A Performance and Profit Oriented Data Replication Strategy for Cloud System

Uras Tos, Riad Mokadem, Abdelkader Hameurlain, Tolga Ayav, Sebnem Bora

A Performance and Profit Oriented Data Replication Strategy for Cloud Systems

Uras Tos*†‡, Riad Mokadem*, Abdelkader Hameurlain*, Tolga Ayav† and Sebnem Bora‡
*Institut de Recherche en Informatique de Toulouse (IRIT), Paul Sabatier University, Toulouse, France
{tos, mokadem, hameurlain}@irit.fr
†Department of Computer Engineering, Izmir Institute of Technology, Izmir, Turkey
tolgaayav@iyte.edu.tr
‡Department of Computer Engineering, Ege University, Izmir, Turkey
sebnem.bora@ege.edu.tr

***Abstract*—In today's world, tenants of cloud systems expect timely responses to queries that process ever-increasing sizes of data. However, most cloud providers offer their services without any performance guarantees to their tenants. In this paper we propose a data replication strategy that aims to satisfy performance guarantees for the tenant while ensuring profitability of the cloud provider. Our strategy estimates the response time of the queries, as well as the expenditures that affect the profitability of the cloud provider. The decision of whether to perform replication is determined by the fulfillment of these two criteria. Validity of the proposed strategy is provided by means of a simulation study.**

***Keywords*—cloud computing, data replication, performance, economic benefit**

I. Introduction

Cloud computing has established itself as a popular computing paradigm. With datacenters filled with commodity hardware, cloud providers offer seemingly infinite amount of resources to meet ever-increasing storage and computational needs of the tenants. Moreover, the set of resources are abstracted and delivered in an economy-based manner [1]. This shift towards the economy based systems brings new challenges to the interactions between cloud providers and tenants. As the tenant requirements change, cloud resources are elastically adjusted and the monetary cost for the tenant is determined according to the pay-as-you-go model [2]. As expected from any economic enterprise, cloud providers aim to maximize their profits. At the same time, tenants expect cloud providers to keep a certain, agreed upon set of *service level objectives* (SLO), defined in a *service level agreement* (SLA) [3]. SLA is a legally binding contract between the cloud provider and tenant. In case of an SLA breach, the provider pays a penalty to the tenant.

Data replication is a very well-known optimization technique that has been commonly adopted by many traditional systems, including (i) database management systems (DBMS) [4], (ii) parallel and distributed systems [5], (iii) mobile systems [6] and (iv) other large-scale systems including P2P [7] and data grid systems [8]. Benefits of data replication include increasing data availability, improving performance, and achieving fault tolerance. In a data replication strategy three major questions of what to replicate, when to replicate, and where to replicate must be answered [9]. In the traditional systems, many available replication strategies create as many replicas as possible to achieve maximum performance. However, such an approach may not be economically feasible for the provider in cloud computing, since the creation of an unnecessarily high number of replicas can result in degraded performance and reduced profit. Hence, new replicas should be added in order to satisfy SLA requirements, while the removal of replicas occurs when these objectives are satisfied over time.

There is a number of efforts in the literature that studied data replication in the cloud systems. Many of them focus just on satisfying the availability SLO [10], [11]. In a typical cloud environment, where frequent queries are placed on a large-scale data, having low response time is crucial for the tenants. However, performance guarantees, e.g. response time, are not offered by cloud providers as a part of the SLA. In order to resolve this issue, there are several works proposed [12], [13] in the literature to include the response time guarantees in the SLA. Dealing with data replication, only a few studies are particularly interested in improved response time [14]–[17]. In addition, even fewer of those studies [18], [19] are taking economics of the cloud into account. To the best of our knowledge, lack of performance SLO is also true for the commercial clouds offered by Amazon[1], Google[2], and Microsoft[3].

In this paper we propose *Performance and Profit Oriented Data Replication Strategy* (PEPR) that ensures SLA guarantees, e.g. availability and performance, to the tenant while maximizing the economic benefit of the cloud provider. For the measure of performance, we consider response time guarantee as an integral part of the SLA. In PEPR, when evaluating a query, if an estimated response time value is greater than the SLO response time threshold, this means that a replication process may be triggered. At that time, economic benefit, i.e. profitability, of the cloud provider is also estimated. Replication decision is made only when both the response time and economic benefit of the provider are satisfied. In replica placement, new replicas are placed on the cloud node that is closest to the most amount of queries. The number of replicas is dynamically adjusted following whether the SLA objectives are satisfied over time. Moreover, a minimum number of replicas are always kept to ensure minimum availability [14].

[1]http://aws.amazon.com/s3/sla/
[2]https://cloud.google.com/storage/sla
[3]http://azure.microsoft.com/en-us/support/legal/sla/

We evaluated performance of PEPR with a simulation study. In the simulations, we show that PEPR not only satisfies the SLA, but also ensures profitability of the cloud provider.

The organization of the rest of the paper is as follows. Section II provides a summary of related work on data replication strategies in the cloud. Section III gives some background information on the aspects of the cloud that are relevant to our study. Sections IV and V describes the details of PEPR and the implemented economic cost model, respectively. Section VI evaluates the performance of PEPR. Section VII concludes the paper and reflects on possible future work.

