A review on data replication strategies in cloud systems

Riad Mokadem*

Institut de Recherche en Informatique de Toulouse (IRIT),
Paul Sabatier University,
Toulouse, France
Email: mokadem@irit.fr
*Corresponding author

Jorge Martinez-Gil

Software Competence Centre Hagenberg,
Softwarepark 21,
Hagenberg, Austria
Email: jorge.martinez-gil@scch.at

Abdelkader Hameurlain

Institut de Recherche en Informatique de Toulouse (IRIT),
Paul Sabatier University,
Toulouse, France
Email: hameurlain@irit.fr

Josef Kueng

Institute for Applied Knowledge Processing,
Johannes Kepler University,
Linz, Austria
Email: Josef.Kueng@jku.at

Abstract: Data replication constitutes an important issue in Cloud data management. In this context, a significant number of replication strategies have been proposed for Cloud systems. Most of the studies in the literature have classified these strategies into static vs. dynamic or centralised vs. decentralised strategies. In this paper, we propose a new classification of data replication strategies in Cloud systems. It takes into account several other criteria, specific to Cloud environments: (i) the orientation of the profit, (ii) the considered objective function, (iii) the number of tenant objectives, (iv) the nature of the Cloud environment and (v) the consideration of economic costs. Dealing with the last criterion, we focus on the provider's economic profit and the consideration of energy consumption by the provider. Finally, the impact of some important factors is investigated in a simulation study.

Keywords: cloud systems; data replication; data replication strategies; classification; SLA; economic profit; performance.

Reference to this paper should be made as follows: Mokadem, R., Martinez-Gil, J., Hameurlain, A. and Kueng, J. (2022) 'A review on data replication strategies in cloud systems', *Int. J. Grid and Utility Computing*, Vol. 13, No. 4, pp.347–362.

Biographical notes: Riad Mokadem is currently an Associate Professor in Computer Science at Paul Sabatier University, Toulouse, France, and a Member of the IRIT laboratory. His current research interests include query optimisation in large-scale distributed environments, data replication and database performance. He is involved in several International conferences and journals.

Jorge Martinez-Gil is currently a Senior Researcher position within the area of Data Analysis Systems from the Software Competence Centre Hagenberg (SCCH) in Austria. He is involved in several applied and fundamental research projects related to knowledge-based technologies.

Abdelkader Hameurlain is a Full Professor in Computer Science at Paul Sabatier University (IRIT Lab.) Toulouse, France. His current research interests include query optimisation in parallel and large scale distributed environments, and mobile databases. He has been Guest Editor of several international journals. He is Co-editor in Chief of the International Journal *'Transactions on Large-Scale Data- and Knowledge-Centred Systems'* (LNCS, Springer).

Josef Kueng is a Full Professor in the Institute for Application Oriented Knowledge Processing at Johannes Kepler University (JKU) Linz, Austria. His core competencies in research and development cover information systems, knowledge-based systems and knowledge processing.

1 Introduction

Data replication is a well-known technique that aims to increase data availability, reduce bandwidth consumption and achieve fault-tolerance. It has been commonly used in: (i) Database Management Systems (DBMS) (Pérez et al., 2010; Tos et al., 2021), (ii) parallel and distributed systems (Benoit and Rehn-Sonigo, 2008), (iii) mobile systems (Guerrero-Contreras et al., 2015) and (vi) large scale systems, including P2P (Spaho et al., 2015) and data Grid systems (Tos et al., 2015). However, data replication strategies proposed for these systems are difficult to adapt to cloud systems. They aim to obtain the best performance from the system by creating as many replicas as possible without taking into account the economic cost of replication (Mokadem and Hameurlain, 2020). Creating as many replicas as possible in cloud systems is not realistic. This cannot be economically feasible as it can lead to unnecessary use of resources and reduced profits for the provider. Indeed, a Cloud provider aims to generate a profit like any economic enterprise in addition to ensuring a certain Quality of Service (QoS) through meeting the tenant's requirements.

On the other hand, existing data replication strategies in the literature have focused on the performance of read-only queries (Mansouri and Buyya, 2019; Tos et al., 2016) as well as the management of updates (Hsu et al., 2018). However, data replication can be costly and performance can be degraded when the data is updated frequently. In fact, the advantages of replication can be neutralised by the overhead associated with maintaining consistency between several replicas. Throughout this paper, we only focus on data replication strategies proposed for Online Analytical Processing (OLAP) applications, i.e., the consistency management is not the focus of this work.

In the literature, a certain number of synthesis works have been interested in the enumeration and classification of the main data replication strategies in Cloud systems (Gilland Singh, 2016; Gopinath and Sherly, 2018; Milani and Navimipour, 2016; Mansouri and Javidi, 2019). Most of them have classified these strategies as: (i) static vs. dynamic strategies; the number of replicas and placement nodes are predetermined during the design phase in static strategies (Begum and Sirisha, 2019; Long et al., 2014) whereas replicas of each object are created, placed and managed dynamically when the system is already operational in dynamic strategies (Mansouri and Javidi, 2018; Tos et al., 2018), or (ii) centralised (Huang et al., 2014; Zhang et al., 2018) vs. decentralised (Mansouri and Buyya, 2019; Wei et al., 2010) replication strategies, depending on the mechanism for controlling the creation of replicas. Furthermore, these works often did not provide any performance evaluation.

In this paper, we classify data replication strategies in Cloud systems according to several other criteria, specific to Cloud environments:

1) *The orientation of the profit*: A replication strategy aims either to reduce the monetary costs paid by tenants to the provider or to increase the profit of the provider. Thus, some strategies are considered as tenant-oriented strategies (Limam et al., 2019; Sakr and Liu, 2012) when most of the proposed strategies in the literature are considered as provider-oriented strategies (Liu et al., 2018; Sousa and Machado, 2012). To the best of our knowledge, only a few studies (Tabet et al., 2017) take (partially) into account this criterion when classifying data replication strategies in Cloud systems.

2) *The objective function for which a data replication strategy is designed*: Considering the fact that a data replication strategy aims to maximise/minimise some objective, it is possible to propose a classification with regard to the considered objective function. In this context, we distinguish data replication strategies that aim to: (a) improve data locality (Lee et al., 2015), (b) improve the network bandwidth locality (Mokadem and Hameurlain, 2020), (c) reduce the cost of replication based on cost models (Pu et al., 2015) or economic behaviours such as auctions (Zhang et al., 2014).

3) *The number of tenant objectives that a replication strategy aims to satisfy*: Data replication strategies can be classified as: single-objective vs. multi-objective strategies. Most of the existing strategies aim to satisfy a single tenant objective such as availability (Sun et al., 2012), energy consumption (Xu et al., 2015), reliability (Bui et al., 2016), performance (Mansouri and Buyya, 2019; Vulimiri et al., 2015) and fault tolerance (Li et al., 2019). On the other hand, some strategies aim to simultaneously meet several tenant objectives. The design of such strategies implies taking into account certain compromises since satisfying one objective can often conflict with another objective. Examples of these satisfied objectives are latency and reliability in Hassan et al. (2009), data availability and load balancing in Edwin et al. (2019) and, availability, response time, network latency, energy consumption and load balancing in Long et al. (2014).

4) *The nature of the cloud environment for which a data replication strategy is designed*: Most of the existing strategies in the literature have been proposed for a single cloud provider (Mokadem and Hameurlain, 2020). Most of the time, they investigated the problem of placing replicas or finding the optimal number of replicas in order to optimise costs while meeting QoS for tenants. On the other hand, only some replication strategies were deployed on several cloud providers (Abu-Libdeh et al., 2010; Mansouri and Buyya, 2019; Wu et al., 2013; Khelifa et al., 2022). In this context, most of the proposed strategies take advantage of price differences of different resources between cloud providers when deciding to replicate in order to satisfy the Service Level Agreement (SLA).

5) *Taking into account the economic costs of data replication*: A significant part of the existing replication strategies in Cloud systems neglects the economic costs of data replication. Most of them aim to satisfy the tenant's objectives while only reducing the cost of replication, e. g., storage and data transfer. Gill and Singh (2016) and Liu et al. (2013) mentioned cost-aware although the considered cost of replication is not necessarily an economic cost. On the other hand, only a few strategies take into account the economic cost of replication by modelling the replication cost, the provider profit, the energy consumption or penalty costs as monetary costs. In this paper, we focus on replication strategies that consider the economic profit of the provider (Mansouri and Buyya, 2019; Mokadem and Hameurlain, 2020; Wu et al., 2013) and the energy consumption of the provider (Alghamdi et al., 2017; Boru et al., 2015; Long et al., 2014; Seguela et al., 2021).

Obviously, the proposed classification can cause some overlap, i.e., some strategies (Mokadem and Hameurlain, 2020; Wu et al., 2013) may be cited in different classes. We also study the impact of some important factors, e.g., bandwidth consumption, cloud topology, user's access pattern and the number of tenants, on the performance of these strategies. In the performance evaluation section, we provide feature comparison of some strategies by measuring some important metrics such as the average response time, the average replica factor and the number of SLA violations. We measure the impact of some factors, e.g., the arrival rate of queries, on the performance and resource consumption of these strategies. The simulation analysis proves that strategies which take into account the compromise between the tenant's objectives and the economic provider profit are more realistic in Cloud systems.

The rest of this paper is organised as follows: Section 2 introduces some specifications of replication strategies in cloud systems and the existing classifications of these strategies. Section 3 presents our proposed replication strategy classification. Section 4 points out some important factors when data replication strategies achieve performance. Section 5 deals with a simulation analysis that measures the impact of some factors on performance. Finally, Section 6 contains conclusion and future work.

2 Data replication in cloud systems: state of the art

An elastic resource management is critical to minimise operating cost while ensuring performance during high loads (Hameurlain and Mokadem, 2017). For this aim, scaling up/down allows adding/removing resources when the workload increases/decreases beyond a given threshold (Hwang et al., 2016). In contrast, scaling out/in (that we consider here through data replication), adds/removes VMs in order to satisfy the tenant requirements. Hwang et al. (2016) affirmed that scaling out has lower over-provisioning of resources than the scaling up. In this context, the provider and its tenants agree on a QoS via an SLA contract. Mainly, an SLA includes: (a) one or several Service Level Objectives (SLOs), (b) a validity period, (c) a Billing Period (BP) during which the provider rents services to its tenants, (d) an agreed monetary amount paid by the tenant to the provider for the processing of its queries during a BP and (e) an agreed monetary penalty amount paid by the provider to its tenant in case of breach of the SLA (Sousa and Machado, 2012).

