The Scientific Journal of King Faisal University

Basic and Applied Sciences

Efficient and Cost-effective Service Broker Policy Based on User Priority in VIKOR for Cloud Computing

Mohammed Radi [1], Ali A. Alwan [2], Abedallah Zaid Abualkishik [3], Adam Marks [4], Yonis Gulzar [5]

[1] Department of Computer and Information Sciences, Faculty of Computers and Information Technology, Al-Aqsa University, Gaza, Palestine
[2] School of Theoretical and Applied Science, Ramapo College of New Jersey, New Jersey, United States of America
[3] College of Computer Information Technology, American University, Dubai, United Arab Emirates
[4] Higher Colleges of Technology, Dubai, United Arab Emirates
[5] Department of Management Information Systems, College of Business Administration, King Faisal University, Al Ahsa, Saudi Arabia

LINK	RECEIVED	ACCEPTED	PUBLISHED ONLINE	ASSIGNED TO AN ISSUE
https://doi.org/10.37575/b/cmp/210032	29/04/2021	12/12/2021	12/12/2021	01/12/2021
NO. OF WORDS	**NO. OF PAGES**	**YEAR**	**VOLUME**	**ISSUE**
8756	8	2021	22	2

ABSTRACT

Cloud computing has become a practical solution for processing big data. Cloud service providers have heterogeneous resources and offer a wide range of services with various processing capabilities. Typically, cloud users set preferences when working on a cloud platform. Some users tend to prefer the cheapest services for the given tasks, whereas other users prefer solutions that ensure the shortest response time or seek solutions that produce services ensuring an acceptable response time at a reasonable cost. The main responsibility of the cloud service broker is identifying the best data centre to be used for processing user requests. Therefore, to maintain a high level of quality of service, it is necessity to develop a service broker policy that is capable of selecting the best data centre, taking into consideration user preferences (e.g. cost, response time). This paper proposes an efficient and cost-effective plan for a service broker policy in a cloud environment based on the concept of VIKOR. The proposed solution relies on a multi-criteria decision-making technique aimed at generating an optimized solution that incorporates user preferences. The simulation results show that the proposed policy outperforms most recent policies designed for the cloud environment in many aspects, including processing time, response time, and processing cost.

KEYWORDS

Cloud computing, data centre selection, service broker, VIKOR, user priorities

CITATION

Radi, M., Alwan, A.A., Abualkishik, A.Z., Marks, A. and Gulzar, Y. (2021). Efficient and cost-effective service broker policy based on user priority in VIKOR for cloud computing. *The Scientific Journal of King Faisal University: Basic and Applied Sciences*, **22**(2), 1–8. DOI: 10.37575/b/cmp/210032

1. Introduction

Nowadays, the amount of digital data is increasing dramatically, and it is expected to continue growing by more than 25% in the USA and more than 30% in Western Europe in one year. This indicates that the size of digital data will double every three years (Gantz and Reinsel, 2012). Big data is defined as a collection of a huge amount of data with a great variety of types, generated based on velocity. Various fields are involved in big data contexts, such as economics, social networks, e-Science, scientific disciplines, and web applications. It has been proven that adopting traditional data processing platforms to process big data is inadequate and undesirable for many reasons. This is due to the fact that big data has unique characteristics that lead to many difficulties when processing data using these traditional platforms (Chen and Zhang, 2014). Cloud computing has become a practical solution for big data applications. It has been argued that soon, around 40% of digital data will be hosted or processed by cloud computing (Gantz and Reinsel, 2012). Considering the significant growth in the volume, variety and velocity of data, cloud storage is a successful platform for servicing big data (Chen and Zhang, 2014).

Typically, cloud storage systems comprise several data centres (DCs) (Amazon S3, 2018; Google Cloud Storage, 2018; Windows Azure, 2018; IBM Cloud, 2018) connected through a network (Wu, 2016). In other words, DCs often contain many heterogeneous machines distributed around the world. The heterogeneity is derived from the different capabilities, varying communication channel specifications and diverse loads of DCs. The processing cost for each DC varies and is determined based on certain factors within the context of each particular DC. Among the factors that influence the processing cost of each DC are the type of job the client offers and the time at which the client's job was submitted for processing. Another factor that impacts the processing cost is the user preferences specified when the job is submitted to the cloud broker. Certain users might prefer a plan that ensures the minimum cost for running a job, whereas other users might prefer a plan that accomplishes their request and guarantees the shortest response time or a plan that fulfils their request by balancing the processing cost and the response time. Thus, they are seeking a plan that offers running the given job at an affordable cost while maintaining an acceptable response time. Therefore, the main task of a service broker in the cloud computing paradigm is to identify and select the DC that offers the best plan in terms of improving the response time and minimizing the cost when carrying out users' jobs (Benlalia *et al.*, 2019; Khan, 2020; Manasrah and Gupta, 2019; Youssef, 2020).

Most of the service broker policies introduced in the literature focus on improving limited aspects, such as cost, response or time when running a user's job. However, other aspects, such as user preferences, are also important and should be reflected in the proposed plan for running the jobs. Nevertheless, it would be very challenging to identify and design an ideal service broker policy that fulfils both requirements, namely, minimum response time and minimum cost. From the literature, we can conclude that three groups of researchers work on service broker policies in cloud computing. The first group concentrates on designing service broker policies to minimize the response time of the user's job (Mehdi *et al.*, 2012; Radi, 2014; Sharma, 2014; Trabay *et al.*, 2021). The second group aims to produce service broker policies to reduce the processing cost when running users' jobs (Chudasama *et al.*, 2012; Rekha and Dakshayini, 2018; Sun *et al.*, 2019). The third group focuses on developing and incorporating service broker policies that offer a trade-off between the response time and the processing cost when processing users' jobs (Khan, 2020; Kofahi *et al.*, 2019; Manasrah *et al.*, 2017; Manasrah and Gupta, 2019; Mehdi *et al.*, 2012; Mehdi *et al.*, 2011; Subramanian and Savarimuthu, 2016). The work presented by Arya and Dave (2017) introduces a new service broker policy for the fog computing environment that identifies the best plan for DC selection,

Corresponding Author: Ali A. Alwan 0060173546110, aaljuboo@ramapo.edu

taking into account desirable user preferences while maintaining a reasonable cost without compromising the performance of running users' jobs (Chauhan *et al.*, 2018).

From the reviewed literature, we observe that a limited number of previous works addressed the issue of incorporating the user's preferences when running the jobs (Al-Tarawneh and Al-Mous, 2019; Arya and Dave, 2017; Chauhan *et al.*, 2018; Manasrah and Gupta, 2019; Zakaria *et al.*, 2019). We argue that user preferences are an essential factor that should be considered when designing a service broker policy. This is due to the fact that the user preferences reflect the degree of satisfaction of the user, who is looking for a service with a high level of quality of service (QoS), which depends mainly on the user's preferences (Arya and Dave, 2017). Since the resources and services for cloud users is based are provided on a pay-per-use basis, it is important to take into consideration the user's priorities when selecting and assigning a DC to meet the user's requirement. It has been argued that many factors could optimize services in cloud computing. Therefore, dynamic policies are better suited to the ever-changing nature of cloud computing (Kofahi *et al.*, 2019).

