

A Cost Model for Hybrid Storage Systems in a Cloud Federations

Amina Chikhaoui, Kamel Boukhalifa
University of Science and Technology Houari
Boumediene, Algiers, Algeria
{achikhaoui, k.boukhalifa}@usthb.dz

Jalil Boukhobza
Univ. Bretagne Occidentale
UMR 6285, Lab-STICC F-29200, Brest, France
boukhobza@univ-brest.fr

Abstract—A cloud federation gives to cloud service providers (CSP) the opportunity to collaborate in order to offer a better QoS to customers at a lower cost. To do so, CSPs make some spare resources available to others at a reduced cost. One of the most critical resources is the storage system as it represents the main system bottleneck. From this point of view, how to efficiently place data in a federation of Clouds with heterogeneous storage systems is a real challenge. To address this issue, one needs to accurately estimate the data placement cost. In this paper, we propose a cost model for hybrid storage systems in a cloud federation for a Database as a Service (DBaaS) application. It takes into account the storage system characteristics, customers I/O workloads and SLA. The proposed cost model considers both 1) Internal customers data placement cost including local placement, outsourcing, back-migration and penalty costs, and 2) External customers data placement cost including insourcing and geo-migration costs. It can be used to help in the decision-making process which aims to enhance customers QoS and reduce CSPs costs in a federation. Simulation results showed the relevance of the considered costs. We have shown that mis-considering some sub-costs may lead to a 95% cost error for external customers data placement and 80% for outsourcing customers. This may cause significant financial loss.

Index Terms—Cloud, federation, hybrid storage, cost model.

I. INTRODUCTION

CLOUD federation [1], [2], [3], [4] is a computing paradigm that consists in making several Cloud Service Providers (CSPs) cooperate by sharing resources. These CSPs insource and outsource their customers' data and services to provide continuous provisioning by exploiting temporal and spatial availability of resources [2] while reducing their cost. Cloud federation need was driven by novel applications such as mobile cloud, IoT, and big data. It came to address many limitations such as resource contention [5], [6], service interruption [7] and Quality of Service (QoS) degradation that may be due to the geographical distance to cloud resources.

Database as a Service (DBaaS) is one of the most important application processing Cloud service offered to cut the IT costs. For such a service, I/O performance and network latency are the two main metrics considered by the customers. Indeed, I/O system is one of the main system bottlenecks [8]. Moreover, it takes about 90% of the transaction execution time in some database queries [9].

Some cloud companies [10] already include latency guarantees in their Service Level Agreements (SLAs) and customers may pay extra charges for reducing I/O latencies.

In order to handle I/O bottlenecks, CSPs rely on the heterogeneity of storage devices. While Hard Disk Drives (HDDs) are used mainly to provide large storage volumes thanks to their low cost per GB, flash based Solid State Drives (SSDs) are integrated to reduce access latency and increase the I/O bandwidth [27], [28]. However, their higher cost does not allow for a massive use [29]. So, according to customers' QoS requirements, a CSP may use this heterogeneity to migrate or replicate their data between different storage classes or to other partner CSPs in the case of a Cloud federation. Dealing with such local and external storage system heterogeneity makes data placement strategies very challenging and needs an accurate cost estimation in order to make adequate decisions when placing data objects.

The cost of operating a Cloud is too significant to be ignored [30], as a matter of fact, using an accurate cost model is a critical issue. In effect, cost models are frequently used for optimization sake. Several state-of-the-art studies dealt with cost estimation of storage systems. We classified them in two categories. The first category is related to studies on centralized Clouds such as [12], [13], [16], [15], [11], [31]. They investigated several issues related to I/O efficiency in case of heterogeneous storage systems, see Table Ia. The second category is related to the cost estimation in the case of interconnected Clouds. These cost models dealt with different operations such as: initial placement of VMs [6], [17], [25], storage cost and geo-migration [21], [18], [32]. However, none of these studies took into account hybrid storage systems and I/O related cost, see Table Ib. They mainly used a fixed storage cost related to data volume without considering I/O operations cost (IOPS or latency).