Replicating data in all nodes is not realistic because of the storage and bandwidth constraints. In consequence, a replication strategy is required. Although many data replication strategies were proposed for classical systems, e.g., data grid systems, they are not suitable for cloud systems since the economic aspects are not taken into account. Creating as many replicas as possible in the Cloud may not be economically feasible. As a result, replication strategies in cloud systems must ensure certain QoS to tenants (without aiming to have the best QoS) while taking into account the economic profitability for the provider. Thus, a strategy for replicating data in cloud systems must not only address the classic data replication issues: (i) *when* should replication take place? (ii) *what* data should be replicated? (iii) *how* many replicas must be created? (iv) *where* to place these replicas? and (v) *which* replica is selected?, but also, it must take into account *economic aspects* such as the cost of replication and the provider profit.

In the literature, a certain number of synthesis works have been interested in the enumeration and classification of the main data replication strategies in Cloud systems (Gill and Singh, 2016; Milani and Navimipour, 2016; Tabet et al., 2017; Gopinath and Sherly, 2018). Most of them classified these strategies as: (i) static vs. dynamic or (ii) centralised vs. decentralised strategies.

2.1 Static vs. dynamic strategies

In a static data replication strategy (Begum and Sirisha, 2019; Ghemawat et al., 2003; Hassan et al., 2009; Long et al., 2014; Zeng and Verravelli, 2014), the number of replicas for a data set is determined in advance during the design phase. Deterministic policies are applied to decide in advance the location of each replica and the cost associated with replication is directly proportional to the number of active replicas (Liu et al., 2013). Ghemawat et al. (2003) proposed a data replication algorithm in GFS (Google File System) offering a reduced response time and high availability. The limitation of this algorithm is that a fixed number (3) of replicas is used for all files. This generates high consumption of resources, particularly in terms of storage and energy consumption (Long et al., 2014). MORM (Long et al., 2014) is another static replication strategy. It is based on an offline algorithm that aims to improve data availability, response time and network latency. Access statistics are used and the replication scheme is established in advance for a long period depending on the initial capacity of the system. In the MOE replication strategy (Hassan et al., 2009), a scalable method reduces storage and latency as well as improves data reliability. This solution explores a research space in which a potentially good solution can be found. The strategy proposed in Zeng and Verravelli (2014) manages the replication of metadata in order to minimise the average response time. An optimal load balancing technique is considered for large-scale Data Centres (DC).

In a dynamic replication (Edwin et al., 2019; Kumar et al., 2014; Lazeb et al., 2019; Mansouri et al., 2017; Tabet et al., 2019; Tos et al., 2016; Wei et al., 2010), replicas of each object are created, placed and managed dynamically when the system is already operational. It is done according to the user's access modes and the availability of resources, e.g., storage and CPU. In the CDRM strategy (Wei et al., 2010), a replica is placed on the node with the lowest blocking probability in order to reduce the data access overhead. A blocking probability is calculated on each Virtual Machine (VM) and the overloaded VMs are blocked for the reception of new queries. This improves load balancing between nodes. However, the satisfaction of the SLA is not taken as a decision criterion. Sousa and Machado (2012) proposed RepliC, a database replication strategy in a multi-tenant environment. It dynamically adjusts the number of replicas by monitoring the system usage. RepliC manages the workload change by directing the queries to replicas with sufficient resources or by creating new replicas. In the PEPR strategy (Tos et al., 2018), a replication is considered only if: (i) the response time of a query, estimated before its execution, is greater than a threshold response time established in the SLA. Furthermore, (ii) the replication must be profitable for the provider. For this aim, the provider's expenses and revenues are also estimated. However, replication is done by query, which generates a high replication cost. The DCR2S strategy (Gill and Singh, 2016) aims to create a replica for data whose popularity exceeds a certain threshold. Based on the concept of knapsack, replicas are re-replicated from more expensive DCs to cheaper DCs in order to reduce the cost of replication. Finally, the DPRS strategy (Mansouri et al., 2017) replicates only a small part of the frequently requested data on the best locations according to the number of users' interests and the available storage space.

In general, static strategies are simple to implement. The choice of a static strategy depends in particular on the stability of the user's access mode, the storage capacity of nodes and the available bandwidth. The static creation of a maximum number of replicas can guarantee the required performance at the cost of a high operating cost (Liu et al., 2013). These strategies are suitable for applications that predetermine user demands. However, user access to data varies widely in Cloud systems in addition to a very heterogeneous workload and bandwidth. As a result, dynamic replication strategies are considered more desirable due to the dynamic aspects of Cloud systems. However, they have certain drawbacks such as the difficulty of collecting precise execution information from all nodes (Long et al., 2014).

2.2 Centralised vs. decentralised strategies

In addition to the nature of replication, replication strategies have also been classified according to the mechanism of controlling the creation of replicas. It depends on what entity controls the replication decision process. Then, data replication strategies can be viewed as centralised vs. decentralised strategies.

Each approach has its advantages and drawbacks. Centralised replication strategies (Begum and Sirisha, 2019; Huang et al., 2014; Sun et al., 2012; Zhang et al., 2018) are easier to implement. The strategy proposed in Zhang et al. (2018) is based on a central controller node that maintains an up-to-date global view of data in the system. This permits to quickly react to network dynamics and workload variations. However, the presence of a central authority is not ideal for reliability and fault tolerance since a single entity is responsible for all the decisions and has knowledge about every aspect of the Cloud system. On the other hand, decentralised replication strategies (Ghemawat et al., 2003; Mansouri and Buyya, 2019; Sousa and Machado, 2012; Tos et al., 2016; Wei et al., 2010) constitute a guarantee of reliability, since there is no single point of failure in the system. The system can behave predictably, even if a number of nodes are lost. However, the fact that some nodes may have incomplete information about the state of the system can lead to non-optimal results with excessive replications.

2.3 Other classifications

Other works classified these strategies according to other criteria. Tabet et al. (2017) provided a survey based on 15 strategies. In addition to the static vs. dynamic classification, other criteria are considered, e.g., the nature of the workload balancing (proactive vs. reactive strategies) and the replica factor determination (optimal replica factor vs. dynamic adjustment of this factor). Tos et al. (2018) provided another survey that takes into account the economic impact of replication. However, they only focus on strategies that consider auction models (Zhang et al., 2018) or virtual economy (Bonvin et al., 2010). Except the strategy proposed in Tos et al. (2016), the cost models used in the mentioned strategies do not necessarily consider monetary costs. Furthermore, these strategies do not consider economic aspects such as the monetary profit of the provider or the cost of penalties paid by the provider to its tenants.

3 Proposed classification

We propose another classification of data replication strategies in Cloud systems according to other criteria more specific to Cloud environments: (i) the orientation of the profit, (ii) the achieved objective function for which a strategy was designed, (iii) the number of tenant's objectives, (iv) the nature of the Cloud environment for which a strategy is designed, (v) the consideration of economic concepts when replicating data. For this last criterion, we focus on (a) strategies that consider the economic cost of replication and the provider's economic profit and (b) strategies that take into account the energy consumption cost. Of course, this classification can cause some overlap. Thus, some strategies can be cited in different classes.

3.1 The orientation of the profit

In the literature, most of the replication strategies proposed for cloud systems focused on the prospect of minimising the consumption of resources, e.g., data transfer or storage, for the cloud provider while satisfying the objectives of tenants (Boru et al., 2015; Khelifa et al., 2020; Liu et al., 2018; Sousa and Machado, 2012; Wei et al., 2010; Zeng and Verravelli, 2014) . Then, the provider profit is increased since its expenditures are reduced. Sousa and Machado (2012) aimed to adjust the consumption of resources in a multi-tenant database environment. By adapting to the workload variation of the nodes, the queries use replicas located on less loaded nodes. This permits to meet the objectives of the tenants, which reduces SLA violations. Boru et al. (2015) aimed to reduce energy consumption as well as bandwidth consumption. Mansouri et al. (2017) replicated the most popular data based on the Pareto principle (80-20%), which allows a significant reduction in the consumption of storage resources.

On the other hand, only some data replication work focus on reducing the monetary costs paid by tenants to the provider (Limam et al., 2019; Magalhaes and Silva, 2013; Sakr and Liu, 2012; Sakr et al., 2011; Sharma et al., 2011; Zhao et al., 2015). Sakr et al. (2011) offered a provisioning tool by declaratively defining application-specific rules. Resources are provisioned adaptively according to the needs of the consumer. The Kingfisher resource management system (Sharma et al., 2011) allows a tenant to optimise their resources in terms of capacity by choosing the server configuration that best suits them in terms of performance and costs. Magalhaes and Silva (2013) collected application-level data and detects performance anomalies motivated by workload variation, resource consumption or application changes. However, it neglects the minimisation of the price paid by the tenant to its provider. Zhao et al. (2015) allowed a dynamic and adaptive supply of resources, including replicas, based on rules defined by the applications. It aims to satisfy the performance requirements of tenants while reducing the monetary cost of the resources allocated to each tenant. Finally, the allocation of resources to tenants in Limam et al. (2019) is done according to the initial budget of the tenant.

