

A Dynamic Approach to Load Balancing in Cloud Infrastructure: Enhancing Energy Efficiency and Resource Utilization

Shadman Sakib

*Department of Computer Science
Missouri State University
Springfield, USA
ss4587s@missouristate.edu*

Ajay Katangur

*Department of Computer Science
Missouri State University
Springfield, USA
ajaykatangur@missouristate.edu*

Rahul Dubey

*Department of Computer Science
Missouri State University
Springfield, USA
rahul.dubey@missouristate.edu*

Abstract—Cloud computing has grown rapidly in recent years, mainly due to the sharp increase in data transferred over the internet. This growth makes load balancing a key part of cloud systems, as it helps distribute user requests across servers to maintain performance, prevent overload, and ensure a smooth user experience. Despite its importance, managing server resources and keeping workloads balanced over time remains a major challenge in cloud environments. This paper introduces a novel Score-Based Dynamic Load Balancer (SBDLB) that allocates workloads to virtual machines based on real-time performance metrics. The objective is to enhance resource utilization and overall system efficiency. The method was thoroughly tested using the CloudSim 7G platform, comparing its performance against the throttled load balancing strategy. Evaluations were conducted across a variety of workloads and scenarios, demonstrating the SBDLB's ability to adapt dynamically to workload fluctuations while optimizing resource usage. The proposed method outperformed the throttled strategy, improving average response times by 34% and 37% in different scenarios. It also reduced data center processing times by an average of 13%. Over a 24-hour simulation, the method decreased operational costs by 15%, promoting a more energy-efficient and sustainable cloud infrastructure through reduced energy consumption.

Index Terms—Cloud Computing, Dynamic Load Balancing, Task Scheduling, Virtual Machine, CloudSim, Data Center

I. INTRODUCTION

Cloud computing has gained widespread adoption due to its scalable and flexible nature. More organizations are moving away from traditional on-premises infrastructure in favor of remote cloud solutions that offer greater cost efficiency, enhanced security, and improved accessibility [1]. Despite its growth and long-term benefits, cloud adoption presents challenges, particularly in process management, legal compliance, data security, and system reliability [2], [3]. These barriers are especially pronounced for small and medium-sized enterprises, which often lack the resources to maintain their own cloud infrastructure.

Infrastructure as a Service (IaaS) has emerged as one of the most adopted cloud models, providing scalable and cost-efficient alternatives [4]. In IaaS, cloud service providers (CSPs) such as AWS, Azure, IBM Cloud, and Google Cloud

operate under service level agreements (SLAs), which establish performance expectations [5]. CSPs must also maintain Quality of Service (QoS), ensuring reliable resource allocation even as user demand grows. As cloud usage increases, load balancing becomes a critical challenge [6], essential for distributing workloads to prevent server overload, minimize latency, and ensure consistent performance.

Addressing these challenges, this research proposes a Score-Based Dynamic Load Balancer (SBDLB) to efficiently distribute workloads across virtual machines (VMs). Unlike traditional methods, SBDLB dynamically evaluates resource availability and assigns tasks using a computed score, promoting optimal resource utilization and balanced workload distribution. It also integrates a VM task threshold to prevent overloading, with each VM capped at a maximum number of concurrent tasks based on its capacity—determined through empirical testing under varying loads. SBDLB is compared against the widely used throttled load balancer, which has demonstrated strong performance in prior studies [7], [8].

Four simulation scenarios were developed to evaluate SBDLB in terms of average response time, data center processing time, and operational cost. Extensive experiments and statistical analysis, including p-value calculations, confirm the significance of performance improvements over the throttled method. Results show that SBDLB consistently outperforms the throttled load balancing strategy across all scenarios, reducing average response times by 34% and 37% in two key tests, and lowering data center processing times by an average of 13%. In a separate experiment, SBDLB demonstrated its ability to achieve better performance with fewer active data centers. Furthermore, a 15% reduction in operational costs was observed over a 24-hour simulation period, highlighting the efficiency of the method. By minimizing execution delays and avoiding resource overuse, SBDLB not only improves performance but also reduces energy consumption. This dual benefit enhances cost efficiency and contributes to more sustainable, environmentally friendly cloud computing operations.

II. RELATED WORK

Data generation has surged in recent years [9], with social media being one of the leading contributors [10]. As more and more users access the cloud, load balancing has become crucial in managing large surges of data requests. Consequently, efficient load balancing techniques are essential to ensure optimal resource utilization and maintain high performance in the face of growing demand. Researchers have increasingly recognized the significance of this problem, leading to extensive studies aimed at mitigating these challenges. Various approaches have been explored, including data center selection policies as well as advancements in load balancing techniques to enhance efficiency and resource utilization.

Load balancing is said to be of two types [11]: (1) static and (2) dynamic. In static methods, system characteristics are predefined and do not consider real-time data, making it simple but inflexible. Dynamic methods, though more complex, adapt to current system status, enabling more efficient load balancing [12]. Several researchers have explored variations of well-known load balancing techniques, including Round Robin [13], Threshold [14], and Throttled [8]. In a recent study, the authors implemented a priority-weighted Round Robin strategy to effectively manage incoming tasks with varying priorities [15]. However, the Round Robin approach may lead to VM overload in scenarios where a high volume of tasks arrives concurrently, as it fails to account for the resource capacity and current load of individual VMs. Another study introduced a dual-threshold approach, where one threshold value identifies underloaded VMs and another detects overloaded VMs [16]. Both [7] and [8] introduce variations to the throttled approach, leading to modest improvements in response times. Additionally, both studies demonstrate significant enhancements over other load balancing methods, such as Round Robin, Active Monitoring, and Equal Load Distribution.