In the case of a Cloud federation, for an accurate storage cost estimation, CSPs must consider both local and external storage costs. Indeed, it might be more cost effective to store a customer's data on a distant HDD than on a local SSD in case of customer mobility for instance. To consider this issue, the cost estimation must take into account both the details of the storage system and the properties of the Cloud federation. To the best of our knowledge these issues were not considered simultaneously in the existing work.

In this paper, we propose a model to evaluate the storage cost of object placement for DBaaS applications in a federated Cloud. We define an object as any logical entity of a database such as a table, a view, or an index. Our model consists of:

TABLE I: Related work classification

work	Hybrid storage system based cost models				
	Occupation	Energy	Wearout	Migration	Penalty
[11]			x		
[12]			x		
[13], [14]	x	x			
[15]		x	x	x	x
[16]	x	x	x	x	x

(a)

work	Interconnected cloud cost models				
	Fixed storage	Traffic	Geo-migration	Heterogeneity	Federation
[17]	x				x
[18], [19], [20]	x	x			
[21]	x	x		x	
[22], [23], [24]			x		
[6]				x	x
[25], [26], [24]		x		x	x

(b)

- Internal customers object placement cost: It takes into account (1) the cost of object placement for internal customers in the local infrastructure, (2) the cost of outsourcing local customers objects, that is buying resources from the other federation CSPs, (3) the cost of prospective back-migration that is the cost induced by bringing back previously outsourced internal customers objects and finally (4) the penalty cost related to the violation of customers SLAs.
- External customers placement cost: this is the cost of managing external customer objects. It consists of (1) the cost of insourcing external customers objects and (2) the cost of sending back these objects to their home or other clouds of the federation when requested.

We have evaluated the cost model and showed that the considered sub-costs are all relevant. With a simple use case, the lowest considered sub-cost part was 6%. Our cost model makes it possible to investigate whether outsourcing is relevant in a federation according to the resource cost and workload properties. It also provides the CSP with a mean to evaluate the penalty cost related to outsourcing some objects and to tune the resources to provide for the federation.

The paper is organized as follows. In Section 2, we discuss some related work, then, we define the system model and formulate the problem in Section 3. In Section 4, our cost model is presented and it is evaluated in Section 5. Finally, we conclude and give some perspectives in Section 6.

II. RELATED WORK

In this section we discuss existing studies about hybrid storage systems and interconnected clouds cost models.

A. Hybrid storage system cost models for the Cloud

Many efforts have been made to estimate the cost of data placement in hybrid storage systems. Existing work focused mainly on single clouds. In [13], [14] the storage system cost was estimated using occupation and energy costs. Authors of [11] considered only the wear out while migration cost has been modeled in [12]. Recently, in [15], a cost model has been proposed taking into account four factors: energy, wear out limit, migration between storage classes and penalties. The authors in [16] have extended the previous study with the storage occupation cost. This is summarized in Table Ia.

B. Interconnected cloud cost models

Many existing studies [18], [19], [20] have dealt with the cost of running big data applications, such as social networks,

and scientific computing [33]. These studies consider storage, computation and bandwidth to evaluate the cost of data geo-migration and storage in distributed data-centers. All these cost models have used a fixed storage cost mainly related to the storage occupation. The running I/O workload has not been considered. Other studies investigated the cost of distributed clouds (eg: Amazon, Google, etc.). Cost models in this category addressed different aspects: Data center construction [34], energy cost [22], [23], [24], and bandwidth cost [24]. Finally, various cost models for federated clouds have been proposed. Toosi et al [6] designed a cost model which sets dynamically the cost of federated VMs according to the amount of idle resources of the CSPs, as well as a revenue model to maximize the CSP profit. In [17] a cost model, including insourcing and outsourcing costs of virtual resources was proposed. In [25], [26], the cost model consists in local, insourcing, outsourcing and network traffic costs.

Existing hybrid storage cost models are not applicable when it comes to interconnected clouds because data placement in this case implies other factors related to the distributed environment. Regarding federated clouds, the cost of I/O system has not been considered in detail. Indeed this cost has been generally considered as constant and was not related to application workloads.