3.2 The considered objective function

An objective function is a criterion that serves to optimise the system performance (Mokadem and Hameurlain, 2015). It determines the approach of each strategy in order to achieve a performance objective. In the following section, we propose a replication strategy classification according to the achieved objective function:

1) *Strategies that improve data locality*: Lee et al. (2015) exploited data popularity when deciding to replicate. Most of the time, data are replicated as close as possible to the nodes that generate the most demand, assuming that these data will probably remain popular in the future (Jayalakshmi and Ranjana, 2015). To cope with the wave of rapid popularity, Ridhawi et al. (2015) and Lazeb et al. (2021) took into account the variation in data popularity which peaks for a short period and then gradually decreases. These strategies are often based on data access history. This corresponds to the temporal and geographic localities. Other strategies even propose mechanisms for predicting future access based on historical access records in order to preventively replicate data (He et al., 2018; Ridhawi et al., 2015). Thus, data related to recently accessed data will probably be requested soon. This corresponds to spatial locality.
2) *Strategies that exploit the network bandwidth (NB) locality*. These strategies aim to reduce the consumption of the NB by replicating on the nodes that have bigger NB with the tenant that requires data (Park et al., 2004). Many replication strategies aim to reduce the NB resource consumption (Ardekani and Terry, 2014; Limam et al., 2019; Tabet et al., 2019, 2016). Mokadem and Hameurlain (2020) exploited the NB locality in order to reduce the response time of a tenant query, i.e., a replica of a required remote data *d* is placed at a node having a larger NB toward the node requiring *d*.
3) *Cost model-based strategies that reduce the replication cost*. In such strategies (Mansouri et al., 2017; Pu et al., 2015; Xiong et al., 2011), the replication decision is made accordingly to the output of a mathematical model that takes into account parameters such as the file access statistics or replica sizes. They aim to reduce the consumption of resources, e.g., storage or NB resources. In the strategy proposed in Tos et al. (2018), the replication decision relies on both cost model that estimates response time *RT* for a query *Q* and economic cost model that estimates the provider profit when a replica creation is considered. A replica creation is considered only if *RT* exceeds a threshold response time defined in the SLA. Then, a new replica is really created if and only if a node is found so that the response time SLO is satisfied again while this replication is profitable for the provider.
4) (iv) *Economic behaviours-based strategies that reduce the replication cost*. Some strategies exploit some economic behaviours used in trading (Belalem et al., 2011;

Marcus et al., 2018; Shi et al., 2017; Zhang et al., 2018). They are based on some market-like mechanisms in which data are regarded as tradable goods. Zhang et al. (2018) improved data availability by exploiting an auction model when placing new replicas. If the desired level of availability is not reached, an auction is launched to determine the location of a new replica. The auction price depends on several properties, including the NB and available storage space. Marcus et al. (2018) aimed to balance the load between the nodes of the system. It is based on an economic model that considers data as goods, queries as customers and nodes as companies. The value of the goods is estimated based on the frequency of access to these data and the monetary costs that the user pays for the execution of the requests.

3.3 The number of tenant's objectives

Most of the replication strategies proposed in the literature aim to guarantee a single specific objective for the tenants, e.g., availability (Wei et al., 2010), reliability (li et al., 2017), low latency (Ma and Yang, 2017), reduced response time (Tos et al., 2018; Khelifa et al., 2021), data durability (Liu et al., 2018), security (Ali et al., 2018) or reduction of the energy consumption (Boru et al., 2015). Other strategies aim to simultaneously meet several objectives for the tenants (Boru et al., 2015; Hassan et al., 2009; Long et al., 2014; Edwin et al., 2019; Mansouri and Javidi, 2018; Mokadem and Hameurlain, 2020).

3.3.1 Single-objective strategies

Data availability is the most answered objective that many data replication strategies aim to satisfy (Gill and Singh, 2016; Long et al., 2014; Liu et al., 2020). In what follows: we cite some strategies that aim to satisfy a single SLO objective:

Availability: Data availability often depends on the availability of the nodes that host data. An overloaded node cannot execute tenant queries on time. As a result, it is often necessary to increase the number of replicas so that the requested data is available on less busy nodes. Wei et al. (2010) emphasised that having too many replicas does not necessarily increase availability, but rather results in a decrease in performance. In this context, they calculate and maintain a minimum number of replicas to satisfy a minimum level of availability. The study of the relationship between the number of replicas and the level of availability has also been the subject of the work in Sun et al. (2012). The proposed strategy establishes a minimum number of replicas for the most popular data.

Fault tolerance: In the case of a node failure, data replicas available in other nodes facilitate the retrieval of that data. In this way, data replication significantly improves the system fault tolerance (Li et al., 2019; Selvi et al., 2018). Xiong et al. (2011) proposed a strategy called Resilient, Fault-tolerant and High-efficient (RFH) that allows the number of replicas to be adapted according to the network traffic. If a data set becomes popular, more replicas are created. Otherwise, unwanted replicas are erased to save resources. As a result, access errors to such replicas are reduced in the case of a failure. Furthermore, the cost of replication is relatively low. Mokadem and Hameurlain (2020) ensured fault tolerance through the creation of at least one replica for each data set on nodes geographically distributed across different regions.

Reliability: It refers to the property that a system can run uninterruptedly without failure. In contrast to availability, reliability is defined in terms of a time interval instead of an instant in time. Data replication is the most widely used technique, along with the erasure code technique (Bui et al., 2016), to guarantee data reliability. It allows creating and storing multiple replicas of data to reduce the likelihood of data loss. Liu et al. (2018) relied on techniques such as data compression to ensure data sustainability.

Performance: Performance guarantees, e.g., in terms of response time, are often not offered by cloud providers as a part of the SLA because of the heterogeneous workloads in cloud systems (Mokadem and Hameurlain, 2020). For example, Google Cloud SQL[1] only provides downtime and error guarantees without an RT guarantee. Thus, satisfying performance can often conflict with the goal of obtaining a maximum economic benefit at minimal operating costs (Long et al., 2014). Only some replication strategies integrate the performance objective, generally in terms of response time, in the SLA (Bai et al., 2013; Dabas and Aggarwal, 2019; Kumar et al., 2014; Li et al., 2018; Mansouri and Buyya, 2019; Sousa and Machado, 2012; Tose t al., 2016) . The RTRM strategy proposed in Bai et al. (2013) aimed to reduce the response time. Using a response time threshold, RTRM creates new replicas when it exceeds this threshold. RTRM dynamically predicts bandwidth and selects a replica accordingly. Kumar et al. (2014) improved performance by proposing a replica placement algorithm. Zhao et al. (2015) tenant-oriented, it meets the performance objective by creating replicas whenever the response time exceeds a threshold set in advance in the SLA. The PEPR strategy (tos et al., 2018) takes advantage of the hierarchy in terms of bandwidth in order to reduce bandwidth consumption and consequently reduce response times. Data access time is reduced in the DPRS strategy (Mansouri et al., 2017) due to the parallel reading of partitions.

Reduction of energy consumption: A number of data replication strategies (Alghamdi et al., 2017; Boru et al., 2015; Edwin et al., 2019; Séguéla et al., 2019; Xu et al., 2015; Zhang et al., 2015) aim to reduce energy consumption. Some of them aim to reduce energy consumption as an objective when others try to reduce the carbon footprint by replicating in a greener DC. Some strategies take into account the profit made by reducing energy consumption. The strategy proposed in Boru et al. (2015) models the energy consumption and the network usage in resource consumption. If the number of data accesses exceeds a certain threshold, a replication is made on the DC which consumes the least energy and uses the least bandwidth than the central databases. The aim of the strategy proposed in Zhang et al. (2015) is to group the workload by placing replicas on a few nodes. Then, put on standby or even stop the inactive nodes. Zhang et al. (2015) reduced carbon emissions, e.g., the emission of greenhouse gases, thanks to the knowledge of the carbon footprint of each energy source and each of the sources used by the DC. Then, replication takes place in DCs that consume less energy.

3.3.2 Multi-objective strategies

The strategy proposed in Hassan et al. (2009) aims to keep storage below certain limits, minimise latency and optimise reliability while trying to find a compromise between these objectives. The MORM strategy (Long et al., 2014) is based on mathematical models that aim to satisfy several objectives, including availability, response time, network latency, energy consumption and load balancing. The placement of data is done according to an optimisation function such as a weight associated to each objective and decided by the administrator. This function aims to capture the compromise between these objectives by taking into account the relationship between the number of replicas and performance. In the strategy proposed in the Iridium system (Pu et al., 2015), the query frequency and the data access statistics are used when placing replicas. In consequence, the consumption of bandwidth between DCs is reduced and latency is minimised. Boru et al. (2015) aimed to reduce both energy consumption and the use of bandwidth. The PEPR strategy (Tos et al., 2016) aims to reduce response time and ensure minimum availability. The EIMORM strategy proposed in Edwin et al. (2019) aimed to balance data availability, load balancing and the cost of replication. In order to balance performance and storage efficiency, Li et al. (2017) first stored data with replication, followed by encoding the replicated data through the erasure coding. Finally, Mansouri and Javidi (2018) aimed to improve availability and load balancing.

On the other hand, even if certain strategies do not directly consider performance as a main objective, i.e., not specified in the SLA, an improvement in performance, e.g., in terms of reduction in response time, can be observed as a consequence of other targeted objectives. Lee et al. (2015) and Wei et al. (2010) claimed that performance can be improved by balancing the load between different nodes. Favouring data locality also reduces the consumption of network bandwidth, which improves system performance (Kloudas et al., 2015; Vulimiri et al., 2018).

3.4 The nature of the cloud environment

Although most of the replication strategies mentioned above have been proposed for a single cloud provider, some replication strategies were deployed on multi-provider clouds (Abouzamazem and Ezhilchelvan, 2013; Abu-Libdeh et al., 2010; Bessani et al., 2013; Chen et al., 2014; Mansouri and Buyya, 2019; Wu et al., 2013).

One of the first replication strategies to consider multiple cloud providers is the strategy proposed in the DepSky system (Bessani et al., 2013). However, it only deals with security aspects without taking into account the economic aspects such as the cost of replication. In the strategy proposed in Abouzamazem and Ezhilchelvan (2013), tenants rent services from several providers, according to a pricing policy and resource prices provided by each cloud provider. The data replication strategy integrated into the *SpanStore* system (Wu et al., 2013) covers several cloud providers. The price difference between providers is exploited in order to minimise costs when taking into account fault tolerance and latency requirements. The resource price difference is also exploited in Mansouri and Buyya (2019) to minimise the monetary cost of replication with the assumption that the workload is known in advance. Khelifa et al. (2022) proposed 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 NCC system proposed in Chen et al. (2014) can interconnect different Clouds and transparently stripe data across them. The proposed strategy in such system is mainly designed for providing a fault-tolerance when the monetary cost of repair is reduced compared to the erasure code technique. Liu, G. and Shen (2017) exploited a nonlinear integer programming model to maximise data availability in both types of failures and minimise the cost of replication. Also, it avoids the vendor lock-in problem, i.e., a tenant may not be free to switch from one provider to another. Finally, the data replication strategy proposed in the RACS system (Abu-Libdeh et al., 2010) retrieves data from the Cloud that is about to fail and move them to the new Cloud while the vendor lock-in problem is also avoided.

