancing, finding that space-shared scheduling performs better than time-shared scheduling in simulations [14].

In clinical trials, the study addresses the need for a new load balancing approach that uses functional tissue thermometry (FT), as traditional LB algorithms do not include FT efficiency factors. This paper suggests a new FT-based LB algorithm, emphasizing the importance of considering efficiencies, resource utilization, quality of service (QoS), and performance in developing scheduling algorithms [15].

Experiments demonstrate that workload prediction lowers the number of service level objective (SLO) violations and migrations, enhancing data center load balancing performance. The RL-based VM migration technique outperforms heuristic-based solutions in heavily loaded systems, showcasing significant improvements [16].

V PROPOSED METHOD

Several methods exist for balancing the queue load in accessible cloud computing. Here, we discuss some notable techniques:

A. Round-Robin Algorithm:

The round-robin algorithm is one of the simplest and most commonly used load balancing techniques in cloud computing. It cycles through available servers, assigning incoming requests to each in turn. Key terms include:

- Burst Time (BT): The time required to complete a request.
- Time Quantum (TQ): The allotted time for a request to access the virtual machine (VM).

B. Least-Connection Method:

This method utilizes the VM status list (BUSY/AVAILABLE) to assign resources. When a request is received, it identifies and assigns the resource to an available VM. This method enhances performance without high complexity.

C. Equally Spread Current Execution (ESCE):

ESCE operates on a queue basis, distributing incoming loads to VMs with the lightest workload. This method ensures balanced workload distribution and is implemented in the CloudAnalyst simulator.

D. Honey Bee Algorithm:

Inspired by the foraging behavior of honey bees, this algorithm uses scout bees to find new "food sources" (tasks) and worker bees to exploit these sources. Key steps include:

1. Initialization: Scouts explore and initialize flower patches (task groups).
2. Recruitment: Bees are recruited based on the quality of flower patches.
3. Local and Global Search: Bees perform local searches around the best patches and global searches to find new patches.
4. Optimization: The algorithm iterates, improving the distribution of tasks.

E. Particle Swarm Optimization (PSO) Algorithm:

PSO simulates the social behavior of particles (tasks) to find optimal solutions. Steps include:

1. Initialization: Randomly determine positions and velocities of particles.
2. Fitness Evaluation: Assess the fitness of each particle and update pbest (personal best).
3. Global Best (gbest): Identify the particle with the highest fitness score.
4. Update: Adjust particles' velocities and positions based on pbest and gbest.
5. Iteration: Repeat until the stopping condition is met.

F. Proposed Hybrid Algorithm:

Combining PSO with load balancing, this hybrid approach involves:

1. Initialization: Identify overloaded and underloaded VMs.
2. Rescheduling: Use PSO to reschedule cloudlets (tasks).
3. Fitness Evaluation: Determine the fitness value for each particle.
4. Update: Adjust pbest and gbest based on fitness values.
5. Iteration: Repeat the process to optimize load distribution.

VI. RESULT

In this scenario, we will be using Eclipse OXYGEN.1 as the integrated development environment (IDE) with Java as the programming language. The hardware setup includes an Intel® Core™ i7 Processor with 8GB RAM. The application will simulate large-scale internet usage scenarios akin to platforms like YouTube, Facebook, Instagram, and other similar websites. We assume that this application is accessed by more than 500 million users globally, with the entire operation being managed in a single data center.

Figure 2: Overview of Cloud Computing

Figure 2 illustrates various network scenarios executed considering the experiment utilized a single data center receiving requests from multiple User Bases. The load balancer method employed in this experiment is Throttled.

For this scenario, we assume the clock has a two-hour accuracy. Evening usage of the app is common among users, and it is also assumed that users load new requests at regular five-minute intervals.

Table 2: Time Required to Respond on Average

	Average Response Time (ms)					
LB Algorithm Tasks	Hybrid	PSO	Honey Bee	Round Robin	Throttled	ESCE (Equally Spread Current Execution Load)
100	138.95	141.35	143.72	152.94	156.82	159.6
200	145.24	149.34	153.62	159.45	164.32	168.3
300	151.82	162.78	165.82	168.21	171.28	176.3
400	157.22	167.29	171.52	175.56	178.24	182.5
500	162.47	172.51	177.85	181.97	184.69	187.3

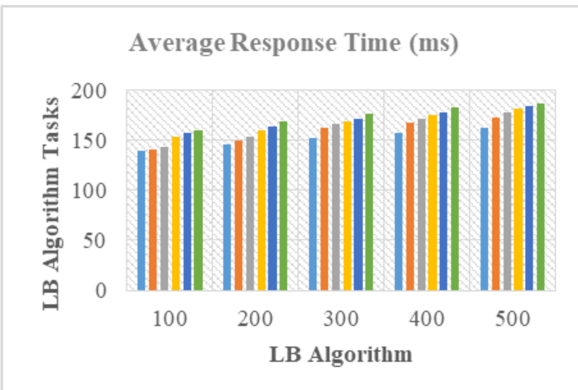

Figure 3 Illustrates a Comparative Analysis of the Mean Response Times of Various Load Balancing Algorithms