In this paper, we propose a cost model for the storage system in a federated Cloud taking into account in one hand, the detailed cost of the storage system and in the other hand, the distributed nature and federation properties.

III. SYSTEM MODEL AND PROBLEM DEFINITION

In this section, we first describe the system model and then we formulate the cost modeling problem.

A. System model

We consider a federation F composed of D CSPs $F = \{CP^d, d \in [1..D]\}$ cooperating in a peer-to-peer intercloud fashion as depicted in Figure 1. This architecture is inspired from [35] and was used in several studies [6], [25], [17]. The figure emphasizes on the d cloud provider view.

Each CP^d has one datacenter and accommodates different types of storage classes $SC = \{sc_j, j \in [1..J]\}$. We limit our study to two storage classes in this work: SSDs and HDDs. Each sc_j has a limited capacity c_{sc_j} , a maximum IOPS per operation $iops_{op,sc_j}$, a wear out wo_{sc_j} , and a purchase cost p_{sc_j} . We assume that a CSP buys a bandwidth bw from an Internet provider with a purchase cost p_{bw} .

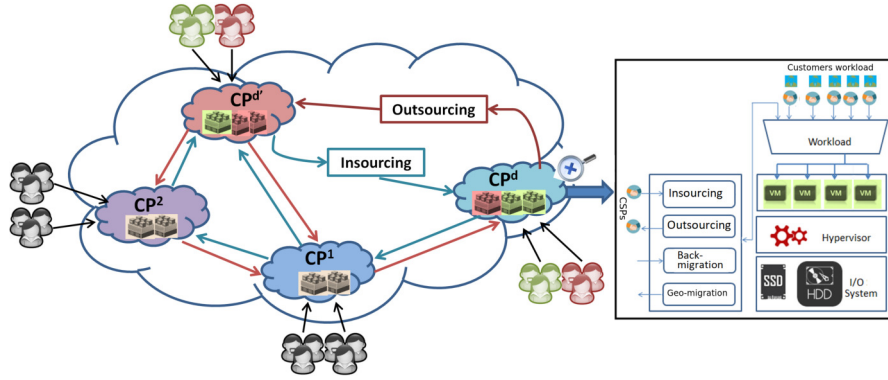


Fig. 1: Federated cloud based hybrid storage system

CSPs may outsource their customers objects to other federation members (called partners), for instance, when they cannot meet customers QoS requirements.

Therefore, each CP^d has two types of customers:

- 1) internal customers: that are the customers of CP^d , $U_{int} = \{u_k, k \in [1, K]\}$,
- 2) external customers: that are partner CSPs $\{CP^{d'}\}$ customers whose objects are outsourced to CP^d , $U_{ext} = \{u_{k'}, k' \in [1, K']\}$, $CP^{d'} \in F - \{CP^d\}$. These customers' objects are insourced by CP^d .

We assume that each internal customer u_k :

- 1) has a set of objects $\{o_{i,k}, i \in [1..N]\}$. Each object has a size denoted $s_{o_{i,k}}$.
- 2) generates an I/O workload wl_k representing the I/O operations generated by the set of queries issued by the customer. According to the access pattern (sequential, random) and the operation type (read, write), We distinguish four I/O patterns as in [16]: sequential read (sr), sequential write (sw), random read (rr) and finally, random write (rw).
- 3) a requested QoS (SLA) in terms of IOPS and latency $iops_{sla,k}, lt_{csla,k}$.
- 4) A penalty function pn_k composed of two parts, one related to storage performance : $pn_{iops,k}$ and the other to the latency: $pn_{ltc,k}$.

The same notation is used for the external customers.

Based on customers' SLA requirements and the amount of available storage resources, objects are served locally (placed and migrated between different storage classes), insourced, geo-migrated or geo-replicated to partner clouds. To clarify the system model, we introduce the following definitions:

Outsourcing [6], [36], [25], [3]: the ability that CSPs have to send some internal customers objects to other federation members. Outsourcing maybe achieved through either geo-migration or geo-replication.