Given its established effectiveness and widespread use, the throttled approach serves as a natural baseline for comparison in this study, providing a benchmark against which the performance of the proposed dynamic algorithm can be evaluated. In another study, the authors randomly assigned values for task length and completion time before allocating these tasks to random VMs. The completion time was then calculated based on the characteristics of both the task and the VM. If the calculated time resulted in a SLA violation for the VM, the task was migrated to another VM [17]. Random allocation of tasks to VMs is an inefficient strategy as it disregards the resource requirements of tasks and the available capacity of VMs.

Several nature-inspired algorithms have been applied to both data center selection policies and load balancing. The genetic algorithm-based DC service broker policy proposed in [18] presents an innovative approach to minimizing network delays. However, it suffers from a prolonged convergence time, and since the genetic algorithm (GA) is executed only once at the start of every hour, it is not well-suited for dynamic environments requiring real-time adaptability. An-

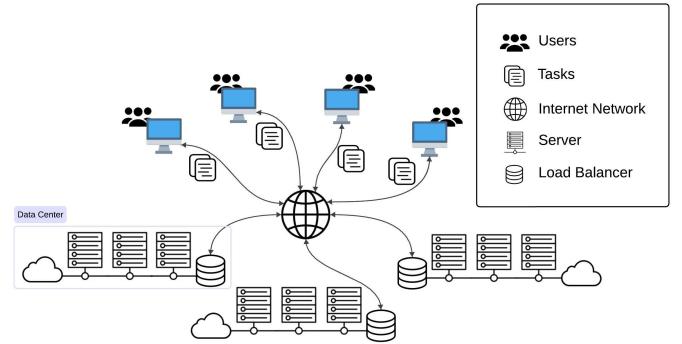


Fig. 1: Simple Cloud Infrastructure

other approach employs GA for load balancing by allocating tasks in bulk from a queue. However, this method presents several drawbacks, including a lack of real-time adaptability, increased waiting times for tasks, and inefficient handling of heterogeneous workloads [19].

A survey confirms that Meta-Heuristic approaches, such as ACO, Cuckoo Search, Honey Bee Optimization, and others, effectively balance cloud workloads. However, these algorithms have drawbacks in convergence rate, affecting exploration or exploitation [20], [21]. Recent studies have increasingly focused on green cloud computing, recognizing data centers as a major source of carbon emissions [22]. As companies and industries prioritize eco-friendly technological solutions, such optimizations align with broader sustainability goals, making the dynamic approach both economically and environmentally favorable [23].

III. SYSTEM MODEL

A. Cloud Environment

Figure 1 illustrates a simplified cloud infrastructure, where users send requests through a central gateway to distributed data centers. Load balancing ensures these requests are efficiently routed, preventing overload and maximizing resource utilization [24]. Consider a cloud service provider with globally distributed data centers: $DC = \{DC_1, DC_2, DC_3, \dots, DC_d\}$, each consisting of hundreds of physical machines. These machines host multiple virtual machines: $VM = \{VM_1, VM_2, VM_3, \dots, VM_n\}$, each with different hardware configurations. When a user requests access to cloud resources, the load balancer directs the task T to the most suitable data center and VM. The decision is based on factors such as resource availability, current load, and task requirements.

B. System Configuration

For executing and evaluating the performance of the proposed load balancing technique, a simulation environment was set up using CloudSim 7G [25]. It is a tool for modeling and testing cloud-based infrastructures and resource allocation strategies. Testing new techniques in a real cloud environment is impractical, as it may impact end-user service quality.

TABLE I: Data Center Specifications

Attribute	Details
Architecture	x86
Operating System	Linux
Virtual Machine Monitor	Xen
CPU Usage Cost	\$3/sec
Memory Cost	\$0.004/MB
Bandwidth Cost	\$0.01/Mbps
Storage Cost	\$0.0001/MB

TABLE II: Physical Machine Specifications

Attribute	Type 1	Type 2
RAM (MB)	1024	2048
Storage (GB)	10	20
Bandwidth (MB/s)	1000	2000
Processing Cores	4	8

Therefore, a reliable simulator like CloudSim is essential for tasks like scheduling and load balancing. The data centers are designed based on the specifications outlined in Table I, while the physical machines adhere to the configurations detailed in Table II. The VMs are created in accordance with the specifications provided in Table III. For simplicity and to facilitate clearer performance comparisons, the simulation utilizes only two distinct server configurations, consistent with the methodology described by Razali [26]. Likewise, the virtual machine setup is divided into two types each designed to meet varying computational requirements. This streamlined heterogeneous configuration enables a more focused analysis of how the load balancing algorithm manages diverse workloads. Unlike prior studies [27], [26], which typically employ uniform VM configurations, this approach offers a more realistic simulation of real-world scenarios.

IV. PROPOSED SCORE-BASED DYNAMIC LOAD BALANCER

A. Score-Based Dynamic Load Balancer

1) *Score-Based Dynamic Load Balancer (SBDLB):* The Score-Based Dynamic Load Balancer (SBDLB) allocates tasks by evaluating virtual machines (VMs) based on resource availability and workload to ensure efficient task distribution. Figure 2 illustrates the flow of the proposed approach. When a task arrives, the system first scans the available VMs, excluding those that exceed a predefined task threshold. For the remaining VMs, key parameters such as available CPU utilization (MIPS), RAM, and bandwidth are retrieved.

