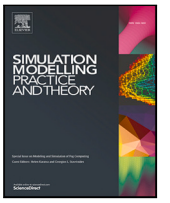
Simulation Modelling Practice and Theory 115 (2022) 102428

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat



Data correlation and fuzzy inference system-based data replication in federated cloud systems

Amel Khelifa [a,*], Riad Mokadem [b], Tarek Hamrouni [a], Faouzi Ben Charrada [c]

[a] LIPAH, Faculty of Sciences of Tunis, Tunis El Manar University, University Campus, Tunis, Tunisia
[b] Institut de Recherche en Informatique de Toulouse (IRIT), Paul Sabatier University, Toulouse, France
[c] LIMTIC, Faculty of Sciences of Tunis, Tunis El Manar University, University Campus, Tunis, Tunisia

ARTICLE INFO

Keywords:
Federated cloud systems
Cloud provider
Data replication
Correlation
Clustering
Fuzzy inference system
Economic model
Service Level Agreement
Profit

ABSTRACT

Federated cloud is a promising solution for cloud providers that enables collaboration and leasing resources among multiple cloud providers. However, resource management in such system is very challenging, as each cloud provider must maintain its own economic profit while meeting the requirement of service level agreement (SLA). In this respect, we propose a dynamic and periodic data replication strategy in federated cloud systems. It aims to guarantee the monetary profit of a cloud provider while satisfying its users' requirements in terms of response time and minimum availability. To identify replicas, we perform a periodical analysis of the users' tasks using the spectral clustering technique to extract the existing correlations between remote data related to SLA violations. The adaptation of such correlations can significantly reduce data transfer amount and the time required to transfer it, thereby reducing tasks' response time and the number of future SLA violations. Then, we rely on a fuzzy inference system to place the groups of replicas considering four main parameters to choose replicas placements among his owned or leased resources from other providers. Furthermore, a replicas number adjustment is performed when SLA is satisfied over time. To demonstrate the efficiency of our strategy, performance of the proposed strategy are compared alongside existing single-clouds-based and interconnected-clouds-based data replication strategies. The obtained results indicate that our strategy decreases the amount of SLA violations while preserving the monetary profit of providers.

1. Introduction and motivations

The market of cloud services is rising rapidly where cloud providers offer multiple facilities geographically distributed around the world to be shared by their users according to the Service Level Agreement (SLA) [1]. However, with the growth of the users number, heavy demands are put on cloud providers. Hence, it becomes difficult to maintain the agreed-upon quality of service (QoS) described in the SLA relying only on the facilities owned by a single cloud provider. This latter deploys its data centers (DCs) in order to execute the users' tasks and store their huge amount of data. Virtual machines (VMs) are connected according to a network hierarchy, where the bandwidth capacity and cost varies with the geographical location of the machines [2,3]. Thus, to maintain the QoS when performing users' tasks, data should be available as close as possible to the users' tasks region, which allows more local data accesses and then reduces data transfer delay and cost.

* Corresponding author.
E-mail addresses: amel.khelifa@fst.utm.tn (A. Khelifa), riad.mokadem@irit.fr (R. Mokadem), tarek.hamrouni@fst.rnu.tn (T. Hamrouni), faouzi.bencharrada@fst.utm.tn (F. Ben Charrada).

https://doi.org/10.1016/j.simpat.2021.102428
Received 31 July 2021; Received in revised form 1 October 2021; Accepted 21 October 2021
Available online 12 November 2021

In this regard, data replication is a well-known technique that involves creating multiple copies of data (replicas) on the system. Multiple replication strategies have been proposed in the clouds with the objective of increasing availability and fault tolerance, and improving the performance so that the amount of SLA violations is reduced [4–8].

Despite the benefits of performing replication, the choice of data to be replicated and the appropriate VMs to hold replicas is not trivial given the variation in user demands, fluctuating resource loads, and bandwidth conditions, combined with the resources owned by a cloud provider in each geographic region. A single cloud provider has not the financial ability to deploy resources all over the world to cope with the geographic dispersion of its users [9]. In this regard, both providers and users resort to the use of interconnected clouds that connect various resource options offered by multiple providers such as the federated cloud in which interconnection is operated by them and transparent to their users. Usually, this interconnection is managed according to a set of rules that define the independence of each cloud provider in terms of autonomy, privacy, and protection [10–12]. The federated cloud enables multiple advantages when performing replication [13–16] including (1) satisfaction of QoS requirements by leasing the needed resources from other cloud providers to place data replicas (2) availability and low latency by taking advantage of the geographic dispersion of resources and placing data near the users in order to adapt to the regional behavior of users (3) cost optimization where a provider can rent resources from the other providers of the federated cloud while taking advantage of the variation of the pricing policies among them, in addition to increasing its revenue by renting out the idle resources to them [17].

From cloud providers' perspective, it is essential that data replication strategies consider the cost involved in replication while simultaneously ensuring their users' QoS requirements and maintaining their monetary profit. However, balancing these competing goals is more challenging when dealing with interconnected-clouds systems. For the QoS satisfaction, some replication strategies resort to exploiting knowledge about the correlations between replicas that can be expressed by the common or simultaneous accesses to them (by users, applications, tasks, services, etc.) as well as by the semantic of data attributes (data identifier, date of creation, date of modification, etc.). These correlations cloud be extracted using several methods such as data mining and machine learning techniques [18–22]. Several studies have shown that considering these correlations during replication reduces the research space and allows the respect of SLAs. In fact, placing correlated data into the same resource that requests it or into a neighboring resource improves data locality, reduces overload caused by the distributed access to these data groups, and provides better latency [18,19,21–25]. Most of these studies have been carried out on single cloud systems, which motivates the adoption of such correlations in interconnected-clouds systems. Dealing with the replication cost optimization, some research efforts have investigated the preservation of cloud providers' profits in a single cloud system [7,21] where the majority of them are interested in reducing the provider's expenses without being interested in the economic profit of this provider. Very little effort has been devoted to the interconnected-clouds systems. Most of them are consumer-centric [13,14] where other strategies are provider-centric [26].

This work is motivated by the following observations from the literature. Federated clouds bring many advantages to the cloud business, where the main objective of the cloud provider is to maintain its monetary profit while meeting the SLA requirements for its tenants. Replicating data among the shared resources of federated clouds can benefit the cloud provider in order to reduce the amount of SLA violations as well as operational costs. In this regard, the correlations between data can be exploited to optimize its access time and its transfer cost. Despite the effectiveness of data correlations in improving performance and reducing the amount of SLA violations, interconnected cloud-based replication strategies do not widely exploit it. In addition, their efforts of cost optimization are limited to lowering the operating cost without considering the monetary profit of the provider. Therefore, we address the problem of data replication in a federated clouds system with the aim of maintaining the profit and satisfying the SLA requirements by exploiting data correlations.

In this regard, we propose a dynamic and periodic data replication strategy called DCRF (Data Correlation and fuzzy inference system-based data Replication in Federated cloud systems). Our strategy considers users' SLOs in terms of response time and minimum availability. To this end, it is periodically triggered as a response to SLA violations. It uses the spectral clustering technique [27] that allows an unsupervised data partitioning, enabling then the extraction of the correlations that exist between remote data that are frequently and jointly accessed by tasks causing SLA violations on a managed VM. Then, to place the replicas groups, the proposed strategy relies on a Fuzzy Inference System (FIS) [28] considering four main parameters. The latter is used to estimates the potential of placing the correlated data groups, either on owned or rented VM. Furthermore, a replicas number adjustment is performed when SLA is satisfied over time. Our strategy attempts to reduce the amount of SLA violations and preserve the monetary profit of the cloud provider, unlike the existing strategies in the related work that ignore the existence of data correlations and focus on lowering the operating cost of replication only.

In brief, the contributions of this paper are as follows:

1. A dynamic and periodic data replication strategy (DCRF) is proposed in federated cloud systems. It deals with main issues of data replication in a federated cloud system, namely: what data to replicate, and where to place replicas while taking into consideration the existing correlations between remote data frequently accessed by users' tasks causing SLA violations. In addition, a replicas number adjustment is applied when SLA is satisfied over time.
2. A fuzzy inference system (FIS) is used to estimate the potential of placing groups of correlated data among the owned resources of a cloud provider and the offered resources by other providers in the federated system. The proposed FIS considers four important parameters: (1) data transfer time ratio, (2) virtual machine load, (3) data availability, and (4) cloud provider profit. Furthermore, an economic model is presented to estimate the monetary profit of cloud provider that includes both revenues and expenditures.

3. The proposed strategy is validated through intensive comparative simulations using the cloud simulation toolkit CloudSim [29]. The effectiveness and the efficiency of our strategy have been proven compared to recent single-clouds-based strategies and interconnected-clouds-based replication strategies considering several performance evaluation metrics. A significant reduction in the amount of SLA violations has been observed, while achieving an improvement in the total monetary provider profit despite adopting different costs models of service providers.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 defines the considered federated cloud system. Section 4 describes the proposed data replication and related algorithms. Section 5 describes the proposed fuzzy inference system that our strategy relies on. Section 6 presents the experimental evaluation of the proposed strategy on the CloudSim [29]. Section 7 concludes our study and provides an outlook on our future work.

2. Related work

Data replication strategies are widely studied in the single cloud environments [4,5]. They focus on increasing data availability, improving quality of service, and optimizing replication cost. In this regard, MORM strategy [30] uses a bio-inspired algorithm (artificial immune algorithm) during replication to take into account data unavailability, service time, load variance, power consumption, and average latency as optimization objectives. It lowers the operating cost by reducing the resources energy consumption. Some profit-aware strategies rely on economic models such as RSPC [3], which seeks to reduce SLA violations in terms of response time while taking into account the provider profit. Others exploit data correlations, such as CEMR [21], aiming to preserve provider profit and reduce service level violations.

With the growing trend towards interconnected-clouds environments [10], research studies have focused on solutions for these environments to better leverage the computing and storage infrastructure they provide [15,16]. In the following, we examine the trade-off between satisfying the quality of service (QoS), optimizing cost, and maintaining the provider's profit through data replication and data placement strategies designed for interconnected clouds systems.

2.1. Consumer-centric approaches

Consumer-centric approaches aim to maximize the level of performance required by users, such as availability, reduce latency and response time at the same time minimizing the cost that users pay to cloud providers. This is done by exploiting the varied prices and features of the resources made available by the cloud providers.

RACS [13] and DepSky [14] are among the first strategies to manage user data through interconnected clouds storage. These approaches aim to enable fault tolerance and minimize the monetary cost paid by users. Their data storage mechanisms consist in distributing user data among several cloud providers to avoid cases of provider lock-in and cloud outage. While cost reduction is obtained by exploiting differences in pricing policies between cloud providers. However, the cost models used by both RACS and DepSky are limited to the cost of storage only.

TripS [31] is a lightweight system to determine data placements on behalf of users' applications running on geo-distributed interconnected clouds system. For this aim, it considers data center locations and storage tiers. To this end, it considers data center locations and storage tiers. TripS requires several inputs, including SLA requirements, consistency model, fault tolerance, latency, storage, and bandwidth costs information for the corresponding storage tier. It then uses mixed-integer linear programming to determine the appropriate data locations to meet the SLA requirements at a low cost.

A similar work called DAR is presented in [32] where the storage framework exploits the variation of Get/Put latency and resource prices offered by cloud providers of the system in order of satisfy SLO requirements including data retrieval latency, availability, and optimization of the paid cost by cloud users. DAR uses integer programming to model the problem of data placement and resource allocation. Hence, two heuristic solutions were presented, including a dominant-cost based data allocation algorithm and an optimal resource reservation algorithm while the used cost model includes storage, transfer, Get and Put costs during resource reservation time.

Meeting availability and latency requirements with cost minimization has also been the subject in [33]. An adaptive data placement framework is proposed called ADPA, which consists of deciding on the appropriate data placement based on the expected frequency of access. The framework predicts the frequency of data access based on historical workload using LSTM (Long short-term memory). It then uses a learning-based data placement algorithm where data can be migrated from one cloud provider to another as workload changes. The cost model used covers storage, network, operations, and migration costs.

In the same context, the correlations between data were extracted and exploited. In [34], a data placement strategy for social services was proposed. The framework seeks to reach multiple objectives including maintaining the quality of service for users, reducing resource usage, and reducing their carbon footprint. Therefore, it associates a certain weight to each objective to allow service providers to seek trade-offs between objectives according to their specific requirements. They focused on reducing latency while lowering the operational cost of users' services when placing users' data across multiple clouds. To this end, social relationships between users were taken into account. Hence, graph-cut methods were used to place data. Taking into account the connections between cloud users represented by shared data access enabled the data placement strategy to meet latency requirements and reduce bandwidth consumption more efficiently than random replication. In [35], data correlations were also taken into account during replication. The replication strategy presented aims to balance the trade-off between reducing data access latency and optimizing the associated cost. To achieve this goal, the strategy relies on extracting data correlations with respect to their location and frequency of access. Indeed, data with high dependencies and high access frequency are selected for replication. Next, the strategy places the replicas according to a cost model that includes the cost of storage and the cost of data transfer. The placement of replicas is done in such a way that the current placement yields a lower cost compared to the non-replicating state.