Geo-migration [21], [18]: the process of moving objects to other clouds without caring about local copy synchronization.

Geo-replication [21], [37]: the process of maintaining multiple copies of objects on multiple sites (CSPs) for a better performance, availability, and reliability.

Insourcing [25], [26], [36], [3]: the opposite process of outsourcing. CSPs make available part of their unused resources to respond to requests from other members.

Inner migration [16], [15]: is the movement of some objects between different storage classes within the same infrastructure. A CSP CP^d migrates periodically some objects between the different storage classes. $O_{m,int}$, $O_{m,ext}$: represent the set of internal (external, respectively) customers objects to migrate between internal storage classes.

Back migration: This operation consists in bringing back previously outsourced objects to the local infrastructure.

The primary objective of a CSP is to reduce the used resources cost while meeting customers QoS needs. The cloud administrator has to take decisions about: (1) which objects need to be moved locally between different storage classes, (2) which ones need to be outsourced and to which partner CSP, i.e. which ones to be replicated or migrated, (3) which other CSP customer objects need to be insourced, (4) and finally, which previously outsourced internal customers objects, need to be brought back to the local infrastructure.

Periodically, the cloud administrator makes decisions about object placement. We note by T the time period during which monitoring is executed to extract objects I/O patterns and to compute the cost of outsourced objects in order to evaluate the overall placement cost as in [16], [15]. We assume that each cloud maintains two matrices, one for internal customers objects (A) and the other for external customers objects (B) such as $A[i, j](T) = 1$ when an internal object is placed in the cloud CP^j and $B[i, j](T) = 1$ when the object is an insourced object to cloud CP^j .

In our study, we assume that a given CSP charges its partners for insourcing actions. Reduced prices are used in order to foster cooperation within the federation. As in [6], each CP^d dynamically adjusts the price of its contributed storage resources according to the amount of idle resources. Let $Cap_{max_{rsc}}$ and $Cap_{idl_{rsc}}$ be the total and idle capacities for a given resource rsc of the provider CP^d . rsc maybe the storage occupation (occ) or performance in IOPS ($iops$). If p_{rsc} is the price paid by internal customers for the resource rsc , its insourcing price F_{res}^d is obtained from the expression

in eq. (1) from [6]. For each time period T , CSPs will use eq.1 to adjust their insourcing prices.

$$F_{rsc}^d = \frac{Cap_{max_{rsc}} - Cap_{idl_{rsc}}}{Cap_{max_{rsc}}} * (p_{rsc}) \quad (1)$$

From the network resource point of view, only outgoing network (Internet) traffic is charged for customers in a given CSP, be it a local or an outsourced customer (in this case, the hosting CSP charges the external network traffic cost for outsourcing one).

In this paper, we assume that each CP^d dedicates a fixed part of storage and bandwidth resources for its internal customers ($sc_{j,int}, bw_{int}$) and another part for the external customers $sc_{j,ext}, bw_{ext}$ as noted in eq. 2.

$$\forall CP^d, \begin{cases} sc_j = sc_{j,int} + sc_{j,ext} \\ bw = bw_{int} + bw_{ext} \end{cases} \quad (2)$$

B. Problem formulation

In the following, we define our problem by giving system inputs and outputs.

Input: A CSP of the federation that has : (1) a set of storage classes, (2) an Internet bandwidth with maximum capacity and purchase cost, (3) a storage and outgoing network bandwidth costs for partner Clouds. (4) a set of internal customers, (5) a set of external customers, (6) two sets of objects $O_{m,int}$ and $O_{m,ext}$ to move between the local storage classes. (7) a monitoring period T , and (8) two matrices: $A(T)$ and $B(T)$.

Output: The monetary placement cost of all customers' objects for a given time period T .

IV. STORAGE COST MODEL IN A CLOUD FEDERATION

A. Overview

We model the cost of object placement for a cloud CP^d , $d \in [1..D]$, belonging to a federation F for a given period of time T . The model is built hierarchically, see Figure 2.

