Journal of Computing Theories and Applications ISSN:3024-9104

Research Article

Optimizing Cloud Computing Performance by Integrating the Novel PSBR Service Broker Policy and Load Balancing Algorithms

Nan Kham Mon*

Faculty of Computer Systems and Technologies, University of Computer Studies, Myitkyina, Myanmar; email: nankhammon29@gmail.com
* Corresponding Author : Nan Kham Mon

Abstract: As cloud computing advances, organizations' IT infrastructure and application deployment processes are moving to the cloud because cloud computing provides everything as a service over the Internet. The performance of a cloud-based application is based on proper datacenter selection and workload distribution within the selected datacenter. Service broker policies are used for suitable datacenter selection, and load balancing algorithms(LBA) are applied to distribute workloads. This paper is to evaluate the effect of a proposed service broker policy (PSBR) on the performance of cloud-based applications with LBA. To achieve the objective, the behavior of the TikTok application was modeled using the worldwide users' statistics on the cloud simulation framework, namely CloudAnalyst. As a result, the average response time and data center processing time are measured. Next, the PSBR provides better results than the existing service proximity-based policy. This paper supports cloud service providers' benefits, from coordination between data center configuration, data center selection, and workload distribution to cloud users' identification of the appropriate procedures for their organization or application. PSBR with Active Monitoring had the best average response time of 75.1 ms, while SPR consistently exhibited higher average times across all algorithms, with the highest being 84.5 ms for Round Robin. Under the PSBR policy, Throttled had the lowest average processing time (4.67), while Round Robin had the highest (5.72). Similarly, under the SPR policy, Throttled maintained its efficiency with the lowest average (4.8), while Round Robin showed the highest (5.79).

Keywords: Cloud computing; Datacenter selection; CloudAnalyst; Load balancing; Service broker

Received: July, 22nd 2024
Revised: September, 8th 2024
Accepted: September, 27th 2024
Published: October, 3rd 2024

1. Introduction

Cloud computing leverages virtualization and grid computing technologies to offer users flexible and virtually unlimited computing resources. The widespread adoption of cloud computing by commercial and industrial users can be attributed to its five fundamental characteristics. With cloud computing, users can independently manage their computing infrastructure and services without interacting with the service provider. Thanks to its broad connectivity options, they can access these resources from various client platforms through the Internet [1]–[4]. This technology allows for scaling services according to demand and provides measurable usage, enabling users to share resources from a centralized pool, such as data centers. As a result, cloud computing is extensively used for developing and hosting applications in diverse fields like social media and e-commerce due to its cost-effectiveness and operational efficiency. Nonetheless, challenges such as selecting the appropriate data center and balancing loads must be addressed for optimal application deployment in the cloud environment. Cloud service brokers play a crucial role by utilizing specific policies to choose the right data center.

However, cloud users' needs, constraints, and demands can vary significantly. For instance, while some users prioritize quick response times regardless of cost, others may be more budget-conscious [5]–[8]. When multiple users have similar demands, the designated data center might become overloaded, leading to diminished performance. Factors such as the geographic locations of data centers, the distribution of users, the number of available data centers in a given area, and the suitability of these data centers all critically influence the overall performance in a cloud environment. Consequently, selecting an appropriate data center enhances performance and satisfies diverse user requirements. Some service broker policies include the Service Proximity Based Policy (SPR), Best Response Time Service Broker Policy, and Dynamic Service Broker Policy.

The SPR operates by selecting the data center nearest to the user's request, considering factors like network latency and bandwidth. However, in cases where multiple data centers exist within the same region, this policy resorts to random selection. This random selection approach can lead to undesirable outcomes such as Increased user response times and Elevated datacenter processing times.

This system proposes a new service broker policy to fix these issues. To address the aforementioned problems, this policy expands on the SPR current random selection feature. This paper investigates how the proposed service broker policy (PSBR) affects cloud-based applications' performance. The use of this suggested strategy is then evaluated using a variety of load-balancing algorithms(LBA). The following is a summary of this system's contributions:

- Resolving the issue of selecting data centers at random and minimizing user response times
- Shortening processing times in data centers

Statistics from TikTok users [9] are used as an example application in the CloudAnalyst simulation framework during implementation. The performance of this application is then assessed using the PSBR alongside existing service proximity-based policies. Three LBAs available within the CloudAnalyst simulation framework have been applied for evaluation. The proposed PSBR selects the most suitable data center based on factors such as response time, processing time, and cost. This proposed policy assists cloud service providers and application designers in effectively determining an optimal service broker policy and load-balancing algorithm for cloud-based applications.