To determine the resource requirements for each incoming task, the system considers the task length, which falls within a predefined range corresponding to specific task types. The task length is then normalized using a min-max scaling technique (Equation 1), mapping it to the range of the available resources for each VM. This ensures that the task's resource demands are expressed in terms of the VM's available resources, such as MIPS, RAM, and bandwidth. If a VM lacks sufficient resources to accommodate the task's normalized demands, it is assigned a score of -1. Otherwise, a suitability score

TABLE III: Virtual Machine Specifications

Attribute	Low-Spec VM	High-Spec VM
Processing Power (MIPS)	500	1000
Storage (GB)	10	20
RAM (MB)	1024	2048
Bandwidth (MB/s)	1000	2000
CPU Cores	1	2

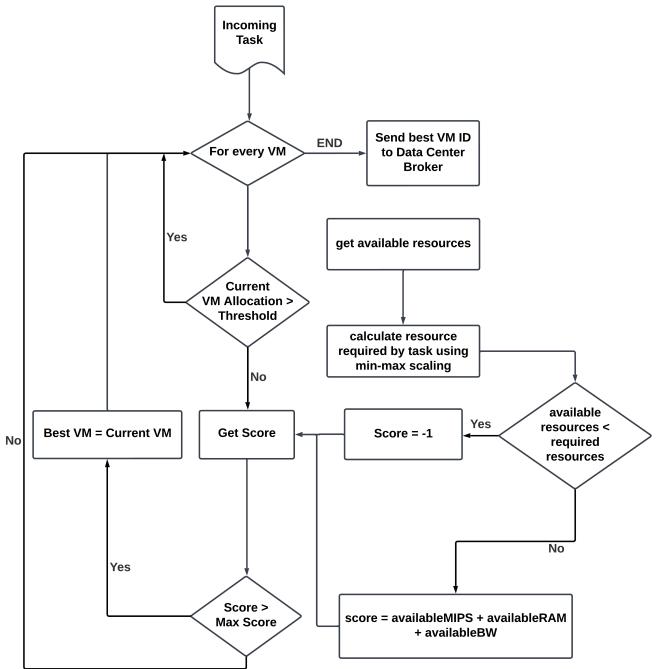


Fig. 2: Score Based Dynamic Load Balancer

is computed by summing the available MIPS, RAM, and bandwidth as follows:

$$\text{Score} = \text{availableMIPS} + \text{availableRAM} + \text{availableBW}$$

The VM with the highest suitability score is selected for task allocation and passed to the Data Center Broker. Once assigned, the VM utilizes its available resources (MIPS, RAM, and bandwidth) in proportion to the task's normalized requirements.

$$y = \frac{(x - x_{\min})}{(x_{\max} - x_{\min})} \times (y_{\max} - y_{\min}) + y_{\min} \quad (1)$$

Min-max scaling maps an input value x from the original range $[x_{\min}, x_{\max}]$ to a target range $[y_{\min}, y_{\max}]$, enabling consistent evaluation of heterogeneous task sizes and resource needs.

B. Performance Metrics

To evaluate the efficiency of the proposed load balancing algorithm, three core performance metrics are used: average response time, average data center processing time, and operational cost.

1) *Average Response Time*: Average response time measures the time taken to process a task from the moment it is acknowledged by the load balancer until completion. It reflects both scheduling efficiency and VM performance. Given n tasks, the average response time is calculated as:

$$\bar{R} = \frac{1}{n} \sum_{i=1}^n R_i$$

where R_i is the response time of the i -th task. Lower values indicate quicker task handling and better system responsiveness.

2) *Average Data Center Processing Time*: This metric captures the average time a data center spends processing all assigned tasks. The total processing time for a data center is given by the difference between the time the last task finishes and the time the first task starts, i.e., $T_{DC} = T_{last_finish} - T_{first_start}$. To calculate the average processing time across n data centers, we sum the total processing times of each data center and divide by the number of data centers:

$$P_{DC} = \frac{1}{n} \sum_{i=1}^n T_{DC,i}$$

where $T_{DC,i}$ is the total processing time for the i -th data center.

3) *Operational Cost*: The operational cost of a data center is directly tied to how long it remains active for task processing. Using the total processing time calculated earlier, the cost to operate the data center is:

$$C_{DC} = T_{DC} \times CostPerSecCPU$$

where $CostPerSec$ is the cost of CPU usage per second. This metric is useful for comparing the cost-efficiency of different load balancing strategies.

C. Setting Up Task Threshold

To prevent VM overloading from bursts of small tasks, a task threshold was set to limit active tasks per VM. Extensive testing across workloads (Figure 3) using 1 to 8 data centers, 2000 tasks across 250 batches, and thresholds of 2, 3, and 4, revealed that a threshold of 3 offered optimal performance. It matched the efficiency of threshold 4 while yielding lower response times than threshold 2. A threshold of 5 caused overload in single DC setups due to limited capacity.