The total object placement cost $Cost_{plc,T}$ is the sum of the placement cost of the internal customers ($Cost_{plc_{int},T}$) and the external customers ($Cost_{plc_{ext},T}$) as shown in eq. 3.

Note that non-recurring costs which do not depend on the objects placement like maintenance cost, human resources cost, air-conditioning costs are not considered in this paper. We do not also consider the cost related to data security.

$$Cost_{plc,T} = Cost_{plc_{int},T} + Cost_{plc_{ext},T} \quad (3)$$

B. Internal customers object placement cost (see (2), Figure 2)

This is the cost of placing internal customers objects. It includes the local placement cost $Cost_{lcl_{int},T}$, the outsourcing cost ($Cost_{out_{src},T}$), the back-migration cost ($Cost_{bck_{mgr},T}$) and the penalty cost $Cost_{pnt_{int},T}$ as shown in eq. 4.

The local placement cost is the storage cost of internal customers objects in the local infrastructure. The outsourcing cost is related to the placement of internal customers objects in partner CSPs. The back-migration cost represents the cost of bringing back the previously outsourced objects to the

home infrastructure. Finally, the penalty cost represents the additional monetary cost caused by SLA violations.

$$Cost_{plc_{int},T} = Cost_{lcl_{int},T} + Cost_{out_{src},T} + Cost_{bck_{mgr},T} + Cost_{pnt_{int},T} \quad (4)$$

1) *Local placement cost (see (4), Figure 2):* It is obtained from the storage cost $Cost_{stg_{int},T}$ of internal customers objects and the inner migration cost (between storage classes within the CSP infrastructure) $Cost_{mgr_{int},T}$ of the set of objects $O_{m,int}$ as shown in eq. 5.

$$Cost_{lcl_{int},T} = Cost_{stg_{int},T} + Cost_{mgr_{int},T} \quad (5)$$

The storage cost: As in [16], $Cost_{stg_{int},T}$ is related to the occupation $Cost_{occ,T}$, the energy $Cost_{erg,T}$ and the wear out cost $Cost_{edr,T}$ due to I/O workload (see eq. 6).

$$Cost_{stg_{int},T} = max[Cost_{occ,T}, Cost_{edr,T}] + Cost_{erg,T} \quad (6)$$

The occupation cost is the amortized cost of the storage system over the period T . The energy cost is the energy consumed by the storage system to execute the I/O workload multiplied by the energy unitary price which we consider constant. Finally, the wear out cost is caused by the execution of the I/O workload which impacts the lifetime of the devices. For SSD, this cost is related to the amount of written data while for HDD, it depends of the amount of both read and written data. For more details see [16].

The migration cost: it consists in reading the set of objects $O_{m,int}$ from the source storage device and writing them to another destination storage device. It includes the energy and endurance costs.

2) *Outsourcing cost (see (5), Figure 2):* It is composed of geo-migrating and geo-replicating internal customers objects and the storage cost in the partner CSPs generated by these two geo-operations as in eq. 7.

$$Cost_{out_{src},T} = Cost_{geo_{mgr},T} + Cost_{geo_{rpl},T} + Cost_{ext_{plc},T} \quad (7)$$

The geo-migration cost (see (11), Figure 2): geo-migration of objects incurs local read operations $Cost_{rd,T}$ and internet bandwidth usage $Cost_{bw,T}$.

The set of objects concerned by the geo-migration O_{gmgr} is obtained from the difference between the matrix A at the period $T - 1$ and the current period T where $o_i, A[i, d](T - 1) = 1 \wedge A[i, d](T) = 0$. Geo-migration cost is shown in eq. 8.

$$Cost_{geo_{mgr},T} = Cost_{rd,T} + Cost_{bw,T} \quad (8)$$

The local read operation cost incurs energy $Cost_{rd_{erg},T}$ and wear out $Cost_{rd_{edr},T}$ costs.

$$Cost_{rd,T} = Cost_{rd_{erg},T} + Cost_{rd_{edr},T} \quad (9)$$

The Internet bandwidth cost is related to the bandwidth consumed by the geo-migration of objects to external clouds.