The remainder of the document will be outlined as follows: Section 2 delves into the existing literature related to the proposed policy. Following that, Section 3 provides an overview of the theoretical background underlying the proposed policy. Section 4 introduces the proposed system, while Section 5 presents the proposed policy's simulation findings and performance results. The document concludes with a summary of the system in Section 6.

2. Related Works

The cloud brokering architecture involves five key entities: cloud suppliers, merchants, evaluators, transporters, and buyers, all essential components in cloud computing processes [10]. Buyers request tasks from cloud suppliers, and evaluators gather essential data. Intermediaries then provide buyers with the virtualized infrastructure and service descriptions, including optimization criteria, virtual machine configurations, Service Measurement Index (SMI) traits, and data center choices. Challenges such as higher processing and response times, workload issues, and underutilization of resources may arise. To address these challenges, selecting data centers based on factors like response time, processing time, workload, and cost is advisable. The authors [11] introduced a novel approach where we devised access policies for authentication and authorization.

Additionally, we integrated an option to check for endpoint security. The setup allows for creating multiple profiles catering to various access methods, each with its distinct access policy. For instance, a web access authentication policy can be tailored for dynamic Access Control List (ACL) connections. Through our experimental methodology, it becomes feasible to swiftly identify users, their locations, prevailing network conditions during access, and server status. Armed with these metrics and insights, we enhance the capability to fortify mission-critical applications operating within intricate corporate environments. With the exponential growth of internet users and the increasing load of IoT devices on networks, the demand for cloud computing services has surged [12]. Cloud technology offers on-demand services to customers, albeit with certain latency overheads. To address this, fog-integrated cloud computing emerges as a promising solution, given its proximity to users. However, this innovation presents challenges, such as efficient task scheduling and load balancing among servers, owing to resource constraints. In this study, the authors proposed two dynamic service broker policies: the modified-service proximity and modified-optimize response time.

These policies are designed to address the limitations of existing approaches, namely service proximity (SP) and optimize response time (ORT). Through simulations, they demonstrated that the modified-SP policy achieves superior re-sponse and processing times compared to SP, while the modified-ORT policy outperforms ORT in terms of response time, albeit with slightly increased processing overhead.

Cloud brokering mechanisms are crucial facilitators in managing cloud resources by mediating between cloud service providers and users [13]. The primary objective is to efficiently fulfill user requests in real-time while minimizing monetary costs by selecting suitable data centers and virtual machines. A pioneering approach—a normalization-based hybrid service brokering method integrated with throttled round-robin load balancing—is introduced to enhance resource management by considering cost and performance factors in cloud service provisioning. This approach evaluates the impact of cost and performance-oriented parameters in a multi-cloud environment by employing a hybrid evaluation criterion, which incorporates static and dynamic elements, and utilizing normalization techniques. Subsequently, selecting the most suitable service provider and throttled round-robin load balancing optimizes cloud resource allocation. Experimental results across various user bases and data centers demonstrate the superiority of this approach, showcasing significant enhancements in response time (up to 17.39%), data center processing time (up to 31.35%), and monetary cost (up to 7.06%) compared to established methods. The paper [14] introduced a novel Priority-based Shortest Job First Policy for Data Center Brokers in cloud environments, termed Priority_DCB. The policy selects processes for execution in ascending order of priority, with tiebreakers resolved using the Shortest Job First principle. It offers a straightforward solution for executing Cloudlets based on their priorities. Implementation and simulation of the proposed approach are conducted using CloudSim. Results indicate that it outperforms other state-of-the-art Data Center Broker Policies regarding Cloudlets' turna-round time and waiting time. The paper presented a novel and economical service broker policy for Data Center (DC) selection in heterogeneous cloud environments, leveraging the VIKOR method while accommodating users' specified priorities[15]. Based on these priorities, it aims to minimize response time and overall cost in user-oriented cloud systems. Experimental results across diverse scenarios show that the proposed solution consistently outperforms existing policies in terms of response time, DC processing time, and total cost.

