Journal of Theoretical and Applied Information Technology
15th March 2024. Vol.102. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

METAHEURISTIC ALGORITHM-BASED LOAD BALANCING IN CLOUD COMPUTING

B. EDWARD GERALD[1], DR.P. GEETHA[2]

[1]Research scholar, Department of Computer Science, Alagappa University, Karaikudi, Tamilnadu, India.

[2]Associate Professor & Head, PG Department of Computer Science, Dr. Umayal Ramanathan College for Women, Karaikudi, Tamilnadu, India.

Email: geraldmay18@gmail.com, geeth.ganesan@gmail.com

ABSTRACT

Cloud computing has emerged as a revolutionary paradigm for delivering computing resources and services on-demand. To ensure the efficient utilization of cloud resources and provide high availability and reliability to users, load balancing is a critical component. Load balancing aims to distribute incoming network traffic or computational tasks across multiple cloud servers, preventing overloading on specific servers and optimizing resource utilization. Traditional load balancing techniques, such as round-robin and least-connections, are often not sufficient to handle the dynamic and complex workload characteristics of cloud environments. In this context, metaheuristic algorithms have gained prominence as an effective approach to address the load balancing problem in cloud computing. This paper presents a comprehensive study on load balancing using metaheuristic algorithms in cloud computing. We explore the key challenges in load balancing for cloud environments and discuss how metaheuristic algorithms, including genetic algorithms, particle swarm optimization, and simulated annealing, have been applied to tackle these challenges. We investigate the theoretical underpinnings of these algorithms and their practical implications for cloud load balancing. Furthermore, we present a comparative analysis of the performance of various metaheuristic algorithms in different cloud computing scenarios. We evaluate their effectiveness in terms of reducing response time, optimizing resource utilization, and enhancing fault tolerance. Real-world experimental result is presented to illustrate the practicality and efficiency of metaheuristic-based load balancing solutions.

Keywords: *Load Balancing Techniques, Cloud Environment, Overloading, Metaheuristic Algorithm*,

1 INTRODUCTION OF LOAD BALACNING IN CLOUD SERVICES

Cloud computing (CC) continues to evolve, demanding better resources for enhanced user experiences. Load balancing (LB) plays a pivotal role in maintaining efficient CC environments. When cloud components fail, LB becomes essential to ensure service continuity by provisioning and de-provisioning resources. LB involves distributing the computational load evenly among all nodes in the system, effectively mitigating issues related to overloading and under-loading in networks. This process strengthens the operational principle of CC and optimizes resource usage. Figure 1 illustrates a scenario without LB where job requests are randomly allocated to nodes, leading to performance degradation. To counter these issues, a robust and efficient LB model is imperative. This study introduces a cloud LB process that encompasses diverse LB models, addressing performance degradation concerns. It efficiently distributes workloads across nodes, optimizing resource utilization and user convenience. LB also incorporates the concept of workload sharing among Virtual Machines (VMs), ensuring equitable distribution and effective resource utilization. This approach enhances throughput, reduces response times, minimizes resource waiting, and prevents resource overload. Consequently, resource allocation becomes effective, contributing to lower power consumption and carbon emissions, promoting Green computing.

2. REVIEW OF LITERATURE

This study Jena et al 2022, proposes the load balancing in the environmentof cloud computing by using the algorithm of hybridization of metaheuristics. Here the over loading of data has been found using the dynamic computing via internet. This dynamic computing is done using the modified particle swarm optimization and the algorithm of improved Q-learning algorithm. This simulates the task waiting time and balances the load by priority [6].In order to find the task scheduling and the time taken for the scheduling purpose is done in this study and this is analyzed by the author Mangalampalli in 2023. Here in this study the degradation of cloud service, energy consumption and the scheduling has been calculated using the method of whale optimization. Then the algorithm task scheduling has been used for scheduling the task. In this study the work load has been analysed from the NASA and HPC2N and then the simulation has been done. The metaheuristic algorithm has been compared for the proposed system[7].In this manuscript the data security in the cloud environment has been enabled using the machine learning technique. Then the implementation is done using the data mining technique so that the data security will be very effective. By using the algorithm of Radom forest an the decision tree the combination of the secure cloud environment has been done in this study. This helps in detecting the patterns, and the data analysis in cloud computing this is proposed by Ige et al 2022 [8].In this paper the load balancing cloud computing technique has been enabled for the optimization of the data in cloud. Then based on much failure the task scheduling in the cloud environment has been enabled in virtual machines. Here the process implementing the hybrid optimization for the balancing of the load is proposed in this study. For the optimization the technique of modified objective harris hawk optimization algorithm has been used in this study [9]. In this paper the identification of the work load in the cloud computing and the balancing of the data is done using the method of multi objective task scheduling algorithm. Then with the help of reinforcement learning the algorithm of hybrid artificial bee colony has been used for overloading and unloading of the data in the cloud computing is done [10]. Imene et al 2022 proposes the task scheduling in the NSGAIII for managing the data in the cloud computing. So they introduced the third generation genetic algorithm for the analysis of the data in the cloud computing and this increases the scheduling task and the efficient problem solution is done in this study [11].Gabi et al 2022 proposes the scheduling purpose of mobile edge cloud computing is done.For simulating the optimization the process of fruitfly based method has been used in this study. For meeting the cloud platform accuracy and the time management the mobile edge cloud has been used for the analysis of time [12].

3. ISSUES AND CHALLENGES LOAD BALANCING

Cloud data centers are notorious for their power consumption. It's imperative to manage this energy usage efficiently. As the demand for cloud services skyrockets, the necessity for an effective load balancing model becomes critical. By distributing workloads intelligently, a load balancer can optimize resource utilization, leading to energy savings and environmentally friendly practices. In the pursuit of cost-effectiveness and energy efficiency, the concept of deploying smaller data centers across various locations gains traction. These smaller data centers offer localized solutions that not only reduce operational costs but also minimize power consumption. However, ensuring uniform resource allocation and fast response times across these distributed centers demands a sophisticated load balancing approach. Centralized load balancing models provide dynamic solutions to distribute workloads. However, the Achilles' heel of such models is the single point of failure. If the central controller crashes, the entire system suffers. This necessitates the development of load balancing strategies that can function effectively without relying on a single point of control, thus enhancing the system's reliability and fault tolerance.

The challenge of load balancing intensifies when cloud nodes are spread across the globe. While load balancing within a close proximity may not account for factors like network latency and communication delays, long-distance distribution of resources raises performance concerns. To address this, load balancing methods need to be devised that consider these challenges and ensure optimized distribution even across extensive geographical distances. Virtualization is a cornerstone of cloud environments, enabling the creation of virtual machines (VMs) for resource

sharing. Dynamic load balancing is pivotal in redistributing workloads among VMs, ensuring that no single VM becomes a performance bottleneck. Effective load balancing facilitates smoother VM migration, aiding in maintaining optimal performance across the cloud infrastructure. Cloud services' hallmark is their on-demand availability. Users can access services as needed, and load balancers must adapt promptly to fluctuations in demand.