D. Task Scheduling and Load Balancing

Figure 4 illustrates the workflow of the proposed simulation framework implemented in CloudSim 7G. The simulation models a cloud environment where tasks of varying complexity are dynamically scheduled to virtual machines via a data center broker and a load balancing mechanism. The simulation initializes data centers and VMs with heterogeneous configurations. Tasks are generated in batches and categorized by computational complexity. The data center broker manages task assignment using two separate load balancing

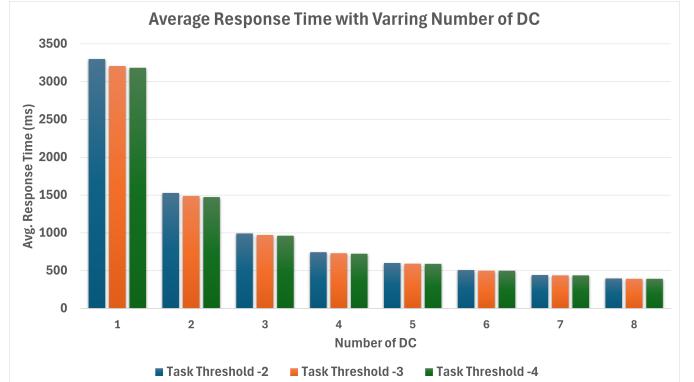


Fig. 3: Average Response Time with Varying Task Threshold

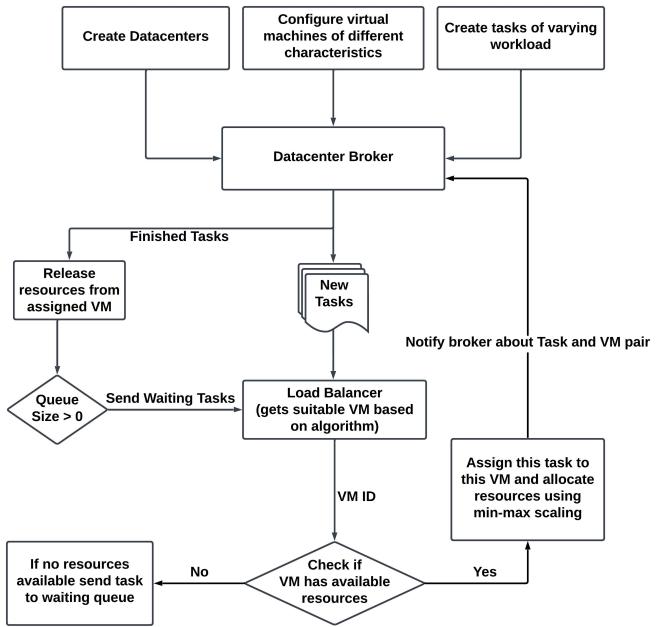


Fig. 4: Task Scheduling and Load Balancing Flow in CloudSim 7G

strategies: (1) the Throttled Load Balancer, which distributes tasks sequentially across VMs, and (2) the proposed Score-Based Dynamic Load Balancer, which allocates tasks based on a real-time scoring mechanism. If a VM has sufficient resources, the task is assigned proportionally based on normalized task length (Equation 1). Longer tasks consume more resources, ensuring balanced distribution. If no VM meets the requirements, the task is queued until capacity is available. Upon task completion, resources are released and queued tasks are reassessed for execution.

V. EXPERIMENTAL SETUP

A. Simulation Environment

The simulation environment is designed to approximate the infrastructure of social media platforms. As of February 2025, Meta operates 24 data center campuses globally [28].

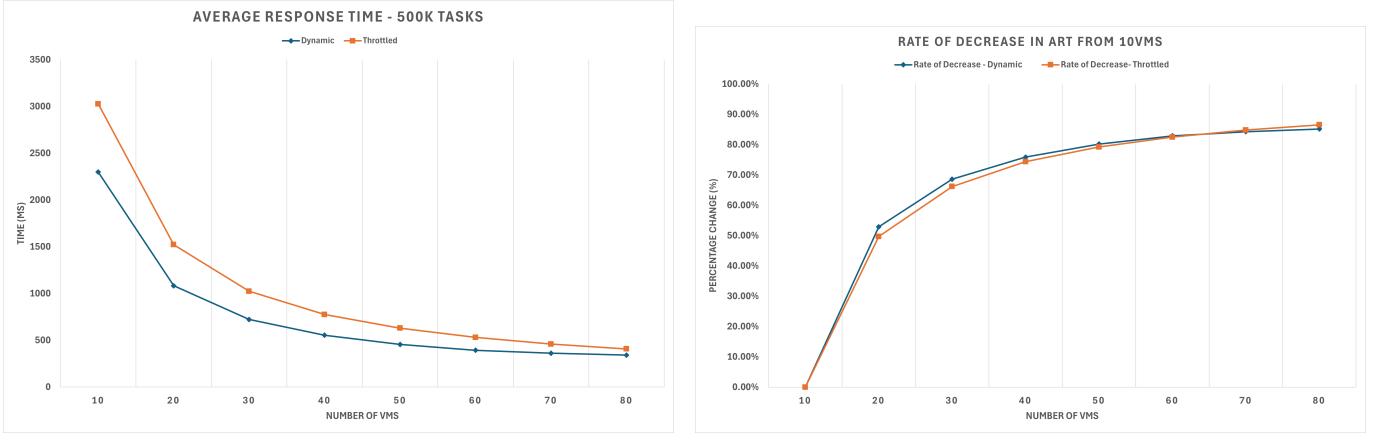


Fig. 5: Performance Metric Over 10 - 80 VMs for 500K Tasks

To maintain a manageable yet realistic model, this number is scaled down by a factor of three, resulting in eight data centers used across simulation scenarios. To enhance realism, tasks are categorized into three types based on data size, complexity, and CPU requirements: *Reels*, *Images*, and *Text Posts*. *Reels* are the largest, ranging from 10 MB to 1 GB [29], followed by *Images* (1–30 MB) [30], and *Text Posts* (10–100 KB).

Task distribution follows current media consumption trends: 60% *Reels*, 30% *Images*, and 10% *Text*. Studies show video content dominates user engagement and retention [31], [32], while images remain crucial due to high visual processing efficiency [33]. This breakdown reflects real-world usage and provides a practical basis for cloud task simulation in CloudSim.