Although only some data replication strategies have been proposed for a multi-provider cloud environment, Mansouri and Buyya (2019) affirmed that more and more cloud customers often use more than one provider. Amazon Web Services and Microsoft Azure are most often chosen by customers. The choice of one provider or another depends on the pricing policies applied by a particular provider. These public cloud service providers are often linked to existing data centres located across the world.

3.5 Consideration of economic aspects

As surprising as it may seem, a significant part of the existing replication strategies in Cloud systems neglects the economic costs related to data replication. In what follows, this is considered as a criterion for classifying data replication strategies.

3.5.1 Replication strategies without taking into account the economic cost of replication

Most of the strategies described here focus on reducing the consumption of the required resources during data replication without taking into account the economic cost of replication (Bai et al., 2013; Dabas and Aggarwal, 2019; Edwin et al., 2019; Kumar et al., 2014; Lee et al., 2015; Long et al., 2014; Liu et al., 2013; Mansouri and Javidi, 2018; Pu et al., 2015; Sakr et al., 2011; Tan and Babu, 2016; Wei et al., 2010) . Most of them were interested in finding the optimal number of replicas (He et al., 2018) or replica placement (Zhang et al.,

2015) because of their effects on performance. In the strategy proposed in Sakr et al. (2011), an elastic database management is considered. Replication decisions are based on rules that ignore the economic aspects. Although the RTRM strategy (Bai et al., 2013) aims to satisfy the response time objective, it neglects the economic aspects. The SWORD strategy (Kumar et al., 2014) minimises a new metric called the query span and the number of nodes involved in executing a query. This reduces the consumption of resources without quantifying this consumption from an economic point of view. It is also the case of the strategy proposed in Dabas and Aggarwal (2019) that aims to satisfy a response time guarantee without considering the cost of replication. Finally, the strategy proposed in Edwin et al. (2019) aimed to meet several objectives. Except the energy consumption, it does not include other economic costs such as the provider profit or penalty costs.

3.5.2 *Data replication strategies taking into account the economic cost of replication*

Some strategies are mentioned as taking into account the economic cost of data replication. However, the cost model used is not necessarily a monetary cost (Bonvin et al., 2010; Gill and Singh, 2016; Liu et al., 2013) . Skute (Bonvin et al., 2010) is a strategy based on a virtual economy. VMs act autonomously and periodically announce their rental to other VMs. They accumulate gain by responding to requests and spend this gain by storing replicas on the resources specific to other VMs, according to their rent. The objective is to minimise communication costs while maximising virtual economic profit. Xiong et al. (2011) presented a resource management strategy that uses machine learning techniques to generate an optimal amount of virtual profit in a multi-tenant DB environment. They use a predictive model to determine CPU configuration and memory allocation, which generates minimum penalty costs for a given workload. Liu et al. (2013) affirmed that the replication cost is taken into account by modelling it in terms of time. This is also the case in the DCR2S strategy proposed in Gill and Singh (2016). Replication is only possible if the cost does not exceed a budget value. It corresponds to a value cost, initially assigned to DCs. Thus, the considered cost of replication is not a real monetary cost. In what follows, we particularly focus on: (i) strategies that take into account the economic profit of the provider and the penalties paid by the provider to its tenants while modelling monetary costs in their cost model and (ii) strategies that aim to reduce the provider's costs from the energy consumption point of view.

Data replication strategies considering the provider economic profit: Only a few data replication strategies (Casas et al., 2017; Liu and Shen, 2017; Mansouri and Buyya, 2019; Mokadem and Hameurlain, 2020; Tos et al., 2018; Tos et al., 2021; Wu et al., 2013; Zeng and Verravelli, 2014; Zeng et al., 2016) model the cost of replication and the provider profit as monetary costs while satisfying the tenant performance objective, e.g., in terms of response time. In the strategy proposed in the SpanStore system (Wu et al., 2013), price differences between different cloud providers are exploited during data replication to minimise replication costs when taking into account fault tolerance and latency requirements. Zeng and Verravelli (2014) taken into account the trade-off between the execution time and the monetary cost of replicating metadata. In Zeng et al. (2016), providers buy services from sellers in the cloud and then resell them to tenants. The proposed strategy aims to minimise the provider's resource expenditures while maximising the use of storage resources for tenants. Then, the number of replicas and their placement depend on the compromise between performance and monetary cost in each node.

The monetary provider's expenses and incomes are also estimated in Tos et al. (2018) before the execution of each tenant query. A new replica is created only if this replication is profitable for the provider. In addition, the location of the replicas is based on the selection of the cheapest DC that meets the response time SLO. Casas et al. (2017) taken into account some characteristics of workflows such as execution time, dependency models and file size. The number of replicas is increased as long as the monetary cost of an application does not exceed a monetary threshold fixed in advance in the SLA. The replication strategy proposed in Mansouri and Buyya (2019) takes advantage of the heterogeneous pricing between DCs, in terms of storage and bandwidth, by replicating according to the future variation of the workload. Finally, replication costs and penalties are modelled as monetary costs included in the provider's expenses in Mokadem and Hameurlain (2020). A new replica is created only if a node placement is found such as the response time objective is satisfied while this replication must be profitable for the provider.

Data replication strategies considering the cost of energy consumption: Most of the data replication strategies cited above neglect the power and energy required for data replication. However, DCs consume a large portion of the world's global electricity consumption and energy bills have become the second highest costs in cloud provider budgets (Zakarya and Gillam, 2017). This is due to the development of the internet and the increasing storage needs of businesses. Some studies, e.g., Shehabi et al. (2016) predicted that the energy consumption of DCs in USA will increase by 13% by 2030. They had already predicted that electricity consumption of DCs will increase by 140 billion KWh/year in 2020, which will cost companies 13 billion dollars/year in electricity bills and the emission of nearly 100 million tonnes of carbon pollution per year. In this context, some data replication strategies have focused on the energy consumption.

The strategy proposed in Alghamdi et al. (2017) takes into account energy consumption in order to reduce the monetary cost of a user query. The provider's profit corresponds to the difference between the state without replication and the state resulting from such replication. The authors claimed that the proposed algorithm reduces the total power consumption of data access compared to an optimal replication solution. Séguéla et al. (2019) compared different replication strategies, mainly (Boru et al., 2015), MORM (Long et al., 2014) and PEPR (Tos et al., 2018). The performance evaluation shown that the strategy proposed in Boru et al. (2015) reduced the provider's expenditures at the cost of higher energy consumption while PEPR reduces expenditures of the provider without any effect on the energy consumption. In contrast, MORM consumes significant energy when a high number of replicas are created. Finally, Seguela et al. (2021) proposed a static and multi objective data replication strategy that aims to reduce both energy consumption and expenditure of the provider. It leverages on heterogeneity, sleep states and consolidation while considering performance for tenants.

4 Important factors for ensuring performance for tenants while considering the economic profit for the provider

Adopting one strategy rather than another depends on several factors that significantly impact both tenant QoS and provider profit. We also need to always ensure that the benefit of a given strategy is higher than the cost of replication (Van Steen and Pierre, 2010). In this context, the cost of replication depends on the decision to favour one factor over others. In what follows, we enumerate some important factors that impact performance of any replication strategy in Cloud Systems.

- *Optimal granularity*: Determining the appropriate data granularity is a very important aspect when replicating in order to achieve the performance objective. Van Steen and Pierre (2010) demonstrated that the optimal granularity depends on the applications. They conclude that meeting the performance objective requires strategies that favour the replication of small data units. This is also valid in cloud systems, especially since the performance objective must be respected so that the provider does not pay penalties to its tenants.
- *Data consistency:* When data are updated frequently, changes should be propagated to all replicas of the system in order to ensure consistency. Consequently, a global synchronisation with appropriate protocols is necessary between different nodes containing these replicas (Chen et al., 2014). There are many consistency models and protocols providing different levels of performance guarantees (Campêlo et al., 2020). However, there is no universal solution and these protocols are generally not scalable. This is why most replication strategies with performance objective generally compromise on data consistency (Van Steen and Pierre, 2010), especially when the provider wants to have an economic profit.
- *Bandwidth consumption:* Many replication strategies aim to reduce the consumption of network resources. In fact, more and more big data are scattered across DCs distributed around the globe. In consequence, replicating these data requires significant NB, which becomes increasingly expensive the further away the replicas are located from users that require them. For this aim, most of replication strategies aim to reduce bandwidth consumption in order to reduce the consumption of provider resources.
- *Access pattern:* The choice of an access pattern constitutes an important factor that impacts performance of any replication strategy. To prove the impact of the access pattern on replication strategy performance, Mansouri and Javidi (2018) evaluated the performance of seven replication strategies under two data distributions: (i) uniform distribution that provides a naive baseline and (ii) non uniform distribution, e.g., zipf (Breslau et al., 1999) that is designed to react to data popularity. They prove that the average query response time is significantly reduced when a given strategy is dynamically adapted to the users' preferences (Mokadem and Hameurlain, 2020). This avoids the SLA violation, which impacts the profit of the provider.
- *Cloud topology:* There is no standard architecture in cloud environments. However, it has been observed that the topology of a given system significantly affects the design of a data replication strategy for which it was designed (Tos et al., 2019). Some companies consider the transferring of all data to a single DC/cluster when executing a tenant query. This generates a significant data transfer. In order to reduce NB consumption, other solutions (Ardekani and Terry, 2014; Tos et al., 2016) model a two level hierarchy, i.e., a region is composed of a single DC, while optimising the cross-region data transfer consumption. However, links between DCs are heterogeneous. In consequence, most of commercial solutions[2] as well as some recent strategies (Mokadem and Hameurlain, 2020; Tos et al., 2021) consider within a region, several DCs that communicate through an intermediate NB. This leads to a system topology with three levels: regions, DCs and nodes that host data.
- *The number of tenants:* In order to improve their profits, providers implement a resource sharing among multiple tenants by consolidating various tenants' applications on a single system. In return, each tenant pays the rent of resources to the provider according to the 'pay as you go' model, i.e., a tenant only pays what it consumes (Armbrust et al., 2010). Hence, serving an optimal number of tenants through the 'pay as you go' model while satisfying tenant objectives results in an optimal profit for the provider.