2.2. Provider-centric approaches

On the other hand, provider-centric approaches mainly focus on the minimization of the operational cost of cloud providers while ensuring the agreed upon requirements to users.

SPANStore (Storage Provider Aggregating Networked Store) [26] is a unified storage service that is built as an interconnected clouds environment for application providers. It aims to meet service level objectives (SLOs) in terms of fault tolerance and latency while minimizing the cost of application execution. SPANStore takes advantage of the geographical dispersion of the data centers of several storage providers to manage workload changes. It then replicates data closer to customers, reducing both latency and SLO violations. It also benefits from the cost diversity of Put/Get operations between regions to reduce replication costs, including storage, network, and update costs.

Reducing the monetary cost paid by cloud application providers was also addressed in [36]. Dynamic and linear programming techniques were used to enable satisfying users' latency requirements while minimizing the cost of applications running over Geo-distributed data centers. This monetary cost includes replica creation, storage, Put/ Get operations, and potential migration costs. A lightweight heuristic solution was presented to determine data locations and redirect users' requests so access latency can be guaranteed at a minimum cost. It relies on the exploitation of the price differences across data storage classes to minimize the cost paid by application providers. In addition, it considers the characteristics of data objects related to their access frequency, which can be hot or cold point objects.

A Preventive Disaster Recovery Plan with Minimum Replica Plan (PDRPMR) is presented in [37]. The plan aims to balance the trade-off between data reliability and the cost of storing replicas. Therefore, it relies on deploying a minimum number of replicas while taking into account the importance of the data and the duration of data storage. Indeed, the plan envisages maintaining reliability requirements for both short and long-term data duration while ensuring low storage consumption. In addition to the storage cost, the adaptive replication placement strategy presented in [38] considers the cost of the network when accessing the data. The strategy relies on workload prediction to place replicas with cost optimization. Future user demands are forecasted using the ARIMA time-series technique and then replicas are placed to meet the corresponding availability and reliability requirements. It relies also on a cost model that considers the differences in pricing policies between the geographic regions of the system.

The correlations between the users have been taken advantage in [39], where data placement and replication strategies are proposed for the social network over geo-distributed cloud system to optimize cost while guaranteeing latency requirements. The problem was formulated as a dynamic set cover problem. Greedy algorithms were combined to guarantee the latency requirements of the social network users while maintaining a minimum cost for the service providers including data storage, transfer, and update. Relationships between social network users were taken into account, where a replica is placed close to users linked to the data owner.

2.3. Discussion

Following the review of the above-discussed strategies, the trade-off between optimizing costs and meeting the needs of the users as described in the SLA is of great importance in the interconnected clouds.

On the one hand, meeting users' needs focused on availability, which was often maintained with a minimum number of replicas. While latency and response time were improved by exploiting the diversity of interconnected clouds resources and their geographical dispersion. Only a few strategies took advantage of correlations between data or users. These correlations proved effective in meeting SLA requirements, encouraging their adoption for data management [23].

On the other hand, cost optimization has mainly focused on the cloud consumer perspective. Most of the above-mentioned studies have considered the cost variation between interconnected clouds resources in order to minimize the operating cost was often limited to network and storage costs. Only a few works consider the perspective of the cloud providers where none of them has considered the monetary profit of the providers. It is important to note that among the reviewed strategies, there is no single approach taking advantage of data correlations to ensure the economic profit of the cloud provider while taking into account the satisfaction of the SLOs of its users. Therefore, in this paper, we propose a data replication strategy based on data correlations and a fuzzy inference system to address the trade-off between the preservation of the cloud provider's monetary profit and SLA compliance in a federated cloud as an interconnected clouds environment in which the interconnection is initiated and managed by the cloud providers.

Tables 1 and 2 identify a non-exhaustive list of some consumer-based and provider-based approaches, respectively. We investigate some characteristics including: the type of a strategy (static (S) or dynamic (D)), the periodicity, data correlation consideration, the addressed replication issues, the QoS metrics to satisfy, cost model consideration, the evaluation method, and the comparison details.

3. System model

In this section, we present a description the federated cloud system on which relies our proposed data replication strategy and which is inspired by the InterCloud project [40]. We consider a federated cloud system that consists of integrating the facilities that multiple cloud providers offer. Each provider is autonomous and has its own clients who must manage their tasks and data in accordance with a service level agreement. We assume that providers are allowed to offer only a subset of their actual resources to be shared among the federated cloud, so that sensitive provider's information cannot be revealed to other federated cloud members. In addition, we assume that interoperability and coordination issues between the federated clouds members are taken care off, since these latter are not the focus of this paper. The users' data is initially stored in the set of VMs owned by their cloud provider (that

Table 1
Consumer centric approaches.

Strategy	[13]	[14]	[34]	[31]	[32]	[35]	[33]
Year	2010	2013	2014	2017	2017	2018	2020
Approach	Inter-cloud storage proxy	Inter-cloud storage system	Data placement strategy	Data placement strategy	Data storage and resource allocation policy	Data replication strategy	Data placement strategy
Type	D	D	D	D	D	D	D
Periodicity	–	+	–	+	–	+	+
Correlations	–	–	+	–	–	+	–
Replication issues	–	Replicas number	Replicas placement	Replicas placement, Replicas selection	Replicas placement, Replicas selection, Replicas number	Replica decision, replicas placement, replicas number	replicas placement
QoS and SLA	Availability, Fault tolerance	Fault tolerance, Security	Reduced latency	Availability, Fault tolerance	Availability, Reduced latency	Reduced latency	Availability, Reduced latency
Cost consideration	+	+	+	+	+	+	+
Profit	–	–	–	–	–	–	–
Evaluation	Simulation	Real implementation	Simulation	Simulation	Real implementation	Simulation	Simulation
Compared with	Single provider storage	Single provider storage	Random replication and near user replication	[26], TripS variations	[26], Random replication, cheapest replication	No replication	Colony optimization algorithm and genetic algorithm based replications

Table 2
Provider centric approaches.

Strategy	[26]	[36]	[38]	[37]	[39]
Year	2013	2019	2019	2020	2020
Approach	Data replication strategy	Data replication strategy	Replicas placement strategy	Data replication strategy	Data placement strategy
Type	D	D	D	D	D
Periodicity	+	+	+	+	+
Correlations	–	–	–	–	+
Replication issues	Replicas placement, Replicas selection, Replicas number	Replicas placement, Replicas selection, Replicas number	Replicas placement, Replicas selection, Replicas number	Replicas placement, Replicas number	Replicas placement
QoS and SLA	Reduced latency Fault tolerance	response time, Availability	Reduced latency availability	Reliability	Reduced latency
Cost consideration	+	+	+	+	+
Profit	–	–	–	–	–
Evaluation	Real implementation	Simulation	Simulation	Simulation	Simulation
Compared with	Single provider storage Fixed replication	Benchmark algorithm	Random replication and cache replication	3 replicas	Random replication, Full replication, 2 and 3 replicas

are not leased to other cloud providers) to be accessed by the users' tasks as they run. The data in this system are of read-only nature and used for OLAP purposes, so there are no consistency issues involved.

When executing the set of users' tasks T and managing their data D, a cloud provider must meet service level objectives (SLOs) in terms of response time and minimum availability (SLO_{RT} and SLO_{MA} respectively). In fact, the response time of a task $t \in T$ denoted $t.Rt$ must not exceed SLO_{RT}, while data availability must be greater than SLO_{MA}. Users of each cloud provider pay for their use of the cloud service according to the pay-per-use model. For simplicity, we assume that a tenant is charged for each task execution. On the other side, the cloud provider has to bear the cost of performing its users' tasks. This cost includes the cost of tasks' execution, the cost of data replication to both owned and leased resources, and the cost of SLA violation. There are other expenses such as licensing, security, power consumption, and the administration that are not the focus of this paper.

Let F be a federated cloud system formed by the aggregation of all the sets of data centers offered by M cloud providers that are distributed among L geographical regions RG. Consequently, the federated cloud is denoted $F = \cup_i^M DC_i$. Let $DC_i = \cup_l^L DC_i^l$ be the set of data centers belonging to a cloud provider Cp_i and let dc_{ij}^l be the jth data center of cloud provider Cp_i located at the lth

geographic region. Each data center dc_{ij}^{l} is virtualized to an heterogeneous set of virtual machines (VMs) denoted as VM_{ij}^{l} hence the set of VMs belonging to the cloud provider Cp_i located at the lth geographic region is given as VM_i^l. Hence, the set of VMs of the federated cloud belonging to the same geographic region is denoted by $VM^l = \cup_i^M VM_i^l$ and $VM = \cup_l^L VM^l$

Considering the ownership of resources among cloud providers, we define the set of VMs owned by a cloud provider Cp_i as VM_i^{Owned}. However, a provider may offer and rent its idle resources to other cloud provider. At the same time, it may rent idle VMs from other cloud providers in the system. Therefore, we define the set of idle VMs offered by the cloud provider CP_i to other cloud providers as $VM_i^{Offered}$ hence the set of idle VMs offered to be rent in the federated cloud is defined as

$$VM^{Offered} = \cup_i^M VM_i^{Offered} \ with \ (VM_i^{Offered} \subset VM_i^{Owned})$$

While we define the set of VMs that the cloud provider CP_i rents to other provider CP_c as $VM_{i\Rightarrow c}^{Rented}$ hence the set of CP_i's VMs rented to other providers is defined as

$$VM_{i\Rightarrow}^{Rented} = \cup_{c\neq i}^{M} VM_{i\Rightarrow c}^{Rented} \ with \ (VM_{i\Rightarrow}^{Rented} \subseteq VM_i^{Offered})$$

As for the set of VMs rented by the cloud provider CP_i from other cloud provider CP_c is given as $VM_{i\Leftarrow c}^{Rented}$ hence the set of rented VMs to CP_i from other providers is defined as

$$VM_{i\Leftarrow}^{Rented} = \cup_{c\neq i}^{M} VM_{i\Leftarrow c}^{Rented}$$

Consequently, the set of virtual machine to be used and managed by a cloud provider CP_i to store its users' data D and execute their tasks T is given as $VM_i^{Managed}$. This set of managed VMs includes the owned VMs by the cloud provider Cp_i that are not leased to other cloud providers and the set of VMs rented from other cloud providers in the system and it is defined as

$$VM_i^{Managed} = \{VM_i^{Owned} \backslash VM_{i\Rightarrow}^{Rented}\} \cup VM_{i\Leftarrow}^{Rented}$$

Each VM $vm_{\in}VM$ is associated with an operating cost according to its functional characteristics, its location RG_l, and the cloud provider owning the VM Cp_i. We represent a VM vm_k by the seven-uple $\{Cap_{CPU}, Cap_{WQ}, Cap_{Storage}, Cost_{Process}, Cost_{Storage}, Price_{Rent}(vm_k), P_{Failure}(vm_k)\}$, the description of each uple is indicated in Table 3.

Each user's task $t \in T$ is submitted for execution to a VM vm_k^l located in the region RG_l closest to the user. Each task requires accessing to a data set $D_t \subseteq D$. Each data $d \in D_t$ is accessed by the task t with a given access frequency $freq(d,t)$ during its execution. We note that the user's tasks are heterogeneous with respect to their size $t.Size$ (the number of instructions of the tasks), the number of their required data D_t, and their access frequency. Local access to data d is performed if this latter is stored in the VM executing t. Otherwise, remote access to d is performed, resulting in higher cost and access time compared to local data access according to the network links between the VMs.

All VMs forming the system $vm_k \in VM$ are connected via network facilities where VMs in the same data center are connected with intra-data center links allowing more bandwidth and lower cost compared to VMs hosted by different data centers. Whereas, VMs hosted by different data centers within the same geographic region are connected via intra-regional links allowing abundant bandwidth and lower cost compared to VMs that are connected with inter-regional links. Fig. 1 (A) illustrates an example of the proposed system and the associated network links between the facilities of four cloud providers spread over three regions. Fig. 1 (B) illustrates an example of an intra-data center network links.