II. Related work

In this section we give an overview of some existing works on data replication in the cloud. These include studies that are interested in the economic aspect of the cloud or improved response time of queries.

Wei et al. [14] proposed Cost-effective Dynamic Replication Management (CDRM) for increasing availability in cloud storage. They make a justification that having too many replicas does not increase availability but results in higher costs. Therefore, the main contribution of CDRM is finding a minimal number of replicas to satisfy the availability requirement. Authors mention that nodes can only serve a limited number of queries and overloaded nodes are blocked from new query arrivals. As a result, replica placement is performed in such a way that the nodes with lowest blocking probability will host the new replicas.

Bonvin et al. [18] presented Skute, a scattered key-value store. In Skute, virtual nodes act as autonomous agents and decide on behalf of the data owner, without any control from the outside. Proposed economic model is based on a virtual economy. Nodes pay rent to other nodes to host replicas according to storage use and query load. In addition, nodes also generate revenue by the amount of queries they answer. In the proposed work, first the availability and then the net benefit is considered by placing replicas to nodes with respect to their economic fitness. Even though performance guarantees are not regarded as an integral part of the SLA used in Skute, reduced average query load per node is achieved over time.

Sakr and Liu [13] introduced an SLA based provisioning strategy for cloud databases. Their approach is a customer-centric strategy, in which the database servers are scaled in and out according to the SLA requirements. As the main SLA objective for the decision process, they chose the total execution time of transactions. In the proposed strategy, cloud system is closely monitored and cloud providers declaratively define application specific rules to adaptively scale the resources. While the SLA-aware provisioning is beneficial for scaling, economic impact of the replication for the cloud provider is not mentioned.

Janpet and Wen [16] designed a data replication strategy to minimize data access time by finding the shortest access path to data objects. They model access frequency, delay and replication budget to find the closest, most suitable node for replica placement. Replication budget is predefined and it is only used as a limiting factor for the users in such a way to regulate number of replicas. A detailed economic relationship between the users and the cloud provider is not addressed. The experimental study shows that by placing data objects closer to the nodes with high access frequency, response time is improved.

Kumar et al. [20] proposed SWORD, a workload-aware data placement and replica selection scheme. Authors introduced a new metric named query span, which is the average number of nodes used in execution of a query. Their approach aims to minimize query span in order to reduce the communication overhead, resource consumption, energy footprint, and transaction cost. They claim that SWORD deals with performance degradation with incremental repartitioning of data. Although provider profit is not a focus of this study, the authors show the effectiveness of their work by doing an experimental analysis to measure query span and transaction times.

Zhang et al. [17] present an auction model to implement a replica placement policy. Proposed work aims to satisfy only availability in a large-scale cloud storage environment. If the desired availability level cannot be maintained, a bidding is held to determine the placement for a new replica. Bidding price is dependent on several properties of the nodes including failure probability, network bandwidth and available space. While response time is not included in the objective function, in the experiments authors observe that performance is improved alongside satisfied availability.

Sousa and Machado [21] proposed RepliC, an elastic multi-tenant database replication strategy. RepliC takes performance SLA into account and elastically adjusts the number of replicas by monitoring the system utilization. When workload changes, RepliC can handle the variation by directing transactions to the replicas with available resources. In an experimental study, RepliC is compared with a rule-based scaling scheme. The results indicate that the proposed strategy satisfies QoS with minimal SLA violations.

Boru et al. [22] introduce a data replication strategy that focuses on improving the energy efficiency of cloud datacenters. Their strategy optimizes energy consumption, bandwidth use and network delays at both inter-datacenter and intra-datacenter levels. The authors modeled datacenter power usage and bandwidth consumption of database operations. A periodical analysis determines the replication decision and estimates the power and bandwidth usage of the replicas in the upcoming periods. With a simulation study they showed that by placing replicas closer, power consumption and response time is improved. Economic benefit however, is not a focus of this study.

III. Background

Cloud providers often establish multiple facilities in separate geographical regions for a multitude of reasons, including providing services that span across the globe. Each region may contain several other subregions that are distributed inside a region. These subregions are cloud facilities that host a number of nodes that provide computational power and storage to the tenants. All nodes in the cloud are interconnected by network links. Nodes in the same subregion of a particular geographical region are interconnected via local and relatively cheaper links. In a similar manner, intra-region bandwidth is comparatively

less abundant and more expensive. As the network hierarchy goes from inter-node links to inter-region links, bandwidth abundance decreases and bandwidth cost increases [23]. A typical example of this cloud hierarchy is depicted in Figure 1.

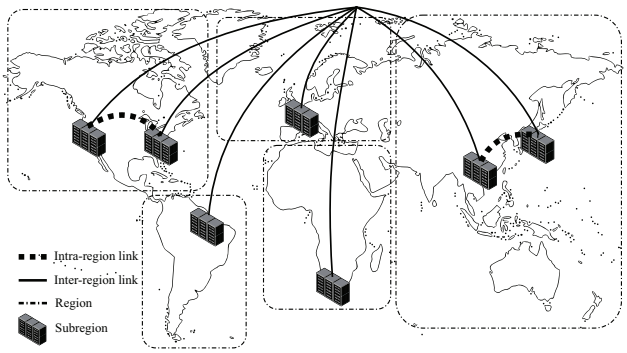

Fig. 1. An example cloud topology showing regions and subregions. Nodes and links inside subregions are not depicted in this figure.