A well-designed load balancing mechanism must handle sudden spikes in user requests efficiently, without compromising quality of service. Scalability is the key to ensuring seamless user experiences even during peak usage times. Load balancing algorithms need to strike a balance between effectiveness and complexity. While sophisticated algorithms may offer advanced load distribution strategies, they could also lead to delays in processing due to their intricate nature. A simpler approach that efficiently allocates resources can often outperform complex algorithms, thereby enhancing the overall efficiency of the cloud system.

4. BACKGROUN OF LOAD BALANCING OPTIMISATION IN CLOUD ENVIRONMENT

Cloud computing provides on-demand access to a distributed pool of resources, requiring efficient management of workloads and available resources. To achieve this, an optimal load balancing (LB) mechanism is essential. This mechanism is responsible for allocating tasks to virtual machines (VMs) while adhering to Quality of Service (QoS) requirements. Cloud systems continually monitor variations in user requests, necessitating a dynamic approach to task execution. The traffic in the internet has been grown rapidly nowadays hence to avoid the traffic based on the data the resource for the distribution and the resource sharing has been involved in the internet. To increase the server speed and to develop the virtual machines has been enabled for the problem that is done in multiple servers in cloud platform. Some of the algorithm that is mostly used in the cloud platform is the static, dynamic and the round robin algorithm. This helps in shifting the data for performing the better services in the cloud platforms. Also here the load balancing is done in different types that in by networks, hypertext transfer protocol and the internal protocol in load balancing. The load balancing is the backend of the multiple region in the cloud computing for processing the data in the nodes which is sent from the various devices. This data is identified using the IP address in the data or the task in the virtual machines. This produces in the seven layers that makes the single IP address that is the Google search engine, cloud run platform, computer engine etc.

5. RESOURCE ALLOCATION PROCESS

When a user request's a resource in the cloud, a service broker steps in the cloud service is proposed. This broker identifies resources that are currently free from ongoing operations and evaluates alternative brokers based on factors like resource functionality and cost. Once a suitable resource is found, the broker directs the user's request to a chosen Data Center (DC). Inside the DC, the Data Center Controller (DCN) takes over. It receives the user's request for processing.

5.1 Task Processing Flow

The Data Center (DC) houses physical resources that handle incoming user requests. The DC forwards these requests to the Load Balancer (LB). The LB is responsible for distributing tasks among available VMs for further processing. However, in cases where no VM exhibits minimal workload, the DCN establishes a queue, awaiting resource availability. Once a VM becomes available, the LB allocates tasks accordingly.

VM Load Management:

If a VM completes its assigned task, the LB may opt to reassign it to another VM for further computations. This flexibility in VM allocation helps prevent any single VM from becoming overloaded, thus ensuring efficient resource utilization across the cloud infrastructure.

5.2 Role of the Load Balancer (LB)

The Load Balancer (LB) plays a pivotal role in deciding which VM is suitable to execute a given task. This decision-making process is crucial because the efficient allocation of tasks represents a significant challenge in cloud computing environments. The LB's objective is to achieve balanced resource distribution, minimize response times, and maintain optimal performance.

Hypervisor Operations:

The hypervisor, a critical component in

virtualized environments, offers several key operations that impact load balancing and VM management:

Provision:

Creating and deploying VM instances based on resource demands.

Multiplexing:

Efficiently utilizing physical resources among multiple VMs.

Suspension:

Temporarily pausing VM operations to reallocate resources.

Live Migration: Moving VMs between physical hosts while minimizing downtime.

Optimal Load Balancing Benefits

Achieving optimal load balancing yields several benefits:

Proper distribution of tasks ensures that resources are utilized efficiently, reducing wastage. Balanced workloads prevent bottlenecks, leading to faster response times for user requests. Effective load balancing contributes to overall system performance by preventing overloading on specific components. In summary, load balancing is a crucial aspect of cloud computing that enables efficient resource allocation, reduced response times, and improved system performance. Through intelligent load balancing mechanisms, cloud environments can ensure seamless user experiences, optimal resource utilization, and streamlined task execution across the distributed infrastructure .Load balancing involves the allocation of tasks to available resources, be it virtual machines, nodes, or data centers. The goal is to distribute workloads evenly, reduce response times, and ensure resource utilization optimization.

5.3 Classification of Load Balancing Strategies based on System State:

In this approach, an overloaded node actively seeks out lightly loaded nodes and transfers some of its tasks to achieve load distribution. However, this approach can lead to congestion on lightly loaded nodes if not managed carefully. Here, lightly loaded nodes identify heavily loaded counterparts and volunteer to take on some of their tasks. The intention is to alleviate the burden on overloaded nodes. However, this can lead to an imbalance in the system if not orchestrated properly. This approach integrates both sender-initiated and receiver-initiated strategies. It attempts to create a harmonious balance between distributing tasks from overloaded nodes and sharing the workload with lightly loaded nodes.

6. STATIC VS DYNAMIC MODELS

Static Models:

These are rule-based models that make decisions without considering the current state of the system. While simple, they might not adapt well to dynamic changes in the system. They can be optimal if accurately configured or suboptimal if the configuration is not precise.

Dynamic Models:

Dynamic models consider the current system state, making decisions based on real-time information. These models are more flexible and capable of adapting to changing conditions. They can be distributed, where each node collaborates in decision-making, or non-distributed, where a single node is responsible for decision-making.

Load-Balancing Metrics:

Metrics are used to assess the effectiveness of load balancing strategies are as follows:

Performance:

The overall efficiency of the system post-implementation.

Response Time:

The time taken to complete a user request.

Throughput:

The rate at which tasks are completed within a specified time frame.

Scalability:

The ability of the system to maintain balanced load distribution as the number of nodes increases.

Fault Tolerance:

The capability of the load balancing system to manage workloads even in the presence of node failures or network issues.

Migration Time:

The time required to migrate tasks from overloaded nodes to under-loaded ones.

Resource Use

Efficient utilization of cloud resources, leading to cost savings and reduced carbon

emissions.

Degree of Imbalance:

A measure of workload inequality among nodes.

Makespan:

The total time taken for task allocation across VMs.

Load-Balancing Policies:

These are strategies that guide the allocation and migration of tasks:

Selection Policy:

Determines which tasks should be transferred from one node to another. It takes into account factors like task overhead, non-local system calls, and task execution time.