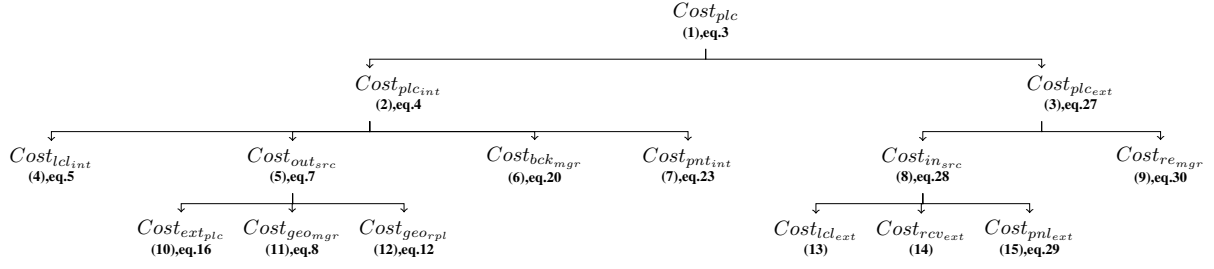


Fig. 2: The overall cost model

We calculate this cost by multiplying the size of all objects to be migrated by the amortized bandwidth over one unit of time $Cost_{bw_{amz},1}(bw_{int})$.

$$Cost_{bw,T} = \frac{\sum_{o_i \in O_{gmgr}} (S_{o_i}) * Cost_{bw_{amz},1}(bw_{int})}{bw_{int}} \quad (10)$$

We calculate the amortized bandwidth cost over one unit of time by distributing the purchase cost of bw_{int} over the subscription time period T_{sp} .

$$Cost_{bw_{amz},1}(bw_{int}) = \frac{p_{bw} * \frac{bw_{int}}{bw}}{T_{sp}} \quad (11)$$

The geo-replication cost (see (12), Figure 2): geo-replicating objects over multiple CSPs consumes supplementary storage and network costs. The geo-replication consists in adding, deleting and synchronizing replicas. So, its overall cost is the sum of the aforesaid operations costs as shown in eq. 12. Deleting replica cost is assumed to be nil.

$$Cost_{geo_{rpl},T} = Cost_{add_{rpl},T} + Cost_{sync_{rpl},T} \quad (12)$$

The cost of adding replicas is the same as the geo-migration cost without deleting the original copy and considering the set of objects $\{O_{grpl}^{d'}, d' \neq d\}$. They are obtained from matrix A .

$$Cost_{add_{rep},T} = \sum_{CP^{d'} \in F - \{CP^d\}} Cost_{geo_{mgr},T}(O_{grpl}^{d'}) \quad (13)$$

Concerning replica synchronization either CP^d forwards the synchronization or receives it. $O_{srpl}^{d'}$ is the list of objects sent to, or received from $CP^{d'}$ for synchronizing replicas. When CP^d forwards the synchronization, its cost is the sum of the read operations induced and the consumed Internet bandwidth costs, see eq. 14.

$$Cost_{syn_{rep},T} = \sum_{CP^{d'} \in F - \{CP^d\}} Cost_{geo_{mgr},T}(O_{srpl}^{d'}) \quad (14)$$

Whereas when CP^d receives the synchronization, this cost consists of writing operations, local Internet bandwidth use, and outgoing network costs (i.e back-migration cost, see later in eq. 20) as shown in the next equation 15.

$$Cost_{syn_{rep},T} = \sum_{CP^{d'} \in F - \{CP^d\}} Cost_{bck_{mgr},T}(O_{srpl}^{d'}) \quad (15)$$

The external placement cost (see (10), Figure 2): Once the objects have been migrated to other clouds, the CSP needs to pay partner clouds for hosting the objects. Therefore, the external storage cost includes the external occupation cost $Cost_{ext_{occ},T}$, the external workload execution cost $Cost_{ext_{wld},T}$ and the external penalty cost $Cost_{ext_{pnl},T}$. The penalty cost here is the charge paid by the partner clouds in case SLAs are violated for the outsourced objects.