From the literature, we observe that certain specific factors have been used to determine the best DC. These factors are cost, DC capacity, current load, communication channel specifications, and user requirements. We also noticed that most of the existing policies in the literature have considered a very limited number of these factors. It is essential to design a service broker policy that is capable of negotiating between cost and performance (response time and processing time) based on user priorities, also taking into account the most critical factors affecting the services in cloud paradigms, such as cost, DC capacity, current load, and communication channel specifications. This problem can be formulated as a multi-criteria decision-making (MCDM) problem.

This paper proposes an efficient and cost-effective service broker policy for DC selection in a heterogeneous cloud environment using VIKOR. The proposed policy relies on the idea of exploiting user priorities when assigning the service to the designated DC. The proposed service broker policy strategy takes into consideration the response time and the overall cost to optimize users' specified priorities. The proposed service broker policy has been developed using a cloud analyst simulator (Limbani and Oza, 2012) to evaluate its performance and efficiency. The experimental result demonstrates that our service broker policy solution outperforms the previous approaches in terms of the total cost, response time and DC processing time for different scenarios (Al-Tarawneh and Al-Mous, 2019; Arya and Dave, 2017; Manasrah and Gupta, 2019).

The remainder of the paper is organized as follows. Section 2 presents a detailed discussion of the previous works related to cloud service broker policies for DC selection. Section 3 explains the detailed steps of the proposed service broker policy based on users' specified priorities. Section 4 describes the experiment setting and the experimental results of the proposed approach compared to the most recent existing service broker policy approaches. Finally, Section 5 concludes the paper and outlines some potential future work directions.

2. Related Work

The main concerns, from the user's perspective, when selecting the best DC are the response time and the cost. It has been reported that producing the best plan to select a DC that best serves the user, with the shortest response time and the cheapest cost, is a challenging process (Al-Tarawneh and Al-Mous, 2019). In cloud computing, a large number of researchers have focused on developing service broker policies for optimum DC selection. One of the most famous static service broker routing policies is the Proximity Service Broker policy, which routes the user's job to the closest DC. If many DCs have the same network delay, the algorithm randomly selects one of them (Limbani and Oza, 2012). Nevertheless, the proposed algorithm in Limbani and Oza (2012) ignored many critical factors, such as DC specification, cost, and DC overhead. Many research studies have attempted to further enhance the Proximity Service Broker policy by avoiding the random selection of a DC when many have the same delay. The work described in Chudasama *et al.* (2012) and Mishra *et al.* (2014) considers the cost of the DC and routes the job to the DC with lower cost if more than one DC has the same network delay. Additionally, the work presented in Al Sukhni (2016), Kapgate (2014) and Radi (2015) attempts to improve the performance by avoiding random selection. These proposed policies consider DC specification and use a round-robin policy with weights based on DC specifications to route the user jobs. The work presented in Rafieyan *et al.* (2020) modifies the randomization in the Proximity Service Broker policy. It attempts to select the DC via minimum distance based on the k-nearest neighbour. However, static policies do not consider the present state of the network or the status of the DC, which could increase the job response time.

The work proposed in Nandwani *et al.* (2016) aims to improve DC selection by giving specific weights to each DC depending on the number of virtual machines (VMs) and selecting the DC in a circular manner based on the weights. However, performance-aware static strategies lead to increased service cost, while cost-aware strategies increase the processing time. Moreover, none of the previous static strategies considered the issue of the dynamic changes in the cloud environment. Most importantly, none of these previous approaches considered the anticipation of the user preferences in designing a broker policy for DC selection in a heterogeneous cloud environment. Conversely, other researchers concentrate on dynamically evaluating the resources and the incoming jobs to reduce response and processing time by considering bandwidth, latency and the size of the job to route the job to a DC in the minimum time to transfer the job and the minimum expected processing time (Manasrah *et al.*, 2017). However, the work proposed by Manasrah *et al.* (2017) does not take into consideration the DC cost and user preferences. The work in Benlalia *et al.* (2019) suggests using the ratio of efficiency that depends on a set of efficiency parameters over the cost of the VM and threshold value to select the best DC with the lowest ratio and less than the threshold value. However, the idea of their approach relies on defining the threshold manually, which has a negative impact on the performance. Furthermore, their work does not consider the user preference, and the work lacks the experimental result to justify the effectiveness of the proposed solution. More recently, a dynamic service broker policy improved the DC selection process by using the concept of test jobs to evaluate the DCs and then used a vector space model and a multi-objective optimization technique to dynamically select the best DC (Kofahi *et al.*, 2019). Taking into account static and dynamic parameters, a normalization-based hybrid service broker (NHSB) approach is proposed by Khan (2020). The NHSB approach considers several factors, such as the number of VMs, VM image size, VM memory, VM bandwidth, cost per VM/s, cost per VM memory, storage cost, bandwidth cost/GB, total memory, total storage, machine bandwidth and total processor speed as static parameters. Moreover, it considers a set of dynamic parameters, such as request load, network delay, and last recorded processing time. The NHSB approach computes the normalized values of those static and dynamic parameters and then selects the DC with the minimum sum of normalized values for distributing load among the DCs. Additionally, a heuristic service broker policy approach is proposed by Rekha and Dakshayini (2018). The proposed approach aims to achieve an acceptable response time with a minimum processing time and total cost. The approach is capable of selecting the DC based on the lowest network delay, expected processing time, and minimum cost. However, the idea of the work presented in Rekha and Dakshayini (2018) did not consider the user preferences. Additionally, the work proposed by Manasrah and Gupta (2019) attempts to trade-off the expected cost and the performance of the selected DC by introducing an optimized service broker policy using a differential evolution algorithm based on a set of parameters to select the DC where the response, processing time and total cost are optimized.

In this regard, various service broker algorithms that have been proposed in the literature incorporate user preference in their policies. The work proposed in Arya and Dave (2017) considers the users' priorities and the current load on the DCs to select the optimal DC in the fog computing environment. The idea of the proposed policy relies on calculating a value γ, based on users' priorities and DC characteristics, and selecting the DC with the highest value of γ for each. Moreover, the value of γ is dynamically changed when aiming to improve load balancing and reflecting the current latency of the DC. Similarly, the work in Al-Tarawneh and Al-Mous (2019) proposes an adaptive fuzzy-based cloud service broker (AFBSB) algorithm. The idea of their work relies on selecting the DC based on user cost and performance preference, request processing requirements, and currently available bandwidth. The algorithm focuses on user preference and does not balance cost and performance. Moreover, the proposed algorithm considers the heterogeneity of the DC and some important specifications of the DC, such as the number of processors and each processor's speed. The proposed work presented in this paper also considers the number of processors and the processor speed as crucial factors in selecting the best DC. Nevertheless, our work differs from that of Al-Tarawneh and Al-Mous (2019) in that it uses a multi-objective optimization approach to balance user cost and performance.