Location Policy:

Decides where tasks should be migrated based on the availability of under-loaded nodes. It can involve arbitrary selection, probing multiple nodes, or negotiation between nodes.

Transfer Policy:

Identifies tasks that should be moved from a local node to another local node. It may consider the current and last received tasks to make decisions.

Information Policy:

Involves gathering data from nodes through methods like broadcasting, agent-based collection, or centralized polling. This data helps in making informed load balancing decisions.

By understanding and effectively implementing these load balancing strategies, cloud environments can ensure optimal resource utilization, reduced response times, and enhanced performance. Load balancing plays a pivotal role in realizing the full potential of cloud computing, delivering seamless services to users while minimizing operational costs and environmental impact.The provided load scheduling techniques exhibit various strengths and focuses on addressing different aspects of cloud computing, such as task scheduling, resource allocation, load balancing, and optimization.

The techniques encompass a range of approaches, including hybridizations of heuristic and meta-heuristic methods, multi-objective optimization, fuzzy logic integration, and innovative algorithms like Dragonfly Optimization. These methods aim to achieve better execution time, optimal resource utilization, improved LB, and enhanced VM application. The comparison table captures the key features and objectives of each technique, showcasing their unique contributions to the field of cloud load scheduling.

7. CLIENT AND SERVER SIDE LOAD BALANCER

The other way of approaching the load balancing is the client side load balancing. This randomly helps the IP to make the connection over the internet. Then based on load across the server the distribution of the data using the method of round robin has been used in the past. So the main contribution of this thesis is to make the load balancing using the fruitfly based optimal resource sharing algorithm and the Meta heuristic algorithm is proposed. The delivery of the data from node to node is done and this is detected for managing the traffic among the nodes. These are the main reason to choose the client side server as a load balancer. Here the process of maintaining the random detection of the data in the nodes is done.

Server side load balancer is used for managing the traffic and maintaining the simple algorithm for balancing the load and maintaining the methods for least connection in the data traffics. Here the less response time and the scheduling time has been enabled for the sophisticated factor for enabling the geographic connection is done. Here the response time and the location of the data is tracked from the server time for the analysis of the data balance in the cloud environments. Some of the features that is used as a features in the load balancing is the symmetric and the asymmetric data that has been done in the cloud. Another features that is used in the cloud are the Hypertext Transfer Protocol (HTTP) compression, TCOP offload, Transmission Control Protocol (TCP) buffering, health checking and the HTTP security is done. These enables the calculation of the response time and the data functionality in the cloud services. This helps in managing the load balancing in the cloud client and the server side balancing.

Journal of Theoretical and Applied Information Technology
15th March 2024. Vol.102. No 5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

8. PROPOSED METHODOLOGY FOR METAHEURISTIC ALGOROITHM IS CLOUD

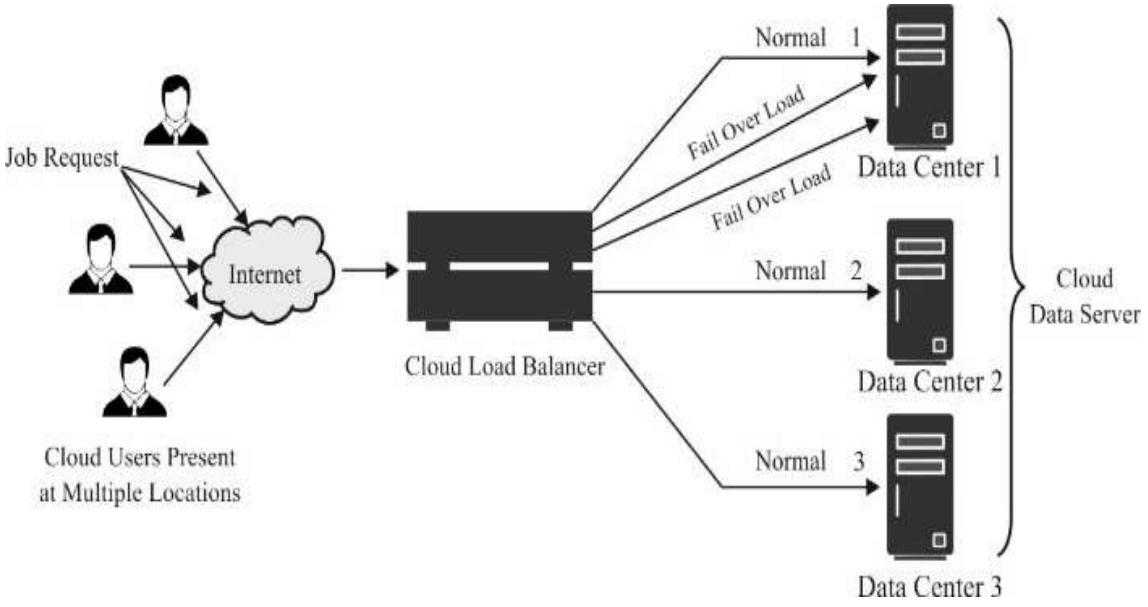

Figure 1 Showcases The Performance Of A Superior LB Approach That Assesses Node

In the above figure the job request is given from the cloud users from the different locations through the internet. Then by using the cloud balancer the data is splited and sent to the different server. These data manage the data and then the accurate and the effective output has been done for the analysis. In the recent context, as the demand for cloud resources and services grows, LB models are continually enhanced to accommodate system overhead. Effective load management directly influences cloud performance Effective CC performance hinges on various LB factors: resource utilization, higher throughput, reduced response time, overload prevention, system stability, fault tolerance, user satisfaction, and overall performance enhancement.

. Failure in the LB process can lead to system overloads and CC platform performance degradation. Load distribution in CC indirectly refers to task allocation for VMs, which relies on distributed computing. LB ensures uniform load allocation, eliminating server overload and trapping. Thus, proficient LB mechanisms are crucial for CC platforms. With rising concerns about power usage and carbon emissions, the need for effective computing resources has intensified. LB, in this regard, aids in avoiding VM overheating by optimizing energy consumption, resulting in efficient load distribution and fostering Green Computing. To achieve optimal VM results and load balancing solutions, various heuristics and metaheuristic algorithms are employed. Heuristic models like Heterogeneous earliest finish time (HEFT), Parallel Execution Time (PET), and Parameter-efficient Fine-tuning (PEFT) offer enhanced outcomes compared to HEFT alone. Metaheuristic techniques such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Simulated Annealing (SA), Greedy randomized adaptive search procedure (GRASP), and Bat Algorithm (BAT) further optimize convergence rates. Numerous research efforts have focused on developing LB techniques for CC using metaheuristic algorithms. This paper offers an extensive survey of LB concepts, design considerations, and existing methodologies, along

with a detailed comparative analysis of their performance.