B. Estimating Computational Demand by Task Type

Task size, measured in Million Instructions (MI), is determined by the workload size and the computational intensity (CI), which represents the number of CPU instructions required to process one byte of input. CI reflects the computational demand of a task and varies across task categories: lightweight tasks (e.g., text processing) typically have low CI (10–100 instructions per byte), moderate tasks (e.g., image processing) require a moderate CI (500–1,000 instructions per byte), and heavy tasks (e.g., video transcoding or compression) involve a high CI (1,000–10,000 instructions per byte). The instruction length for a given task is calculated by multiplying the data size by the task's CI, resulting in the following formula for MI:

$$MI = \frac{\text{Data Size (Bytes)} \times CI}{10^6}$$

VI. EXPERIMENTAL ANALYSIS

This section presents four cloud-based simulation scenarios comparing SBDLB with throttled load balancing. Configurations and metrics are summarized in Table IV. Experiments varied task loads from 100K to 500K in 100K steps, with a

batch size of 2000. Due to space constraints, only one representative result per scenario is shown. To confirm consistency and statistical significance, p-values are included. Detailed results follow in the subsections.

A. S-1: VM Scalability

This experiment evaluated how different load balancing strategies affect system performance under varying workloads and resource configurations. It focused on assessing the scalability and effectiveness of SBDLB versus throttled-based load balancing by varying the number of VMs per data center (10 to 80) and total incoming tasks, measuring average response time as the primary metric. The system used eight active data centers. The VM range was based on preliminary results showing response time plateaus beyond 80 VMs.

SBDLB consistently outperformed throttled. At 500K tasks and across the full VM range, SBDLB achieved a 34% lower average response time (Figure 5a). This improvement is statistically significant ($p = 3.54 \times 10^{-10}$), highlighting SBDLB's superior efficiency in handling large-scale, distributed workloads.

A key finding was that increasing VMs from 10 to 20 cut average response time by 50%, but gains diminished with further increases. Response time plateaued around 60 VMs (Figure 5b), which was chosen as the standard for later experiments. This trend illustrates the point of diminishing returns: while initial VM increases yield major gains, beyond 60, added resources provide minimal benefit. This insight supports efficient infrastructure planning by balancing performance with cost.

B. S-2: Varying DCs

This experiment assessed the scalability and efficiency of the SBDLB algorithm by varying the number of data centers (1 to 8) while keeping 60 VMs per center, as identified optimal in Scenario 1. Key metrics included average response time, processing time, and operating costs.

TABLE IV: Cloud Scenarios And Analyzed Metrics

Scenario Name	Description	Variable Factors	Analyzed Metrics
Scenario 1: VM Scalability	Evaluates how changing the number of VMs per DC affects system performance, with the total number of DCs fixed at eight	The number of virtual VMs assigned to each DC and the total volume of tasks to be processed.	Average response time
Scenario 2: Varying DC	Analyzes the effect of varying the number of data centers while keeping the number of VMs per DC constant at 60	The number of data centers actively participating in task processing	average response time, DC processing time, and DC operating cost
Scenario 3: Task Allocation	Examines how two load balancers distribute tasks between high-spec and low-spec virtual machines	The total number of tasks entering the system over a given period	Task distribution across high-spec and low-spec VMs
Scenario 4: 24-Hour Variation	Assesses how effectively two load balancers distribute workloads across 60 virtual machines over a continuous 24-hour simulation period	The timing and intensity of peak usage periods, including how many tasks are received during those high-traffic intervals	Hourly breakdowns of average response time and data center processing time and operational cost of DC for a full 24-hour period

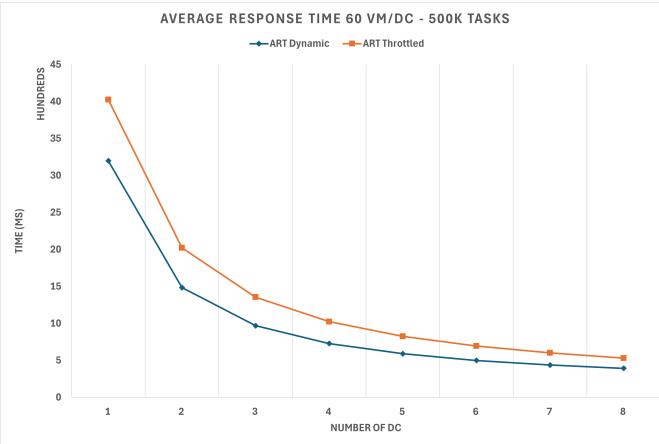


Fig. 6: Average Response Time For Varying Number Of DCs Using 60VM/DC For 500k Tasks

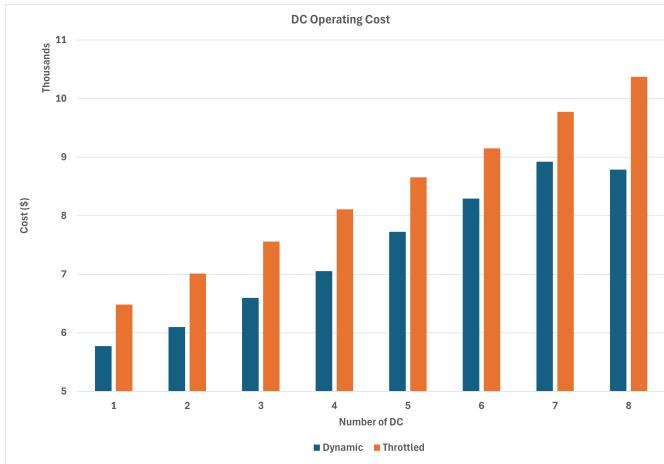


Fig. 7: Data Center Operating Costs for 500K Tasks

As shown in Figure 6, SBDLB consistently outperformed throttled load balancing across all configurations. For a 500K task load, it achieved a 37% lower average response time, with a highly significant p-value of 3.35×10^{-12} . Notably, SBDLB maintained better performance even with fewer data centers. At 500K tasks, it completed processing in 970 ms