4. Our proposition: data correlation and fuzzy inference system-based data replication strategy

In this section we propose a novel dynamic data replication strategy that targets a cloud provider CP_i belonging to the federated cloud. Our strategy aims to take advantage from the resources offered by other members of the federated cloud with the objective of preserving its monetary profit while satisfying SLOs of its users in terms of response time and minimum availability (SLO_{RT} and SLO_{MA} respectively).

Our strategy is periodically triggered on a virtual machine (VM) $vm_k \in VM_i^{Managed}$ of the cloud provider CP_i in response to SLA violations in terms of response time. Hence, the replication period P is represented by the number of tasks that were close to causing an SLA violation while their execution. We note that a task t executed on a VM $vm_k \in VM_i^{Managed}$ is considered to be close to causing an SLA violation if its response time noted $t.RT$ is close to a weight w to violate the SLA. We refer to this task as a violating task. We define Th_{RT} a threshold to identify the tasks that are close w to exceed the SLO_{RT} as described in Eq. (1).

$$Th_{RT} = w * SLO_{RT} \ with \ 0 < w < 1 \tag{1}$$

Fig. 2 and Algorithm 1 summarize our proposed strategy that deals with the following main issues:

1. identification of the groups of correlated replicas
2. placement of the groups of replicas
3. replica number adjustment

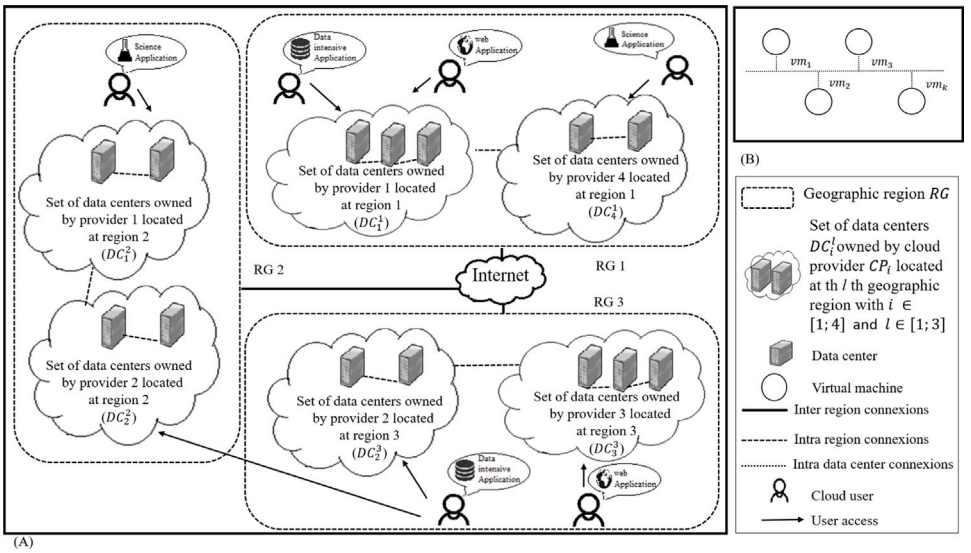


Fig. 1. (A) Example of the considered federated cloud system and the associated network links between the facilities of four cloud providers distributed among three regions (B) Example of an intra-data center network links.

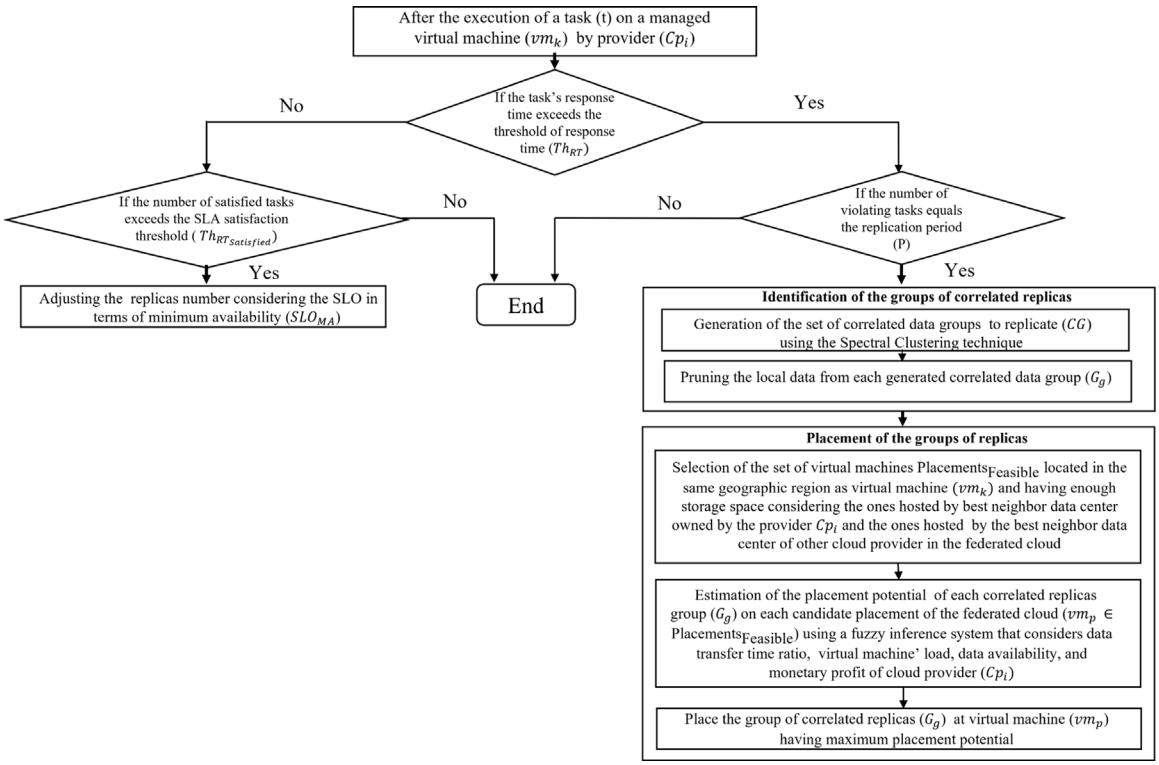


Fig. 2. Flowchart of the proposed strategy.

Table 3
Used notations by replication strategy.

Notation	Description
F	Federated cloud system formed by a set of data centers offered by M cloud provides and distributed on L regions
Cp_i	The ith cloud provider belonging to the federated cloud
DC_i	The set of data centers of cloud provider Cp_i
DC_i^l	The set of data centers of cloud provider Cp_i located on RG_l
dc_{ij}^l	the jth data center of cloud provider CP_i located on RG_l
VM	the set of all VMs (VMs) of the federated cloud
VM^l	Set of VMs belonging to a geographic region RG_l, $VM^l = \cup_i^M VM_i^l$
VM_i^l	Set of VMs belonging to cloud provider Cp_i and located on RG_l
VM_{ij}^l	Set of VMs belonging to cloud provider Cp_i and hosted by dc_{ij}^l located on RG_l
VM_i^{Owned}	Set of VMs belonging to cloud provider Cp_i
$VM_i^{Offered}$	Set of VMs offered to be rent by CP_i to other cloud providers ($VM_i^{Offered} \subset VM_i^{Owned}$)
$VM_{i\Rightarrow}^{Rented}$	Set of VMs rented from Cp_i to other providers Cp_c $(i \neq c)$ thus $VM_{i\Rightarrow}^{Rented} = \cup_{c\neq i}^{M} VM_{i\Rightarrow c}^{Rented}$ ($VM_{i\Rightarrow}^{Rented} \subseteq VM_i^{Offered}$)
$VM_{i\Leftarrow}^{Rented}$	Set of VMs rented by Cp_i from other providers Cp_c $(i \neq c)$ thus $VM_{i\Leftarrow}^{Rented} = \cup_{c\neq i}^{M} VM_{i\Leftarrow c}^{Rented}$
$VM_i^{Managed}$	Set of all VMs managed by cloud provider Cp_i used during the execution of users tasks, $VM_i^{Managed} = \{VM_i^{Owned} \setminus VM_{i\Rightarrow}^{Rented}\} \cup VM_{i\Leftarrow}^{Rented}$
vm_k	The k^{ih} VM $vm_k \in VM_i^{Managed}$
Cap_{CPU}	Computational capacity of VM vm_k
Cap_{WQ}	Waiting queue capacity of VM vm_k
$Cap_{Storage}$	Storage capacity of VM vm_k
$Cost_{Process}$	Processing cost per MI on a VM vm_k
$Cost_{Storage}$	Storage cost on a VM vm_k
$Price_{Rent}(vm_k)$	Rent price of an idle VM vm_k to other cloud providers
$P_{Failure}(vm_k)$	Probability of failure of a VM vm_k
$Cap_{Bw}(vm_s, vm_r)$	Bandwidth capacity between two VMs vm_s holding data and vm_r requesting data
$Cost_{Transfer}(vm_s, vm_r)$	Transfer cost between two VMs vm_s holding data and vm_r requesting data
D	Set of data of the users of the cloud provider CP_i
T	Set of tasks of the users of the cloud provider CP_i
P	Replication period indicated by the number of tasks that were close to violate the SLA
$t.RT$	Response time of task t
Th_{RT}	Threshold on tasks response time to identify the tasks considered for analysis
$Th_{RT_{satisfied}}$	Threshold associated to the number of executed tasks that satisfied the SLA objectives in terms of response time
$CPenalty$	SLA violation penalty per each task execution
$RevT$	Provider revenue per user task execution
SLO_{RT}	The users' SLO in term of response time
SLO_{MA}	The users' SLO in term of minimum availability
K	Number of correlated data groups to generate
CG	Set of extracted groups of correlated data
$DTTR(d, vm_p)$	Data transfer time ratio of a data d if placed on VM vm_p
$Load(vm_p)$	load of a VM vm_p
$Av(d, vm_p)$	Data availability of a data d if placed on VM vm_p
$Profit(vm_p)$	Monetary profit of a VM vm_p
$PlacementPotential(d, vm_p)$	Placement potential of a data d on VM vm_p
$PlacementPotential(G_g, vm_p)$	Placement potential of a set of correlated data G_g on VM vm_p

Algorithm 1: Proposed Replication Algorithm

Input: vm_k: the virtual machine triggering replication, P: the replication period, SLO_{RT}: the response time SLO, SLO_{MA}: minimum availability SLO,$Th_{RT_{satisfied}}$: threshold associated to the number of executed tasks that satisfied the SLA objectives in terms of response time, **K**: number of correlated data groups to generate

Output: Replicas management

$Th_{RT} = w * SLO_{RT}$ *with* $0 < w < 1$;
After the execution of a task t on vm_k
if *(response time of a task* $t.RT > Th_{RT}$*)* **then**
    $Counter_{SLA\_violations} += 1$;
    $Counter_{SLA\_satisfactions} = 0$;
else
    $Counter_{SLA\_satisfactions} += 1$;
if *(*$Counter_{SLA\_violations} == P$*)* **then**
    CG = Replica_Identification(vm_k,K);
    for ***each*** *correlated data group* $G_g \in CG$ **do**
        Fuzzy_Replicas_Placement(vm_k,G_g);
if *(*$Counter_{SLA\_satisfactions} == Th_{RT_{satisfied}}$*)* **then**
    Adjust_Replicas_Number(vm_k) ;

Our strategy performs a replicas number adjustment when the SLA in terms of response time is satisfied over time. The adjustment consists of deleting unnecessary replicas from the VMs. Less frequent replicas having a replica factor RF^d higher than the minimal replicas factor MRF^d enabling SLA satisfaction in terms of minimal availability are identified. These replicas will then be deleted from the system. Algorithm 2 summarizes the replicas factor adjustment process.

Algorithm 2: ADJUST_REPLICAS_NUMBER(vm_k)

Input: vm_k: virtual machine triggering replication
Output: Replicas number adjustment
Select R_{vm_k} the set of stored replicas on vm_k;
for ***each*** *data* $d \in R_{vm_k}$ **do**
    Estimate the minimal replicas factor MRF^d;
    Estimate total access frequency;
Sorting the set R_{vm_k} in ascending order according to their total access frequency;
for ***each*** *data* $d \in R_{vm_k}$ **do**
    if $(RF^d \geq MRF^d)$ **then**
        Delete replicas of d from vm_k;

4.1. Identification of the groups of correlated replicas

To identify the set of data to replicate at each replication period, the strategy uses the spectral clustering algorithm [27] due to its powerful clustering capability and efficiency compared to other clustering algorithms such as K-means [41] as well as its wide use in the field of data management [22,42,43]. Given the correlation (similarity) matrix CM of the accessed data and a number **K** indicating the number of data groups to generate, the groups of replicas are selected. Algorithm 3 summarizes the process of replicas groups' identification.