The main objective of a service broker policy is to balance cost and performance (response time and processing time) based on user priorities, also taking into account the most critical factors affecting the services in cloud paradigms, such as cost, DC capacity, current load, and communication channel specifications. However, the service broker policy process has multiple conflicting criteria. MCDM methods can be used to evaluate conflicting criteria to find the best solution. Recently, MCDM methods have been utilized in cloud computing to evaluate cloud services. For example, Patiniotakis *et al.* (2015) used the fuzzy analytical hierarchical process (AHP) method for ranking cloud services. Additionally, TOPSIS has been employed to compute the trust value of a cloud service provider (Sidhu and Singh, 2017), and TOPSIS with a triangular fuzzy number was employed to rank cloud services in Kumar *et al.* (2018). The VIKOR technique falls under the MCDM approach, which has been used by many researchers over various applications, and it is preferred due to its characteristic (Alabool *et al.*, 2013). Chauhan *et al.* (2018) and Otay and Yıldız (2021) utilized VIKOR methods to find ranks of given service alternatives within given QoS constraints.

Most of the service broker policies introduced in the previous works focused on optimizing certain parameters while ignoring other critical factors, such as specified user priorities. Unfortunately, an optimal service broker policy that meets both minimum response time and minimum cost has not yet been found. As mentioned previously, the objective of some users is to minimize the response time, while other users are interested in minimizing the total cost. It has been found that most of the previous works did not consider users' priorities. Since cloud providers offer resources to users on a pay-per-use basis, it is important to consider the user's priorities when selecting the DC to process the user's request. Since in cloud computing there are many factors to be considered to optimize the process of running the user's task, the dynamic policies are more suitable for the continuously changing nature of cloud computing (Kofahi *et al.*, 2019). We have also noticed that a set of factors are used to select the best DC, such as cost, DC capability, current load, communication channel specification, and users' requirements. This paper aims to propose an efficient service broker policy that can trade-off between cost and performance (response time and processing time) based on user priorities considering the most important factors, such as cost, the DC's capability, current load and communication channel specifications.

3. The Proposed Service Broker Policy

This section presents and discusses the proposed service broker policy. The proposed approach has four components: Cloud Service Broker, Modelling Service Brokering Problems, Service Broker Policy Based on VIKOR, and VIKOR-based Service Broker Algorithm. These components are further elaborated in the following subsections.

3.1. Cloud Service Broker (CSB)

This component is responsible for identifying and determining the most suitable DCs that are located in different regions around the world to execute the requests submitted by clients. The process flow of this component works as follows. First, the user submits the request, based on the CSB, aiming to collect the current metadata of the factors for all DCs. Next, the brokering algorithm attempts to identify the best DC based on the collected information and the user's specified priorities. This user's request will be routed to the designated DC for execution. Finally, the cloud service broker attempts to send back the reply to the user who is requesting the service. Figure 1 illustrates the detailed process of the service broker policy component.

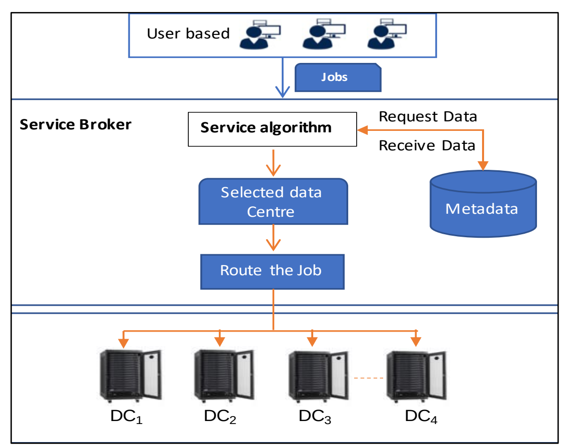


Figure 1: Service Broker Policy

3.2. Modelling Service Brokering Problem

The main function of this component is modelling the service brokering problem as a decision problem. It aims to determine the best DC among the alternative heterogeneous DCs that could be selected to run the given task, considering response time, cost or balance. Then, the alternatives, represented here by a set of DCs, leave various options open to the users to be considered in the decision. The criteria of a decision problem are a set of factors affecting the selection process. The alternative cloud DCs will be evaluated by comparing the factors (criteria) to measure their potential fit in the problem. In a heterogeneous cloud system, there is a collection of n DCs defined as a set: DC = $\{DC_1, DC_2, DC_3,..., DC_n\}$. Each DC is characterized by a set of criteria: Cost, DC Specification, Current Load, and Communication Channel Specifications. A detailed explanation of these criteria is given below. Table 1 describes the criteria that have been adopted for the service broker decision problem.

Table 1: The criteria of a decision problem

Criteria	Symbol	Parameters	Aim
Cost	DCC	Data transfer cost and processing costs	Minimize
DC Specification	DCS	Processor speed and number of processors	Maximize
Current Workload	DCCL	Number and size of loaded jobs	Minimize
Communication Channel Specification	DCCS	Transmission delay time	Minimize

These criteria are defined and computed as outlined below.

- **Cost:** Each DC has a different cost, which comprises the data transfer cost and the processing cost. The essential task of the service broker algorithm is to determine the DC that will introduce the lowest cost that accomplishes the user's request. The formula given in equation 1 describes the computation of the cost of the DC. The cost of the DC is computed by calculating the cost per VM in one hour's time and the cost of data transfer in GB.
  DCC = cost per VM $/Hr + data Transfer cost $/GB (1)
- **DC Specification:** DCs in the cloud have different hardware specifications, such as different processor speeds and a varying number of processors. Therefore, the proposed service broker algorithm attempts to determine the best DC that has the highest value of DC specification. Equation 2 represents the formula for computing the value of the DC specification for each active DC in the cloud.
  DCS = number of processors $\times$ processor speed (2)
- **Current Workload:** The third criterion that has been taken into consideration for selecting a DC is the current workload. During the run time, each DC is loaded with a varying number of user requests, and the current workload is dynamically computed based on the service broker algorithm for each DC. The service broker algorithm aims to select the DC that produces the lowest workload.
- **Communication Channel Specification:** The last criterion considered in this component is the transmission delay between the user region and the DC region. The proposed service broker algorithm needs to compute the transmission delay time for each DC to determine the best DC. The algorithm chooses the DC that introduces the lowest transmission delay. The formula for the transmission delay time is given in equation 3.
  DCD = Delay_Matrix(User region, DC_Region_i) (3)

Since the decision matrix contains DCC, DCS, DCCL and DCD, which have different measurement units, computational problems can occur. Therefore, it is necessary to compute normalized values from the original value to perform attribute comparison. We use linear normalization, which scales the original value to be between [0, 1]. In linear normalization, the normalized values nor_{ij} of each attribute, *xij*, are calculated based on the formula given in equation 4.

$$nor_{ij} = \frac{x_{ij}}{\sum_1^m x_{ij}^2} \qquad \text{where } i = 1 \dots m, \;\; j = \cdots n \tag{4}$$

The values of DCS, DCC, DCCL and DCD are normalized based on equations 5, 6, 7, and 8, respectively.

$$norDCS_{ij} = \frac{DCS_{ij}}{\sum_1^m DCS_{ij}^2} \tag{5}$$

$$norDCC_{ij} = \frac{DCC_{ij}}{\sum_1^m DCC_{ij}^2} \tag{6}$$

$$norDCCL_{ij} = \frac{DCCL_{ij}}{\sum_1^m DCCL_{ij}^2} \tag{7}$$

$$norDCD_{ij} = \frac{DCD_{ij}}{\sum_1^m DCD_{ij}^2} \tag{8}$$

Finally, it is clear that the model service brokering problem favours a DC that introduces minimum cost and communication delay while maintaining the highest capability of workload to produce a fast response time. Therefore, it is necessary to evaluate the *n* DCs with more than one criterion to select the DC. This type of problem is called a multiple attribute decision-making (MADM) problem (Wickremasinghe *et al.*, 2010). This paper implements the VIKOR method to design a service broker policy, which is explained in the next subsection.