Tenants utilize the services they rent from the provider by placing queries to the cloud. These queries require data sets that may reside on multiple cloud nodes scattered around different geographical regions. From the tenant's perspective, it is essential for the response time of an average query to be within the threshold defined in the SLA. The cloud provider aims to satisfy the SLA with the maximum amount of profit. The essence of the proposed strategy contains two models. Former is the cost model based on a response time estimation and latter is the economic cost model based on a provider profit estimation. It should be noted that, processing data in the cloud has many challenges including distributed execution and partitioning. The focus of our study is only on the data replication aspect of cloud data.

An important characteristic that differentiates the cloud from traditional business models is the penalty mechanism. Should an SLA breach occur, the provider is obligated to pay an agreed upon monetary sum to the tenant [12]. In our case, when the actual response time is found to be greater than the threshold defined in the SLA, there exists an SLA violation. It is therefore important to note that, penalties play an important role in the economics of the cloud.

IV. Data replication strategy

Replication strategies are expected to determine *what* data are concerned by the replication, *when* a replica should be created/deleted, *how many* replicas to create and *where* to place the replicas [9]. Creating as many replicas as possible cannot be economically feasible in cloud systems. Hence, PEPR gives a prime importance to issues such as timing of starting the replication, keeping an optimal number of replicas, a good replica placement and economic profitability.

A. When to replicate

In PEPR, a replication event occurs if and only if it is necessary to replicate to meet the response time guarantee and it is a profitable action for the provider (Figure 2). At the beginning of the query execution, when the query is submitted to the node, response time estimation is calculated for that particular query. If the estimated response time is greater than the response time threshold (denoted $\mathcal{T}_{SLO}$), namely the response time SLO, the required remote data set may be required to be replicated. However, response time estimation is not enough on its own to trigger a replication. In the second step, the replication of that particular remote data set is evaluated from an economic standpoint by profit estimation model. Calculated profit estimation indicates, if the provider would still be profitable after a particular replication event is carried out. If and only if the provider is still estimated to be profitable, the replication event starts. Decision to trigger a replication involves both the response time and provider profit estimations.

1: $\mathcal{T}_{\mathcal{Q},n} \leftarrow$ estimated response time of executing $\mathcal{Q}$ on node n
2: **if** $\mathcal{T}_{\mathcal{Q},n} > \mathcal{T}_{SLO}$ **then**
3: $p \leftarrow$ estimated profit by placement of new replica
4: **if** $p > 0$ **then**
5: PlaceReplica()
6: **end if**
7: **end if**

Fig. 2. Replication decision algorithm.

1) Response time estimation: Queries are placed to the cloud for processing. In order to find the estimated response time ($\mathcal{T}_{\mathcal{Q},n}$) of a particular query ($\mathcal{Q}$) executed on a node (n), it is necessary to evaluate the amount of time contributed by computation, I/O, and network. Any particular query may require data sets from both the local node (n) and a number (i) of remote nodes. In the case when some remote data (d) is required, it is necessary to access each remote data set with the I/O throughput of t from the remote nodes and migrate each over a bandwidth (b) to the local node. Once the required total data ($\mathcal{D}$) is ready to be processed on node n, the query is executed. During query execution, a computational load of l_n is observed on node n, alongside with the I/O throughput capability of local node (t_n). α is a variable denoting the overhead of executing the query on node n and variation between queries. While response time estimation for similar scenarios has been studied in traditional systems [5], Equation 1 shows the proposed response time estimation.

$$\mathcal{T}_{\mathcal{Q},n} = (\frac{\mathcal{D}l_n}{t_n} + \sum_{1}^{i}(\frac{d_i}{b_i} + \frac{d_i}{t_i}))\alpha \quad (1)$$

Response time estimation takes processing of the query and data transfer into account. As a result, $\mathcal{T}_{\mathcal{Q},n} > \mathcal{T}_{SLO}$ condition may occur in two situations. First, during the execution of the query, some portion of total data is shipped from remote nodes to the node that is executing the query. If the nodes that host the remote data are in a region that is accessed via a low bandwidth availability, this will have an impact on data transfer times and in turn the response time. Therefore, it is beneficial to replicate data sets that reside in remote nodes. Second, the average load of the node that executes the query directly contributes to the response time of the query. Normally, queries are not redirected to the nodes with high average load. However, if all replicas of any particular data set reside on nodes with high average

load, these data sets may be replicated to new locations with low system load to prevent over-utilization.

2) Profit estimation: In PEPR, we assume that the tenant pays for a particular service with an agreed upon quality. As a result, for any amount of revenue, the provider must pursue the route of most economical replication decisions in order to decrease expenditures. In other words, ensuring profitability for the provider lies in minimizing expenditures (Equation 2).

$$profit = revenue - expenditures \quad (2)$$

Before the execution of a query, the expenditure of the provider is calculated alongside the response time estimation. Calculated expenditure is compared against the revenue to estimate the profit of provider.