4.1.1. Generation of the correlated data groups

Our strategy applies a spectral clustering algorithm to the correlation matrix ($n * n$ matrix) where n represents the total number of data required by the violating tasks executed on the virtual machine (VM) vm_k representing the replication period P. Hence, the correlation between a pair of data (d_i, d_j) is estimated by the number of common tasks accessing them during their execution in a VM vm_k as indicated by Eqs. (2) and (3).

$$If(i \neq j) \quad then \quad Corr_{d_i,d_j} = \frac{|T_k(d_i) \cap T_k(d_j)|}{P} \tag{2}$$

$$If(i == j) \quad then \quad Corr_{d_i,d_j} = 0 \tag{3}$$

where $T_k(d_i)$ ($T_k(d_j)$ respectively) is the set of tasks accessing data d_i (d_j respectively) and P is the number of violating tasks executed on the VM vm_k representing the replication period.

After the application of the spectral clustering algorithm, we obtain a set $CG = \{G_1, G_2, \ldots, G_{\mathbf{K}}\}$ of **K** correlated data groups that are accessed jointly by the violating tasks.

4.1.2. Selection of data to replicate

We consider replicating remote correlated data that violating tasks access jointly and frequently. Therefore, we prune the local data from the generated data groups belonging to the VM vm_k where the replication is triggered. Afterwards, all groups will be sorted in descending order according to their average access frequency estimated using Eq. (4).

$$AvgFreq(G_g) = \frac{\sum_{d \in G_g} freq(d, vm_k)}{|G_g|} \tag{4}$$

Algorithm 3: REPLICAS_IDENTIFICATION

Input: vm_k: virtual machine triggering replication, **K**: number of correlated data groups to generate
Output: G The set of correlated replicas groups
$CG \leftarrow \emptyset$;
Extraction of CG the group of correlated replicas groups using the spectral clustering technique;
for ***each*** *correlated data group* $G_g \in CG$ **do**
    for ***each*** *data* $d \in G_g$ **do**
        if *(data* d *is stored in* vm_k*)* **then**
            $G_g \leftarrow G_g \backslash \{d\}$
for ***each*** *correlated data group* $G_g \in CG$ **do**
    Calculate $AvgFreq(G_g)$ the average access frequency of data group G_g;
Sorting the set CG in descending order according to their average access frequency ;

4.2. Placement of the groups of replicas

Replica placement among virtual machines (VMs) of the federated cloud has been proven to be NP-hard [44]. The dynamic nature of clouds and the large amount of data and managed virtual machines make the replica placement problem large, complex and in most cases difficult to model. In this respect, we propose an approximate solution that considers a reduced searching space represented by a subset of the VMs that exist within the same geographic region of the VM vm_k triggering the replication. The proposed heuristic selects a replica placement vm_p to receive the group of correlated replicas in order to satisfy SLO_{RT} and SLO_{MA} in a profitable way. For this aim, the placement virtual machine vm_p should satisfy various constraints. Constraints can be expressed by: (1) Storage constraint, which refers to the VM's ability to store all the replicas. (2) Hardware constraint, which refers to the processing capacity and network capacity of a VM to execute the users' tasks. (3) Performance constraint, which indicates the quality of service (QoS) or service level objectives (SLOs) that should be met in terms of response time and minimum data availability. (4) Cost constraint, which indicates that the monetary profit of the provider should be maintained.

We define the set of VMs $Placements$ as the candidate VMs to host a group of correlated replicas ($G_g \in CG$). This set of VMs considers only the VMs within the same geographic region RG_l as the VM vm_k triggering replication and it covers (1) the VMs hosted by dc_{ij}^{l} the data center hosting vm_k (2) the VMs hosted by the best neighbor among the data centers owned by Cp_i (3) the ones offered and hosted by the best neighbor among the data centers of other cloud providers. We define the best neighbor as the data center having the minimum product of bandwidth capacity and the transfer cost to the data center hosting vm_k triggering replication. In addition, we define the set of VMs $Placements_{Feasible}$ as the set of candidate VMs for placement having enough storage space to store the set of replicas $G_g \in CG$.

To select the suitable VM $vm_p \in Placements_{Feasible}$ among the set of feasible placements, our replication strategy relies on a fuzzy inference system (FIS) [28] that is described in the next section. It takes into account four parameters, which are: (1) data transfer time ratio, (2) virtual machine load, (3) the availability of data, and (4) the monetary profit of the cloud provider. The FIS is used due to its ability to enable multi-criteria decision by allowing the consideration of the above-mentioned parameters. Hence, it allows the estimation of the potential of placing each data group ($G_g \in CG$) into a VM vm_p. This placement potential is denoted as $PlacePotential(G_g, vm_p)$. We will explain in Section 5 how the potential of placement $PlacePotential(G_g, vm_p)$ is estimated. Whereas Algorithm 4 summarizes the replicas placement process.

Algorithm 4: Fuzzy_Replicas_Placement

Input: vm_k: virtual machine triggering replication located at the l^{th} geographic RG_l, CG set of correlated data groups to replicate
Output: Replicas placement management

Select $VM_{ij}^{(Managed,l)}$ the set of managed VMs hosted by data center dc_{ij}^{l} hosting vm_k located at RG_l ;
Select $VM_{ij\prime}^{(Managed,l)}$ the set of managed VMs hosted by the best neighbor $dc_{ij\prime}^{l}$ owned by cloud provider CP_i $(j \neq j\prime)$ located at RG_l;
Select $VM_{cm}^{(Offered,l)}$ the set of offered VMs hosted by the best neighbor dc_{cm}^{l} owned by cloud provider $CP_c (i \neq c)$ located at RG_l;
$Placements \leftarrow VM_{ij}^{(Managed,l)} \cup VM_{ij\prime}^{(Managed,l)} \cup VM_{cm}^{(Offered,l)}$;
for ***each*** *correlated data group* $G_g \in CG$ **do**
    $Placements_{Feasible} \leftarrow$ Set of the VMs belonging to $Placements$ and having enough storage space to store the group of replicas G_g;
    for ***each*** *virtual machine* $vm_p \in Placements_{Feasible}$ **do**
        Estimate $PlacePotential(G_g, vm_p)$ using Fuzzy Inference System ;
    $vm_{placement} \leftarrow$ the virtual machine vm_p belonging to $Placements_{Feasible}$ and having maximum placement potential ;
    Place G_g in $vm_{placement}$;

In the following we indicate how the four replicas placement parameters namely: (1) data transfer time ratio, (2) load of the virtual machine, (3) data availability, and (4) monetary profit of cloud provider, are estimated for each VM ($vm_p \in Placements_{Feasible}$) and a data ($d \in G_g$).

These parameters are chosen in accordance with the replication strategy objectives namely preserving the monetary profit of the cloud provider and reducing the SLA violations amount in terms of response time and minimum availability. Regarding response time SLA violations, the latter are mainly related to long data transfer time and overload on virtual machines that execute the clients' tasks. To reduce the amount of response time SLA violations, replicas should be placed on a low-loaded virtual machine that serves replica requests with low data transfer time. The chosen replicas placement should meet the minimum data availability SLO. In addition, it should preserve the monetary profit of the provider.

1. ***Data transfer time ratio***
   We define a data transfer time ratio of a data d if placed in a VM vm_p to estimate the potential of a VM to serve the future requests with a low data transfer time. To this end, we divide the data transfer time of d if placed in vm_p by the longest data transfer time of d [45] as indicated by Eq. (5):

$$DTTR(d, vm_p) = \frac{DTT(d, vm_p)}{LDTT(d)} \tag{5}$$

   with d is a candidate data to replicate $d \in Gg$, $DTT(d, vm_p)$ is the data transfer time if replica d is placed in vm_p, $LDTT(d)$ is the longest data transfer time of data d. Formulas of DTT and LDTT are given hereafter:

$$DTT(d, vm_p) = \frac{\sum_{vm_r \in Req_d (r \neq p)} \left(\frac{d.Size}{Cap_{Bw}(vm_p, vm_r)} + Delay(vm_p, vm_r) \right)}{|Req_d|} \tag{6}$$

with $d.Size$ is the size of data d, $Cap_{Bw}(vm_p, vm_r)$ and $Delay(vm_p, vm_r)$ are the bandwidth capacity and the delay between VMs vm_p and a vm_r, and Req_d is the set of VMs requesting data d the most.

$$LDTT(d) = \frac{d.Size}{Min_{Bw}} + Max_{Delay} \tag{7}$$

with Min_{Bw} is the minimum bandwidth $vm_p \in Placements_{Feasible}$, and Max_{Delay} is the maximum delay between two VMs belonging to the federated system.

2. ***Virtual machine's load***

The load of a VM vm_p is estimated by combining its queuing and its processing capacities [46] as given by Eq. (8):

$$Load(vm_p) = \frac{1}{2}(QueueCap(vm_p) + ProcessCap(vm_p)) \tag{8}$$

with $QueueCap(vm_p)$, and $ProcessCap(vm_p)$ are the queuing and the processing capacities of the VM vm_p, respectively, and are computed as follows:

$$QueueCap(vm_p) = \frac{\sum_{t \in T_p^{WQ}} t.Size}{Cap_{WQ}} \tag{9}$$

with $t.Size$ is the size of task t, T_p^{WQ} is the set of tasks in the waiting queue of vm_p, and Cap_{WQ} is the waiting queue capacity of the VM vm_p.

$$ProcessCap(vm_p) = \frac{BusyMips}{Cap_{CPU}} \tag{10}$$

with $BusyMips$ is the size of the busy Mips of the processing element of vm_p, and Cap_{CPU} is the processing capacity of vm_p.

3. ***Availability***

Maintaining a minimum expected data availability can be done by maintaining a minimum number of replicas [37]. Indeed, we can rely on a minimum replica factor MRF_d of data d to ensure that the availability of data d exceeds a given minimum expected availability SLO_{MA}. In this respect, we use the equation presented in [47] to estimate the availability of data d if placed on a VM vm_p denoted as $Av(d, vm_p)$

4. ***Profit***

Besides SLOs satisfaction, the profit of cloud provider is the most important factor that should be considered during replicas placement. A cloud provider Cp_i is required to manage its resources (owned and rented) while preserving its monetary profit. We estimate the profit value of each VM vm_p managed by the cloud provider Cp_i by subtracting the operating cost from revenues related to task execution and data management [48] as indicated in Eq. (11).

$$Profit(vm_p) = Revenues(vm_p) - Expenditures(vm_p) \tag{11}$$

$$Revenues(vm_p) = N_p * RevT \tag{12}$$

with N_p is the number of total executed tasks on a VM vm_p, and $RevT$ is the provider revenue per user task execution.
The expenditures related to the management of a VM vm_p is estimated according to the cloud provider Cp_i it belongs to. Indeed, if vm_p is owned by the cloud provider Cp_i, its expenditures will cover the task processing cost, data placement cost, and penalties cost. Otherwise, if vm_p is rented by the cloud provider Cp_i from cloud provider Cp_c, its expenditures will include the price of rent besides to the cost of penalties. Eq. (13) and Eq. (14) distinguish both cases as follows:
If $vm_p \in VM_i^{Owned}$

$$Expenditures(vm_p) = TPC(vm_p) + DPC(vm_p) + Penalties(vm_p) \tag{13}$$

If $vm_p \in VM_{i\Leftarrow}^{Rented}$

$$Expenditures(vm_p) = Price_{Rent}(vm_p) + Penalties(vm_p) \tag{14}$$

with VM_i^{Owned} is the set of VMs owned by cloud provider CP_i, $VM_{i\Leftarrow}^{Rented}$ is the set of VMs rented by Cp_i from other providers. TPC is the tasks processing cost, DPC is the data placement cost, $Penalties$ is the cost of SLA violations, and $Price_{Rent}(vm_p)$ is the price of renting a VM from its cloud provider Cp_c.

$$TPC(vm_p) = \sum_{t \in T_p} \left(\frac{t.Size}{Cap_{CPU}} * Cost_{Process}\right) \tag{15}$$

tasks processing cost is estimated with T_p the set of tasks executed on vm_p, $t.Size$ is the size of task t, Cap_{CPU} is the processing capacity of VM vm_p, and $Cost_{Process}$ is the processing cost on a VM vm_p.

$$DPC(vm_p) = \sum_{d \in Gg} (d.Size * Cost_{Storage} + d.size * freq(d, vm_p) * Cost_{Transfer}(vm_{best}, vm_p)) \tag{16}$$

Data placement cost can be estimated by the sum of storage cost and transfer cost of replicas from their best sites vm_{best}. We determine the best site for getting replicas as the site vm_s holding replica and having the minimum product $Cap_{Bw}(vm_s, vm_p) * Cost_{Transfer}(vm_s, vm_p)$.