3.3. VIKOR-based Service Broker Policy

This section presents the proposed VIKOR-based technique to resolve the issue of multi-criteria optimization in complex systems. The idea of the proposed technique relies on employing a set of conflicting criteria to characterize, rank and select the best DC from the set of alternative DCs in the cloud. The VIKOR strategy produces a ranking index based on the measure of closeness to the ideal solution (Wickremasinghe *et al.*, 2010). There are *n* alternative DCs (DC_1, DC_2, DC_3,..., DC_n), and each DC is characterized by *m* criteria, in which *m* comprises up to four criteria: DCC, DCS, DCCL, and DCD. The preferred ratings of each DC (alternatives) in each criterion are described in Table 2.

Table 2: The preferred ratings of the data centres

Data Centre	Cost: DCC	DC Specification DCS	Current Workload DCCL	Communication Channel Specification DCD
	Weight $= W_{DCC}$	Weight $= W_{DCS}$	Weight $= W_{DCCL}$	Weight $= W_{DCD}$
DC_1	DCC_1	DCS_1	$DCCL_1$	DCD_1
DC_2	DCC_2	DCS_2	$DCCL_2$	DCD_2
DC_3	DCC_3	DCS_3	$DCCL_3$	DCD_3
DC_4	DCC_4	DCS_4	$DCCL_4$	DCD_4
...	...	...	...	...
DC_n	DCC_n	DCS_5	$DCCL_5$	DCDn

Since all MADM-based methods assume that every criterion should have a predefined weight, the proposed VIKOR-based service broker policy assigns a predefined weight value for all the criteria. A subjective method has been incorporated to determine the weights for the considered criteria. The weight value for the criteria will be set by the client to reflect the preferred priorities, which could be one of the following: cost minimization, shortest response time, or balance between cost minimization and shortest response time. The proposed service broker policy sets the weight for each criterion based on the formula given in equation 9.

$$\sum_{j=1}^{n} W_j = 1 \tag{9}$$

where W_j is the weight of the criterion j.

The detailed steps of the adopted VIKOR-based service broker algorithm to solve the service broker issue are shown in Figure 2 (Wickremasinghe *et al.*, 2010). The input of the algorithm comprises the details of the user-based (UB) properties and the weight details of DCC, DCS, DCCL and DCD for the available DCs, while the output of the algorithm identifies the most appropriate DC to be selected for the user's task. The algorithm starts by computing the given criteria pertaining to DCC, DCS, DCCL and DCD for each involved DC, using equations 1, 2, 3 and 4 in Section 3.2 (steps 1–5). Next, the values of these criteria for each DC are normalized based on equations 5, 6, 7 and 8 in Section 3.2 (steps 6–10). This is followed by determining the values of the best f_j^* and the worst f_j^- functions of all the criteria, *j* = 1, 2,..., *m*, as depicted in steps 11–19. It should be noted that if the *j*th function represents a benefit, then $f_j^* = \max_k f_{jk}$ and $f_j^- = \min_k f_{jk}$. If the *j*th function represents a cost, then $f_j^* = \min_k f_{jk}$ and $f_j^- = \max_k f_{jk}$. Since DCC, DCCL and DCD represent a cost function, then f_j^* *and* f_j^- are computed as shown in the algorithm:

- $f_1^* = \mathrm{DCC}^* = \min\,(\mathrm{DCC}_1, \mathrm{DCC}_2, \mathrm{DCC}_3 \dots\dots \mathrm{DCC}_n)$
- $f_3^* = \mathrm{DCCL}^* = \min\,(\mathrm{DCCL}_1, \mathrm{DCCL}_2, \mathrm{DCCL}_3 \dots\dots \mathrm{DCCL}_n)$
- $f_4^* \mathrm{DCD}^* = \min\,(\mathrm{DCCL}_1, \mathrm{DCCL}_2, \mathrm{DCCL}_3 \dots\dots \mathrm{DCCL}_n)$
- $f_1^- = \mathrm{DCC}^- = \max\,(\mathrm{DCC}_1, \mathrm{DCC}_2, \mathrm{DCC}_3 \dots\dots \mathrm{DCC}_n)$
- $f_3^- = \mathrm{DCCL}^- = \max\,(\mathrm{DCCL}_1, \mathrm{DCCL}_2, \mathrm{DCCL}_3 \dots\dots \mathrm{DCCL}_n)$
- $f_4^- = \mathrm{DCD}^- = \max(\mathrm{DCCL}_1, \mathrm{DCCL}_2, \mathrm{DCCL}_3 \dots\dots \mathrm{DCCL}_n)$.

Furthermore, the maximum and minimum values f_j^* *and* f_j^- of the function for DCS are computed as follows:

- $f_2^* = \mathrm{DCS}^* = \max(\mathrm{DCS}_1, \mathrm{DCS}_2, \mathrm{DCS}_3 \dots\dots \mathrm{DCS}_n)$
- $f_2^- = \mathrm{DCS}^- = \min(\mathrm{DCS}_1, \mathrm{DCS}_2, \mathrm{DCS}_3 \dots\dots \mathrm{DCS}_n)$

Steps 20–22 demonstrate the details of computing the values of S_k and R_k, k = 1, 2,..., n, using the formulas in equations 9 and 10, respectively:

$$S_k = \sum_{j=1}^{m} W_j \left|f_j^* - f_{jk}\right| / \left|f_j^* - f_j^-\right| \tag{10}$$

$$R_k = \max_j \{W_j \left|f_j^* - f_{jk}\right| / \left|f_j^* - f_j^-\right|, j = \; 1{,}2\;3 \dots m\} \tag{11}$$

where W_j is the weight of the j^{th} criterion, S_k represents the utility measure and R_k represents the regret measure. The value of Q_k, k = 1, 2,...,n, is computed utilizing the formula in equation 11, as described in steps 23–24:

$$Q_k = \frac{v(S_k - S^*)}{S^- - S^*} + \frac{(1-v)(R_k - R^*)}{R^- - R^*}, K = 1,2,3 \dots n \ alternatives \quad (12)$$

where $S^* = \min_j S_k$, $S^- = \max_j S_k$, $R^* = \min_j R_k$, $R^- = \max_j R_k$, and ν is the weight for the strategy of the 'majority of criteria', while 1 – ν is the weight of the individual regret. In step 25, the values of S, R and Q are ranked in ascending order, producing three ranking lists. Finally, a compromise solution is produced and returned (step 26).