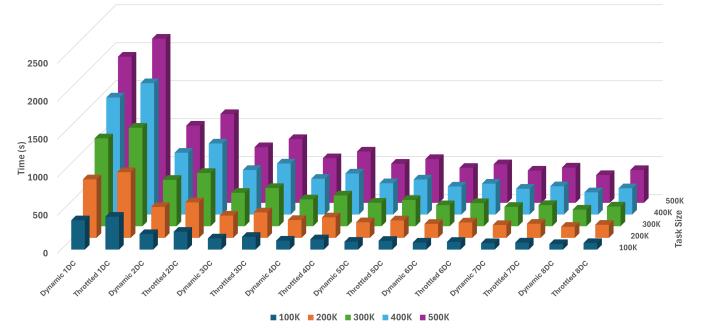


Fig. 8: DC Processing Time Over 1-8 DCs And Task Size of 100K - 500K

using 3 DCs, while throttled required 4 DCs and still had a slower response time of 1024 ms. This trend held across all workloads from 100K to 500K, highlighting SBDLB's efficiency in reducing resource use and operational overhead.

These gains translate to lower energy use, infrastructure demands, and cost—supporting sustainable, scalable cloud deployment. Figure 7 shows that SBDLB reduces operating costs, driven by faster task completion and shorter resource active time. Figure 8 further shows that SBDLB cut data center processing time by 13% on average, with a statistically significant p-value of 4.48×10^{-9} . These results reinforce SBDLB's advantages in performance, cost-efficiency, and sustainable resource management.

C. S-3: Task Allocation

This experiment evaluated task distribution strategies in a heterogeneous cloud setup, showing that SBDLB achieves better performance and resource utilization than the throttled approach.

As shown in Figure 9, SBDLB assigns more tasks to high-spec VMs (crests) and fewer to low-spec ones (troughs), optimizing processing power and avoiding bottlenecks. In contrast, the throttled method distributes tasks uniformly, overloading weaker VMs and underusing stronger ones. These results confirm that intelligent, capability-aware scheduling significantly improves efficiency and system performance.

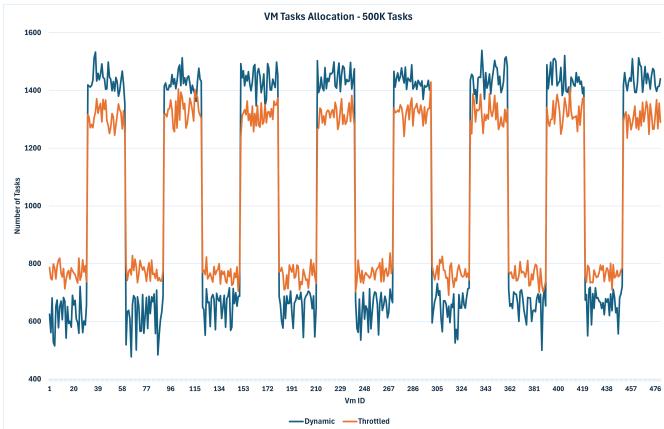


Fig. 9: VM Task Distribution for 500K Tasks

D. S-4: 24-Hour Variation

To provide a more comprehensive evaluation, this study analyzes SBDLB over a 24-hour period, distinguishing between peak and non-peak hours based on user demand [34], [35]. Hourly configurations are detailed in Table V. To better reflect real-world workloads, the simulation introduces random variations in batch size and batch count per hour. This stochastic setup accounts for natural demand fluctuations, enhancing the robustness and realism of the results.

TABLE V: Batch Processing Details by Hour Type

Hour Type	Hours	Batch Sizes	Total Batches
Peak	8-10, 13-14, 17-22	5K, 5.5K, 6K	18,19,20
Non-Peak	0-7, 11-12, 15-16, 23-24	3K, 3.5K, 4K	9,10,11

1) *Average Response Time Per Hour*: Figure 10 shows hourly average response times for SBDLB and throttled load balancing under varying workloads. SBDLB consistently outperforms throttled across the 24-hour period, adapting more effectively to workload fluctuations and optimizing resource use. Throttled shows noticeable response time spikes during peak hours (8–10 AM, 1–2 PM, and 5–10 PM), reflecting its struggle with high traffic due to static resource allocation. While both methods perform better during non-peak hours, throttled still lags slightly. Though the gap narrows under lower load, SBDLB maintains a performance edge. Faster task completion with SBDLB also reduces congestion, operational costs, and energy use, emphasizing its advantages in both efficiency and sustainability.