The paper [16] identifies a significant gap in existing scheduling algorithms: the lack of an adaptive approach capable of simultaneously addressing load balancing and optimizing Quality of Service (QoS) requirements. A novel adaptive method was introduced to bridge this gap, merging the best-worst multi-criteria decision-making method (BWM) with the compromise ranking method (VIKOR). VIKOR functions as the decision maker, prioritizing tasks within this innovative approach. Through numerical experiments, the efficacy of the proposed method is validated, with comparisons made against existing scheduling algorithms across a range of performance metrics. The simulation results underscored substantial improvements in metrics like throughput, makespan, waiting time, virtual machine (VM) utilization, and VM usage cost across all experimental scenarios, showcasing its superiority over existing approaches. The article [17] offered a thorough examination of a variety of Cloud Service Broker (CSB) policies, simultaneously tackling the challenges faced by current policies. Its primary goal is to identify research gaps and suggest solutions to improve future policy development. It also elucidates different methodologies for Data Center (DC) selection in Cloud Computing (CC), catering to practitioners and researchers. Synthetic analysis systematically assesses and compares numerous DC selection techniques, providing decision-makers with a practical framework to choose the most appropriate technique for their needs. Ultimately, the article strongly underscores the vital role of effective CSB policies in DC selection, emphasizing their importance in enhancing CC performance. By highlighting the significance of these policies and their modeling implications, the article contributes to both the broader modeling discourse and its practical implementation in the CC domain. The paper [18] highlighted the effectiveness of the differential evolution algorithm, a renowned metaheuristic known for its speed and robustness, in the context of selecting optimal cloud data centers. It introduces a customized version of this algorithm-based cloud service broker policy, specifically designed for selecting the most suitable data center within cloud computing environments. Incorporating a novel mutation technique enhances the differential evolution algorithm to improve performance and accuracy in data center selection. Its effectiveness is assessed using the CloudAnalyst simulator, with comparative analyses against state-of-the-art cloud service broker policies.

The authors addressed [19] the increasingly complex task of selecting optimal cloud services amidst the expanding array offered by multi-cloud providers. It delves into multi-ti-objective location-aware Service Brokering (MOLSB), a methodology aimed at minimizing both cost and latency while providing a spectrum of solutions. Various heuristics have been proposed to handle dynamic resource requirements efficiently, yet they often lack consistency in delivering optimal performance across diverse scenarios. Recognizing this limitation, the paper advocates for developing a suite of effective heuristics capable of balancing multiple objectives with varying trade-offs. Leveraging Genetic Programming hyper-heuristics (GPHH), traditionally used in solving multi-objective dynamic optimization problems like workflow scheduling, the paper introduces GPHH-MOLSB. It automatically generates a repertoire of Pareto-optimal heuristics to cater to diverse Quality of Service (QoS) preferences. Evaluation of real-world datasets reveals significant performance enhancements of GPHH-MOLSB over several existing methods, underscoring its potential in addressing the challenges of cloud service selection in multi-cloud environments. The article [20] thoroughly examined cloud storage utilization, focusing on exploring opportunities, motivations, and challenges associated with cost optimization from the user's standpoint. It is initiated by scrutinizing the complete process of employing cloud storage, addressing aspects such as enhancing storage efficiency, leveraging cloud storage service features, and adopting emerging storage paradigms like edge storage to minimize costs. Through this exploration, the article presents a detailed taxonomy and overview of recent advancements in these domains.

Additionally, it outlines potential future directions for cost optimization from the user's viewpoint and provides concluding insights. By providing an in-depth survey of recent developments aimed at optimizing cloud storage costs for users, this article is poised to attract a diverse audience within the cost-effective cloud storage market. The paper [21] offered a comprehensive survey of the resource provisioning challenges encountered by cloud brokers. It commences by presenting a representative examination of cloud broker architecture. Subsequently, it delineates the resource provisioning problem into two primary categories: resource selection and resource management, discussing each in detail. This survey introduces a novel taxonomy of cloud broker research, accompanied by an analysis of current research endeavors. The paper concludes by providing valuable insights into potential future research avenues and identifying unresolved issues within the domain of cloud brokerage.