9. EFFICIENT META HEURISTIC SCHEDULING OF VIRTUAL MACHINES IN CLOUD PLATFORM

The scheduling of the virtual machines in the cloud environment is done using the IaaS by running the task using some of the optimization methods. Here the scheduling, response time calculation and the load is calculated by different approaches in the dynamic environment [31]. Some of the heuristic and the hybridization technique has been enabled for the expressing the efficient output for the prediction of the optimization and the load maintaining problems. Some of the main factors that helps in finding the problem of execution enables the makespan and the approach of the optimal performance and the load balancing in the cloud environment is proposed in this thesis [32]. Some of the distribution among the load balancing related problem has been found using the hybrid Meta heuristic method to clear the execution of work flow and the determination of load distributions. Some prediction regarding the response time and the execution cost can be analysed using the hybridization cost and the utilization of the resources is done. Here the value of threshold and the value of the multiple computed virtual machines is also been maintained in the cloud computing. Some of the load balancing optimization helps in utilizing the dynamic number and priority based scheduling in the cloud environments.

10. ANALYSIS WHICH COMPARES THE LOAD BALANCING USING THE META HEURISTIC IN CLOUD COMPUTING

Some of the serious problem in the environment of cloud computing is the challenges that make sure of the services functions in the CPS that is the cloud service provider. Then based on the prediction the data has been analysed using the flow time, processing time, its performance, and the resource time. Based on the results the performance of the load balancing has been improved using this Meta heuristic in the cloud services. Here the mapping of the workload has been done to relate and classify the difficulties in the cloud environments. Here the process of maintaining the random optimum solution for the scheduling purpose is done in this paper. The heuristic and the meta heuristic is used for the load balancing that helps in searching of the optimum solution and the problem of the solution is done.

11 PROPOED METHODOLOGY OF THE META HEURISTIC ALGORITHM IN CLOUD

This proposed system enables the dynamic technique for managing the method for balancing the load is round robin. This helps in the min and the max load balancing. This makes the transfer of data in all the nodes and makes the task migrations. This migration helps in determining the data that has been scheduled in the cloud. Some of the global problems that is done are evaluator selection based recombination methods. This enables the propagation of the data which is sent through the virtual machine by the helps of nodes is done. This algorithm is most wanted because of the function objective that helps in the optimization. This is similar to the Accuracy, precision and finding of speed of data that is travelled in the cloud computing. Here the process of managing the meta heuristic and the challenges in the nonlinear function is managed using the meta heuristic algorithm.

Some of the scheduling algorithm has been enabled for making the cloud environment more efficient. So, to sort out the issues regarding the data scheduling and the distribution of the data is done using the technique of meta heuristics.

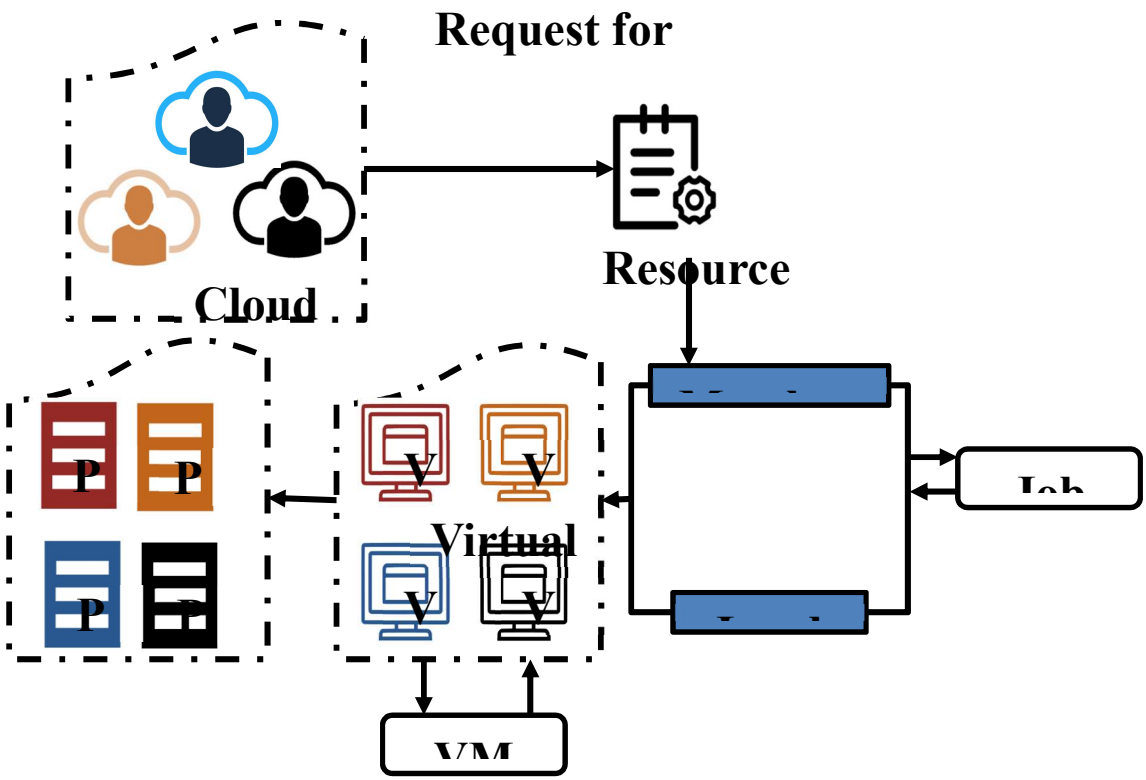

Figure 2 Process Of Job Migration In Cloud Platform

In this study, the method that is involved in the algorithm of the meta heuristic algorithm is analyzed and drawn. Here figure 2 proposes the flow of the job migration in the cloud environment is done. First of all the resource manager makes the request to the request manager regarding the task and any other process regarding the data transmissions. The meta heuristic method manages the methods involves in the method heuristic model and in the model of load balancing. This makes the virtual machines act more effective to migrate the jobs in the cloud environments.

All the jobs that is sent by the user is analysed virtual machine allocation that enables the formation of the job migration is done. These load balancer and the meta heuristic model enables the job migration and rectifies the problem regarding the job and the distribution of the job is done.

12 EVALUATION METHODS AND RESULTS FOR META HEURISTIC ALGORITHM FOR LOAD BALANCING

In the analysis of different load balancing (LB) algorithms, the average response time is a crucial performance metric. The presented data in Table 1 and Figure 2 highlights the comparison of average response times for various LB methods, including Throttled, RR (Round Robin), ACO (Ant Colony Optimization), Exact Algorithm, and MPSO (Modified Particle Swarm Optimization). Initial and the ending of the data proceeded in the cloud has been calculated using the meta heuristic algorithm. The optimization of the data is also done using the communication among the data in the virtual machines. Some of the resource allocation framework has been enabled simultaneously for scheduling the cycle in the cloud computing and makes the reallocation of the data in the spaces.