Input: requested user base, W_{DCC}, W_{DCS}, W_{DCCL}, W_{DCD}

Output: Target data centre name

1. **For each data centre i**
2. DCC_i = cost per VM \$/Hr + data transfer cost \$/GB
3. DCS_i = number of processors × processor speed
4. DCD_i = Delay_Matrix(User region, DC_Region_i)
5. $DCCL_i$ = number of current requests
6. **For each data centre i**
7. $norDCS_{ij} = \frac{DCS_{ij}}{\sum_1^m DCS_{ij}^2}$
8. $norDCC_{ij} = \frac{DCC_{ij}}{\sum_1^m DCC_{ij}^2}$
9. $norDCCL_{ij} = \frac{DCCL_{ij}}{\sum_1^m DCCL_{ij}^2}$
10. $norDCD_{ij} = \frac{DCD_{ij}}{\sum_1^m DCD_{ij}^2}$
11. Determine the best $\boldsymbol{f_j^*}$ and the worst $\boldsymbol{f_j^-}$
12. $f_1^* = DCC^* = \min(norDCC_1, norDCC_2, norDCC_3 \dots norDCC_n)$
13. $f_2^* = DCS^* = \max(norDCS_1, norDCS_2, norDCS_3 \dots norDCS_n)$
14. $f_3^* = DCCL^* = \min(norDCCL_1, norDCCL_2, norDCCL_3 \dots norDCCL_n)$
15. $f_4^* DCD^* = \min(norDCCL_1, norDCCL_2, norDCCL_3 \dots norDCCL_n)$
16. $f_1^- = DCC^- = max\ (norDCC_1, norDCC_2, norDCC_3 \dots norDCC_n)$
17. $f_2^- = DCS^- = min(norDCS_1, norDCS_2, norDCS_3 \dots norDCS_n)$
18. $f_3^- = DCCL^- = max\ (norDCCL_1, norDCCL_2, norDCCL_3 \dots norDCCL_n)$
19. $f_4^- = DCD^- = max(norDCCL_1, norDCCL_2, norDCCL_3 \dots norDCCL_n)$
20. Compute the values S_k and R_k, k = 1, 2,..., n,
21. $S_k = \sum_{j=1}^{m} W_j |f_j^* - f_{jk}| / |f_j^* - f_j^-|$
22. $R_k = \max_j \{W_j |f_j^* - f_{jk}| / |f_j^* - f_j^-|, j = 1,2\ 3 \dots m\}$
23. Compute the value Q_k, k = 1,2,...,n,
24. $Q_k = \frac{v(S_k - S^*)}{S^- - S^*} + \frac{(1-v)(R_k - R^*)}{R^- - R^*}$
25. Sorting the values S, R and Q in ascending order
26. Return the first data centre in the Q-sorted list

Figure 2: VIKOR-based service broker algorithm

4. Results and Discussion

4.1. The Experimental Settings

To evaluate the performance and prove the efficiency of our proposed solution, the VIKOR-based service broker policy, in generating an optimized solution that incorporates user preferences when ranking and selecting the best DC from the set of alternative DCs in the cloud, several extensive experiments were designed. These experiments were conducted on an Intel Core i7 3.6GHz processor with 32GB of RAM on a Windows 8 Professional operating system. The proposed service broker policy was applied and tested using the Cloud Analyst simulator and then compared to the performance of two well-known broker policies, namely, the closest DC policy and the optimized routing policy. The comparison was based on three crucial parameters: total cost, total response time, and processing time. The comparison included five different cases, which were as follows:

- The first case used the closest DC.
- The second case used the optimal response time.
- The third case used the proposed approach with the high user priority to reduce total cost.
- The fourth case used the proposed approach with the high user priority to reduce response time.
- The fifth case used the proposed approach to balance the response time and total cost.

For the simulation, there were six heterogeneous DCs and three user bases; the configurations for the DCs, user bases and other simulator parameters are outlined in Tables 3(a), 3(b), and 3(c), respectively. In addition, the default parameters were set for internet characteristics, and the evaluation process was in different configuration scenarios. In all the scenarios, the duration of the simulation was set as one day. The locations of the DC and the user base varied in each case. Below are the details of the three scenarios considered in this work.

- **First scenario:** Six heterogeneous DCs were located in the same region, while three UB DCs were distributed over three different regions.
- **Second scenario:** Six heterogeneous DCs were located in different regions, while three UB DCs were located in the same region.
- **Third scenario:** Six heterogeneous DCs and six UB DCs were distributed across all regions.

Table 3: The parameter settings of the simulation

(a) Data centre configurations

DCs	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units	Memory (MB)	Storage (MB)	Available BW	Number of Processors	Processor Speed
DC1	1.6	0.05	0.1	0.2	2	512	100,000,000	1,000	3	500-1000
DC2	2.4	0.05	0.1	0.7	1	512	100,000,000	1,000	4	1000
DC3	5	0.05	0.1	3	3	512	100,000,000	1,000	4	10000
DC4	0.1	0.05	0.1	0.1	1	512	100,000,000	1,000	3	100
DC5	0.24	0.05	0.1	0.11	1	512	100,000,000	1,000	3	200
DC6	0.13	0.05	0.1	0.2	1	512	100,000,000	1,000	3	2000

(b) User-based data centre properties

UB	User Requests/Hour	Request Size (KB)	Start of UB's Peak Hours GMT	End of Peak Hours GMT	Avg Peak Users	Avg Off-Peak Users
UB1	120	1000	3	9	5000	500
UB2	60	100	3	9	1000	100
UB3	60	100	3	9	1500	150

(c) Other parameter settings

Parameters	Value
User grouping factor	1000
Request grouping factor	50
Request size (bytes)	100
Load balancing policy	Throttled

4.2. The Experimental Results

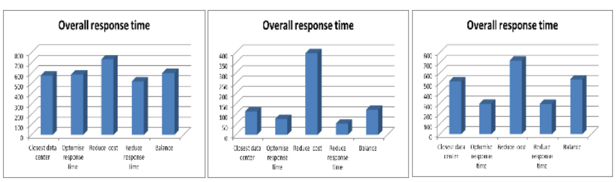
This section presents the experimental results of the VIKOR-based service broker policy solution for DC selection in a heterogeneous cloud environment, in which the proposed policy identifies the most appropriate DC to handle the user-specified service based on the user's priorities. In the experiment, three crucial performance metrics involving various scenarios were considered, namely the response time, the overall cost and the DC processing, to measure and evaluate the performance and the efficiency of the proposed solution.