3. Methodology

A technique known as cloud computing uses the Internet to offer computer resources as services. Users can obtain cloud services, like computational power, storage, and databases, with pay-as-you-go pricing from cloud service providers like Amazon Web Services (AWS), Microsoft Azure, etc., instead of investing in and maintaining actual data centers and servers. Cloud computing provides consumers with everything as a service, including email, streaming, software delivery, application development and deployment, data analysis and retrieval, data backup and restoration, and more. However, these services can be broadly divided into three categories: platform as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS). A cloud represents the Internet and can be either private or public. The general public makes the public cloud available for anybody to use. A private cloud is one that a company offers for personal usage only. The primary distinction between public and private sectors is that the former employ common infrastructure, whereas the latter utilizes infrastructure that is exclusively their own.

3.1. Service Broker Policy

Cloud service brokers help cloud users negotiate the terms of service between cloud service providers and users and manage several cloud services within an organization. Aggregation, service intermediation, and service arbitrage are the three main responsibilities of a cloud service broker. A cloud service broker can assist cloud users in effectively obtaining computing resources while drastically lowering process costs, improving flexibility, and decreasing downtime. Additionally, cloud service brokers are in charge of choosing the right data center. They do this for incoming user requests by applying service broker policies like the following:

The SPR chooses the data center closest to the user. It will randomly choose a data center if multiple data centers are in the closest region. The broker policy extends the closest datacenter policy: Best Response Time Service Broker Policy. This policy assesses the current response time of the data centers when the nearest data center's response time starts to deteriorate. The data center with the fastest reaction time is then sought after. However, the closest and fastest data center selection process is based only on a 50:50 lottery and ignores the circumstances surrounding multiple data centers.

3.2. Load Balancing Algorithm

To ensure that no server is underloaded, overburdened, or idle, load balancers distribute the workload among the servers in a data center. In order to improve execution and response times as well as resource consumption, load balancers employ load-balancing algorithms. Load balancing techniques are used by server hubs in cloud computing to distribute workloads among available virtual machines. As of right now, CloudAnalyst has three LBA implemented:

- Round Robin Load Balancer: It uses a straightforward round-robin technique to distribute the virtual machines.
- Active Monitoring Load Balancer: It distributes the workload across the accessible virtual machines so that each virtual machine has an equal number of active jobs simultaneously.
- Throttled Load Balancer: It can only handle a certain number of jobs at once, allotted to one virtual machine. Some requests may need to be queued until the next virtual machine becomes available if there are more requests than there are available virtual machines.

4. Proposed Service Broker Policy

The policy for service brokers is being proposed. PSBR extended the closest data center policy. The closet data center policy randomly selects the data center when one or more are in the same region. The randomly selected data center can give unsatisfactory results. Therefore, when the region has one or more data centers, the PSBR chooses one depending on the price and the number of virtual machines housed in each data center.

Consequently, the PSBR chooses the data center with the shortest response time, data center processing time, and overall cost. Figure 1 is a flow diagram for the proposed PSBR. The flow diagram shows that PSBR gathers a list of data centers from the closest area in response to a request from the user to choose a data center. In the event that there is just one data center listed, the closest data center policy is used to choose that data center. When several data centers are on the list, the PSBR determines which one has the fastest response time by weighing each one. The weight value of each datacenter is depicted by Equation (1).

$$W_i = \frac{no\ of\ VM\ on\ each\ DC_i\ from\ closest\ region}{no\ of\ VM\ in\ all\ DC\ from\ closest\ region} \tag{1}$$

Where W_i refers to the weight of each data center, DC refers to the data center, DC_i refers to each data center, and VM refers to the virtual machine.

Then, the PSBR calculates the total cost of each data center based on the virtual machine cost and data transfer cost using Equation (2).

$$C_i = \left(\frac{MIPS_{total}}{VM_{MIPS}} \right) \times VM_{cost} + UDS \times DT_{cost} \tag{2}$$

Where, C_i refers to the total cost of each data center that considers both data transfer and virtual machine processing costs, ($MIPS_{total}/VM_{MIPS}$) refers to the total number of instructions per average processing power assigned to the data center. VM_{cost} denotes the virtual machine cost of each data center; UDS denotes the size of requests; and DT_{cost} denotes the data transfer cost. Then, the PSBR searches for the optimal values using weight and total cost with Equation (3).

$$Opt_i = w1 \times W_i + w2 \times C_i \tag{3}$$

Where, Opt_i refers to the optimal value of each data center, W_i refers to the weight value of each data center, and C_i refers to the total cost value of each data center. According to the general formula of multi-objective scalarization, the coefficient values will determine the solution of the fitness function and show the priority.