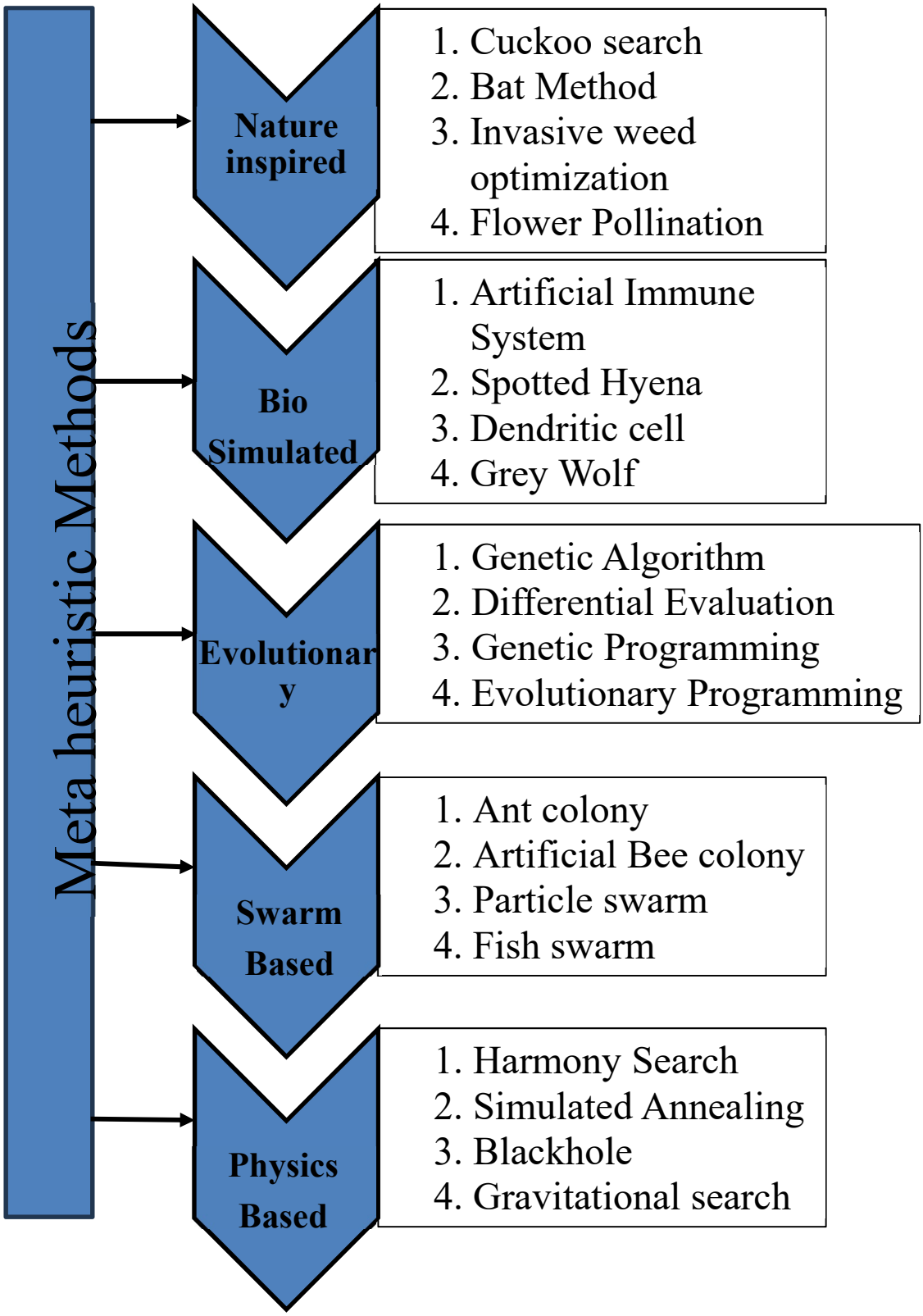

Figure 3 Methods Involved In Meta Heuristic Algorithm

The above diagram evaluates the method involved in the algorithm of meta heuristics for the simulation enabled in the cloud services. Here in this diagram the meta neuristic method involves nature inspired, bio simulated, evolutionary based, swarm based and physics based in the method of meta heuristics. This has many built in method for the better and effective cloud service in the cloud platform.

The nature inspired has some of the built in method of cukoo search, BAT method, invasive weed optimization, and flower pollination. This makes the cloud computing a very good nature recommended platform for the load balancing using the algorithm of meta heuristic is used. Then for the bio simulated the method of artificial immune system, spotted hyena, dendritic cell and the gray wolf is used as the effective method.

Then in the evolutionary based for the purpose of evaluating the data in the load balancing the genetic algorithm, differential evaluation, the genetic programming, and the evolutionary programming has been used in this study. Then in the swarm based analysis the ant colony, artificial bee colony, particle swarm and the fish swarm are the best methods for the analysis is done in the cloud computing is done.

Based on the physics based the method of harmony search, simulated annealing, black hole and the gravitational search has been used for the better data transmission of the data using the algorithm of the meta heuristic algorithm in the cloud computing. This manage the data in each nodes that is get from the different IoT devices. And this data has been evaluated for the better load balancing in the cloud platform.

Table 1 Average Response Time Analysis Of Different Algorithms

Methods	**Average Response Time (ms)**
Throttled	365.52
RR	364.85
ACO	362.67
Exact Algorithm	365.87
MPSO	360.11

Table 1 presents the average response times in milliseconds for each method. The results indicate that the Throttled model achieved an average response time of 365.52 ms, followed closely by RR with 364.85 ms, and ACO with 362.67 ms. The Exact Algorithm, however, exhibited the highest average response time of 365.87 ms, indicating its inferior performance. On the other hand, the MPSO algorithm demonstrated a relatively lower average response time of 360.11 ms, positioning it as the superior method among those analyzed.

The variation that is among the entire algorithm why is because each algorithm plays a vital role for calculating the response time. Comparing to all the algorithm the average response time is between 360 to 380 ms .