B. How many replicas

Regarding the decision of how many replicas of a data set to maintain a system, the first critical issue is maintaining availability. While PEPR aims to satisfy response time SLO, keeping availability SLO cannot be ignored. As discussed in Section II there are many existing studies in the literature that focus on satisfying availability SLO. In PEPR, we create a minimum number of replicas to maintain a minimum given availability [14] at the initial placement stage of data sets. At no point in time, the number of replicas for any particular data set is allowed to decrease below this minimum number of replicas.

How many replicas of each data set to keep in the system do not have a strictly defined upper limit. As long as satisfying SLA requires so and it is still profitable to have more replicas, new replicas will be created. In PEPR, we implement an incremental approach for creating replicas. In other words, at each step, one replica is created. As a result, the degree of replication is merely a consequence of the data replication strategy. Therefore, the number of replicas is not a statically defined number indicating the degree of replication, but rather a dynamically performed adjustment.

```
1: updateAccessCounts()
2: actualRespTime ←measureRespTime(Q)
3: if actualRespTime > T_SLO then
4:    slaBreached ← true
5: end if
6: if ExecutedQCount = QPerEpoch and
   slaBreached = true then
7:    slaBreached ← false
8:    ExecutedQCount ← 0
9:    removeLeastRecentlyUsed(Num_LRU)
10: end if
```

Fig. 3. Replica retirement algorithm.

While creation of new replicas are straightforward, replica retirement operation requires some apriori information. In essence, replica retirement is performed as the SLA is satisfied over time. After execution of each query, the actual response time of that particular query is logged. In addition, how many times a particular data set has been accessed is also recorded in a list, ordered by the number of accesses. Let $QPerEpoch$ show a system parameter that indicates the number of queries that defines an epoch. After every $QPerEpoch$ queries executed, if there are no SLA breaches observed in that epoch, a predefined number of least recently used replicas, denoted by Num_LRU, are removed from the system. Keeping unused replicas in the cloud would result in wasted storage space and in turn, unnecessary expenditure for the provider. Using this approach, the cloud system will converge to keep replicas that only have high number of access. Figure 3 is evaluated after each query $\mathcal{Q}$ and performs the described retirement tasks, after each epoch.

C. Where to replicate

Given that the creation of a new replica is determined to be necessary, finding a node to place the new replica is the next issue to address. Similar to the previous replication decisions, this task is performed with the consideration of response time and profit estimation.

Finding an optimal placement should be done in a timely fashion to minimize the overhead caused by the search. Searching among all of the nodes in the cloud may take a significant time, resulting in violation of response time SLO. Taking just the nodes in the local region into account reduces the search space for finding optimal placement. As a result, PEPR evaluates the nodes only in the local region to come up with the placement for the new replica. Also, in terms of bandwidth availability, response time guarantees can be met when the remote data is replicated closer to the requesting nodes. Ideally, required data is preferred to be placed on the node executing a particular query. However, if that is not possible, the reality may require placing replicas in the same subregion or at least in the same region. Increased data locality will improve data transfer times and positively affect response times. Moreover, the system load of the query executing nodes is also important. High average system load of the candidate node for replica placement is may to prevent meeting the response time SLO by resulting in I/O bottleneck.

```
1: i ← 0
2: while i do
3:    N ← getNeighborsAtHops(n, i)
4:    for each node m in N do
5:       if m.freeSpace ≥ sizeOf(d) then
6:          if m.load ≤ loadThreshold then
7:             place replica of remote data d on m
8:             terminate
9:          end if
10:      end if
11:   end for
12:   increment i
13: end while
```

Fig. 4. Replica placement algorithm.

In PEPR, bandwidth capability, storage space and computational load of the nodes are evaluated to find a placement. The search for optimal placement starts from the requestor node itself, and continues firstly in the local subregion and then the other subregions in the same region until a placement location is found. At each step, each node is checked to determine

whether it has the necessary storage space. If a node with enough available storage to host a new replica is found, it is then checked for the computational load. As placing a new replica on a particular node will result in more load for that node, only a node having less load than a predefined load threshold (denoted $loadThreshold$) is selected. At this point, the selected node is considered the best node to place the replica. Details of the process of selecting a node for replica placement is given in Figure 4.

It is important to mention that, at any time some data migrate, there is a cost associated with the migration. In data migration, bandwidth is consumed during transfer and storage is consumed at the destination. Monetary aspect of the migration cost is dealt with the profit estimation by regarding migration cost a part of bandwidth and storage costs. In addition, data migration caused by pre-replicating data also results in an adverse effect on response time of execution of a query.

V. Economic cost model

Unlike a one-time payment for an unlimited use of the software in a traditional business model, a cloud tenant pays the provider only for its own consumption [1]. For this aim, an efficient provider economic cost management is required. That leads us to realize an economic cost model that deals with the estimation of expenditures of the provider when evaluating a query.

A. Provider's revenue

A tenant periodically pays the rent associated with the services acquired from the provider. Exact amount to pay is determined according to the utility metrics defined and monitored during a billing period. Hence, the revenue of a provider when evaluating a query is essentially composed of the amount received from the tenant.

Considering that the provider offers various service plans with different capability levels, we assume that both parties agree upon the most suitable one for a particular application of the tenant. In the case when the tenant is supplied with an inadequate service level, obviously it would be problematic to satisfy the SLA. For this reason, in our strategy we assume that the tenant is supplied with a suitable service which is scaled according to the changing requirements.