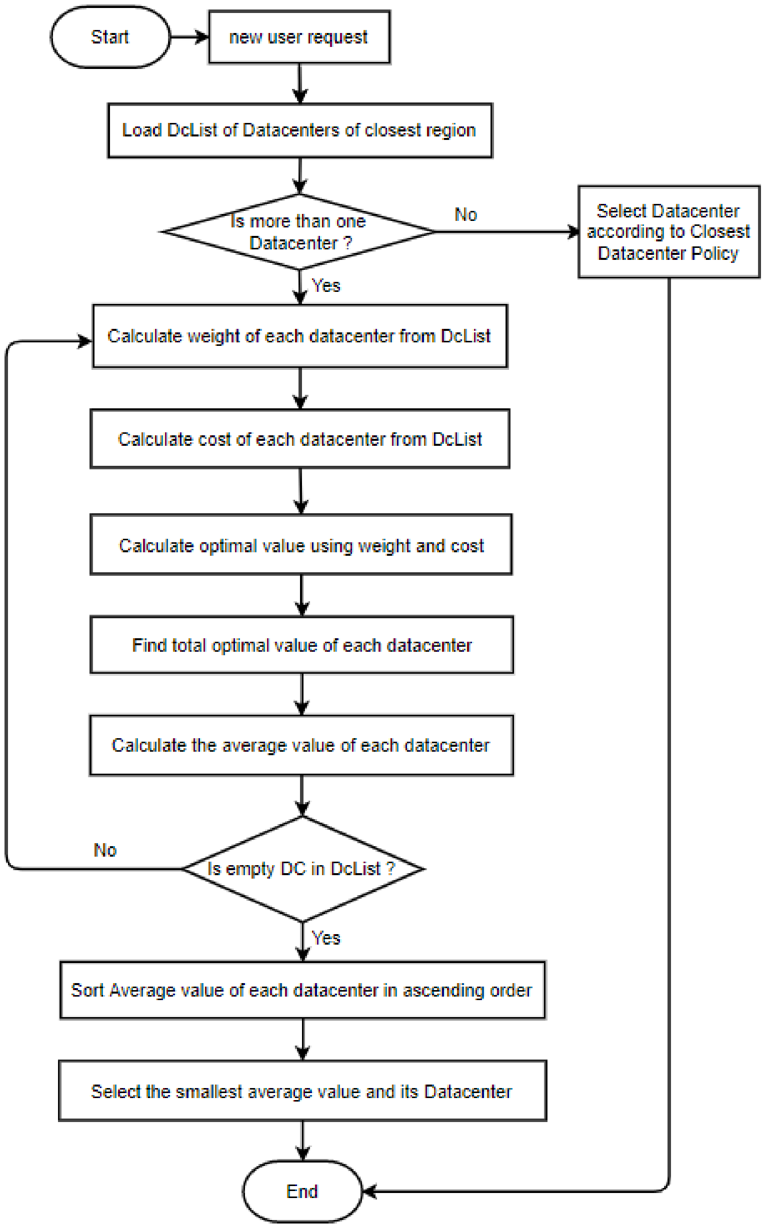

Figure 1. PSBR Flow Diagram

The larger the coefficient value, the higher the priority, and the sum of coefficient values must be 1. Therefore, $w1$ and $w2$ are the coefficient values. Then the total optimal and average values of each data center is calculated using Equations (4) and (5), respectively.

$$totalOpt_i = \sum_{i=1}^{N} Opt_i \quad (4)$$

$$Avg_i = \frac{totalOpt_i}{N} \quad (5)$$

Where, $totalOpt_i$ refers to the sum of optimal values of each datacenter, Opt_i refers to the optimal value for each data center, and N is the number of intervals. Avg_i refers to the average value of each data center. When the average values of all data centers have been calculated, the PSBR selects the data center with the smallest average value.