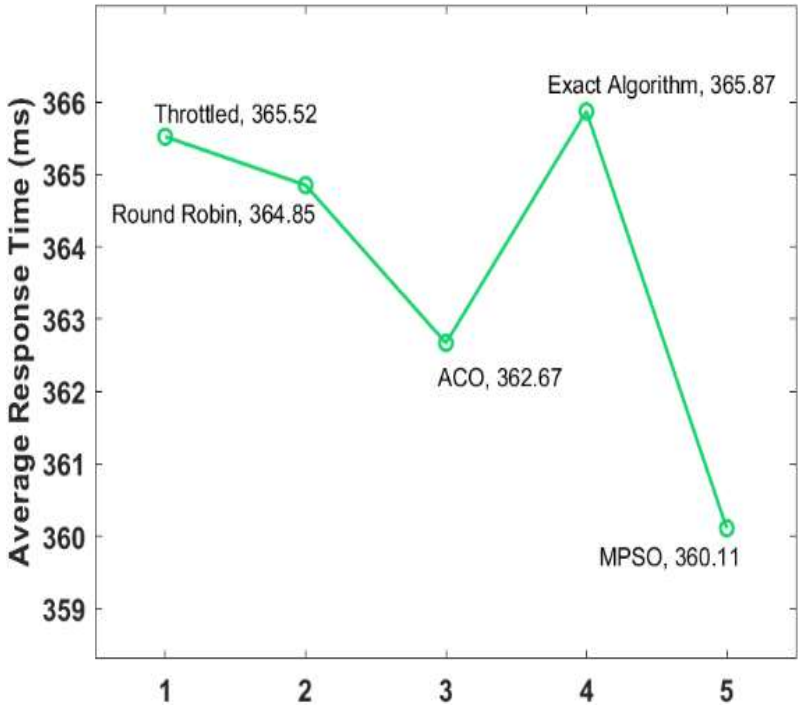

Figure2 Average Response Time Analysis Of Different Methods

In the above response time calculation is analysed using many of the method. In the round robin method the average response time is calculated as the 364.85 ms. ACO calculated as the 362.67. The MPSO is calculated as 360.11 and Exact algorithm is calculated as the 366.87. These are all the values response time has been calculated according to the methods.

CPU Utilization Time Analysis:

The results reveal insights into the efficiency of these methods in utilizing the available CPU resources. The analysis includes RD (Random Distribution), WRR (Weighted Round Robin), DLM (Dynamic Load Management), LB-BC (Load Balancing with Budget Constraint), LB-RC (Load Balancing with Resource Constraints), Improved Particle Swarm Optimization-Firefly (IPSO-Firefly), and FIMPSO.CPU Utilization Time in the system measures how effectively the Load Balancing (LB) method utilizes available CPU resources, ensuring efficient task processing.

$$\text{CPU Utilization Time (\%)} = \frac{\text{Time CPU is actively used by tasks}}{\text{Total Observation}} * 100$$

Table 2. Comparison analysis between various Load Balancing methods based on CPU utilization

Methods	CPU Utilization			
	Small	Medium	Large	Extra large
RD (Random Distribution)	42	54	64	74
WRR (Weighted Round Robin)	48	59	68	78
DLM (Dynamic Load Management)	52	64	73	82
LB-BC (Load Balancing with	59	69	82	86

Budget Constraint)				
LB-RC (Load Balancing with Resource Constraints)	67	75	86	92
Improved Particle Swarm Optimization-Firefly (IPSO-Firefly)	69	79	89	92
FIMPSO	72	82	92	97

This table 2 presents CPU utilization percentages for various Load Balancing methods across different task sizes. For each method, CPU utilization is reported for four task size categories: Small, Medium, Large, and Extra Large. Notably, the FIMPSO method exhibits the highest CPU utilization across all task sizes, reaching 97% for Extra Large tasks, while RD shows the lowest CPU utilization, peaking at 74% for Extra Large tasks.

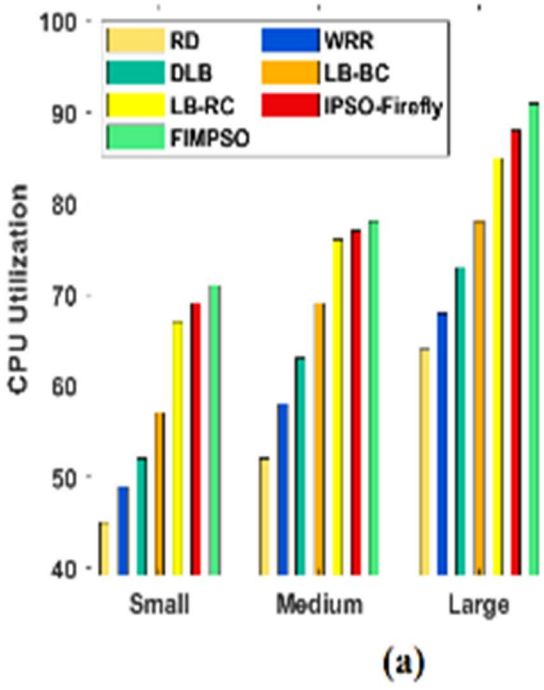

Figure 3a. CPU utilization Analysis

Moving to Figure 3a, this illustrates the CPU utilization time of different LB models under varying tasks. From the analysis, it's evident that the RD model demonstrated the worst LB performance, achieving the lowest CPU utilization. In contrast, the WRR technique managed slightly higher CPU utilization than RD. The DLM approach yielded considerably better CPU utilization compared to traditional methods. LB-BC achieved moderate results, demonstrating reasonable CPU utilization. LB-RC and IPSO-Firefly methods showcased near-optimal outcomes with high CPU utilization. Remarkably, the FIMPSO algorithm stood out, displaying the most efficient performance with maximum CPU utilization.

Memory Application Time Analysis:

Memory Application Time evaluates the proposed LB method's efficiency in managing memory resources, optimizing memory allocation for tasks. The focus here is on the methods' ability to efficiently allocate and manage memory resources. Similar to the previous analysis, RD, WRR, DLM, LB-BC, LB-RC, IPSO-Firefly, and FIMPSO are included in the analysis.

Memory Application Time (%) $= \frac{\text{Time Memory is actively used by tasks}}{\text{Total Observation Time}} *$ **100**

Table 3. Memory application time Analysis for various LB methods

Methods	Memory Utilization			
	Small	Medium	Large	Extra large
RD (Random Distribution)	39	48	60	72
WRR (Weighted	45	53	64	76

Round Robin)				
DLM (Dynamic Load Management)	49	58	69	81
LB-BC (Load Balancing with Budget Constraint)	53	62	73	84
LB-RC (Load Balancing with Resource Constraints)	58	66	76	87
Improved Particle Swarm Optimization-Firefly (IPSO-Firefly)	60	69	80	89
FIMPSO	62	71	84	93

This table 3 illustrates memory utilization percentages for various Load Balancing methods across different task sizes. The FIMPSO method demonstrates the highest memory utilization across all task sizes, peaking at 93% for Extra Large tasks, while RD exhibits the lowest memory utilization, reaching 72% for Extra Large tasks.