2) *DC Processing Time Per Hour*: Figure 11 shows hourly data center processing times for SBDLB and throttled load balancing. SBDLB consistently reduces processing time across all hours, handling both peak and non-peak workloads more efficiently. During peak hours, throttled struggles with workload surges, while SBDLB maintains better performance. In non-peak hours, both improve, but throttled shows greater variability, whereas SBDLB remains stable. By completing tasks faster, SBDLB not only increases efficiency, but also

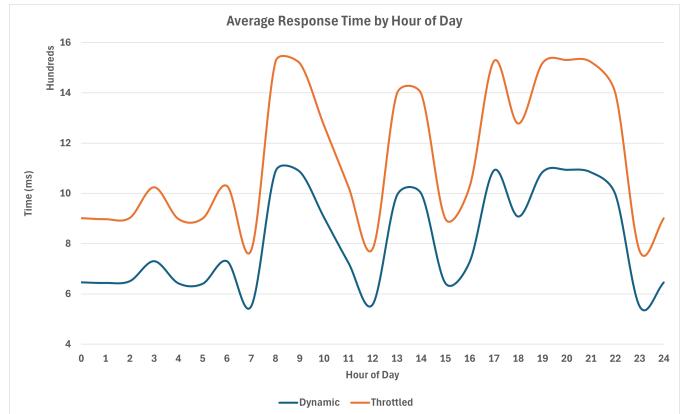


Fig. 10: Average Response Time by Hour

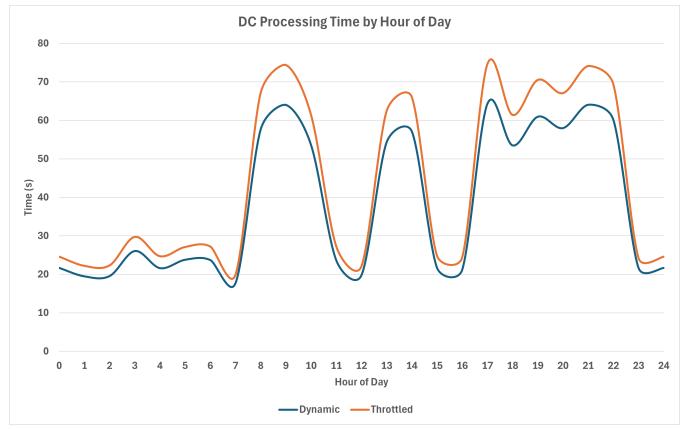


Fig. 11: DC Processing Time by Hour

reduces energy use and operational costs, making it a more cost-effective cloud solution.

3) *DC Operating Cost Per Hour*: Hourly data center costs follow the trend in Figure 11, as cost is proportional to processing time. Both metrics peak during high-load hours and decline during non-peak periods. SBDLB consistently reduces costs compared to throttled, with the greatest savings during peak hours. In non-peak times, the cost difference narrows but still favors SBDLB.

E. Total Cost Analysis Over 24 Hours

Operating the data center for 24 hours (per Table V) costs \$22,818 with SBDLB versus \$26,246 with throttled—a 15.02% reduction. This cut is significant for both cost and environmental impact. By optimizing processing during peak hours, SBDLB lowers energy use, helping reduce carbon emissions from data centers—a major source of global energy demand [36]. Efficient load balancing not only saves money but also supports sustainability by reducing computational overhead and the data center’s carbon footprint.

VII. CONCLUSION

Cloud technology has become central to modern digital infrastructure, with global adoption on the rise. Efficient load

balancing is vital for optimizing performance, enhancing user experience, and reducing operational costs. This study proposes SBDLB, a dynamic load balancing method that adapts to varying conditions and outperforms the traditional throttled approach. Results show SBDLB reduces task response time by 34%–37%, lowers data center processing time by 13%, and completes workloads more efficiently using fewer resources. These gains translate to cost savings and lower energy consumption, promoting both economic and environmental sustainability. Future work will explore scalability across diverse data center setups and VM types, with potential enhancements through heterogeneous VMs and reinforcement learning for self-optimizing performance.