4.2.1. The Overall Response Time

In this section, we present the experimental results of both our proposed solution, the VIKOR-based service broker policy, and the previous approaches for the three scenarios considered in this paper concerning the overall response time. This set of experiments aimed to investigate the impact of incorporating user preferences (priorities) on the overall response time of the process of identifying and selecting the most appropriate DC among the available set of DCs to execute user tasks. Figures 3(a), 3(b), and 3(c) present the overall response time achieved by the proposed service broker policy strategy and the other previous strategies in the context of cloud computing based on scenario 1, scenario 2, and scenario 3, respectively. From the experimental results, it is evident that our proposed policy, the VIKOR-based service broker policy, outperformed the other policies in terms of the overall response time in all three scenarios. The results also demonstrate that the average improvements of 34.74% and 13.86% were obtained by the closest DC and optimized response time approaches, respectively. In the case where the user priority was to balance cost and response time, the proposed approach obtained an overall response time close to the best overall response time of 87%.

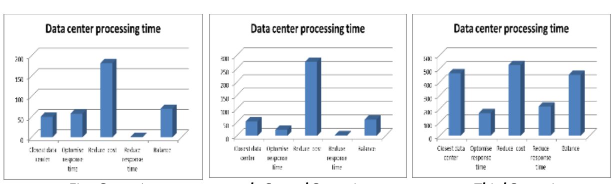
4.2.2. The Data Centre Processing Time

In this section, we discuss the effects of identifying user priorities as one of the crucial factors influencing DC processing time when selecting a DC. We aimed to examine the performance of the VIKOR-based service broker policy and its capability in handling the process of identifying and selecting DCs in the cloud paradigm. In this section, we also illustrate the experimental results of our proposed solution, the VIKOR-based service broker policy, in the three scenarios concerning the DC processing time. The experimental results for the overall DC processing time are presented in Figure 4. Figure 4(a) shows the performance of the VIKOR-based service broker policy based on the first scenario of the processing time of the DC. Similarly, Figures 4(b) and 4(c) illustrate the results of the experiments based on the second and third scenarios taking into consideration the DC processing times, respectively. The results indicate that the proposed solution, the VIKOR-based service broker policy, achieved the best results for both the first and second scenarios compared to the other approaches (closest DC, optimized response time, reduced cost, and balance). This is due to the fact that our proposed solution incorporated the user priority factor, which resulted in reducing the overall response time by producing a lower processing time for the DC in all three scenarios. In the case where the user priority was to balance cost and response time, the proposed approach obtained a DC processing time close to the best processing time.



a. First Scenario b. Second Scenario c. Third Scenario

Figure 3: The results of overall response time of the three scenarios

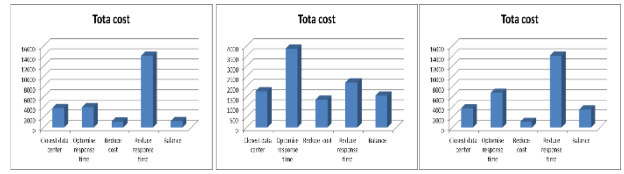


a. First Scenario b. Second Scenario c. Third Scenario

Figure 4: The results of data centre processing time of the three scenarios

4.2.3. The Total Cost

In this section, we illustrate the experimental results of our proposed solution, the VIKOR-based service broker policy, in the three scenarios with respect to the total cost. Figure 5 shows the results obtained for total cost corresponding to the three scenarios. Figure 5a 5b, and 5c present the experimental results for the first scenario, the second scenario, and the third scenario, respectively. From the results, it is evident that the proposed approach, employing user priorities, led to a significant reduction in the total cost for all cases. The results also indicate that the proposed strategy steadily outperformed the other approaches by generating a lower cost for the three scenarios considered in this study. The average improvement of 54.55% and 73.8% were obtained by the closest DC and optimized response time approaches, respectively. Finally, from the results, it can be concluded that the proposed approach obtained total cost data that was better than the closest DC and optimized response time approaches in the case where the user's priority was to balance cost and response time.



a. First Scenario b. Second Scenario c. Third Scenario

Figure 5: The results of the total cost for the three scenarios

4.2.4. Comparison with Other Policies

For evaluation purposes, the proposed approach was compared with the priority-based service broker policy (PBSBP) for a fog computing environment proposed by Arya and Dave (2017), the AFBSB algorithm (Al-Tarawneh and Al-Mous, 2019), and the optimized service broker routing policy (OSBRP) proposed by Manasrah and Gupta (2019). For simplicity and without loss of generality, our work followed the same experimental settings and other environment configurations in the experimental study as those in Arya and Dave (2017) and Manasrah and Gupta (2019). The details of the configurations for the DC, UB, and load balancing and grouping factor are described in Tables 4(a), 4(b), and 4(c), respectively.

Table 4: The configuration of the simulation for the comparison with other policies

(a) Data centre configuration

DCs	Physical Hardware Units	Processor per Hardware Unit	Arch	OS	Cost per VM $/Hr	Memory Cost	Data Transfer Cost($/Gb)	Processor Speed
FDC1	2	3	x86	Linux	1.60	0.05	0.20	500-1000
FDC2	1	4	x86	Linux	2.40	0.05	0.70	1000
FDC3	3	4	x86	Linux	5.00	0.05	0.30	10000
FDC4	1	3	x86	Linux	0.10	0.05	0.10	100
FDC5	1	3	x86	Linux	0.24	0.05	0.11	2000

(b) User base configuration

UB	User Requests/Hour	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg. Peak Users	Avg. Off-Peak Users
UB1	120	3	9	5000	500
UB2	60	3	9	1000	100
UB3	60	3	5	1500	150

(c) Load balancing and grouping factor configuration

Parameters	Value
User grouping factor in userbases	1000
Request grouping factor in datacentres	50
Executable instruction length per request (bytes)	100
Load balancing policy across VMs in a single datacentre (default load balancing algorithm)	Throttled
Simulation duration	24 h
Available memory (MB)	512 MB
Storage	1 TB
Available bandwidth	1000
VM policy	TIME_SHARED
VM image size	10,000

Table 5 illustrates the results of the experiment that concentrates on comparing the proposed broker policy with the most recent broker policies, namely PBSBP (Arya and Dave, 2017), OSBRP (Manasrah and Gupta, 2019), and AFBSB (Al-Tarawneh and Al-Mous, 2019). The main reason for selecting these works is that these works match the objective of this work (i.e. minimizing response time, processing time and the overall cost). From the results, it is evident that our proposed approach outperforms the previous approaches in terms of the total cost, the overall average response time, and the average of the DC processing time.

Table 5: Comparison with other policies

Performance Parameters	PBSBP	OSBRP	AFBSB	Our Proposed Approach
Total cost	2000	1967.6	1125	1047
Overall response time (Avg.)	97	85.8	85.2	84.3
Data centre processing time (Avg.)	25	15.2	15.1	15

The results also demonstrate that the PBSBP technique (Arya and Dave, 2017) is the worst by incurring the highest total cost, the longest average overall response time and the longest average DC processing time. Furthermore, the results of the experiment indicate that the AFBSB technique (Al-Tarawneh and Al-Mous, 2019) is better than PBSBP (Arya and Dave, 2017) and OSBRP (Manasrah and Gupta, 2019) in terms of total cost, the average overall response time and the average DC processing time.