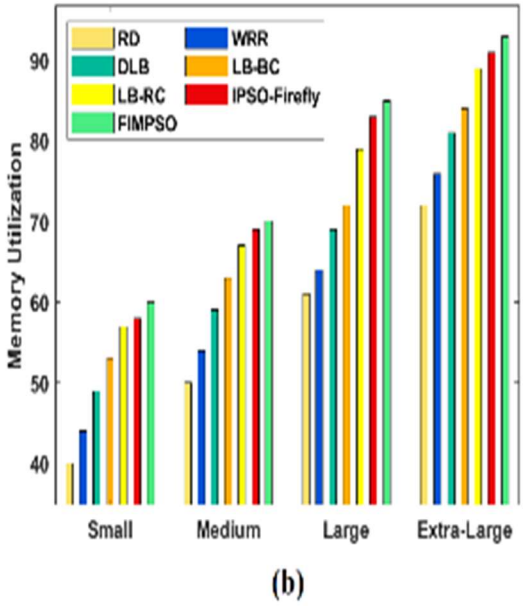

Figure 3b: Memory application time Analysis

Figure 3b provides insights into the memory application time of different LB methods under diverse tasks. The analysis highlights that the RD method was the weakest performer in LB, exhibiting lower memory application. The WRR approach, while better than RD, still displayed moderate memory utilization. The DLM technology showcased considerable memory consumption, outperforming classical methodologies. LB-BC demonstrated slightly better outcomes with acceptable memory consumption time. LB-RC and IPSO-Firefly achieved results close to optimality with maximum memory utilization. Notably, the FIMPSO algorithm excelled in efficient memory utilization.

Makespan Analysis:

Makespan in the proposed system reflects the time taken to complete a set of tasks, showcasing how quickly tasks are processed for efficient task execution.The analysis covers RD, WRR, DLM, LB-BC, LB-RC, IPSO-Firefly, and FIMPSO.

$$\textbf{Makespan} = \frac{\textbf{Time when the last task completes}}{\textbf{Time when the first task starts}}$$

Table 4 Makespan comparison between various methods

Methods	Makespan			
	Small	Medium	Large	Extra large
RD (Random Distribution)	74	150	209	280
WRR (Weighted Round Robin)	67	138	200	260
DLM (Dynamic Load Management)	56	126	192	250
LB-BC (Load Balancing with Budget Constraint)	48	118	183	230
LB-RC (Load Balancing with Resource Constraints)	44	106	177	150
Improved Particle Swarm Optimization-Firefly (IPSO-Firefly)	37	95	169	138
FIMPSO	29	88	158	125

This table 4 presents makespan durations for various Load Balancing methods across different task sizes: Small, Medium, Large, and Extra Large. Notably, the FIMPSO method achieves the lowest makespan durations across all task sizes, with a remarkable minimum of 29 units of time for small tasks, while RD exhibits the highest makespan durations, reaching 280 units of time for Extra Large tasks.

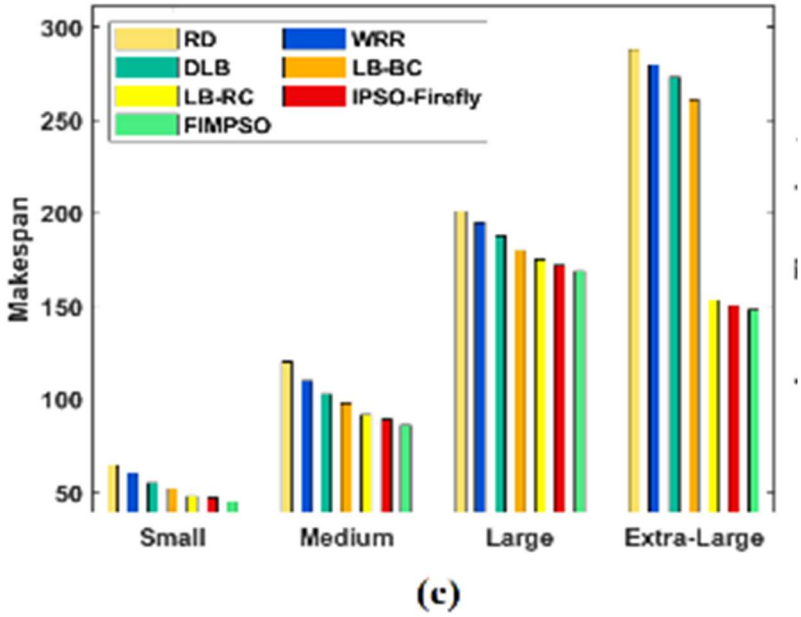

Figure 3c. Makespan Analysis

Figure 3c focuses on the makespan of different LB methods under distinct operations. Results indicate that the RD model resulted in an ineffective LB performance, with the lowest makespan. WRR demonstrated moderate makespan compared to RD. DLM exhibited a substantial makespan due to its consideration of dynamic load management. LB-BC achieved

improved results with acceptable makespan. LB-RC and IPSO-Firefly demonstrated near-optimal outcomes with maximum makespan. Remarkably, the FIMPSO method displayed efficient results by achieving the maximum makespan, which could imply better task completion.

Average Throughput Analysis:

Average Throughput in the proposed system measures the number of tasks completed per unit of time, indicating the system's efficiency in task completion.The analysis includes RD, WRR, DLM, LB-BC, LB-RC, IPSO-Firefly, and FIMPSO.

$$\text{Average Throughput} = \frac{\text{Total Number of Tasks}}{\text{Total Time}}$$

Table 5 Average Throughput Analysis

Methods	Average Throughput			
	Small	Medium	Large	Extra large
RD (Random Distribution)	65	56	42	28
WRR (Weighted Round Robin)	72	64	48	37
DLM (Dynamic Load Management)	79	71	55	45
LB-BC (Load Balancing with Budget Constraint)	82	76	63	56
LB-RC (Load Balancing with Resource Constraints)	91	83	70	67
Improved Particle Swarm Optimization-Firefly (IPSO-Firefly)	95	85	68	70
FIMPSO	98	89	78	73

This table 5 examines the average throughput of various LB frameworks under different operational scenarios. Among the methods analyzed, FIMPSO stands out with the highest Average Throughput across all workloads, indicating its superior performance in task completion. LB-RC and IPSO-Firefly also demonstrate high throughput. In contrast, RD exhibits the lowest throughput, signifying its inefficiency in handling tasks. LB-BC and DLM achieve moderate to high throughput, making them suitable choices for certain workloads.