REFERENCES

- [1] Y. Li, W. Zhao, Y. Su, W. Li, and C. Yuan, “Overview of cloud computing deployment mode and technology development trend,” in *2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE, 2023, pp. 1–5.
- [2] H. Tabrizchi and M. Kuchaki Rafsanjani, “A survey on security challenges in cloud computing: issues, threats, and solutions,” *The journal of supercomputing*, vol. 76, no. 12, pp. 9493–9532, 2020.
- [3] H. Pallathadka, G. S. Sajja, K. Phasinam, M. Ritonga, M. Naved, R. Bansal, and J. Quiñonez-Choquecota, “An investigation of various applications and related challenges in cloud computing,” *Materials Today: Proceedings*, vol. 51, pp. 2245–2248, 2022.
- [4] S. Zoting, “Infrastructure as a service (iaas) market size to hit usd 898.52 bn by 2034,” January 28 2025, accessed: March 5, 2025. [Online]. Available: <https://www.precendceresearch.com/infrastructure-as-a-service-market>
- [5] S. Paul and M. Adhikari, “Dynamic load balancing strategy based on resource classification technique in iaas cloud,” in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018, pp. 2059–2065.
- [6] M. Kumar and S. C. Sharma, “Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing,” *Procedia computer science*, vol. 115, pp. 322–329, 2017.
- [7] S. Y. Mohamed, M. H. N. Taha, H. N. Elmahdy, and H. Harb, “A proposed load balancing algorithm over cloud computing (balanced throttled),” *International Journal of Recent Technology and Engineering*, vol. 10, no. 2, pp. 28–33, 2021.
- [8] H. N. Le and H. C. Tran, “Ita: The improved throttled algorithm of load balancing on cloud computing,” *International Journal of Computer Networks & Communications (IJCNC)*, vol. 14, 2022.
- [9] F. Duarte, “Amount of data created daily (2024),” *Exploding Topics*, June 13 2024, accessed: 2025. [Online]. Available: <https://explodingtopics.com/blog/data-generated-per-day>
- [10] A. K. Sandhu, “Big data with cloud computing: Discussions and challenges,” *Big Data Mining and Analytics*, vol. 5, no. 1, pp. 32–40, 2021.
- [11] S. Jain, “A survey of load balancing challenges in cloud environment proceedings of the smart—2016,” in *IEEE Conference ID*, vol. 39669, 2016.
- [12] M. S. Al Reshan, D. Syed, N. Islam, A. Shaikh, M. Hamdi, M. A. Elmagzoub, G. Muhammad, and K. H. Talpur, “A fast converging and globally optimized approach for load balancing in cloud computing,” *IEEE Access*, vol. 11, pp. 11 390–11 404, 2023.
- [13] C. Gao and H. Wu, “An improved dynamic smooth weighted round-robin load-balancing algorithm,” in *Journal of Physics: Conference Series*, vol. 2404, no. 1. IOP Publishing, 2022, p. 012047.
- [14] N. Rathore, “Dynamic threshold based load balancing algorithms,” *Wireless Personal Communications*, vol. 91, no. 1, pp. 151–185, 2016.
- [15] A. Katangur, S. Akkaladevi, and S. Vivekanandhan, “Priority weighted round robin algorithm for load balancing in the cloud,” in *2022 IEEE 7th international conference on smart cloud (SmartCloud)*. IEEE, 2022, pp. 230–235.
- [16] S. Chowdhury and A. Katangur, “Threshold based load balancing algorithm in cloud computing,” in *2022 IEEE international conference on joint cloud computing (JCC)*. IEEE, 2022, pp. 23–28.
- [17] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, “A load balancing algorithm for the data centres to optimize cloud computing applications,” *Ieee Access*, vol. 9, pp. 41 731–41 744, 2021.
- [18] S. Chowdhury, A. Katangur, A. Sheta, N. R. Psayadala, and S. Liu, “Genetic algorithm based service broker policy to find optimal datacenters in cloud services,” in *2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCBDA)*. IEEE, 2023, pp. 270–278.
- [19] A. Y. Zomaya and Y.-H. Teh, “Observations on using genetic algorithms for dynamic load-balancing,” *IEEE transactions on parallel and distributed systems*, vol. 12, no. 9, pp. 899–911, 2001.
- [20] M. Pai, S. Rajarajeswari, D. Akarsha, and S. Ashwini, “Analytical study on load balancing algorithms in cloud computing,” in *Expert Clouds and Applications: Proceedings of ICOECA 2021*. Springer, 2022, pp. 631–646.
- [21] M. Gokul and M. Balamurali, “Cloud load balancing using metaheuristics,” in *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2022, pp. 589–595.
- [22] G. Sriram, “Green cloud computing: an approach towards sustainability,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 4, no. 1, pp. 1263–1268, 2022.
- [23] U. Demirbaga, “Ecocloud: Green computing through energy and carbon efficient task scheduling in industrial iot-enabled cloud environments,” *IEEE Internet of Things Journal*, 2025.
- [24] X. Sui, D. Liu, L. Li, H. Wang, and H. Yang, “Virtual machine scheduling strategy based on machine learning algorithms for load balancing,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 160, 2019.
- [25] R. Andreoli, J. Zhao, T. Cucinotta, and R. Buyya, “Cloudsim 7g: An integrated toolkit for modeling and simulation of future generation cloud computing environments,” *Software: Practice and Experience*, 2025.
- [26] R. A. M. Razali, R. Ab Rahman, N. Zaini, and M. Samad, “Virtual machine migration implementation in load balancing for cloud computing,” in *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*. IEEE, 2014, pp. 1–4.
- [27] G. Soni and M. Kalra, “A novel approach for load balancing in cloud data center,” in *2014 IEEE international advance computing conference (IACC)*. IEEE, 2014, pp. 807–812.
- [28] M. Zhang, Meta’s data center locations for facebook and instagram. [Online]. Available: <https://dgtlinfra.com/meta-data-center-locations-facebook/>
- [29] Q. Team. (2025, January 27) Facebook reels in 2025: Definitive guide for marketers. [Online]. Available: <https://quickframe.com/blog/facebook-reels-guide-for-marketers/>
- [30] D. Lamaj. (2024, August 14) Facebook image sizes - must-read guide (updated). Publer Blog. [Online]. Available: <https://publer.com/blog/facebook-image-sizes/>
- [31] K. McCormick. (2024, 1) 75 staggering video marketing statistics. WordStream. [Online]. Available: <https://www.wordstream.com/blog/ws/2017/03/08/video-marketing-statistics>
- [32] E. Lukan. (2023, 11) 50 video statistics you can’t ignore in 2025. [Online]. Available: <https://www.synthesia.io/post/video-statistics>
- [33] Crackitt. (2018, 6) State of visual content marketing: the statistics. [Online]. Available: <https://www.crackitt.com/state-of-visual-content-marketing-videos-images-statistics/>
- [34] P. Rajput and S. Kumar, “Simulation of a large scaled web application on the cloud using cloud analyst,” *Simulation*, vol. 10, no. 9, pp. 46–56, 2014.
- [35] A. K. Dubey and V. Mishra, “Performance analysis of cloud applications using cloud analyst,” in *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 2017, pp. 79–84.
- [36] Y. S. Patel, P. Townend, A. Singh, and P.-O. Östberg, “Modeling the green cloud continuum: integrating energy considerations into cloud-edge models,” *Cluster Computing*, vol. 27, no. 4, pp. 4095–4125, 2024.