It has been observed that trade-offs exist between these factors, especially when a replication strategy aims to satisfy the performance objective for the tenants. In consequence, some factors should be taken into account simultaneously. For example, keeping data close to the user, in order to reduce the access cost, should not be done at the expense of network congestion. Also, many works have concluded that a good replication strategy must be based on an efficient replica placement algorithm with an optimal number of replicas while the choice of nodes holding these replicas should not be done at the expense of the system load. On the other hand, it has been observed that a lot of strategies have been proposed for cloud systems. For a more advanced comparison between these strategies, we have identified, in Table 1, a non-exhaustive list of some data replication strategies with respect to existing and proposed classifications. There is not a single one that ensures all the tenant objectives while considering the economic aspects of clouds. In consequence, a good strategy could favour several tenant objectives while trying to find a compromise between the satisfaction of a certain QoS for the tenant and the economic profit for the provider.

Table 1 A comparative analysis of some data replication strategies with regard to the proposed classification

	The proposed classification							
	Static vs. Dynamic	*Centralised vs. Decentralised*	*Provider vs. Tenant oriented*	*The considered Objective Function*	*Single vs. Multi-Objective*	*Single vs. Multi-Provider Cloud*	*Consideration of economic costs*	*Consideration ofenergy consumption*
Ghemawat et al. (2003)	Static	Decentralised	Provider	Data Locality	Single	Single	–	–
Hassan et al. (2009)	Static	Decentralised	Provider	Data Locality	Multi	Single	–	–
Abu-Libdeh et al. (2010)	Dynamic	Decentralised	Provider	Data Locality	Multi	Multi	X	–
Wei et al. (2010)	Dynamic	Decentralised	Provider	Cost Model	Multi	Single	–	–
Bonvin et al. (2010)	Dynamic	Decentralised	Provider	Data Locality	Single	Single	X	–
Sakr and Liu (2011)	Dynamic	Decentralised	Tenant	Cost Model	Single	Single	–	–
Xiong et al. (2011)	Dynamic	Centralised	Provider	Cost Model	Single	Single	X	–
Sun et al. (2012)	Dynamic	Centralised	Provider	Cost Model	Single	Single	–	–
Bai et al. (2013)	Dynamic	Decentralised	Provider	Data Locality	Multi	Single	–	–
Lin et al. (2013)	Dynamic	Decentralised	Provider	Cost Model	Multi	Single	–	–
Wu et al. (2013)	Dynamic	Decentralised	Provider	NB Locality	Multi	Multi	X	–
Long et al. (2013)	Static	Decentralised	Provider	Data Locality	Multi	Single	–	X
Zeng and Veeravalli (2014)	Static	Decentralised	Provider	Cost Model	Multi	Single	X	–
Kumar et al. (2014)	Dynamic	Decentralised	Provider	Data Locality	Multi	Single	–	X
Boru et al. (2015)	Dynamic	Centralised	Provider	Data Locality	Multi	Single	–	X
Zhao et al. (2015)	Dynamic	Decentralised	Tenant	–	Multi	Single	X	–
Gill and Singh (2016)	Dynamic	Decentralised	Provider	Data Locality	Multi	Single	X	–
Zeng et al. (2016)	Static	Decentralised	Provider	Cost Model	Single	Single	X	–
Mansouri et al. (2017)	Dynamic	Decentralised	Provider	Data Locality	Multi	Single	–	–
Edwin et al. (2017)	Static	Decentralised	Provider	Cost Model	Multi	Single	X	X
Shi et al. (2017)	Dynamic	Decentralised	Provider	Economic	Multi	Multi	X	–
Liu and Shen (2017)	Dynamic	Decentralised	Provider	Cost Model	Multi	Multi	X	–
Alghamdi et al. (2017)	Dynamic	Decentralised	Provider	Cost Model	Multi	Single	X	X
Tos et al. (2018)	Dynamic	Decentralised	Provider	NB Locality	Multi	Single	X	–
Mansouri and Javidi (2018)	Dynamic	Decentralised	Provider	Data Locality	Multi	Single	–	–
Sun et al. (2018)	Dynamic	Decentralised	Provider	Cost Model	Multi	Single	–	–
Liu et al. (2018)	Dynamic	Decentralised	Provider	–	Multi	Single	X	–
Mansouri and Buyya (2019)	Dynamic	Decentralised	Provider	Cost Model	Multi	Multi	X	–
Limam et al. (2019)	Dynamic	Decentralised	Tenant	Cost Model	Multi	Single	X	–
Mokadem and Hameurlain (2020)	Dynamic	Decentralised	Provider	NB Locality	Multi	Single	X	–

5 Simulation analysis

We compare the performance of five replication strategies proposed for cloud systems. Two of these strategies do not consider the economic aspects of replication: (i) the Cost-effective Dynamic Replication Management strategy (CDRM) (Wei et al., 2010) considers the objective of load balancing and (ii) the Dynamic Popularity aware Replication Strategy (DPRS) (Mansouri et al., 2017) replicates only the top 20% of frequently accessed data on the best locations. The other three strategies take into account the economic cost of replication: (iii) the Dynamic Cost-aware Re-replication and Re-balancing Strategy (DCR2S) (Gill and Singh, 2016) considers the response time objective while an initial budget for each DC is fixed in the SLA (iv) the PErformance and Profit oriented data Replication strategy (PEPR) (Tos et al., 2018) aims to satisfy the response time objective when data replication can be considered for each query and (v) the Replication Strategy satisfying Performance objective while ensuring an economic profit for the provider in Cloud DCs (RSPC) (Mokadem and

Hameurlain, 2020). RSPC also meets the objective of minimum availability and data replication is considered per group of queries.

CloudSim (Calheiros et al., 2010), a popular and open source cloud computing simulation tool, is used to simulate DCs. In our experiments, we simulated a cloud with three regions. Within each region, we simulated 10 DCs. Then, 1000 heterogeneous VMs are implemented in each DC. We have extended CloudSim to support data replication, query placement and some important requirements. Thus, each VM has storage, memory and computing capacity. Economic concepts are also taken into account: (i) a tenant is charged for a given number of DB queries (here 1000 queries) during a billing period (BP) and (ii) an SLA violation occurs when the query response time exceeds a response time threshold (here 100 s). The threshold value is defined based on preliminary experiments.

The arrival rate of database (DB) queries follows a Poisson distribution. Our experiments dealt with 3000, 12,000, 30,000 and 48,000 queries during a BP. The broker assigns cloudlets (associated to queries) to randomly selected VMs when accessing distributed relations. We considered a subset of TPC-H[3] queries {Q4, Q10 and Q8} for analytical purposes. These queries have different level of complexity {1, 3 and 7 joins, respectively} when a query plan is pre-determined for each query. We call them simple, medium and complex queries, respectively. We simulated a parallel execution of queries launched simultaneously by several tenants. A read-only DB relation constitutes the granularity of replication. We dealt with a simulation since it allows us to directly control some parameters in order to understand their individual impact on performance, e.g., query arrival rate and NB variations. Table 2 describes the main parameters used in our experiments.

Table 2 Some configuration parameters

Parameter	*Value*
#regions	3
#DCs	10
#VMs	1000
Average size of a relation	700 Mb
Avg. available inter-region NB (delay respect.)	500 Mb/s (150 ms respect.)
Avg. available inter-DC NB (delay respect.)	1Gb/s (50 ms respect.)
Avg. available intra-DC NB (delay respect.)	8 Gb/s (10 ms respect.)
Average size of a relation	800 Mb
Average VM processing capability	1500 MIPS
Average storage capacity/ VM	10 Gb
Billing Period (BP) duration	10 min
#queries/ BP	[3000, 48,000]
Response time threshold	100s

5.1 Experimental results

We have measured the following metrics: (i) the average replica factor, (ii) the average measured query Response Time (RT) during a BP, (iii) the impact of the query arrival rate and user access pattern on performance, (iv) the number of SLA violations and (v) consumption of the provider's resources.

Figure 1 shows the average RT and Figure 2 shows the average replica factor obtained with the compared replication strategies when the data distribution is uniform. We deal with simple, medium and complex queries. PEPR presents the most important replica factor when a low number of queries are submitted during a BP, e.g., less than 12,000 queries. It is due to the fact that the replica decision is considered at the per-query level, i.e., each time RT of a query exceeds the RT threshold. This generates an important overhead. The replica factor in CDRM, DPRS and DCR2S is more important compared to RSPC as it is shown in Figure 2. In fact, RSPC replicates data only if the SLA is not satisfied. This occurs if the estimated RT exceeds a RT threshold for a given number of times (here 10 times) or when the estimated RT exceeds a critical RT threshold (here, 180 s). On the other hand, CDRM aims to balance the workload between different nodes by creating more replicas.

Figure 1 Average response time

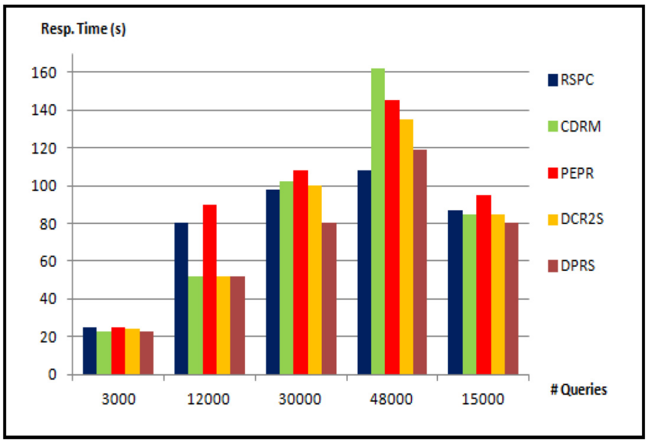
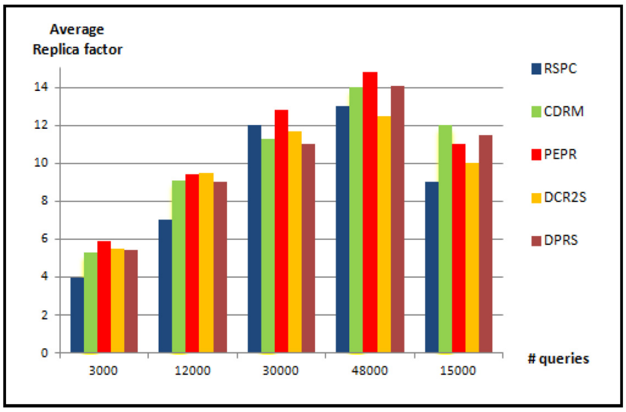