The purchased resources are billed according to eq. 1.

$$Cost_{ext_{plc},T} = Cost_{ext_{occ},T} + Cost_{ext_{wld},T} - Cost_{ext_{pnl},T} \quad (16)$$

The external occupation cost: represents the cost of storing the migrated or replicated internal customers objects on partner clouds. We calculate this cost by using the federated occupation cost $F_{occ}^{d'}$ of $CP^{d'}$ as noted in eq. 17

$$Cost_{ext_{occ},T} = \sum_{d' | CP^{d'} \in F - \{CP^d\}} \left(\sum_i A[i, d'] * S(o_i) \right) * F_{occ}^{d'} \quad (17)$$

The cost of workload execution represents the amount of expected IOPS consumed by internal users. It is calculated the same way as in the eq. 18 where $F_{iops}^{d'}$ is the cost of one IOPS in the cloud $CP^{d'}$

$$Cost_{ext_{wld},T} = \sum_{d' | CP^{d'} \in F - \{CP^d\}} \left(\sum_i A[i, d'] * IOPS(o_i) \right) * F_{iops}^{d'} \quad (18)$$

The penalty cost is a percentage $\alpha\%$ of the total charges paid by CP^p to the partner cloud, as in Amazon Cloud.

$$Cost_{ext_{pnl},T} = \alpha * (Cost_{ext_{occ},T} + Cost_{ext_{wld},T}) \quad (19)$$

3) Back migration cost (see (6), Figure 2): $Cost_{bck_{mgr},T}$ is the cost of bringing back the previously outsourced objects to the local infrastructure. It is composed of the cost of data transfer (outgoing network cost) from the partner clouds $Cost_{ntw_{out},T}$, the local Internet bandwidth cost $Cost_{bw,T}$

corresponding to the bandwidth consumed locally to receive the previously outsourced objects, and the cost $Cost_{wr,T}$ to write the received objects on the local storage, see eq. (20).

$$Cost_{bck_{mgr},T} = Cost_{wr,T} + Cost_{ntw_{out},T} + Cost_{bw,T} \quad (20)$$

The set of internal customers objects to be brought back $O_{b_{mgr}}$ is deduced from the matrix A .

The local write cost: is the cost of placing previously outsourced objects in the local storage system. This cost involves the cost of energy $Cost_{wreg,T}$ consumed by the storage devices and the wear out cost of these devices $Cost_{wredr,T}$ caused by the write operations. Some of these objects are placed on HDD and others on SSD, see eq. 21.

$$Cost_{wr,T} = Cost_{wreg,T} + Cost_{wredr,T} \quad (21)$$

Outgoing network cost: This cost is charged by partner CSPs when CP^d brings back their internal customer objects to the local infrastructure. We calculate this cost by multiplying the size of the objects to bring back by the cost of outgoing network $Cost_{out}(CP^d)$ of the partner.

$$Cost_{ntw_{out},T} = \sum_{CP^d \in F - \{CP^d\}} (Cost_{out}(CP^d) * (\sum_{o_i \in O_{b_{mgr}}} S_{o_i})) \quad (22)$$

Local Internet bandwidth cost is calculated with eq. 10.

4) **Penalty cost** (see (7), Figure 2): The violation of SLA terms by the CSP entails a penalty that should be paid. As in [16], we calculate the overall penalty as the sum of all internal customers penalties.

$$Cost_{pnt_{int},T} = \sum_{u_k \in U_{int}} (Cost_{pnt,T}(u_k)) \quad (23)$$

Internal customer penalty cost is composed of a penalty related to the IOPS $Cost_{pnt_{iops},T}(u_k)$ and another one related to the latency $Cost_{pnt_{ltc},T}(u_k)$.

$$Cost_{pnt,T}(u_k) = Cost_{pnt_{iops},T}(u_k) + Cost_{pnt_{ltc},T}(u_k) \quad (24)$$