Algorithm 1. Proposed Service Broker Policy

```
INPUT: Datacenter List DC_i = {DC_1, DC_2, ..., DC_n}, User Request List UB_i = {UB_1, UB_2, ..., UB_n}, Region List R_i = {R_1, R_2, ..., R_6}
OUTPUT: Optimal Data Center List DC_i
Get a region proximity list
if there is new User Request UB_i then load Datacenter List DC_i of closest region
  if there is more than one Datacenter DC_i at that region R_i then
    for each Datacenter DC_i at that region R_i do
        Calculate the weight value W_i of each Datacenter DC_i using (1)
        Calculate the cost value C_i of each Datacenter DC_i using (2)
        Calculate the optimal value Opt_i of each Datacenter DC_i using (3)
        Find the total optimal value totalOpt_i of each Datacenter DC_i using (4)
        Calculate the average value Avg_i of each Datacenter DC_i using (5)
    end for
    Sort the average value Avg_i of each Datacenter DC_i in ascending order
    Select the smallest average value Avg_i and its Datacenter DC_i
    Return Datacenter DC_i
  else
    Return Datacenter DC_i according to SPR Policy
  end if
end if
```

The PSBR aims to optimize the selection of data centers based on user requests and region proximity. The algorithm takes as input a list of data centers (DC_i), user requests (UB_i), and regions (R_i), and outputs the optimal data center list. When a new user request arrives, the data centers in the closest region are loaded. If multiple data centers exist in the region, the algorithm computes the weight (W_i), cost (C), and optimal value (Opt_i) for each data center. The total optimal value (totalOpti) and average value (Avg_i) are calculated, after which the data centers are sorted in ascending order based on their average values. The data center with the smallest average value is selected and returned. If only one data center exists in the region, it is returned according to the Service Proximity Based Policy. This approach ensures efficient load distribution by prioritizing data centers with the lowest average costs and highest performance.

5. System Experiments

The experiments aim to demonstrate how the PSBR can affect cloud-based application performance when combined with LBA. Given TikTok's vast user base of over 1.4 billion users worldwide, it provides a relevant case study for evaluating cloud-based application performance under varying conditions. Table 1 presents the annual user statistics for TikTok, broken down by region from 2018 to 2021, highlighting the significant growth and regional distribution of the platform's user base. To assess the effects of the cloud-based application TikTok, the CloudAnalyst simulation framework was employed. This framework enables the evaluation of different service broker policies and LBA in a controlled environment, providing insights into their impact on application performance and user experience. Through these experiments, the study aims to identify optimal strategies for managing the load and ensuring efficient resource utilization in cloud environments, particularly for applications with massive and globally distributed user bases like TikTok.

Table 1. TikTok Annual Users by Region 2018 to 2021(million).

Year	Asia	North America	Europe	South America
2018	62	28	21	3
2019	130	55	53	10
2020	198	105	98	64
2021	313	138	158	188

To simulate in CloudAnalyst, we assumed that 1/100 of the size of TikTok's annual user statistics in 2018 were considered simultaneous online users during off-peak hours, and we also assumed that 1/100 of the size of TikTok's annual user statistics in 2021 were considered simultaneous online users during peak hours. Then, we assumed that the hardware specifications of data centers were the same.

Table 2. Userbase Configuration.

Name	Region	Peak Hours (GTM)	Avg Peak Users	Avg Off-Peak Users
UB1	3	13-15	313000	62000
UB2	0	15-17	138000	28000
UB3	2	20-22	158000	21000
UB4	1	1-3	188000	3000

Table 3. Other CloudAnalyst Parameter Values.

Parameters	Values
User Grouping Factor in Userbase	10
Request Grouping Factor	10
Executable instruction length per request	100
Simulation Duration	60 min
Image size	10000
Memory	512
Bandwidth	1000
Datacenter Architecture	X86
Datacenter processor	4
Datacenter Operating system	Linux
Datacenter virtual machine monitor	Xen

The CloudAnalyst simulation framework was used to evaluate the performance of the cloud-based application TikTok as described in the following scenario using the PSBR with three existing LBA. The scenario represents the configuration of data centers within the same region, shown in Table 4.

Table 4. Datacenter Configuration.

Datacenter	Region	No. of VMs	Cost per VM/hr	Data transfer cost/Gb
DC1	0	25	0.15	0.15
DC2	0	50	0.12	0.12
DC3	0	75	0.1	0.1
DC4	2	100	0.05	0.05
DC5	2	50	0.1	0.1
DC6	3	125	0.03	0.03
DC7	3	75	0.1	0.1

Table 5. Experimental Results for Response Time.