It is worthy of noting that, data replication is transparent to the tenant. Therefore, tenant is not billed for the number of replicas kept. It is the provider's responsibility to maintain the replicas in the cloud to satisfy the SLA terms. The tenant pays for the utilized service, and does not care about how many replicas the provider needs to create to meet the demand caused by the query load.

When a provider serves several tenants, it receives revenues from all served tenants. It is obvious that increasing the number of tenants decreases the per tenant performance but reduces the overall operating cost of the provider [24].

B. Provider's expenditures

As an inevitable part of doing business, cloud provider has a number of expenditures. In PEPR, we separately deal with various types of costs that contributes toward the total expenditure. Cloud provider hosts a number of nodes to handle the queries from tenants. These nodes require power and other support hardware to function properly. These expenses contribute towards the computational investment ($\mathcal{C}_i$). Another type of cost is related to network usage ($\mathcal{C}_b$). The data required by the queries perpetually shipped over network links to various destinations in the cloud, globally. Whether it is for accessing remote data set without replication or migrating data for creating a replica, it all requires precious bandwidth resource. Storage is another cost item ($\mathcal{C}_s$) that the provider deals with. Creating new replicas will consume storage space on the nodes and increase the expenditure.

For any replication event, it is possible to comment on the change in any particular type of cost. For example, creation of a new replica will result in a certain amount of increased storage use. However, it is relatively harder to comment on how a change in one unit of expenditure imposes a change in another. Moreover, this relationship between different types of expenditures can vary from one provider to another. Therefore, each individual cost type are treated separately from each other.

On the cost of computational utility, let $T_{\mathcal{Q}}$ be the estimated total time needed to evaluate a query $\mathcal{Q}$. Let $n_{\mathcal{Q}}$ be the number of nodes required when evaluating $\mathcal{Q}$ during a unit time $\mathcal{C}_t$ (e.g. one hour on a node). Please note that, this includes nodes that hold the replicas created when evaluating $\mathcal{Q}$ [21].

$$expenditures = \mathcal{C}_i + \mathcal{C}_b + \mathcal{C}_s + \sum_{1}^{n_{\mathcal{Q}}} T_{\mathcal{Q}}\mathcal{C}_t + \mathcal{P}_{tenant} \tag{3}$$

Provider pays penalties to the tenants ($\mathcal{P}_{tenant}$) if one or more SLOs are not satisfied. As a result, in addition to the operational costs, penalties are also included in the provider cost. Through data replication, the provider aims to minimize the penalties paid to reduce cost [25]. In regard to described cost types, estimation of provider cost, i.e. expenditures, is given in Equation 3.

VI. Performance evaluation

A. Simulation environment

Using a well-designed simulation tool is an accurate way of implementing performance evaluation scenarios without the burden and cost of dealing with an actual cloud environment. For this purpose, several cloud simulators [26] have been used in the literature. Among these simulators, most popular choice is the CloudSim [27]. Being an open source project, CloudSim is also relatively easily extensible.

Originally, CloudSim is suitable for simulation scenarios that require virtual machine (VM) provisioning, task distribution, and power management. However, data replication functionality is not as advanced. As a result, we extended CloudSim to support the requirements of data replication and placement tasks.

In the simulations, we realized a hierarchical cloud environment as described at the beginning of Section III. Simulated cloud consists of three geographical regions. Each region

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Number of regions	3
Number of datacenters per region	2
Number of VMs per datacenter	10
VM processing capability	1000 MIPS
VM storage capacity	20 GB
Intra-datacenter bandwidth	40 Gbit/s
Intra-region bandwidth	5 Gbit/s
Inter-region bandwidth	1 Gbit/s
Avg. Intra-datacenter delay	5 ms
Avg. Intra-region delay	25 ms
Avg. Inter-region delay	100 ms
Response time SLO	180 s
Simulation duration	10 min
Number of queries processed	10000
Query arrival rate	16.67 query/s
Avg. computational load of a query	2500 MI
Number of data sets	20
Avg. size of a data set	550 MB
Intra-datacenter data transfer cost	\$0.05 per GB
Intra-region data transfer cost	\$0.005 per GB
Inter-region data transfer cost	\$0.0001 per GB
Storage cost	\$0.1 per GB
Processing cost	\$1 per 10^6 MI
Penalty cost	\$0.01 per violation
Revenue	\$75 per billing period
Num_LRU	1
$QPerEpoch$	10
$loadThreshold$	90%

contains two subregions represented as datacenters. In turn, each datacenter contains ten nodes realized as VMs. Network topology is established in such a way that, the bandwidth capacity is more abundant and cheaper inside the datacenters but less abundant and more expensive towards the inter-region level. Details of this cloud topology are depicted in Figure 5. Also a complete list of simulation parameters can be found in Table I. We chose the simulation parameters to be in accord with existing studies [28] to realistically represent a typical cloud environment.

Each VM has the necessary computational resources to perform execution of queries. These resources include CPU, RAM, storage, and network connectivity. In order to make an accurate comparison, we use this same computational investment for all evaluated strategies. During execution of queries, VMs can access the data sets in other VMs by remote reads or by replicating them to local storage.