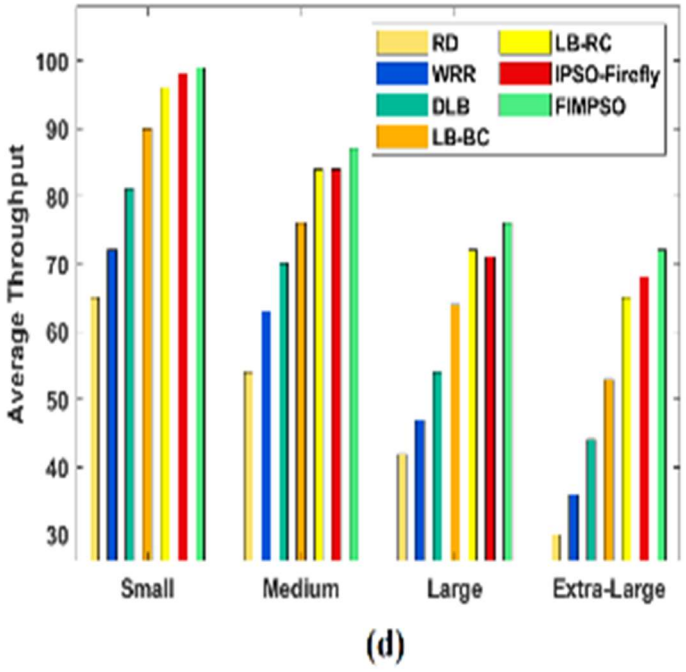

Figure 3 d. Average Throughput Analysis

Figure 3d delves into the average throughput of diverse LB frameworks under different operations. Results indicate that the RD method exhibited the lowest average throughput, signifying its inefficiency as an LB approach. WRR achieved a moderate average throughput over RD. DLM demonstrated considerable average throughput, indicating better task completion rates than traditional methodologies. LB-BC showcased acceptable outcomes with moderate average throughput.

LB-RC and IPSO-Firefly demonstrated similar and improved results, achieving the maximum average throughput. The FIMPSO technology displayed proficient results by accomplishing optimal average throughput.

13. CONCLUSION

This analyses provide a comprehensive understanding of how different LB algorithms perform across various performance metrics. These results serve as a valuable guide for selecting the most suitable LB approach based on specific performance objectives and requirements. This makes the scheduling of the resources in the every virtual machine among the nodes and the task. Also while comparing the existing method this is one of the perfect method for the load balancing in the cloud computing.

REFERENCE

[1] Ansari, Mohd Dilshad, Vinit Kumar Gunjan, and Ekbal Rashid. "On Security and Data Integrity Framework for Cloud Computing Using Tamper-Proofing." ICCCE 2020. Springer, Singapore, 2021. 1419-1427.

[2] Saldamli, Gokay, et al. "Enterprise Backend as a Service (EBaaS)." Advances in Parallel & Distributed Processing, and Applications. Springer, Cham, 2021. 1077-1099.

[3] Ranapana, R. A. A. I. B., and K. P. N. Jayasena. "Novel Approach for Load Balancing in Mobile Cloud Computing." 2021 6th International Conference on Information Technology Research (ICITR). IEEE, 2021.

[4] Ebadifard, Fatemeh, and Seyed Morteza Babamir. "Autonomic task scheduling algorithm for dynamic workloads through a load balancing technique for the cloud-computing environment." Cluster Computing 24.2 (2021): 1075-1101.

[5] Ziyath, S., and S. Senthilkumar. "MHO: meta heuristic optimization applied task scheduling with load balancing technique for cloud infrastructure services." Journal of Ambient Intelligence and Humanized Computing 12.6 (2021): 6629-6638.

[6] Priya, V., C. Sathiya Kumar, and Ramani Kannan. "Resource scheduling algorithm with load balancing for cloud service provisioning." Applied Soft Computing 76 (2019): 416-424.

[7] Pradhan, Arabinda, and Sukant Kishoro Bisoy. "A novel load balancing technique for cloud computing platform based on PSO." Journal of King Saud University-Computer and Information Sciences (2020).

[8] Pourghaffari, Ali, Morteza Barari, and Saeed Sedighian Kashi. "An efficient method for allocating resources in a cloud computing environment with a load balancing approach." Concurrency and Computation: Practice and Experience 31.17 (2019): e5285.

[9] Malik, Manoj Kumar, Ajit Singh, and Abhishek Swaroop. "A planned scheduling process of cloud computing by an effective job allocation and fault-tolerant mechanism." Journal of Ambient Intelligence and Humanized Computing 13.2 (2022): 1153-1171.

[10] Yu, Chaohui, et al. "Transfer learning with dynamic adversarial adaptation network." 2019 IEEE International Conference on Data Mining (ICDM). IEEE, 2019.

[11] Dam, S., Mandal, G., Dasgupta, K., & Dutta, P. (2021). An ant-colony-based meta-heuristic approach for load balancing in cloud computing. In *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing* (pp. 873-903). IGI Global.

[12] Yadav, M., & Gupta, S. (2020). Hybrid meta-heuristic VM load balancing optimization approach. *Journal of Information and Optimization Sciences*, *41*(2), 577-586.

[13] Alghamdi, M. I. (2022). Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO). *Sustainability*, *14*(19), 11982.

[14] Kumar, A., & Devi, R. (2023, April). VM Migration and Resource Management using Meta Heuristic Technique in Cloud Computing Services. In *2023 International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)* (pp. 1-6). IEEE.

[15] Adil, M., Nabi, S., & Raza, S. (2022). PSO-CALBA: Particle swarm optimization based content-aware load balancing algorithm in

cloud computing environment. *Computing and Informatics*, *41*(5), 1157-1185.

[16] Adil, M., Nabi, S., & Raza, S. (2022). PSO-CALBA: Particle swarm optimization based content-aware load balancing algorithm in cloud computing environment. *Computing and Informatics*, *41*(5), 1157-1185.

[17] Adil, M., Nabi, S., & Raza, S. (2022). PSO-CALBA: Particle swarm optimization based content-aware load balancing algorithm in cloud computing environment. *Computing and Informatics*, *41*(5), 1157-1185.

[18] Haris, M., & Khan, R. Z. (2022). A systematic review on load balancing tools and techniques in cloud computing. *Inventive Systems and Control: Proceedings of ICISC 2022*, 503-521.

[19] KALAIVANI, R. Energy Efficient and Load Balanced Optimal Resource Allocation framework for Cloud Environment using ML based Meta heuristic techniques.

[20] Mikram, H., El Kafhali, S., & Saadi, Y. (2023, March). Metaheuristic Algorithms Based Server Consolidation for Tasks Scheduling in Cloud Computing Environment. In *The International Conference on Artificial Intelligence and Computer Vision* (pp. 477-486). Cham: Springer Nature Switzerland.