Service Broker Policy	Load Balancing Algorithms	Response Time of Application (milliseconds)		
		Min	Max	Avg
PSBR	Round Robin	36.5	261.2	78.5
	Active Monitoring	37.3	260.6	75.1
	Throttled	36.6	259.9	78
SPR	Round Robin	37.4	260.6	84.5
	Active Monitoring	37.4	261.2	84.2
	Throttled	37.4	260	84

Table 4 shows Region 0 is North America, Region 1 is South America, Region 2 is Europe, Region 3 is Africa, Region 4 is Asia, Region 5 is Australia, and Region 6 is Middle East. During the scenario simulation, the response and data center processing times are evaluated, and the results are recorded. The analysis of the results is shown in the Table 5 and Table 6.

According to the experimental result in Table 5, the average response time of the proposed PSBR with an active monitoring load balancing algorithm is the best. Next, the proposed PSBR with three load-balancing algorithms is better than the existing SPR in all measures of average response time.

Table 6. Experimental Results for Processing Time.

Service Broker Policy	Load Balancing Algorithms	Datacenter Processing Time (milliseconds)		
		Min	Max	Avg
PSBR	Round Robin	0.18	10.7	5.72
	Active Monitoring	0.19	7.87	5
	Throttled	0.18	6.08	4.67
SPR	Round Robin	0.31	10.85	5.79
	Active Monitoring	0.19	11.25	5.04
	Throttled	0.18	5.77	4.8

The experimental result Table 6 shows the data center processing time. In the result table, the proposed PSBR with throttled load balancing gives the best data center processing time. Then, the proposed PSBR with an active monitoring load balancing algorithm also provides a better processing time than the existing SPR.

6. Conclusions

This paper has presented the effect of the proposed service broker policy (PSBR) on the performance of cloud-based applications with different load-balancing algorithms. The existing SPR randomly selects the data center without considering response time or data center processing time. So, the new service broker policy is proposed to handle the random data center selection part of the existing SPR. Next, the average response time and data center processing time are measured. According to the evaluation results, the response time of PSBR with the active monitoring load balancing algorithm is the best of all, and the processing time of PSBR with the throttled load balancing algorithm is better than the other since its result is shorter. Finally, the proposed PSBR gives better response time and less data center processing time than the existing SPR in all conditions. This system will be experimented with additional measurements such as Throughput, Resource Utilization, and Latency in future work.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] L. Heilig, E. Lalla-Ruiz, and S. Voß, "Modeling and solving cloud service purchasing in multi-cloud environments," *Expert Syst. Appl.*, vol. 147, p. 113165, Jun. 2020, doi: 10.1016/j.eswa.2019.113165.

[2] A. Hussain, J. Chun, and M. Khan, "A novel framework towards viable Cloud Service Selection as a Service (CSSaaS) under a fuzzy environment," *Futur. Gener. Comput. Syst.*, vol. 104, pp. 74–91, Mar. 2020, doi: 10.1016/j.future.2019.09.043.

[3] X. Wang, J. Cao, D. Yang, Z. Qin, and R. Buyya, "Online cloud resource prediction via scalable window waveform sampling on classified workloads," *Futur. Gener. Comput. Syst.*, vol. 117, pp. 338–358, Apr. 2021, doi: 10.1016/j.future.2020.12.005.

[4] R. Achar, P. S. Thilagam, and S. Acharya, "Broker-based mechanism for cloud provider selection," *Int. J. Comput. Sci. Eng.*, vol. 22, no. 1, p. 50, 2020, doi: 10.1504/IJCSE.2020.107247.

[5] S. S. Chauhan, E. S. Pilli, and R. C. Joshi, "BSS: a brokering model for service selection using integrated weighting approach in cloud environment," *J. Cloud Comput.*, vol. 10, no. 1, p. 26, Dec. 2021, doi: 10.1186/s13677-021-00239-5.

[6] D. R. and S. R., "Cloud providers ranking and selection using quantitative and qualitative approach," *Comput. Commun.*, vol. 154, pp. 370–379, Mar. 2020, doi: 10.1016/j.comcom.2020.02.028.