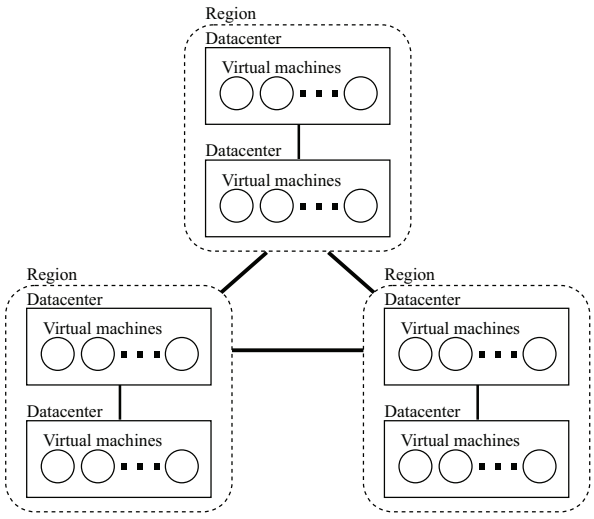

Fig. 5. Simulated cloud topology.

The duration of the simulation is regarded as one billing period. For that period, the tenant is charged for the services obtained, and provider profit is determined by the expenditures of the provider during this billing period. In the simulation, the queries are placed to the cloud with random arrival times. Each query requires a randomly determined data set to be retrieved for processing. The aim is to monitor how the replication strategies cope with the query load to keep the response time SLO and observe provider profit in a billing period.

B. Simulation results

In the simulations, we compared PEPR against no replication (i.e. full remote reads) and *Bandwidth Hierarchy Replication* (BHR) [23] strategies. No replication strategy is used as a baseline to show the impact of replication by how the profit and response time are affected by only performing remote reads. BHR is a data replication strategy that is aimed towards traditional systems in which, the provider profit is not taken into account while making replication decisions.

We measured average response time, the average of actual response times of all queries processed during the simulation, in addition to the total number of replications, number of SLA violations, storage usage, and the amount of data transferred over network as the evaluation metrics. Each type of cost associated with these metrics are calculated according to the simulation parameters. Table II shows the obtained values for each of these metrics, and the total provider cost.

As its name suggests, the no replication strategy did not create any replicas, therefore its storage use only shows the space occupied by the initial data placement. Among the other two strategies, PEPR created less than half of the number of replicas created by BHR. Consequently, this determines the storage use of these strategies. PEPR used almost half of the storage space consumed by BHR, while BHR filled more than half of the entire storage space available in the cloud.

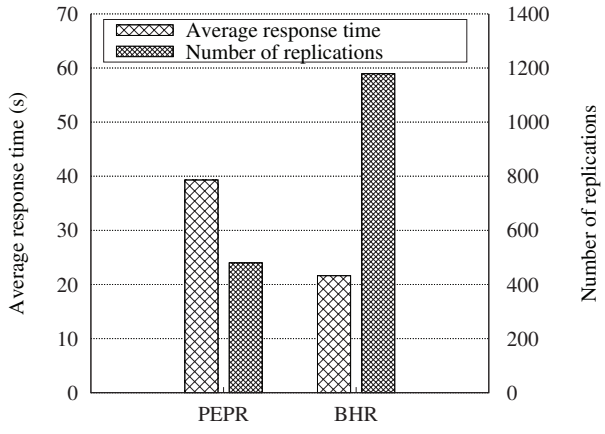

Fig. 6. Average response time and the number of replications.

TABLE II. SIMULATION RESULTS.

Replication strategy	Avg. response time (s)	Number of replications	SLA violations	Storage usage (%)	Inter-region data transfer (GB)	Intra-region data transfer (GB)	Intra-datacenter data transfer (GB)	Total expenditures of the provider ($)
No Replication	630.83	0	7754	0.91	3648.63	915.00	901.17	278.99
BHR	21.62	1179	266	54.79	21.92	382.78	5121.93	85.19
PEPR	39.31	480	259	28.36	21.92	1274.80	4165.58	60.01

In terms of average response time, both PEPR and BHR satisfied the response time SLO while the no replication strategy failed in that respect. BHR achieved slightly better average response time (88% less than the response time SLO) compared to PEPR (78% less than the response time SLO) due to having a much greater number of replicas in the cloud, as depicted in Figure 6. However, it is important to highlight that, the provider's aim is not to get the best response time but instead, to satisfy the SLA with most amount of profit. Please note that, the no replication strategy is omitted in Figure 6 to retain a better axis range for the comparison of other two strategies.

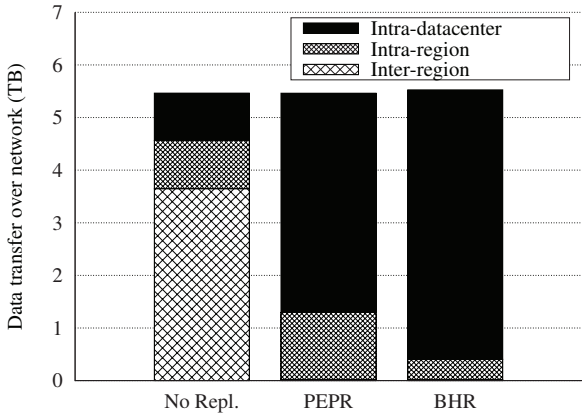

Fig. 7. Breakdown of total data transfers with respect to network hierarchy.