Figure 2 Average replica factor



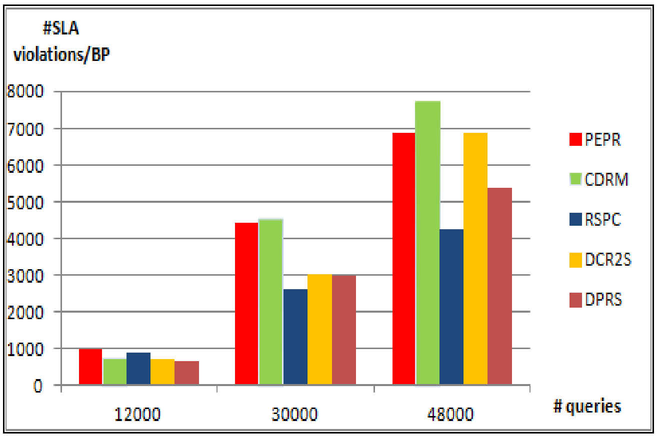
When 30,000 queries are submitted during a BP (50 queries/s), PEPR and RSPC create the most important number of replicas in order to satisfy the RT objective. DPRS presents the best RT since it creates replicas only for the most popular data. CDRM does not create more replicas when a load balance is achieved. As long as the budget is not reached, DCR2S creates additional replicas in order to improve RT. When 48,000 queries are submitted, i.e., it corresponds to 80 queries/s, only a few replicas are created with DCR2S and RSPC strategies. Once the cost of replication exceeds the budget in DCR2S, the knapsack algorithm tries to optimise the cost of replication by re-replicating to lower cost DCs. However, load balancing is not taken into account and replicating outside the local region does not decrease the replication cost. Overloaded VMs are blocked from receiving new queries in CDRM and popular data are updated with DPRS. This generates replica creations outside the region receiving the queries. Hence, DCR2S, DPRS and especially CDRM generate a significant RT increase. PEPR generates a greater RT even it creates more replicas.

In order to measure the replica factor adjustment with the compared strategies, we measure the average replica factor when a significant decrease is observed in the number of queries. At the end of a BP in which 48,000 queries were submitted, we simulate the submission of a low number of queries (15,000 queries) during the next BP as shown in Figure 1. We observed that the replica factor of all strategies is decreased. Since the workload has decreased, all strategies remove some replicas in order to reduce the consumption of resources. The RT objective is satisfied with all strategies. PEPR, DCR2S and especially RSPC remove more replicas than CDRM and DPRS. This proves that CDRM and DPRS continue to use some of the previously created replicas even with a reduced number of queries.

We measure the impact of the user's access pattern on performance. We are interested on average response time obtained with the compared strategies when the data distribution is non uniform (here a zipf distribution), which better responds to data popularity. We have observed that the replica creation is proportional to the data popularity. Compared to results shown in Figure 1, the gains in terms of response time is around 11%, 9% and 7% with DPRS, RSPC and DCR2S, respectively. On the other hand, CDRM creates replicas based on load balancing regardless of data popularity.

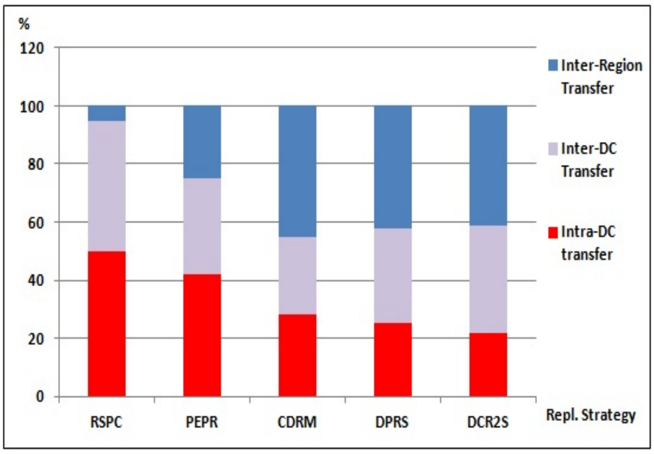
We also analyse the number of SLA violations during a BP. We assume that a penalty amount is paid from a tenant to the provider when the RT exceeds a RT threshold (here 100 s). Figure 3 shows a relationship between the number of submitted queries and the number of SLA violations during a BP. When a low number of queries are submitted during a BP, PEPR generates the most important number of SLA violations as a result of higher RTs while other strategies generate almost the same number of SLA violations. When a high number of queries are submitted, PEPR, CDRM and DCR2S generate even more SLA violations while the number of SLA violations with RSPC and DPRS increases slowly. The number of SLA violations with CDRM is 2.2 times more important than those generated by RSPC.

Figure 3 Average number of SLA violations



Finally, Figure 4 shows the NB resource consumption required by the compared strategies while considering the NB hierarchy. In these experiments, 30,000 queries were submitted during the BP. With RSPC, the majority of data transfers are performed in the intra-region level. Inter-region data transfers are performed only during initial replications that aim to satisfy the minimum availability objective. In contrast, satisfying the RT objective does not require inter-region data transfers. On the other hand, inter-region transfers are more frequent with DCR2S, DPRS and especially CDRM. This highly impacts RTs since inter-region links are slower.

Figure 4 Average NB resource consumption



5.2 Discussion

With a reduced number of queries, strategies that do not take into account the provider's profit, i.e., CDRM, DCR2S and DPRS, create more replicas than RSPC and PEPR. In consequence, slightly lower response times are obtained with these strategies. At the cost of having a slightly higher RT while satisfying the RT objective, RSPC and PEPR generate less provider expenditure costs.

As the number of queries increases, VMs become busier. CDRM satisfies only a load balancing objective which is not sufficient to ensure the response time objective. Based on data popularity, DPRS creates additional replicas. However, these replicas are created outside the region that receives queries. This is not the case with DCR2S due to the limited initial budget. However, the re-replication is not sufficient to balance the workload. Although PEPR considers the provider's profit, it generates an overhead that affects performance since a replication is provided per query when most of replications are provided per set of queries in RSPC. Then, better response times are obtained with RSPC. This is due to the exploitation of parallelism, the replication by group of queries and the replica placement according to the availability of the network bandwidth. Regarding penalty costs, RSPC and DPRS create more replicas to satisfy SLA, which reduces penalty costs compared to CDRM and DCR2S strategies. Furthermore, more unnecessary replicas are removed with RSPC and DPRS through an elastic replica factor adjustment. In consequence, the provider expenditure costs are reduced. With respect to resource consumption, RSPC and PEPR benefit from the NB hierarchy, which reduces the data transfer consumption. In contrast, DPRS and mainly CDRM require inter-region data transfers. Finally, RSPC and PEPR require more storage consumption. However, storage costs are cheaper than data transfer costs.

6 Conclusions

We provided a survey of data replication strategies in Cloud systems. Most of data replication studies in the literature classified these strategies as static vs. dynamic or centralised vs. decentralised strategies. Furthermore, these works often did not provide any performance evaluation. We propose to classify these strategies according to other criteria while taking into account some characteristics specific to Cloud systems: (i) the orientation of the profit (provider-oriented vs. customer-oriented strategies), (ii) the considered objective function (data locality vs. availability of bandwidth vs. cost reduction), (iii) the number of SLO objectives included in the SLA (single-objective vs. multi-objective strategies), (vi) the nature of the Cloud environment for which a strategy was designed (strategies for single-provider vs. multi-provider Cloud systems) and finally, (v) the consideration of economic costs by data replication strategies. Regarding the last criterion, we focus in particular on strategies that take into account the provider monetary profit and strategies that reduce the energy consumption.

From this review, it has been seen that a lot of strategies have been proposed for cloud systems. However, there is not a single one that ensures all the tenant objectives while considering the economic aspects of clouds. Although cloud systems as described in Foster et al. (2008) are based on economics, most of the replication strategies proposed in the literature are only interested in reducing the costs of replication without focusing on economic costs of this replication. On the other hand, only a few strategies aim to satisfy the performance objective while taking into account the economic cost of replication, the provider's economic profit, the energy consumption and the penalties paid to tenants. Furthermore, very few strategies integrate the provider's monetary expenses into the economic cost model.

We performed a simulation study to investigate the impact of some important factors on data replication performance. The simulation study indicates that promoting a tenant's objective at the cost of the provider's profit should not be the goal of data replication strategies in Cloud systems. In consequence, it is important to take into account the trade-off between the QoS satisfaction for the tenants and the provider profit for the Cloud provider. This constitutes a motivation for our research work. As a future work, we aim to design new replication strategies that aim to meet several tenants' objectives including performance while considering the economic cost resulting from replication.

Acknowledgements

This work has been funded by the project FR06/2020 by International Cooperation & Mobility (ICM) of the Austrian Agency for International Cooperation in Education and Research (OeAD-GmbH). We would also thank 'the French Ministry of Foreign and European Affairs' and 'The French Ministry of Higher Education and Research' which support the Amadeus program 2020 (French-Austrian Hubert Curien Partnership – PHC). Project Number 44086TD.

References

Abouzamazem, A. and Ezhilchelvan, P. (2013) 'Efficient inter-Cloud replication for high-availability services', *Proceedings of the IEEE International Conference on Cloud Engineering (IC2E)*, pp.132–139.

Abu-Libdeh, H., Princehouse, L. and Weatherspoon, H. (2010) 'Racs: a case for cloud storage diversity', *ACM SoCC*, pp.229–240.

Alghamdi, M., Tang, B. and Chen, Y. (2017) 'Profit-based file replication in data intensive cloud data centers', *IEEE International Conference on Communications (ICC)*, pp.1–7.

Ali, M., Kashif, B., Khan, U., Bhardwaj, V., Keqin, L. and Albert, Z. (2018) 'DROPS: division and replication of data in cloud for optimal performance and security', *IEEE Transaction on cloud Computing*, pp.303–315.