[7] M. Hosseini Shirvani, "Bi-objective web service composition problem in multi-cloud environment: a bi-objective time-varying particle swarm optimisation algorithm," *J. Exp. Theor. Artif. Intell.*, vol. 33, no. 2, pp. 179–202, Mar. 2021, doi: 10.1080/0952813X.2020.1725652.
[8] P. S. Anjana, R. Wankar, and C. R. Rao, "Design of a Cloud Brokerage Architecture Using Fuzzy Rough Set Technique," in *Multi-disciplinary Trends in Artificial Intelligence*, 2017, pp. 54–68. doi: 10.1007/978-3-319-69456-6_5.
[9] Backlinko Team, "TikTok Statistics You Need to Know," *Backlinko*, 2024. https://backlinko.com/tiktok-users
[10] A. Jyoti, R. K. Pathak, and P. Singh, "Service Broker Algorithm for Datacenter Selection with light and heavy load in Cloud Computing," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 4, pp. 6747–6751, Aug. 2020, doi: 10.30534/ijatcse/2020/372942020.
[11] B. Mukhopadhyay, R. Bose, and S. Roy, "A Novel Approach to Load Balancing and Cloud Computing Security using SSL in IaaS Environment," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 2362–2370, Apr. 2020, doi: 10.30534/ijatcse/2020/221922020.
[12] J. Bisht and V. V. Subrahmanyam, "Improvising service broker policies in fog integrated cloud environment," *Int. J. Commun. Networks Distrib. Syst.*, vol. 28, no. 5, p. 534, 2022, doi: 10.1504/IJCNDS.2022.125360.
[13] M. A. Khan, "Optimized hybrid service brokering for multi-cloud architectures," *J. Supercomput.*, vol. 76, no. 1, pp. 666–687, Jan. 2020, doi: 10.1007/s11227-019-03048-5.
[14] N. K. Mishra, P. Himthani, and G. P. Dubey, "Priority-Based Shortest Job First Broker Policy for Cloud Computing Environments," in *Proceedings of International Conference on Communication and Computational Technologies*, 2021, pp. 279–290. doi: 10.1007/978-981-16-3246-4_23.
[15] M. Radi, A. Alwan, A. Abualkishik, A. Marks, and Y. Gulzar, "Efficient and Cost-effective Service Broker Policy Based on User Priority in VIKOR for Cloud Computing," *Basic Appl. Sci. - Sci. J. King Faisal Univ.*, pp. 1–8, 2021, doi: 10.37575/b/cmp/210032.
[16] E. Rafieyan, R. Khorsand, and M. Ramezanpour, "An adaptive scheduling approach based on integrated best-worst and VIKOR for cloud computing," *Comput. Ind. Eng.*, vol. 140, p. 106272, Feb. 2020, doi: 10.1016/j.cie.2020.106272.
[17] S. Al-E'mari, Y. Sanjalawe, A. Al-Daraiseh, M. B. Taha, and M. Aladaileh, "Cloud Datacenter Selection Using Service Broker Policies: A Survey," *Comput. Model. Eng. Sci.*, vol. 139, no. 1, pp. 1–41, 2024, doi: 10.32604/cmes.2023.043627.
[18] Y. Sanjalawe, M. Anbar, S. Al-E'mari, R. Abdullah, I. Hasbullah, and M. Aladaileh, "Cloud Data Center Selection Using a Modified Differential Evolution," *Comput. Mater. Contin.*, vol. 69, no. 3, pp. 3179–3204, 2021, doi: 10.32604/cmc.2021.018546.
[19] Y. Chen, T. Shi, H. Ma, and G. Chen, "Automatically Design Heuristics for Multi-Objective Location-Aware Service Brokering in Multi-Cloud," in *2022 IEEE International Conference on Services Computing (SCC)*, Jul. 2022, pp. 206–214. doi: 10.1109/SCC55611.2022.00039.
[20] M. Liu, L. Pan, and S. Liu, "Cost Optimization for Cloud Storage from User Perspectives: Recent Advances, Taxonomy, and Survey," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–37, Dec. 2023, doi: 10.1145/3582883.
[21] X. Li, L. Pan, and S. Liu, "A survey of resource provisioning problem in cloud brokers," *J. Netw. Comput. Appl.*, vol. 203, p. 103384, Jul. 2022, doi: 10.1016/j.jnca.2022.103384.