By creating no replicas, most queries processed with the no replication strategy required remote read operations from different regions. On the other hand, PEPR and BHR created replicas and benefited from reduced access time, expectedly. As depicted in Figure 7, PEPR and BHR made the majority of the network transfer at the intra-datacenter level by using replicas in the regions where bandwidth is cheaper. Again, BHR having immensely more amount of replicas, almost always had a replica of required data set in the same datacenter. On the other hand, PEPR more frequently chose to access data sets available in the other datacenters in the same region, as long as the response time is satisfied. In other words, assuming that response time SLO is satisfied, if accessing a data set in another datacenter is more profitable than creating a new replica in local datacenter, PEPR took such an action to achieve optimality.

Figure 8 depicts the total monetary cost of each strategy in the simulated billing period. The accumulated costs are calculated from the unit costs defined as the simulation parameters. All replication strategies are evaluated by the queries generated from the same random distribution, therefore the computational

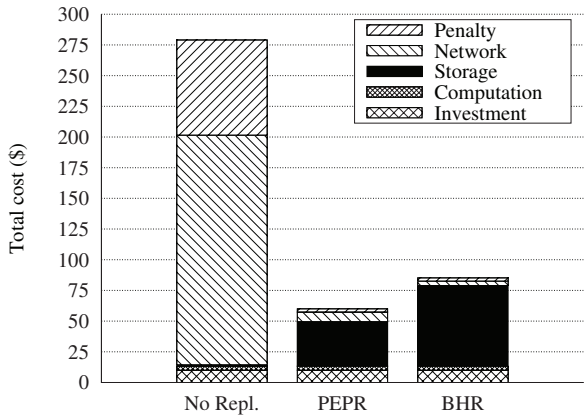

Fig. 8. Total cost (monetary expenditure) of the provider during one simulated billing period.

costs are the same for all strategies. A large sum of inter-region data transfers by the no replication strategy caused a significant network cost. Furthermore, frequent violation of SLA results in a high penalty cost with the no replication strategy. Among the other two strategies, number of SLA violations are very low. As a result, the penalty cost is at a more acceptable level.

In terms of storage costs, no replication strategy only kept the initially placed data sets, hence the storage cost is almost negligible. BHR strategy created a large number of replicas that in turn resulted in a substantial storage cost for the provider. Compared to BHR, PEPR caused less then half the storage cost of BHR. PEPR accumulated slightly more network costs, however this can be regarded as a trade-off move to save on storage costs. When total costs are considered, the PEPR yielded 30% less provider cost than BHR strategy. This difference in cost amount directly contributes towards the profitability of the provider.

Results of the experimental evaluation indicate that, while a data replication strategy is essential to satisfy performance guarantees, simply using a traditional strategy is not enough for the cloud provider to ensure economic benefit. Traditional strategies tend to eagerly replicate to attain best possible performance. However, in pursuit of the best performance, traditional strategies consume more storage and increase provider costs. As a result, as long as the performance SLO is satisfied, it is important to focus on improving the economic benefit of the provider. PEPR works towards this aim by not increasing provider costs due to unnecessary replication once the performance SLO is met.

VII. Conclusion

In this study, we proposed PEPR, *Performance and Profit Oriented Data Replication Strategy* for cloud systems. PEPR not only satisfies the SLA, but also ensures the profitability of the provider. These two criteria are at the core of our replication strategy. As the tenants place queries to the cloud, PEPR estimates the response time of each query and evaluates whether the response time can be satisfied or not. In the case where the estimated response time is greater than the response time SLO, PEPR calculates a profit estimation for creating a replica for that particular data set. Only if the creation of the new replica is profitable, the replication is performed. When response time is satisfied over time, older replicas that are not used are retired from the cloud. We evaluated the validity of PEPR with an experimental evaluation. In a simulation environment, we compared PEPR with two other strategies, a traditional strategy that does not take economics of the cloud into account and a no replication strategy. While both PEPR and the the traditional strategy satisfied the SLA, only PEPR ensured the profitability of the provider. The results confirm that the consideration of economics of the cloud should be an important consideration in any data replication strategy that aims to perform in a cloud environment.

A possible future direction for this study is to take queries that include dependent operations, e.g. join operations, into account. This way, the proposed strategy would be applicable to database queries as well. Hence, we deferred performance comparison with existing database oriented replication strategies to a future study. Moreover, implementing the proposed strategy in a real cloud environment may also present an interesting research opportunity.

Acknowledgment

The work presented in this paper is supported in part by TUBITAK.

References

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *2008 Grid Computing Environments Workshop*. IEEE, Nov. 2008, pp. 1–10.

[2] M. Armbrust, I. Stoica, M. Zaharia, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, and A. Rabkin, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, p. 50, 2010.

[3] V. Stantchev and C. Schröpfer, "Negotiating and enforcing QoS and SLAs in grid and cloud computing," in *Advances in grid and pervasive computing*, 2009, pp. 25–35.

[4] B. Kemme, R. Jimenez-Peris, and M. Patino-Martinez, *Database Replication*. Morgan and Claypool Publishers, 2010.

[5] M. T. Özsu and P. Valduriez, *Principles of Distributed Database Systems*, 3rd ed. Springer New York, 2011.