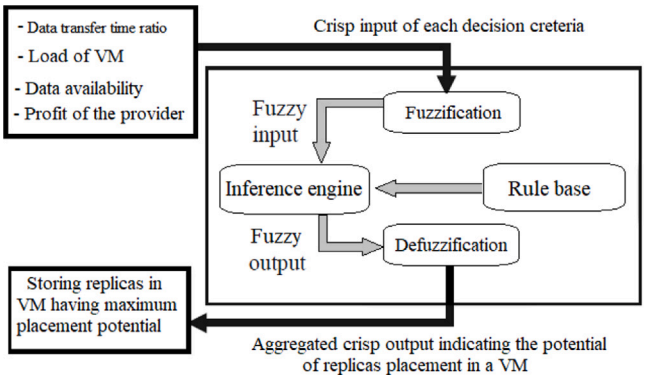


Fig. 3. Overview of the used fuzzy inference system.

G_g is the group of correlated replicas to place in vm_p. $d.size$ is the size of data d, and $freq(d, vm_p)$ is the number of requests of a data d by the executed tasks in vm_p during a replication period P. The storage cost on a VM vm_p is $Cost_{Storage}$, and $CapBw(vm_s, vm_p)$ and $Cost_{Transfer}(vm_s, vm_p)$ are respectively the bandwidth capacity and the transfer cost between vm_s holding d and vm_p.

$$Penalties(vm_p) = V_p * CPenalty \tag{17}$$

with V_p is the number SLA violations tasks on vm_p during replication period P, and $CPenalty$ is the monetary cost paid by provider for each penalty.

We remind that the values of the following parameters: $Cost_{Process}$ ($ per hour), $Cost_{Storage}$ ($ per Gb), and $Cost_{Transfer}(vm_s, vm_p)$ ($ per Gb) represent the infrastructure costs that the cloud provider deal with when making the system operational. Whereas the values of $Price_{Rent}(vm_p)$ ($), $RevT$ ($), and $CPenalty$ ($) are related to the price model of the provider. All these values are determined by the cloud provider [12,49].

4.3. Complexity analysis

Our replication strategy presented in Algorithm 1 relies on three algorithms, having each a complexity computed as follows.
The replicas number adjustment algorithm (Algorithm 2) consists on deleting unnecessary replicas from a virtual machine (VM) vm_k. Its complexity is $O(|R_{vm_k}|)$ where R_{vm_k} is the set of stored replicas on vm_k.
Regarding the replicas identification algorithm (Algorithm 3) that relies on spectral clustering algorithm, the most expensive step is the computation of the eigenvalues of the similarity matrix. This latter depends on the number of analyzed data n hence its complexity is $O(n^3)$. In our case, the number of data to be analyzed n is equal to the number of accessed data by the P violating tasks only.
The replicas placement algorithm (Algorithm 4) uses a fuzzy inference system that is invoked to decide the placement of K replicas groups among the set of candidate virtual machine for placements denoted as $PlacementsFeasible$. According to [50,51], the complexity of a fuzzy inference system is affected by the number of generated rules R and its complexity is $O(R)$. The placement potential of each replicas group is estimated for each candidate virtual machine belonging to the set $PlacementsFeasible$. Thus, the complexity of the replicas placement algorithm is $O(K \times N_{PF} \times R)$ where K is the number of generated replicas groups, N_{PF} is the number of virtual machines belonging to the set $PlacementsFeasible$, and R is the number of fuzzy rules.

5. Replicas placement fuzzy inference system

To estimate the placement potential of a virtual machine, we consider a fuzzy inference system (FIS) [28]. The FIS allows a multi-criteria decision-making by mapping uncertain variables (parameters) from the complex real world (where it is difficult to provide accurate mathematical models) to numerical data. It consists of three main steps: (1) Fuzzification of the crisp values associated to each parameter. (2) Combining the results of the set of rules that have been activated by the inference engine considering the fuzzy values of each parameter. (3) Defuzzification of the aggregated resulting fuzzy value representing the decision. The FIS has been successfully applied in several areas of resource management such as data replication, load balancing, and task scheduling [6,52,53]. It is also known for its low complexity compared to traditional solutions such as linear functions. Therefore, we use it to map the crisp values of the data transfer time ratio, load of a virtual machine (VM), data availability, and monetary profit of cloud provider parameters into a single value indicating the placement potential of a replica in a VM. Fig. 3 depicts an overview of the used fuzzy inference system.

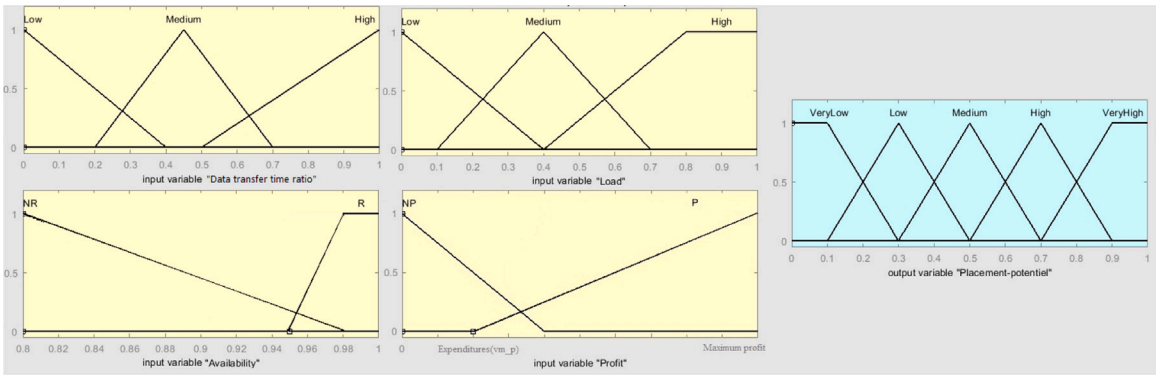


Fig. 4. The used membership functions.

Rule	Data transfer time ratio	Virtual machine's load	Data availability	Provider profit	Placement potential
1.	High	High	NR	NP	Very Low
2.	High	Medium	NR	P	Low
3.	Medium	Medium	NR	P	Medium
4.	Medium	Medium	R	P	High
5.	Low	Low	R	P	Very high

Fig. 5. Example of fuzzy rule base.

During the fuzzification and defuzzification steps, membership functions are used to quantify a fuzzy term. Each decision-making parameter is associated with a membership function indicating how its crisp values will be turned into fuzzy values (linguistic) and vice versa.

Here, we can have for both data transfer time ratio and virtual machine load the following fuzzy values Low, Medium, and High. While the data availability is associated with the values Non Respected (NR) and Respected (R). We remind that data availability is considered as respected if it exceeds the service level objective in term of minimum availability SLO_{MA} (in Fig. 4 $SLO_{MA} = 0.95$). As for the profit of cloud provider Cp_i, it is associated with the values Non Profitable (NP) and Profitable (P). We remind that a data placement is considered as profitable if the profit value exceeds the expenditures as indicated by Eq. (11). Finally, the placement potential is associated with the values: Very Low, Low, Medium, High, and Very High. More details about fuzzification are given in Appendix in order to explain how crisp values are turned into fuzzy values. The way to design fuzzy sets depends on the designer's experience and intuition [54]. We defined our membership functions using the Matlab-based fuzzy logic toolbar by running several experiments on the designed FIS in a way that enables activating most of the rules in the rule base. In this respect, our rule base contains only 36 rules. The membership functions of each input decision criteria and the output indicating the placement potential created are depicted by Fig. 4.

The inference engine evaluates the set of fuzzy rules to activate. These rules are a simple "IF-THEN" rules with a condition represented by the values of the set of the above-mentioned parameters and a conclusion indicating the potential of placing data into the virtual machine. Fig. 5 depicts a subset of our rule base.

The result obtained from the inference engine is fuzzy value resulting from the aggregation of the condition part of each activated rule. This fuzzy value is defuzzified to obtain a crisp value between 0 and 1 that represents the placement potential of a replica into the virtual machine. Among defuzzification methods, we use the centroid method since it is the most used function as it is applicable on triangular and trapezoid membership functions [6]. Hence, the placement potential of a data d in a virtual machine vm_p denoted as $PlacePotential(d, vm_p)$ is calculated using the centroid method indicated in [6]. Then, the placement potential of a group of replicas ($G_g \in CG$) in a virtual machine vm_p is estimated by the average placement potential of each data ($d \in G_g$) as indicated by Eq. (18). Finally, a decision is made to replicate and store each set of data ($G_g \in CG$) in the virtual machine that has maximum placement potential.

$$PlacePotentiel(G_g, vm_p) = \frac{\sum_{d \in G_g} PlacePotentiel(d, vm_p)}{|G_g|} \tag{18}$$

6. Performance evaluation

To validate our proposed replication strategy, we conducted numerous experiments that permit to analyze and compare, according to several evaluation metrics, the performances of our proposed strategy DCRF to six existing strategies in the literature:

Cloud provider	Price										
	CPU Price $ /10^7MI			Storage Price $/ Gb			BW Price $ /Gb				
	US	EU	AS	US	EU	AS	Intra-DC	US	EU	AS	Inter-Regions
Provider 1	0.020	0.025	0.027	0.006	0.006	0.0066	0.001	0.0015	0.002	0.004	0.008
Provider 2	0.020	0. 018	0.020	0.0096	0.008	0.0096	0.001	0,0015	0.002	0.004	0.008
Provider 3	0.0095	0.0090	0.0080	0.0120	0.0096	0.0090	0.001	0.0015	0.002	0.004	0.008

Fig. 6. Operating costs of virtual machine according to each cloud provider.

- Strategies designed for a single cloud in order to highlight the profit maintenance efficiency of the proposed strategy.
- Strategies designed for interconnected clouds in order to highlight its performance and ability to satisfy SLA.

As a single cloud based strategies, MORM replication strategy [30], RSPC [3], and CEMR [21] are considered. MORM uses a bio-inspired algorithm (artificial immune algorithm) during replication by considering data unavailability, service time, load variance, energy consumption, and average access latency as optimization objectives. While RSPC and CEMR rely on economic models with the objective to reduce SLA violations in terms of response time and preserving the profit of provider with the deference that CEMR takes advantage of correlations between data to replicate.

As for the interconnected-clouds based strategies, Xie et al. strategy [35], Mansouri et al. strategy [36] and PDRPMR [37] are considered. Similar to DCRF, Xie et al. strategy [35] takes advantage of existing correlations between user data extracted using a heuristic method. While Mansouri et al. strategy [36] focuses on creating individual replicas with the aim of reducing data access latency and minimizing the cloud provider's monetary expenses. On its side, PDRPMR considers satisfying reliability requirements and reducing storage cost by minimizing the number of replicas. However, Xie et al. strategy, Mansouri et al. strategy, and PDRPMR do not consider the provider's monetary profit.

Five performance evaluation metrics are used in the experiments:

- Average response time metric consists of the average amount of time it takes from tasks' submission to completion [55].
- Number of SLA violations
- Effective network usage (ENU) is a metric that measures the efficiency of bandwidth and network usage. A low ENU value indicates that the replication strategy has stored the replica in the proper location, allowing an efficient use of network bandwidth [55].
- Rented resources percentage is estimated by dividing the number of rented virtual machines by the total number of offered virtual machines in the federated system.
- Total monetary profit of the cloud provider value is estimated by subtracting the operating cost from revenues while the expenditures include the data storage cost, the data transfer cost, tasks execution cost, and SLA violations penalties cost in accordance with the prices indicated in Table 4 and Fig. 6.

We used the open source simulation tool CloudSim [29] after extending its multiple java classes to simulate our considered federated cloud system composed of three cloud provider distributed among three geographical regions similar to the system depicted by Fig. 1.

For simplicity, we consider the following. All cloud providers provide the same VM type where its configuration and its operating cost are inspired by the standard instance of well-known cloud providers (such as Microsoft Azure, Google Cloud, and AWS) mentioned in their web sites with respect to each geographic region. The users of each provider are charged for each task execution by an amount $RevT$ while a penalty amount $Cpenalty$ is paid to the user for each violation. In this respect, the SLO in terms of response time SLO_{RT} is set to 180 s and the value of w is set to 0.8. These values are defined based on preliminary experiments [21]. While the SLO in terms of minimum data availability SLO_{MA} is set to 0.95. This value is defined with accordance to the SLA of real clouds. The users' data of each provider are distributed randomly on its owned VM while the users' tasks are distributed using the Zipf access pattern. To investigate the impact of varying both tasks' number and data centers' number on the performance of the compared strategies, we measured the evaluation metrics while varying the number of data centers during the execution of 5000 tasks and varying the number of executed tasks on 9 data centers. Table 4 summarizes the considered simulation parameters while Fig. 6 indicates the considered operating cost for the used VM by each cloud provider considering the different regions.