The hybrid algorithm combining Honey Bee (HB) and Particle Swarm Optimization (PSO) demonstrates a lower average response time compared to standalone algorithms such as Honey Bee (HB), Round Robin, Throttled, ESCE (Equally Spread Current Execution Load), and PSO. This superiority in quality and effectiveness is attributed to the hybrid algorithm's simplicity and lack of additional computation requirements. In our setup, we configured a data center with 50 virtual machines and tested cloudlets equal to twice the number of VMs. The innovative approach allows cloudlets to be rescheduled and executed on specific VMs. PSO was employed to randomly update the inertia weight for improved performance after updating the fitness value from the initial VM-based scheduling. This optimization enhances overall system functionality. In this context, hybridization outperforms conventional single population-based scheduling, ensuring no single node is responsible for the entire scheduling decision. To improve response time proportionally with the increase in the number of jobs, the workload should be evenly distributed across all available virtual machines.

VII. CONCLUSIONS

Distributed computing primarily manages software, data retrieval, and storage capabilities without requiring the end-user to understand the physical location or structure of the system providing these services. Stack adjustment is crucial in distributed storage, as it enhances resource utilization and overall system performance. Recent algorithms enable proper stack modifications, improving systems through more efficient scheduling and asset allocation.

This article introduces cloud computing and the stack modification process, focusing on the importance of stack adjustment calculations. Various computations in distributed computing have been outlined, highlighting characteristics such as flexibility, enhanced asset utilization, and improved reaction time.

The scheduler identifies the parameter ratio with the highest value by considering the utilization of RAM, bandwidth, and CPU resources during interval processing time. This strategy, rather than focusing solely on CPU load, determines which parameter adjustments are needed to achieve optimal utilization. Among several tactics, the recommended strategy resulted in a reduced workload where applied, demonstrating its effectiveness. The performance analysis approach's validity was indirectly confirmed, suggesting it may surpass other methods in terms of efficiency.

REFERENCES

[1] Mall, Peter and Grance, Tim, "The NIST definition of cloud computing", National Institute of Standards and Technology, 2009,vol53, pages50, Mell2009

[2] http://www.ancoris.com/solutions/cloudcomputing.html, "Cloud Computing |Google Cloud - Acores."

[3] http://www.personal.kent.edu/~rmuhamma/ E-Systems/Myos/prioritySchedule.htm, "Priority Scheduling -Operating Systems Notes."

[4] S. S. Moharana, R. D. Ramesh, D. Powar, "Analysis of load balancers in cloud computing"

International Journal of Computer Science and Engineering,2013, vol 2, pages 101-108

[5] http://blog.nexright.com/?cat=6 “Cloud Computing « Nexright Blog”.

[6] A. Sidhu, S. Kinger, “Analysis of load balancing techniques in cloud computing”, International Journal Of Computers & Technology 4 (2) (2013) pages737–741.

[7] http://www.qualitytesting.info/group/cloudcomputing/forum/topics/software-as-a-s, “Software as a Service (SAAS) - Quality Testing”

[8] http://letslearncloud.wordpress.com/ “Cloud Computing | Learn Cloud and its tips.”

[9] P. Gupta, M. Samvatsar, and U. Singh, “Cloud computing through dynamic resource allocation scheme,” in Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017, 2017, vol. 2017-Janua, doi: 10.1109/ICECA.2017.8212723.

[10] Chaudhari, Anand and Kapadia, Anushka, “Load Balancing Algorithm for Azure Virtualization with Specialized VM”, 2013, algorithms, vol 1, pages 2, Chaudhari

[11] Nayandeep Sran, Navdeep Kaur, “Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing”, vol 2, jan 2013

[12] P. S. Chouhan, M. Samvatsar, and U. Singh, “Energetic SSource allotment scheme for cloud computing using threshold-based,” in Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017, 2017, vol. 2017-Janua, doi: 10.1109/ICECA.2017.8212744.

[13] Chaczko, Zenon and Mahadevan, Venkatesh and Aslanzadeh, Shahrzad and Mcdermid, Christopher, “Availability and load balancing in cloud computing”, International Conference on Computer and Software Modeling, Singapore, chaczko2011availability

[14] http://www.writeaprogram.info/c/os-programs /priority-scheduling/ “Priority Scheduling Algorithm Example – Write a Program.’’

[15] R. Alonso-Calvo, J. Crespo, M. Garcia-Remesal, A. Anguita, V. Maojo, “On distributing load in cloud computing: A real application for very-large image datasets”, Procedia Computer Science 1 (1) (2010) pages 2669–2677

[16] S.-S.Wang, K.-Q. Yan, S.-S.Wang, C.-W. Chen, “A three-phases scheduling in a hierarchical cloud computing network”, in: Communications and Mobile Computing (CMC), 2011 Third International Conference on,IEEE, 2011, pp. 114–117.