Ardekani, M.S. and Terry, D.B. (2014) 'A self-configurable geo-replicated cloud storage system', *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation*, Broomfield, pp.367–381.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D, Rabkin, A., Stoica, L. and Zaharia, M. (2010) 'A view of cloud computing Communication', *Communication of the ACM*, Vol. 53, No. 4, pp.50–58.

Bai, X., Jin, H., Liao, X., Shi, X. and Shao, Z. (2013) 'RTRM: a response time-based replica management strategy for cloud storage system', *Grid and Pervasive Computing*, Vol. 1, pp.124–133.

Begum, S.S. and Sirisha, S. (2019) 'Cloud optimal security using data fragmentation and replication', *International Journal of Computer Science Engineering and Scientific Technology (IJCSEST)*.

Belalem, G., Bouamama, S. and Sakhri, L. (2011) 'An effective economic management of resources in cloud computing', *Journal of Computers*, Vol. 6, No. 3, pp.404–411.

Benoit, A. and Rehn-Sonigo, V. (2008) 'Replica placement and access policies in tree networks', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 19, No. 12, pp.1614–1627.

Bessani, A., Correia, M., Quaresma, B., André, F. and Sousa, P. (2013) 'DepSky: dependable and secure storage in a cloud-of-clouds', *ACM Transactions on Storage*, Vol. 9, No. 4, pp.1–32.

Bonvin, N., Papaioannou, T.G. and Aberer, K. (2010) 'A self-organized, fault tolerant and scalable replication scheme for cloud storage categories and subject descriptors', *ACM Symposium on Cloud Computing*, pp.205–216.

Boru, D., Kliazovich, D., Granelli, F., Bouvry, P. and Zomaya, A.Y. (2015) 'Energy-efficient data replication in cloud computing datacenters', *Cluster Computing*, Vol. 18, No. 1, pp.385–402.

Breslau, L., Cao, P., Fan, L., Phillips, G. and Shenker, S. (1999) 'Web caching and Zipf-like distributions: evidence and implications', *Proceedings of the 18th IEEE Conference of the IEEE Computer and Communications Societies*, New York, USA, pp.126–134.

Bui, D.M., Hussain, S., Huh, E.N. and Lee, S. (2016) 'Adaptive replication management in HDFS based on supervised learning', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 6, pp.1369–1382.

Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C. and Buyya, R. (2010) 'CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms', *Software: Practice & Experience*, Vol. 41, No. 1, pp.23–50.

Campêlo, R.A., Casanova, M.A. and Guedes, D.O. et al. (2020) 'A brief survey on replica consistency in cloud environments', *Journal of Internet Services and Applications*, Vol. 11, No. 1, pp.1–13.

Casas, I., Taheri, J., Ranjan, R., Wang, L. and Zomaya, A.Y. (2017) 'A balanced scheduler with data reuse and replication for scientific workflows in cloud computing', *Future Generation Computer Systems*, Vol. 74, pp.168–178.

Chen, H., Hu, Y., Lee, P. and Tang, Y. (2014) 'NCCloud: a network coding- based storage system in a cloud-of-clouds', *IEEE Transactions on Computers*, Vol. 63, No. 1, pp.31–44.

Dabas, C. and Aggarwal, J. (2019) 'Delayed replication algorithm with dynamic threshold for cloud datacenters', *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, Springer, pp.625–637.

Edwin, E.B., Umamaheswari, P. and Thanka, M.R. (2019) 'An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center', *Cluster Computing*, Vol. 22, No. 5, pp.11119-11128.

Foster, I., Zhao, Y., Raicu, I. and Lu, S. (2008) 'Cloud computing and grid computing 360-degree compared', *Proceedings of the IEEE Grid Computing Environments Workshop*, pp.1–10. Doi: 10.1109/GCE.2008.4738445.

Ghemawat, S., Gobioff, H. and Leung, S-T. (2003) 'The Google file system', *ACM SIGOPS Operating Systems Review*, New York, USA.

Gill, N.K. and Singh, S. (2016) 'A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers', *Future Generation Computer Systems*, Elsevier, Vol. 65, pp.10–32.

Gopinath, S. and Sherly, E. (2018) 'A comprehensive survey on data replication techniques in cloud storage systems', *International Journal of Applied Engineering Research*, Vol. 13, No. 22, pp.15926–15932.

Guerrero-Contreras, G., Rodriguez-Dominguez, C., Diaz, S.B. and Garrido, J. (2015) 'Dynamic replication and deployment of services in mobile environments', *New Contributions in Information Systems and Technologies*, Publishing, pp.855–864.

Hameurlain, A. and Mokadem, R. (2017) 'Elastical data management in cloud systems', *Special Issue on Journal of Computer Systems Science and Engineering*, Vol. 32, No. 4.

Hassan, O., Ramaswamy, L., Miller, J., Rasheed, K. and Canfield, R. (2009) 'Replication in overlay networks: a multi-objective optimization approach', *Collaborative Computing: Networking, Applications and Worksharing*, Springer, pp.512–28.

He, L., Qian, Z. and Shang, F. (2018) 'A novel predicted replication strategy in cloud storage', *Journal of Supercomputing*, Springer, pp.1–19.

Hsu, T-Y., Kshemkalyani, A. and Shen, M. (2018) 'Causal consistency algorithms for partially replicated and fully replicated systems', *Future Generation Computer Systems*, pp.1118–1133.

Huang, K., Li, D. and Sun, Y. (2014) 'CRMS: a centralized replication management scheme for cloud storage system', *Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC)*, pp.344–348.

Hwang, K., Bai, X., Shi, Y., Li, M., Chen, W. and Wu, Y. (2016) 'Cloud performance modeling with benchmark evaluation of elastic scaling strategies', *Transactions on Power and Distribution Systems*, Vol. 27, No. 1, pp.130–143.

Jayalakshmi, D.S. and Ranjana, R.T.P. (2015) 'Dynamic data replication strategy in cloud environments', *Proceedings of the 5th International Conference on Advances in Computing and Communications (ICACC)*, pp.102–105.

Khelifa, A., Hamrouni, T., Mokadem, R. and Ben Charrada, F. (2020) 'Triadic concept analysis-based data replication strategy while satisfying tenant performance and provider profit guarantees', *International Journal of High Performance Computing and Networking (IJHPCN)*, Vol. 16, Nos. 2/3, pp.67–86.

Khelifa, A., Hamrouni, T., Mokadem, R. and Ben Charrada, F. (2021) 'Combining task scheduling and data-replication for SLA compliance and enhancement of provider profit in clouds', *Applied Intelligence*, Vol. 51, No. 10, pp.7494–7516.

Khelifa, A., Mokadem, R., Hamrouni, T. and Ben Charrada, F. (2022) 'Data correlation and fuzzy inference system-based data replication in federated cloud systems', *Simulation Modelling Practice and Theory*, Vol. 115, 102428.

Kloudas, K., Mamede, M., Preguiça, N. and Rodrigues, R. (2015) 'Pixida: optimizing data parallel jobs in wide-area data analytics', *Proceedings of VLDB*, pp.72–83.

Kumar, K.A., Quamar, A., Deshpande, A. and Khuller, S. (2014) 'SWORD: workload-aware data placement andreplica selection for cloud data management systems', *The VLDB Journal*, Vol. 23, No. 6, pp.845–870.

Lazeb, A., Mokadem, R. and Belalem, G. (2019) 'Towards a new data replication management in cloud systems', *International Journal of Strategic Information Technology and Applications (IJSITA)*, IGI Global, Vol. 10, No. 2, pp.1–20.

Lazeb, A., Mokadem, R. and Belalem, G. (2021) 'A new popularity-based data replication strategy in cloud systems', *Multiagent Grid Syst*, Vol. 17, No. 2, pp.159–177.

Lee, J., Chung, J. and Lee, D. (2015) 'Efficient data replication scheme based on hadoop distributed file system', *International Journal of Software Engineering and its Applications*, Vol. 9, No. 12, pp.177–186.

Li, C., Wang, Y., Chen, Y. and Lu, Y. (2019) 'Energy-efficient fault-tolerant replica management policy with deadline and budget constraints in edge-cloud environment', *Journal of Network and Computer Applications*, Vol. 143, pp.152–166.

Li, R., Hu, Y. and Lee, P. (2017) 'Enabling efficient and reliable transition from replication to erasure coding for clustered file systems', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 28, No. 9, pp.2500–2513.

Limam, S., Mokadem, R. and Belalem, G. (2019) 'Data replication strategy with satisfaction of availability, performance and tenant budget requirements', *Journal of Cluster Computing*, Vol. 22, pp.1199–1210.

Lin, J-W., Chen, C-H. and Chang, J-M. (2013) 'Qos-aware data replication for data-intensive applications in cloud computing systems', *IEEE Trans Cloud Computing*, Vol. 1, No. 1, pp.101–115.

Liu, G. and Shen, H. (2017) 'Minimum-cost cloud storage service across multiple cloud providers', *IEEE/ACM Transactions on Network*, Vol. 25, No. 4, pp.2498–2513.

Liu, J., Shen, H., Narman, H.S., Lin, Z. and Li, Z. (2018) 'Popularity-aware multi-failure resilient and cost-effective replication for high data durability in cloud storage', *Transactions on Parallel and Distributed Systems*, Vol. 30, No. 10, pp.2355–2369.

Liu, K., Peng, J., Wang, J., Liu, W., Huang, Z. and Pan, J. (2020) 'Scalable and adaptive data replica placement for geo-distributed cloud storages', *Transactions on Parallel and Distributed Systems*, Vol. 31, No. 7, pp.1575–1587.

Long, S-Q., Zhao, Y-L. and Chen, W. (2014) 'MORM: a multi-objective strategy for cloud storage cluster', *Journal of Systems Architecture*, Vol. 60, No. 2, pp.234–244.

Ma, K. and Yang, B. (2017) 'Stream-based live data replication approach of in-memory cache', *Concurrency and Computation: Practice and Experience*, Vol. 29, No. 11, pp.1–9.

Magalhaes, J.P. and Silva, L.M. (2013) 'A framework for self-healing and self-adaptation of cloud-hosted web-based applications', *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science*, pp.555–565.

Mansouri, N. and Javidi, M.M. (2018) 'A new prefetching-aware data replication to decrease access latency in cloud environment', *The Journal of Systems and Software*, Vol. 144, pp.197–215.