[6] G. Guerrero-Contreras, C. Rodriguez-Dominguez, S. Balderas-Diaz, and J. Garrido, "Dynamic replication and deployment of services in mobile environments," in *New Contributions in Information Systems and Technologies*. Springer International Publishing, 2015, pp. 855–864.

[7] E. Spaho, A. Barolli, F. Xhafa, and L. Barolli, "P2P data replication: Techniques and applications," in *Modeling and Processing for Next-Generation Big-Data Technologies*. Springer International Publishing, 2015, pp. 145–166.

[8] U. Tos, R. Mokadem, A. Hameurlain, T. Ayav, and S. Bora, "Dynamic replication strategies in data grid systems: a survey," *J. Supercomput.*, vol. 71, no. 11, pp. 4116–4140, 2015.

[9] K. Ranganathan and I. Foster, "Identifying dynamic replication strategies for a high-performance data grid," in *Proc. Int. Grid Computing Workshop*, vol. 2242. Springer, 2001, pp. 75–86.

[10] G. Silvestre, S. Monnet, R. Krishnaswamy, and P. Sens, "AREN: A popularity aware replication scheme for cloud storage," in *IEEE 18th Int. Conf. Parallel and Distributed Systems*. IEEE, Dec. 2012, pp. 189–196.

[11] D. Sun, G. Chang, S. Gao, L. Jin, and X. Wang, "Modeling a dynamic data replication strategy to increase system availability in cloud computing environments," *J. Comput. Sci. Technol.*, vol. 27, no. 2, pp. 256–272, Mar. 2012.

[12] Y. Kouki, T. Ledoux, and R. Sharrock, "Cross-layer SLA selection for cloud services," in *1st Int. Symp. Network Cloud Computing and Applications*. IEEE, Nov. 2011, pp. 143–147.

[13] S. Sakr and A. Liu, "SLA-based and consumer-centric dynamic provisioning for cloud databases," in *IEEE 5th Int. Conf. Cloud Computing*. IEEE, Jun. 2012, pp. 360–367.

[14] Q. Wei, B. Veeravalli, B. Gong, L. Zeng, and D. Feng, "CDRM: A cost-effective dynamic replication management scheme for cloud storage cluster," in *IEEE Int. Conf. Cluster Computing*. IEEE, Sep. 2010, pp. 188–196.

[15] X. Bai, H. Jin, X. Liao, X. Shi, and Z. Shao, "RTRM: A response time-based replica management strategy for cloud storage system," in *Grid and Pervasive Computing*, 2013, no. 1, pp. 124–133.

[16] J. Janpet and Y.-F. Wen, "Reliable and available data replication planning for cloud storage," in *IEEE 27th Int. Conf. Advanced Information Networking and Applications (AINA)*. IEEE, Mar. 2013, pp. 772–779.

[17] H. Zhang, B. Lin, Z. Liu, and W. Guo, "Data replication placement strategy based on bidding mode for cloud storage cluster," in *11th Web Information System and Application Conf.*, 2014, pp. 207–212.

[18] N. Bonvin, T. G. Papaioannou, and K. Aberer, "A self-organized , fault-tolerant and scalable replication scheme for cloud storage categories and subject descriptors," in *Proc. 1st ACM Symp. Cloud computing - SoCC '10*, 2010, pp. 205–216.

[19] H. Ghanbari, B. Simmons, M. Litoiu, and G. Iszlai, "Feedback-based optimization of a private cloud," *Future Gener. Comp. Sy.*, vol. 28, no. 1, pp. 104–111, Jan. 2012.

[20] K. A. Kumar, A. Quamar, A. Deshpande, and S. Khuller, "SWORD: workload-aware data placement and replica selection for cloud data management systems," *VLDB J.*, Jun. 2014.

[21] F. R. C. Sousa and J. C. Machado, "Towards elastic multi-tenant database replication with quality of service," in *IEEE/ACM 5th Int. Conf. Utility and Cloud Computing, UCC 2012*, 2012, pp. 168–175.

[22] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Comput.*, vol. 18, no. 1, pp. 385–402, 2015.

[23] S. M. Park, J. H. Kim, Y. B. Ko, and W. S. Yoon, "Dynamic data grid replication strategy based on internet hierarchy," in *Grid and Cooperative Computing*. Springer Berlin Heidelberg, 2004, pp. 838–846.

[24] W. Lang, S. Shankar, J. M. Patel, and A. Kalhan, "Towards multi-tenant performance SLOs," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 6, pp. 1447–1463, 2014.

[25] P. Xiong, Y. Chi, S. Zhu, H. J. Moon, C. Pu, and H. Hacigumus, "Intelligent management of virtualized resources for database systems in cloud environment," in *IEEE 27th Int. Conf. Data Engineering (ICDE)*. IEEE, 2011, pp. 87–98.

[26] A. Ahmed and A. S. Sabyasachi, "Cloud computing simulators: A detailed survey and future direction," in *Proc. IEEE Int. Advance Computing Conf., IACC 2014*, 2014, pp. 866–872.

[27] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011.

[28] L. A. Barroso, J. Clidaras, and U. Hölzle, *The Datacenter As a Computer: An Introduction to the Design of Warehouse-Scale Machines*, 2nd ed. Morgan and Claypool Publishers, 2013.