The results of the experiment illustrate a significant improvement in the proposed policy in terms of the average response time, the average DC processing time and the total cost. Most importantly, the results also demonstrate that the idea of incorporating the user priority in the proposed broker policy led to a balance between the response time and the total cost, which, in turn, improved the efficiency while maintaining a low services cost. Similarly, Table 6(a) and 6(b) describes the experimental result for both cases, namely the third case and fourth case comparisons, for the proposed broker policy against PBSBP (fourth case) (Arya and Dave, 2017), and AFBSB (cost) (Al-Tarawneh and Al-Mous, 2019), respectively. It is clear that our proposed approach for the third case outperforms both PBSBP (fourth case) (Arya and Dave, 2017) and AFBSB (cost) (Al-Tarawneh and Al-Mous, 2019) in terms of the total cost, the average overall response time and the average DC processing time. The results shown in Table 6(a) denote that the PBSBP technique (fourth case) is the worst compared to AFBSB (cost) and our proposed approach (third case) by incurring the highest total cost, the longest overall response time and the longest DC processing time. Likewise, the experimental results reported in Table 6(b) indicate that our proposed technique (fourth case) is the best compared to PBSBP (third case) and AFBSB (performance) in total cost, the overall average response time, and the average DC processing time. Finally, we can conclude that the idea of exploiting the user priority in our proposed service broker policy is very beneficial by selecting the best DC that ensures a significant reduction in the total cost, the overall average response time and the average DC processing time.

Table 6: The result of the experiments for the third and fourth cases

(a) Third case comparison

Performance Parameters	PBSBP Case (iv)	AFBSB Cost	Our Proposed Approach Third Case
Total cost	1350	1150	951
Overall response time (Avg.)	190	520	499
Data centre processing time (Avg.)	110	215	368

(b) Fourth case comparison

Performance Parameters	PBSBP Case (iii)	AFBSB Performance	Our Proposed Approach Fourth Case
Total cost	16000	2400	1601
Overall response time (Avg.)	150	98	76
Data centre processing time (Avg.)	70	50	21

5. Conclusion

In the last decade, cloud computing has become a crucial practical solution for a huge number of big data applications. In a heterogeneous cloud environment, many DCs may implement different user jobs at different times and costs. In real-world environments, cloud users typically have different priorities. For instance, some users seek a solution that best serves their request for minimum cost. Conversely, other users seek a plan that processes the given jobs while ensuring minimum response time or are interested in carrying out their tasks at an affordable cost with an acceptable response time. The primary objective of the cloud service provider is to identify the most suitable DC to process the user request, ensuring a high level of QoS, which depends on the predefined user priorities. In this paper, we proposed an efficient and cost-effective service broker policy for DC selection in a heterogeneous cloud environment based on VIKOR, taking into consideration users' specified priorities. To this end, the proposed service broker policy endeavours to minimize the response time and the overall cost based on the users' specified priorities for user-oriented cloud systems. The results of the experiment, performed in various scenarios, demonstrated that the proposed solution outperformed the current policies in terms of response time, DC processing time and total cost in all cases.

Biographies

Mohammed Radi

Department of Computer and Information Sciences, Faculty of Computers and Information Technology, Al-Aqsa University, Gaza, Palestine, 00972599384779, moh_radi@alaqsa.edu.ps

Dr Radi is a Palestinian Assistant Professor at the Faculty of Computers and Information Technology, Al-Aqsa University, Gaza. He obtained a PhD in Computer Science in 2009 from Universiti Putra Malaysia (UPM). Dr Radi is a former Head of the Computer and Information Sciences Department, and a former deputy dean for planning and development at Al-Aqsa University. His research interests include cloud computing, data storage management in cloud, disaster recovery in multi-cloud, job scheduling, fog computing, high-performance computing, and No-SQL databases.

Ali A. Alwan

School of Theoretical and Applied Science, Ramapo College of New Jersey, New Jersey, USA, 0060173546110, aaljuboo@ramapo.edu

Dr Alwan is an Iraqi Assistant Professor. He obtained a master's degree and a PhD in Computer Science from UPM in 2009, and 2013, respectively. Dr Alwan is an Assistant Professor of Computer Science at the School of Theoretical and Applied Science, Ramapo College of New Jersey, USA. His research interests include preference queries; skyline queries; probabilistic, incomplete and uncertain databases; query processing on incomplete data; location-based social networks (LBSN); recommendation systems; and data management in cloud computing.

Abedallah Zaid Abualkishik

Information Technology Management Department, College of Computer and Information Technology, American University, Dubai, United Arab Emirates, 00971501504133, abedallah.abuakishik@aue.ae

Dr Abualkishik is a Jordanian Associate Professor. He obtained a PhD in Software Engineering from Universiti Putra Malaysia in 2013. Dr Abualkishik is an IBM-certified trainer in Big Data, Data Science, Blockchain, and Artificial Intelligence. His research interests include software functional size measurement, software functional measures conversion, cost estimation, empirical software engineering, software reusability, database systems, data management in big data, and data science. He is serving the scientific community as a regular reviewer for several high-impact journals.

Adam Marks

Educational Technology Department, Higher Colleges of Technology, Dubai, United Arab Emirates, 0097144021785, amarks@hct.ac.ae

Dr Marks is an American professor. He obtained his PhD from the University of Central Florida, USA, in 2001. He is an academic and product advisor with Microsoft, Ellucian and Blackboard. He has worked in industry and academia in different leading capacities and roles, both nationally and internationally. Prof. Marks is a globally minded leader in higher education administration with expertise in developing and implementing strategic plans for higher education institutions, with the emphasis on growth, academic integrity, and efficiency.

Yonis Gulzar

Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa, Saudi Arabia, 00966545719118, ygulzar@kfu.edu.sa

Dr Gulzar is an Indian Assistant Professor at King Faisal University, Saudi Arabia. He obtained a master's degree in Computer Applications from Bangalore University, India, in 2013. He obtained a PhD in Computer Science from the International Islamic University Malaysia in 2018. His research interests include preference queries, skyline queries, probabilistic and uncertain databases, query processing on incomplete data, data integration, LBSN, recommendation systems, data management in cloud computing, artificial intelligence, deep learning, and computer vision.

References

Al Sukhni, E. (2016). K-nearest-neighbor-based service broker policy for data centre selection in cloud computing environment. *Int. Res. J. Electron. Comput. Eng*, **2**(3), 5–9.

Al-Tarawneh, M. and Al-Mousa A. (2019). Adaptive user-oriented fuzzy-based service broker for cloud services. *Journal of King Saud University-Computer and Information Sciences*, (Article in Press).

Amazon S3. (2018). Available at: https://aws.amazon.com/s3/ (accessed on 10/03/2020).

Arya, D. and Dave, M. (2017). Priority based service broker policy for fog computing environment. pp. 84–93. In: *Proceedings of the International Conference on Advanced Informatics for Computing Research*, Jalandhar, India, 17–18/03/ 2017.