Mansouri, N. and Javidi, M.M. (2019) 'A review of replica replacement techniques in grid computing and cloud computing', *Journal of Algorithms and Computation*, Vol. 51, No. 2, pp.134–151.

Mansouri, N., Rafsanjani, M.K. and Javidi, M.M. (2017) 'DPRS: a dynamic popularity aware replication strategy with parallel download scheme in cloud environments', *Simulation Modelling Practice and Theory*, Vol. 77, pp.177–196.

Mansouri, Y. and Buyya, R. (2019) 'Dynamic replication and migration of data objects with hot-spot and cold-spot statuses across storage data centers', *Journal of Parallel and Distributed Computing*, Vol. 126, pp.121–133.

Marcus, R., Papaemmanouil, O., Semenova, S. and Garber, S. (2018) 'NashDB: an end-to-end economic method for elastic database fragmentation, replication, and provisioning', *Proceedings of the International Conference on Management of Data*, pp.1253–1267.

Milani, B.A. and Navimipour, N.J. (2016) 'A comprehensive review of the data replication techniques in the cloud environments: major trends & future directions', *Journal of Network and Computer Applications*, Vol. 64, pp.229–238.

Mokadem, R. and Hameurlain, A. (2015) 'Data replication strategies with performance objective in data grid systems: a survey', *International Journal of Grid and Utility Computing*, Vol. 6, No. 1, pp.30–46.

Mokadem, R. and Hameurlain, A. (2020) 'A data replication strategy with tenant performance and provider economic profit guarantees in cloud data centers', *Journal of Systems and Software (JSS)*, Elsevier, Vol. 159. Doi: 10.1016/j.jss.2019.110447.

Park, S.M., Kim, J.H., Ko, Y.B. and Yoon, W.S. (2004) 'Dynamic data grid replication strategy based on Internet hierarchy', *Grid and Cooperative Computing*, pp.838–846.

Pérez, J.M., García-Carballeira, F., Carretero, J., Calderón, A. and Fernández, J. (2010) 'Branch replication scheme: A new model for data replication in large scale data grids', *Future Generation Computer Systems*, Vol. 26, No. 1, pp.12–20.

Pu, Q., Ananthanarayanan, C., Bodik, P., Kanadula, S., Akella, A., Bahl, P. and Stoica, I. (2015) 'Low latency geo-Distributed data analytics', *SIGCOMM*, pp.421–434.

Ridhawi, I.A., Mostafa, N. and Masri, W. (2015) 'Location-aware data replication in cloud computing systems', *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp.20–27.

Sakr, S. and Liu, A. (2012) 'SLA-based and consumer-centric dynamic provisioning for cloud databases', *Proceedings of the IEEE International Conference on Cloud Computing*, pp.360–367.

Sakr, S., Zhao, L., Wada, H. and Liu, A. (2011) 'CloudDB AutoAdmin: towards a truly elastic cloud-based data store', *Proceedings of the IEEE International Conference on Web Services*, pp.732–733.

Séguéla, M., Mokadem, R. and Pierson, J.M. (2019) 'Comparing energy-aware vs. costaware data replication strategy', *Proceedings of the International Green and Sustainable Computing Conference (IGSC)*, pp.1–8.

Séguéla, M., Mokadem, R. and Pierson, J.M. (2021) 'Energy and expenditure aware data replication strategy', *14th IEEE International Conference on Cloud Computing* (CLOUD 2021), Chicago, IL, USA, 5–10 September, IEEE, pp.421–426.

Selvi, S.A.E. and Anbuselvi, R. (2018) 'RAAES: reliability-assured and availability-enhanced storage for cloud environment', *International Journal of Pure and Applied Mathematics*, Vol. 118, pp.103–112.

Sharma, U., Shenoy, P., Sahu, S. and Shaikh, A. (2011) 'Kingfisher: cost-aware elasticity in the cloud', *Proceedings of the IEEE INFOCOM*, pp.1–5.

Shehabi, A., Smith, S.J., Sartor, D.A., Brown, R.E., Herrlin, M., Koomey, J.G., Masanet, E.R., Horner, N., Azevedo, I.L. and Lintner, W. (2016) *United States Data Center Energy Usage Report*, Energy Technology Area, Report.

Shi, W., Wu, C. and Li, Z. (2017) 'An online auction mechanism for dynamic virtual cluster provisioning in geo-distributed clouds', IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 3, pp.677–688.

Sousa, F.R.C. and Machado, J.C. (2012) 'Towards elastic multi-tenant database replication with quality of service', *Proceedings of the IEEE International Conference on Utility and Cloud Computing*, IEEE Computer Society, Washington, pp.168–175.

Spaho, E., Barolli, A., Xhafa, F. and Barolli, L. (2015) 'P2P data replication: techniques and applications', *Modeling and Proceedings for Next Generation Big-Data Technologies*, Springer, pp.145–166.

Sun, D., Chang, G., Miao, C. and Wang, X. (2012) 'Building a high serviceability model by checkpointing and replication strategy in cloud computing environments', *Proceedings of the IEEE International Conference on Distributed Computing Systems Workshops*, pp.578–587.

Sun, S.Y., Yao, W.B. and Li, X.Y. (2018) 'DARS: a dynamic adaptive replica strategy under high load Cloud-P2P', *Future Generation Computer Systems*, Vol. 78, pp.31–40.

Tabet, K., Mokadem, R. and Laouar, M.R. (2019) 'A data replication strategy for document-oriented NoSQL systems', *International Journal of Grid and Utility Computing (IJGUC)*, Inderscience, Vol. 10, No. 1, pp.53–62.

Tabet, K., Mokadem, R., Laouar, M.R. and Eom, S. (2017) 'Data replication in cloud systems: a survey', *International Journal of Information Systems in the Service Sector*, Vol. 8, No. 3, pp.7–33.

Tan, Z. and Babu, S. (2016) 'Tempo: robust and self-tuning resource management in multi-tenant parallel databases', *Proceedings of the VLDB Endow*, Vol. 9, pp.720–731.

Tos, U., Mokadem, R., Hameurlain, A. and Ayav, T. (2021) 'Achieving query performance in the cloud via a cost-effective data replication strategy', *Soft Computing Journal*, Vol. 25, No. 7, pp.5437–5454.

Tos, U., Mokadem, R., Hameurlain, A., Ayav, T. and Bora, S. (2015) 'Dynamic replication strategies in data grid systems: a survey', *The Journal of Supercomputing*, Vol. 71, No. 11, pp.4116–4140.

Tos, U., Mokadem, R., Hameurlain, A., Ayav, T. and Bora, S. (2016) 'A performance and profit oriented data replication strategy for cloud systems', *Proceedings of the IEEE International Conference on Cloud and Big Data Computing (CBDCom)*, France, pp.780–787.

Tos, U., Mokadem, R., Hameurlain, A., Ayav, T. and Bora, S. (2018) 'Ensuring performance and provider profit through data replication in cloud systems', *Cluster Computing*, Vol. 21, No. 3, pp.1479–1492.

Van Steen, M. and Pierre, G. (2010) 'Replicating for performance: case studies: replication, theory and practice', in Charron-Bost, B., Pedone, F. and Schiper, A. (Eds): *Replications*, Springer, Berlin, pp.73–89.

Vulimiri, A., Curino, C., Godfrey, B., Padhye, J. and Varghese, G. (2015) 'Global analytics in the face of bandwidth and regulatory constraints', *Proceedings of the 12th IEEE International Conference on USENIX Symposium*, pp.323–336.

Wei, Q., Veeravalli, B., Gong, B., Zeng, L. and Feng, D. (2010) 'CDRM: a cost-effective dynamic replication management scheme for cloud storage cluster', *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER)*, pp.188–196.

Wu, Z., Butkiewicz, M., Perkins, D., Katz-Basset, E. and Madhyastha, H.V. (2013) 'Spanstore: cost-effective geo-replicated storage spanning multiple cloud services', *Proceedings of the 24th ACM Symposium on Operating Systems Principles*, pp.292–308.

Xiong, P., Chi, Y., Zhu, S., Moon, H.J., Pu, C. and Hacigumus, H. (2011) 'Intelligent management of virtualized resources for database systems in cloud environment', *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pp.87–98.

Xu, Z., Deng, N., Stewart, C. and Wang, X. (2015) 'CADRE: carbon-aware data replication for geo-diverse services', *Proceedings of the IEEE International Conference on Autonomic Computing*, Grenoble, pp.177–186.

Zakarya, M. and Gillam, L. (2017) 'Energy efficient computing, clusters, grids and clouds: a taxonomy and survey', *Sustainable Computing: Informatics and Systems*, Vol. 14, pp.13–33.

Zeng, L., Xu, S., Wang, Y., Kent, K., Bremner, D. and Xu, C. (2016) 'Toward cost-effective replica placements in cloud storage systems with qos-awareness', *Software: Practice and Experience*, Vol. 47, No. 6, pp.813–829.

Zeng, Z. and Verravelli, B. (2014) 'Optimal metadata replications and request balancing strategy on cloud data centers', *Journal of Parallel and Distributed Computing*, Vol. 74, No. 10, pp.2934–2940.

Zhang, H., Lin, B., Liu, Z. and Guo, W. (2014) 'Data replication placement strategy based on bidding mode for cloud storage cluster', *Proceedings of the IEEE International Conference on 11th Web Information System and Application Conference*, pp.207–212.

Zhang, L., Deng, Y., Zhu, W., Zhou, J. and Wang, F. (2015) 'Skewly replicating hot data to construct a power-efficient storage cluster', *Journal of Network and Computer Applications*, Vol. 50, pp.168–179.

Zhang, Y., Jiang, J, Xu, K., Nie, X., Reed, M.J., Wang, H., Yao, G., Zhang, M. and Chen, K. (2018) 'BDS: a centralized near-optimal overlay network for inter-datacenter data replication', *EuroSys*, pp.1–14.

Zhao, L., Sakr, S. and Liu, A. (2015) A framework for consumer-centric SLA management of cloud-hosted databases', IEEE Transactions on Services Computing, Vol. 8, No. 4, pp.534–549.

Notes

1 Available online at: https://cloud.google.com/sql/docs/

2 Google DC Locations. Available online at: http://www.google.com/about/datacenters/inside/locations/

3 TPC-H benchmark specification, 2019. Available online at: http://www.tpc.org/tpch/