6.1. Comparison of DCRF performance with some single-cloud based strategies

Fig. 7 (A) illustrates the average response time and Fig. 7 (B) illustrates the amount of SLA violations in terms of response time of DCRF alongside single cloud based data replication strategies. The values of both average response time and SLA violations amount increase as the number of users' task increases. The MORM strategy achieves the highest values of average response time and SLA violations compared to other strategies up to 4% and 29% respectively. MORM cannot handle changing workloads since it is a static strategy. It creates replicas based on data unavailability, service time, load variance, power consumption, and average latency. On

Table 4
Configuration of the simulation parameters.

Parameter	Value
Number of cloud provider	3
Number of regions	3
Number of DCs per provider	Between 2 and 5 data centers
Number of VMs within a DC	8
Number of submitted Task	Between 1000 and 10000 tasks
Task size	Between 200 and 1000 MI
Number of data	200
Data size	Between 300 Mb and 1 Gb
Inter-region BW (resp. delay)	500 Mb/s (resp. 150 ms)
Intra-region BW (resp. delay)	1 Gb/s (resp. 50 ms)
Intra-DC BW (resp. delay)	8 Gb/s (resp. 10 ms)
VM processing capability	1500 MIPS
VM number of CPU	2
VM RAM	4 Gb
VM storage capacity	8 Gb
Provider revenues per task execution ($RevT$)	0.7$
Penalty per violation ($Cpenalty$)	0.0025 $
Response time service level objective SLO_{RT}	180 s
Minimum availability service level objective SLO_{MA}	0.95
Specific parameters to our strategy	
Replication period P	32 violating tasks
$w = 0.8$ hence $Th_{RT} = (0.8 \times 180)$	
Number of K clusters to extract using the spectral clustering algorithm K	3

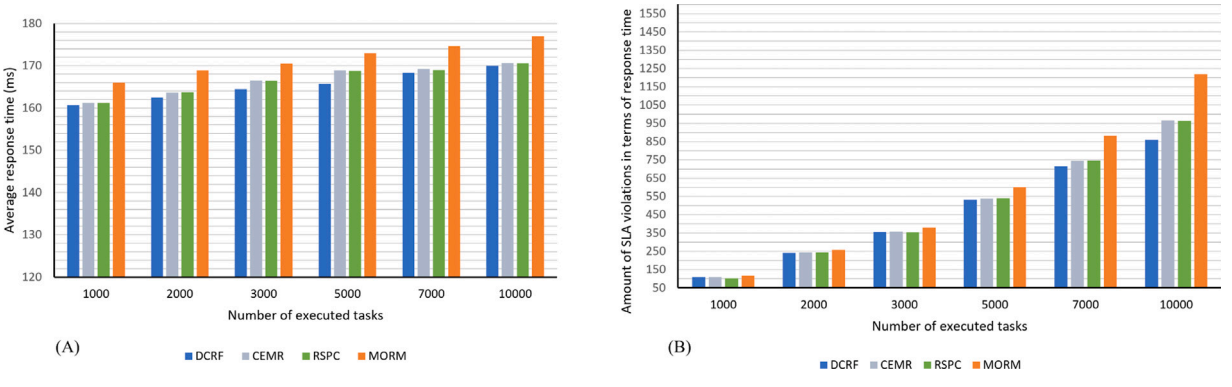


Fig. 7. (A) Average response time (in ms) (B) Amount of SLA violations in terms of response time of single cloud-based strategies.

the other side, RSPC, CEMR and DCRF are dynamic strategies that generate replicas in response to SLA violations, thus achieving lower values. DCRF performance is close to the performance of both RSPC and CEMR with a slight improvement up to 3% in terms of response time and 10% in terms of SLA violations.

Figs. 8 (A) and 8 (B) show the results obtained when measuring both the effective network usage and the average monetary profit of each provider respectively of DCRF alongside single cloud-based data replication strategies. All compared strategies consider the cost of accessing data during replication. MORM strategy considers the energy consumption while RSPC, CEMR, and DCRF use economic models for estimating the monetary profit of cloud providers. Hence, replicas are created when it is profitable for the cloud provider. RSPC places replicas in high-bandwidth VMs within the same geographic region as the VM initiating the replication. CEMR places correlated replicas in the VMs requesting the replicas most and that are related to SLA violations. DCRF places the correlated replicas in the VM having the highest placement potential within the same geographic region as the VM initiating the replication. This is performed using a fuzzy inference system that considers data transfer time, data availability, virtual machine load, and provider monetary profit.

Considering the above, the number of replicas, the number of remote data accesses, and the amount of SLA violations generated by MORM are much greater than that produced by other strategies, which increases the value of ENU and negatively affects the profit of the provider. On the contrary, the number of replicas, the number of remote data accesses, and the amount of SLA violations generated by RSPC, CEMR, and DCRF are few. Thus, the required data by the users' tasks are in most cases accessed locally, which decreases the values of ENU and preserves the monetary profit of cloud providers. Indeed, DCRF records a higher profit up to 16%, 8%, and 2% compared to MORM, RSPC, and CEMR, respectively.

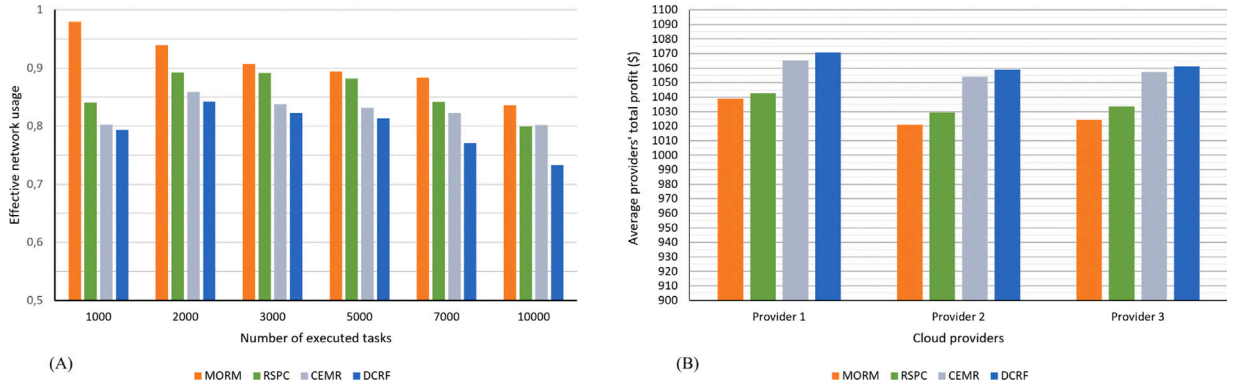


Fig. 8. (A) Effective Network Usage (B) Average total monetary profit per provider of single cloud based strategies.

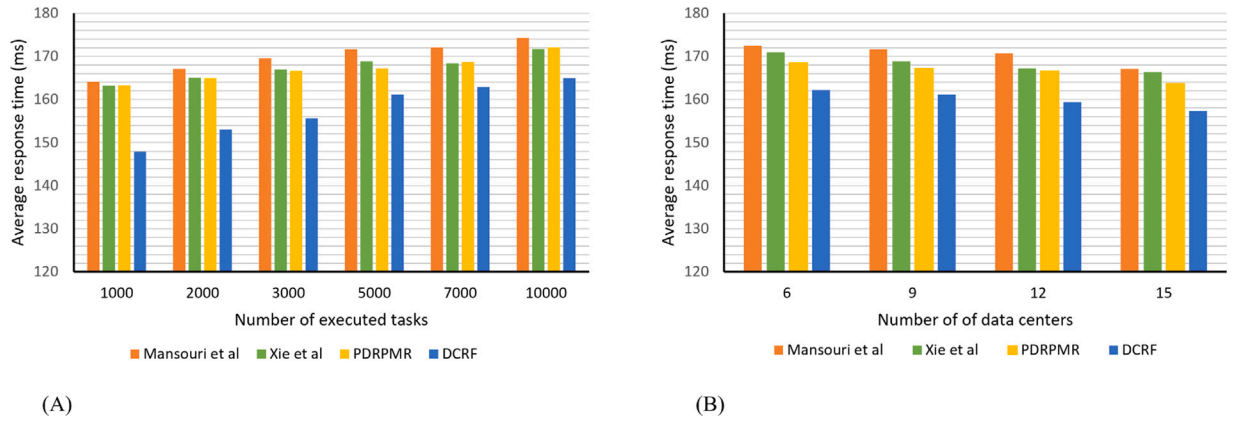


Fig. 9. (A) Average response time (in ms) while varying the tasks number (B) Average response time (in ms) while varying the data centers number.

6.2. Comparison of DCRF performance with some interconnected-clouds based strategies

6.2.1. Average response time

We analyzed the impact of both tasks number and data centers number on the tasks' average response time. Obtained results are depicted in Fig. 9. As the number of tasks increases, the response time values for the compared strategies increases too. The strategy of Mansouri et al. has the highest values compared to other strategies. It performs replication periodically for independent data unlike Xie et al. strategy and DCRF that take advantage of data correlations. As for PDRPMR, it reduces the average response time thanks to the high number of created replicas. In fact, it considers either one replica or two replicas for each requested data depending on the reliability requirements. However, DCRF generates the lowest values of response time. This performance is obtained thanks to the used clustering technique that enables extracting the groups of correlated data causing SLA violations and the used fuzzy inference systems that considers the data transfer time and the VMs' load, lowering then the average response time.

On the other side, when the number of data centers increases, the tasks response time decreases for the compared strategies. DCRF maintains the lowest values (around 4%, 5%, and 6% in average, respectively compared to PDRPMR, Xie et al. strategy and Mansouri et al. strategy). The results also indicate that when the number of data centers increases, the time difference between DCRF and the other strategies increases.

6.2.2. Number of SLA violations

Fig. 10 depicts the results of the amount of SLA violations while varying the number of executed tasks and the number of deployed data centers. The number of SLA violations of the compared strategies increases with the increase of the number of tasks. The results obtained by Xie et al. and Mansouri et al. strategies are close due to the fact that both strategies do not focus on reducing the number of violations. Instead, they focus on reducing the latency. On its side, PDRPMR focus on satisfying reliability requirements ignoring latency and response time SLA violations. On the contrary, DCRF replicates data in response to the SLA violations, and thus it achieves the lowest values. In fact, it records lower values of response time SLA violations amount by around 17%, 18%, and 21% in average, respectively compared to Xie et al. strategy, PDRPMR, and Mansouri et al. strategy.

As the number of data centers increases, the amount of SLA violations increases slightly for all the compared strategies since the users' tasks are distributed over different regions. This indeed increases the time to access data and thus enhances the occurrence of

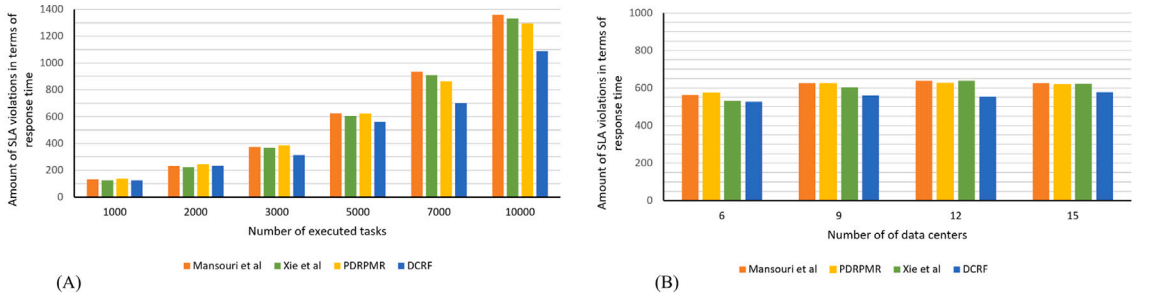


Fig. 10. Amount of SLA violations in terms of response time (A) while varying the tasks number (B) while varying the data centers number.