Benlalia, Z., Beni-hssane, A., Abouelmehdi, K. and Ezati, A. (2019). A new service broker algorithm optimizing the cost and response time for cloud computing. *Procedia Computer Science*, **151**(n/a), 992–7.

Wickremasinghe, B., Calheiros, R.N. and Buyya, R. (2010). CloudAnalyst: A cloudaim-based visual modeller for analysing cloud computing environments and applications. p. 446–52. In: *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, AINA* 2010, Perth, Australia, 20-23/04/2010.

Chen, C.L.P. and Zhang, C.Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, **275**(n/a), 314–47.

Chudasama, D., Trivedi, N. and Sinha, R. (2012). Cost effective selection of data centre by proximity-based routing policy for service brokering in cloud environment. *International Journal of Computer Technology and Applications*, **3**(6), 2057–9.

Gantz, J. and Reinsel, D. (2012). *The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. IDC Iview: IDC Analyze The Future*, 1–16. Available at: https://www.cs.princeton.edu/courses/archive/spring13/cos598C/idc-the-digital-universe-in-2020.pdf (accessed on 24/06/2020)

Google Cloud Storage. (2018). Available at: https://cloud.google.com/storage/ (accessed on 24/06/ 2020).

Kapgate, D. (2014). Efficient service broker algorithm for data centre selection in cloud computing. *International Journal of Computer Science and Mobile Computing*, **3**(1), 355–65.

Khan, M.A. (2020). Optimized hybrid service brokering for multi-cloud architectures. *The Journal of Supercomputing*, **76**(1), 666–87.

Kofahi, N.A., Alsmadi, T., Barhoush, M. and Moy'awiah, A. (2019). Priority-based and optimized data centre selection in cloud computing. *Arabian Journal for Science and Engineering*, **44**(11), 9275–90.

Kumar, R.R., Mishra, S. and Kumar, C. (2018). A novel framework for cloud service evaluation and selection using hybrid MCDM methods. *Arabian Journal for Science and Engineering,* **43**(12), 7015–30.

Limbani, D. and Oza, B. (2012). A proposed service broker policy for data centre selection in cloud environment with implementation. *International Journal of Computer Technology & Applications*, **3**(3), 1082–7.

Manasrah, A.M. and Gupta, B.B. (2019). An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. *Cluster Computing*, **22**(1), 1639–53.

Manasrah, A.M., Smadi, T. and ALmomani, A. (2017). A variable service broker routing policy for data centre selection in cloud analyst. *Journal of King Saud University-Computer and Information Sciences*, **29**(3), 365–77.

Mehdi, N.A., Ali, H., Alwan, A.A. and Abdul-Mehdi, Z.T. (2012). Two-Phase provisioning for HPC tasks in virtualized datacentres. pp. 29–35. In: *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE'2012).* Dubai, United Arab Emirates, 01/03/2012.

Mehdi N.A., Mamat A., Alwan A.A. and Abdul-Mehdi, Z.T. (2011). Minimum completion time for power-aware scheduling in cloud computing. p. 480–9. In: *Proceedings of the International Conference of Developments in e-Systems Engineering (DeSE2011)*. Dubai, United Arab Emirates, 03/12/2011.

Mishra, R.K., Kumar, S. and Naik, B.S. (2014). Priority based round-robin service broker algorithm for cloud-analyst. p. 878–81. In: *Proceedings of the 2014 IEEE International Advance Computing Conference (IACC)*. Gurgaon, India, 21–2/02 2014.

Nandwani, S., Achhra, M., Shah, R., Tamrakar, A., Joshi, K. and Raksha, S. (2016). Weight-based data centre selection algorithm in cloud computing environment. In: S.S. Dash, M.A. Bhaskar, B.K., Panigrahi, and S. Das, (eds.) *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, 515–25. Switzerland: Springer.

Otay, I. and Yıldız, T. (2021). Multi-criteria cloud computing service provider selection employing pythagorean fuzzy AHP and VIKOR. In: *Proceedings of the International Conference on Intelligent and Fuzzy Systems*. Istanbul, Turkey 24–6/08/2021.

Patiniotakis, I., Verginadis, Y. and Mentzas, G. (2015). Preference-based cloud service selection for cloud service brokers. *Journal of Internet Services and Applications*, **6**(1), 1–14.

Radi, M. (2014). Weighted round robin policy for service brokers in a cloud environment. p. 45–9. In: *Proceedings of the International Arab Conference on Information Technology (ACIT2014),* Nizwa, Oman, 9–11/12/2014.

Radi, M. (2015). Efficient service broker policy for large-scale cloud environments. *International Journal of Computer Science Issues*, **12**(1), 85–90.

Rafieyan, E., Khorsand, R. and Ramezanpour, M. (2020). An adaptive scheduling approach based on integrated best-worst and VIKOR for cloud computing. *Computers & Industrial Engineering*, **140**(5), 106272.

Rekha, P.M. and Dakshayini, M. (2018). Dynamic cost-load aware service broker load balancing in virtualization environment. *Procedia Computer Science*, **132**(n/a), 744–51.

Chauhan, S.S., Pilli, E.S., Joshi, R.C. and Singh, G. (2018). UPB: User preference based brokering for service ranking and selection in federated cloud. In: *Proceedings of the IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS).* Indore, India, 16–19/12/2018.

Sharma, V. (2014). Efficient data centre selection policy for service proximity service broker in cloudanalyst. *Int. J. Innovative Comp. Sci. Eng.(IJICSE),* **1**(1), 21–28.

Sidhu, J. and Singh, S. (2017). Improved topsis method based trust evaluation framework for determining trustworthiness of cloud service providers. *Journal of Grid Computing,* **15**(1), 81–105.

Subramanian, T. and Savarimuthu, N. (2016). Application based brokering algorithm for optimal resource provisioning in multiple heterogeneous clouds. *Vietnam Journal of Computer Science*, **3**(1), 57–70.

Sun, L., Dong, H., Hussain, O.K., Hussain, F.K. and Liu, A.X. (2019). A framework of cloud service selection with criteria interactions. *Future Generation Computer Systems*, **94**(n/a), 749–64.

The IBM Cloud. (2018). Available at: https://www.ibm.com/cloud/ (accessed on 28/06/2020).

Trabay, D.W., El-Henawy, I. and Gharibi, W. (2021). *A Trust Framework Utilization in Cloud Computing Environment Based on Multi-Criteria Decision-Making Methods. The Computer Journal.* DOI: 10.1093/comjnl/bxaa138

Windows Azure. (2018). Available at: https://www.azure.cn/ (accessed on 13/05/2020).

Wu, X. (2016). Data sets replicas placements strategy from cost-effective view in the cloud. *Scientific Programming*, **11**(n/a),1–13.

Youssef, A.E. (2020). An integrated MCDM approach for cloud service selection based on TOPSIS and BWM. *IEEE Access*, **8**(n/a), 71851–65.

Zakaria B., Abderahim B., Karim A. and Abdellah E. (2019). A new service broker algorithm optimizing the cost and response time for cloud computing. *Procedia Computer Science*, **151**(n/a), 992–7.