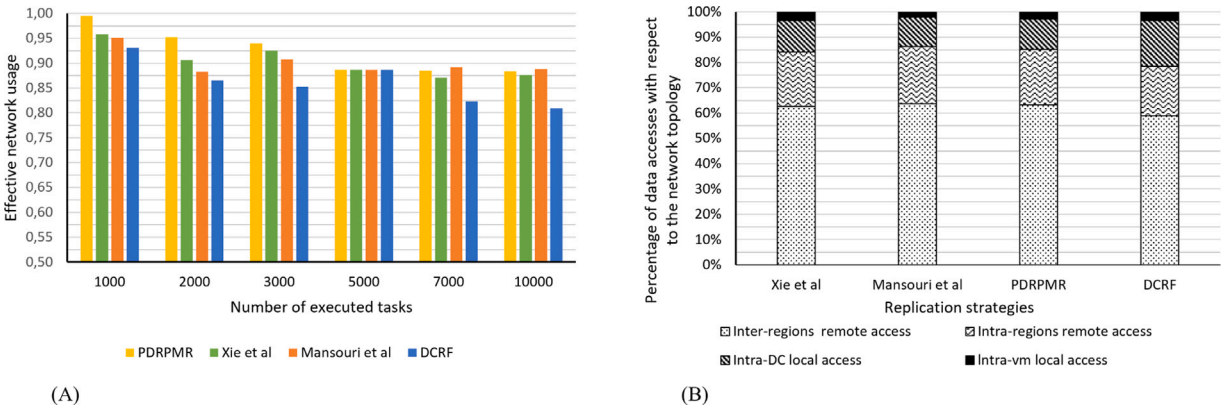


Fig. 11. (A) Effective Network Usage while varying the tasks number (B) Average percentage of data accesses with respect to the network topology during tasks execution.

SLA violations. However, the strategies maintain the same performance compared to the results obtained when varying the number of tasks.

6.2.3. Effective network usage

Replication strategies aim to reduce network traffic and bandwidth consumption. Therefore, we use the ENU metric to compare the performance of compared strategies towards bandwidth and network consumption. Fig. 11 (A) and Fig. 11 (B) illustrate the relationship between the ENU value and the number of data accesses w.r.t. network topology.

All the compared strategies yield higher values for ENU than DCRF. Both Xie et al., and Mansouri et al. place replicas based on a cost model in virtual machines that meet latency requirements not considering the network topology. When performing a small number of tasks (1000, 2000, and 3000), the strategy of Xie et al. records the highest ENU values. However, when executing a large number of tasks (5000, 7000, and 10000), the recorded values decrease because the strategy can better exploit the correlations between the data. PDRPMR records the highest values of ENU due to inter-provider replicas placement as it places the first replicas on VMs of the provider offering the cheapest cost and the second replicas on VMs of the provider offering the shortest data recovery time. This increases both the number of remote data accesses and bandwidth consumption. While DCRF records the lowest values (up to 11% compared to PDRPMR) because it relies on a fuzzy inference system for replicas placement that considers data transfer time, VM's load, data availability, and provider' monetary profit. This latter places the groups of correlated replicas either in intra-data center level or intra-region level hence the number of generated inter-region data accesses and bandwidth consumption are reduced. Fig. 11 (B) indicates how DCRF generates the highest number of local data access (by around 25%) and the lowest number of remote data accesses (by around 32%) during tasks execution compared to the other strategies.

6.2.4. Rented resource percentage

Interconnected-clouds based replication strategies resort to take advantage of the additional available resources by renting the idle offered virtual machines (VMs) among the cloud providers. Here we measure the percentage of rented VMs while varying the number of tasks and the number of data centers. Fig. 12 illustrates the obtained results.

Both strategies of Xie et al. and Mansouri et al. generate a large number of replicas. They have then to rent a large number of VMs that satisfy access latency at low replication cost from other cloud providers. When executing a small number of tasks (1000, 2000, and 3000), the Xie et al. strategy rents out fewer resources than when performing a larger number of tasks since the number of

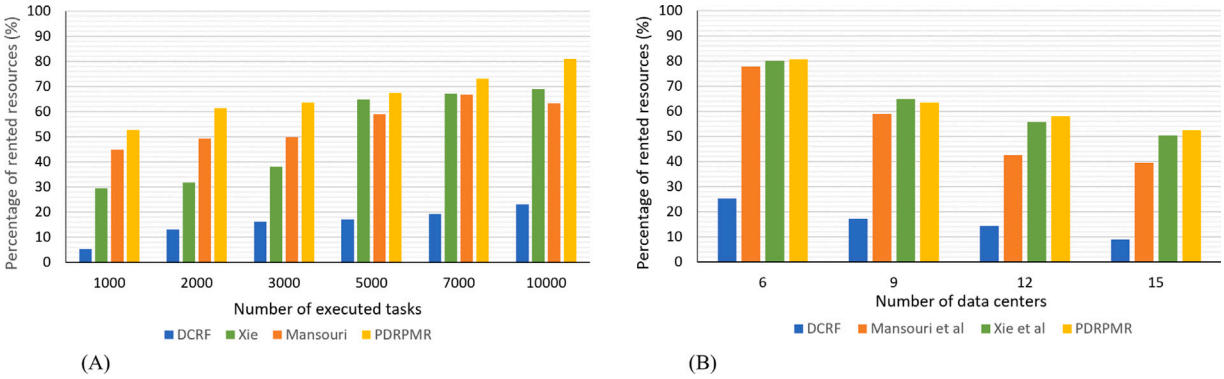


Fig. 12. Percentage of rented resources (A) while varying the tasks number (B) while varying the data centers number.

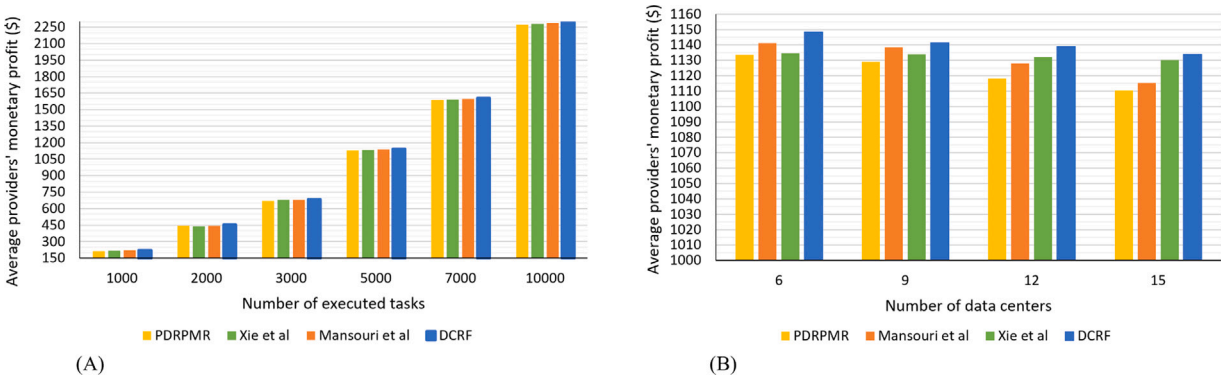


Fig. 13. Average total monetary profit of providers (A) while varying the tasks number (B) while varying the data centers number.

replicas increases and the need to store them increases. PDRPMR records the highest values when varying both the number of tasks and the number of data centers. This is because it produces the largest number of replicas (at least one replicas for each requested data). It stores replicas on the VMs of other cloud providers to maintain reliability requirements. While DCRF records the least value (up to 81%, and 88%, and 89% respectively compared to Xie et al. strategy, Mansouri et al. strategy, and PDRPMR) when varying both the number of tasks and the number of data centers. In addition, it relies on a fuzzy inference system to choose the proper replicas placement among owned and offered resources. This latter focuses during this selection on the offered resources by the nearest provider only. Moreover, DCRF performs a replicas number adjustment when SLA is satisfied over time. This process enables the removal of unnecessary replicas that are not often accessed, reducing then the need for additional resources.

6.2.5. Total monetary profit of a provider

Maintaining the monetary profit of cloud providers is of great importance. Fig. 13 illustrates the average monetary profit of cloud providers of the federated cloud while Fig. 14 illustrates the average monetary profit of each provider while varying the number of tasks and data centers.

Fig. 13 indicates that the providers' profit increases with increasing number of tasks and decreases with increasing number of data centers. The performance of the compared strategies is fairly similar, because both Xie et al. and Mansouri et al. strategies rely on a cost model to reduce the monetary cost of replication. PDRPMR records the least monetary profit values since it generates high amount of replicas, remote data access, and SLA violations. This increases the cost of data storage, the cost of data transfer, and the cost of penalties. DCRF relies on an economic model to maintain the providers' monetary profit not only reducing the replication cost. Indeed, DCRF succeeded in achieving the highest values of monetary profit. This appears more when varying the number of data centers.

Fig. 14 depicts the average monetary profit per each provider when varying the number of tasks and the number of data centers. The results of Xie et al. and Mansouri et al. are close. While PDRPMR continues to record the lowest values, DCRF maintains the highest values of monetary profit per each provider despite the difference between their cost models.

7. Conclusions and future directions

We proposed a dynamic data replication strategy DCRF (Data Correlation and fuzzy inference system-based data Replication in Federated cloud systems) in order to preserve the monetary profit of a cloud provider belonging to a federated cloud system

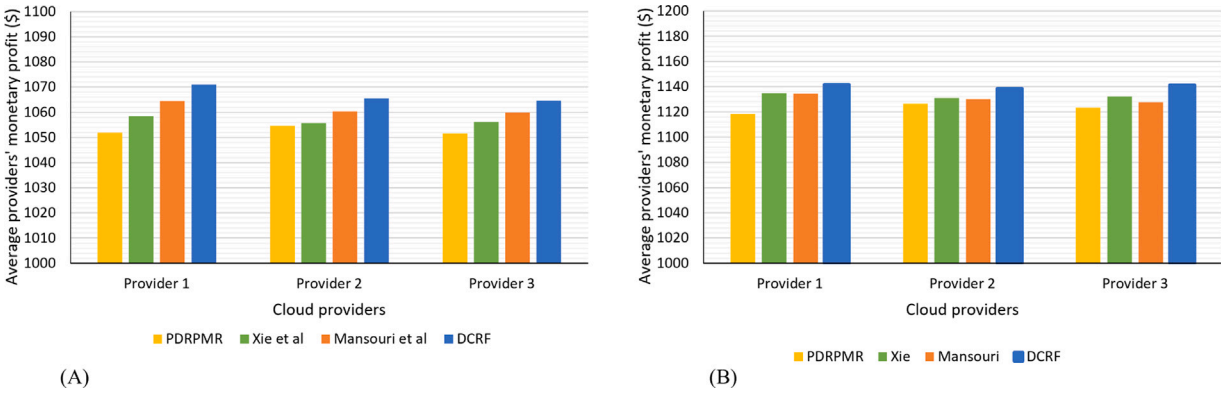


Fig. 14. Average total monetary profit per provider (A) while varying the tasks number (B) while varying the data centers number.

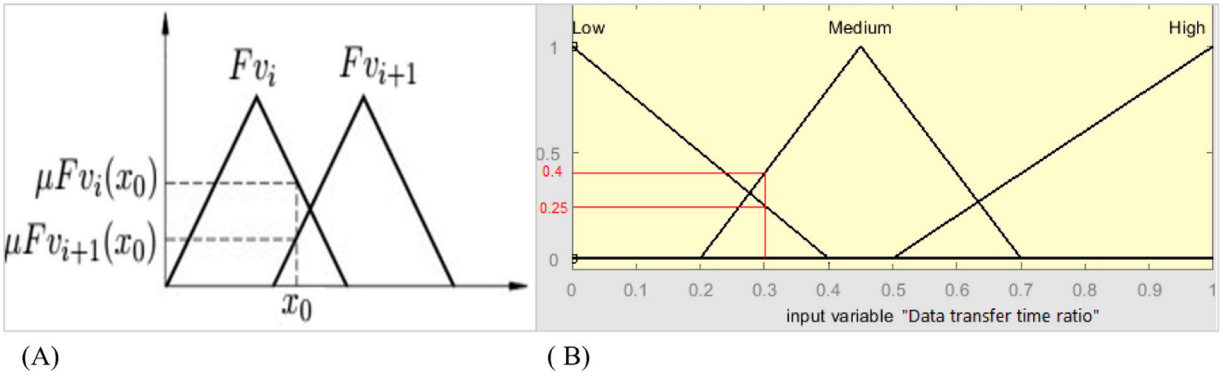


Fig. A.15. Illustrative example of the fuzzification step (A) theoretical example (B) practical example on data transfer ratio parameter.

while satisfying QoS for the users. Dealing with the Service Level Objectives (SLOs) to be satisfied by the provider, we focused on response time and minimum availability. DCRF is triggered periodically as a response to SLA violations. It takes advantage of the data correlations that exist between data accessed frequently by tasks causing SLA violations. Therefore, the spectral clustering technique is used to extracts them. Then a Fuzzy Inference System (FIS) is used to estimates the potential of placing the groups of correlated replicas among the rented and owned resources of the provider. The FIS considers four important parameters: (1) data transfer time ratio, (2) virtual machine load, (3) data availability, and (4) cloud provider profit. Moreover, a replicas number adjustment is performed when SLA is satisfied over time, which enables the removal of unnecessary replicas from the system considering the satisfaction of a minimum data availability. Simulation results with CloudSim [29] indicates that DCRF has better performance in comparison with both single cloud based strategies and interconnected-clouds based strategies in terms of the average response time, number of SLA violations, effective network usage, and the total monetary profit of cloud providers. DCRF decreases the amount of SLA violations (up to 21%) while preserving the monetary profit of providers despite the difference between their cost models.

As part of our future work, we plan to explore the following issues: (1) using probabilistic models and machine learning techniques to predict the future SLA violations based on the users' demand and the workload conditions making DCRF more proactive, (2) considering additional users' perspectives such as limited budget, (3) considering data consistency, and (4) performing evaluations on a real cloud environment.

Appendix. Illustrative example of the fuzzification step

The fuzzification phase consists of translating the crisp value of a parameter to fuzzy values using the membership functions. Let x_0 be the read crisp value of a parameter having two fuzzy values Fv_i and Fv_{i+1}. According to Fig. A.15 (A), this reading corresponds to two constant values $\mu Fv_i(x_0)$ and $\mu Fv_{i+1}(x_0)$ called fuzzy inputs. They can be interpreted as the truth values of x_0 related to Fv_i and to Fv_{i+1}, respectively.

Let take our membership function for the parameter "data transfer time ratio" having three fuzzy values Low, Medium, and High. As illustrated in Fig. A.15 (B), a crisp value of data transfer ratio value $x_0 = 0.3$ corresponds to the fuzzy inputs 0.25 Low and 0.4 Medium. These coefficients are used by the fuzzy engine to estimate the degree of activation of the rules. More details about fuzzification cloud be found in [28].

References

[1] A. Kumar, S. Bawa, A comparative review of meta-heuristic approaches to optimize the SLA violation costs for dynamic execution of cloud services, Soft Comput. 24 (6) (2020) 3909–3922.
[2] L.A. Barroso, U. Hölzle, P. Ranganathan, The datacenter as a computer: Designing warehouse-scale machines, Synth. Lect. Comput. Archit. 13 (3) (2018) i–189.
[3] R. Mokadem, A. Hameurlain, A data replication strategy with tenant performance and provider economic profit guarantees in Cloud data centers, J. Syst. Softw. 159 (2020) 110447.
[4] N. Mansouri, M.M. Javidi, A review of data replication based on meta-heuristics approach in cloud computing and data grid, Soft Comput. (2020) 1–28.
[5] S. Slimani, T. Hamrouni, F. Ben Charrada, Service-oriented replication strategies for improving quality-of-service in cloud computing: a survey, Cluster Comput. (2020) 1–32.
[6] N. Mansouri, B.M.H. Zade, M.M. Javidi, A multi-objective optimized replication using fuzzy based self-defense algorithm for cloud computing, J. Netw. Comput. Appl. 171 (2020) 02811.
[7] U. Tos, R. Mokadem, A. Hameurlain, T. Ayav, Achieving query performance in the cloud via a cost-effective data replication strategy, Soft Comput. 25 (7) (2021) 5437–5454.
[8] M. Séguéla, R. Mokadem, J.-M. Pierson, Comparing energy-aware vs. cost-aware data replication strategy, in: International Green and Sustainable Computing Conference (IGSC), 2019, pp. 1–8.
[9] J. Hong, T. Dreibholz, J.A. Schenkel, J.A. Hu, An overview of multi-cloud computing, in: Workshops of the International Conference on Advanced Information Networking and Applications, Springer, 2019, pp. 1055–1068.
[10] R. Buyya, S.N. Srirama, G. Casale, R. Calheiros, Y. Simmhan, B. Varghese, E. Gelenbe, B. Javadi, L.M. Vaquero, M.A. Netto, et al., A manifesto for future generation cloud computing: Research directions for the next decade, ACM Comput. Surv. 51 (5) (2018) 1–38.
[11] M. Masdari1, M. Zangakani, Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities, J. Supercomput. 76 (1) (2020) 499–535.
[12] M.R.M. Assis, L.F. Bittencourt, MultiCloud tournament: a cloud federation approach to prevent free-riders by encouraging resource sharing, J. Netw. Comput. Appl. 166 (2020) 102694.
[13] H. Abu-Libdeh, L. Princehouse, H. Weatherspoon, RACS: a case for cloud storage diversity, in: Proceedings of the 1st ACM Symposium on Cloud Computing, 2010, pp. 229–240.
[14] A. Bessani, M. Correia, B. Quaresma, F. André, P. Sousa, DepSky: dependable and secure storage in a cloud-of-clouds, ACM Trans. Storage (TOS) 9 (4) (2013) 1–33.
[15] A. Abouzamazem, P. Ezhilchelvan, Efficient inter-cloud replication for high-availability services, in: 2013 IEEE International Conference on Cloud Engineering (IC2E), IEEE, 2013, pp. 132–139.
[16] C. Li1, J. Zhang, H. Tang, Replica-aware task scheduling and load balanced cache placement for delay reduction in multi-cloud environment, J. Supercomput. 75 (5) (2019) 2805–2836.
[17] N. Grozev, R. Buyya, Inter-cloud architectures and application brokering: taxonomy and survey, Softw. - Pract. Exp. 44 (3) (2014) 369–390.
[18] B. Pang, Y. Yang, F. Hao, A sustainable strategy for multi-cloud service composition based on formal concept analysis, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019, pp. 2659–2665.
[19] N. Mansouri, M.M. Javidi, B.M.H. Zade, Using data mining techniques to improve replica management in cloud environment, Soft Comput. (2019) 1–26.
[20] A. Khelifa, T. Hamrouni, R. Mokadem, F.B. Charrada, Cloud provider profit-aware and triadic concept analysis-based data replication strategy for tenant performance improvement, Int. J. High Performance Comput. Netw. 16 (2–3) (2020) 67–86.
[21] A. Khelifa, T. Hamrouni, R. Mokadem, F.B. Charrada, Combining task scheduling and data replication for SLA compliance and enhancement of provider profit in clouds, Appl. Intell. 51 (10) (2021) 7494–7516.
[22] M. Chellouf, T. Hamrouni, Popularity and correlation aware data replication strategy based on half-life concept and clustering in cloud system, Concurr. Comput.: Pract. Exper. 33 (10) (2021).
[23] Q. Zhao, C. Xiong, C. Yu, C. Zhang, X. Zhao, A new energy-aware task scheduling method for data-intensive applications in the cloud, J. Netw. Comput. Appl. 59 (2016) 14–27.
[24] T. Shi, H. Ma, G. Chen, S. Hartmann, Location-aware and budget-constrained application replication and deployment in multi-cloud environment, in: 2020 IEEE International Conference on Web Services (ICWS), 2020, pp. 110–117.
[25] K. Liu, J. Peng, J. Wang, W. Liu, Z. Huang, J. Pan, Scalable and adaptive data replica placement for geo-distributed cloud storages, IEEE Trans. Parallel Distrib. Syst. 31 (7) (2020) 1575–1587.
[26] Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, H.V. Madhyastha, Spanstore: Cost-effective geo-replicated storage spanning multiple cloud services, in: Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles, Association for Computing Machinery, 2013, pp. 292–308.
[27] J. Liu, J. Han, C. Aggarwal, C. Reddy, Spectral clustering, 2013.
[28] R.R. Yager, L.A. Zadeh, An Introduction To Fuzzy Logic Applications in Intelligent Systems, Vol. 165, Springer Science & Business Media, 2012.
[29] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. - Pract. Exp. 41 (1) (2011) 23–50.
[30] S.-Q. Long, Y.-L. Zhao, W. Chen, MORM: a multi-objective optimized replication management strategy for cloud storage cluster, J. Syst. Archit. 60 (2) (2014) 234–244.
[31] K. Oh, A. Chandra, J. Weissman, TripS: Automated multi-tiered data placement in a geo-distributed cloud environment, in: Proceedings of the 10th ACM International Systems and Storage Conference, 2017, pp. 1–11.
[32] G. Liu, H. Shen, Minimum-cost cloud storage service across multiple cloud providers, IEEE/ACM Trans. Netw. 25 (4) (2017) 2498–2513.
[33] P. Wang, C. Zhao, Y. Wei, D. Wang, Z. Zhang, An adaptive data placement architecture in multicloud environments, Sci. Progr., 2020.
[34] L. Jiao, J. Lit, W. Du, X. Fu, Multi-objective data placement for multi-cloud socially aware services, in: IEEE INFOCOM 2014-IEEE Conference on Computer Communications, IEEE, 2014, pp. 28–36.
[35] F. Xie, J. Yan, J. Shen, A data dependency and access threshold based replication strategy for multi-cloud workflow applications, in: International Conference on Service-Oriented Computing, Springer, 2018, pp. 281–293.
[36] Y. Mansouri, R. Buyya, Dynamic replication and migration of data objects with hot-spot and cold-spot statuses across storage data centers, J. Parallel Distrib. Comput. 126 (2019) 121–133.
[37] M.M. Alshammari, A.A. Alwan, A. Nordin, A.Z. Abualkishik, Data backup and recovery with a minimum replica plan in a multi-cloud environment, Int. J. Grid High Performance Comput. (IJGHPC) 12 (2) (2020) 102–120.
[38] T.-Y. Hsu, A.D. Kshemkalyani, A proactive, cost-aware, optimized data replication strategy in geo-distributed cloud datastores, in: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing, 2019, pp. 143–153.
[39] H. Khalajzadeh, D. Yuan, B.B. Zhou, J. Grundy, Y. Yang, Cost effective dynamic data placement for efficient access of social networks, J. Parallel Distrib. Comput. (2020).

[40] R. Buyya, R. Ranjan, R.N. Calheiros, InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services, in: International Conference on Algorithms and Architectures for Parallel Processing, Springer, 2010, pp. 13–31.
[41] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.
[42] R. Jin, C. Kou, R. Liu, Y. Li, Efficient parallel spectral clustering algorithm design for large data sets under cloud computing environment, J. Cloud Comput.: Adv. Syst. Appl. 2 (1) (2013) 1–10.
[43] A. Atrey, G. Van Seghbroeck, H. Mora, F. De Turck, B. Volckaert, Spech: a scalable framework for data placement of data-intensive services in geo-distributed clouds, J. Netw. Comput. Appl. 142 (2019) 1–14.
[44] K.A. Kumar, A. Quamar, A. Deshpande, S. Khuller, SWORD: workload-aware data placement and replica selection for cloud data management systems, VLDB J. 23 (6) (2014) 845–870.
[45] S. Tuli, R. Sandhu, R. Buyya, Shared data-aware dynamic resource provisioning and task scheduling for data intensive applications on hybrid clouds using Aneka, Future Gener. Comput. Syst. 106 (2020) 595–606.
[46] B. Kruekaew, W. Kimpan, Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing, Int. J. Comput. Intell. Syst. 13 (1) (2020) 496–510.
[47] M. Hussein, M. Mousa, A light-weight data replication for cloud DataCenters environment, Int. J. Innov. Res. Comput. Commun. Eng. 2 (2014) 2392–2400.
[48] R. Mahmud, S.N. Srirama, K. Ramamohanarao, R. Buyya, Profit-aware application placement for integrated fog–cloud computing environments, J. Parallel Distrib. Comput. 135 (2020) 177–190.
[49] A.N. Toosi, R.N. Calheiros, R.K. Thulasiram, R. Buyya, Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment, in: 2011 IEEE International Conference on High Performance Computing and Communications, IEEE, 2011, pp. 279–287.
[50] Y.H. Kim, S.C. Ahn, W.H. Kwon, Computational complexity of general fuzzy logic control and its simplification for a loop controller, Fuzzy Sets and Systems 111 (2) (2000) 215–224.
[51] J. Miliauskaitė, D. Kalibatiene, Complexity issues in data-driven fuzzy inference systems: systematic literature review, in: International Baltic Conference on Databases and Information Systems, Springer, 2020, pp. 190–204.
[52] E. Iranpour, S. Sharifian, A distributed load balancing and admission control algorithm based on fuzzy type-2 and game theory for large-scale SaaS cloud architectures, Future Gener. Comput. Syst. 86 (2018) 81–98.
[53] X. Zhou, G. Zhang, J. Sun, J. Zhou, T. Wei, S. Hu, Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT, Future Gener. Comput. Syst. 93 (2019) 278–289.
[54] P. Zhang, Chapter 7 - industrial intelligent controllers, in: Advanced Industrial Control Technology, William Andrew Publishing, 2010, pp. 257–305.
[55] T. Hamrouni, S. Slimani, F. Ben Charrada, A data mining correlated patterns-based periodic decentralized replication strategy for data grids, J. Syst. Softw. 110 (2015) 10–27.