The IOPS penalty cost is calculated as in [16], it is proportional to the ratio between the offered IOPS $iops_{offered}(u_k)$ and the requested one ($iops_{sla,k}$). The requested IOPS is defined in the SLA while we calculate the offered IOPS from the time needed to handle the I/O workload of the customer u_k and the total number of I/O requests issued to their objects. It is calculated as follows:

$$Cost_{pnt_{iops},T}(u_k) = pnt_{iops,k} \left(\frac{iops_{offered}(u_k)}{iops_{sla,k}} \right) \quad (25)$$

The latency penalty cost: In our model, we assume that customers pay extra charges to have a reduced network latency. If the latency offered $ltc_{offered}(u_k)$ to customer u_k is lower

than the one requested $ltc_{sla,k}$, then a penalty is applied to the CSP. This penalty cost is proportional to the violation degree and the number of requests nb_{rqt} , as noted in Table VI. We calculate the latency penalty cost as follows:

$$Cost_{ltc_{iops},T}(u_k) = \sum_{i=1} nb_{rqt} (pnt_{ltc,k} \left(\frac{ltc_{offered}(u_k)}{ltc_{sla,k}} \right)) \quad (26)$$

TABLE II: Latency penalty per request

$ltc_{offered}$	Penalty (%)
$[0..B_0]$	0
$[B_i..B_{i+1}]$	x_i
$> B_n$	x_n

C. **External customers placement cost** (see (3), Figure 2)

This cost $Cost_{plc_{ext},T}$ is the sum of the cost of insourcing external customers objects $Cost_{in_{src},T}$ and the cost generated when some objects of the external customers are geo-migrated to other clouds or taken back by their home clouds (re-migration cost) re_{mgr} , see eq. 27.

$$Cost_{plc_{ext},T} = Cost_{in_{src},T} + Cost_{re_{mgr},T} \quad (27)$$

1) **Insourcing cost** (see (8), Figure 2): The insourcing cost is composed of receiving external customers objects cost $Cost_{rcv_{ext},T}$, the local placement cost $Cost_{lcl_{ext},T}$ and the penalty cost $Cost_{pnt_{ext},T}$.

$$Cost_{in_{src},T} = Cost_{rcv_{ext},T} + Cost_{lcl_{ext},T} + Cost_{pnt_{ext},T} \quad (28)$$

Receiving cost (see (14), Figure 2) of the external customers objects is the cost of writing these objects, calculated from eq. 21, and the cost of the consumed local Internet bandwidth, calculated with eq. 10. The set of external customers objects to be received $O_{e_{rcv}}$ is deduced from the matrix B

The local placement cost (see (13), Figure 2): of the external customers objects is calculated from eq. 5.

The penalty cost (see (15), Figure 2) is the sum of penalty costs of the external customers.

$$Cost_{pnt_{ext},T} = \sum_{u_{k'}^d \in U_{ext}} (Cost_{pnt,T}(u_{k'}^d)) \quad (29)$$

In the federation, resource prices are dynamically set (see eq. 1), so the penalty cost is set accordingly and is related to the unused storage resources β which is calculated as follows:

$$\beta = \frac{Cap_{max_{rsc}} - Cap_{idl_{rsc}}}{Cap_{max_{rsc}}} \quad (30)$$

With rsc being a combination of the storage space and performance requested. The penalty cost of an external customer $u_{k'}^d$ is thus given in eq. 31:

$$Cost_{pnt,T}(u_{k'}^d) = \beta * Cost_{pnt_{iops},T}(u_{k'}^d) \quad (31)$$

2) *The re-migration cost (see (9), Figure 2)*: This cost is driven by the decision of home clouds to geo-migrate their customers back to their infrastructure or to another cloud. This incurs local read operations and Internet bandwidth costs as described in eq. 8. The set of objects concerned by the re-migration is deduced from the matrix B by the difference between the period $T - 1$ and the current period T where for o_i , $B[i, d](T - 1) = 1 \wedge B[i, d](T) = 0$.

V. EVALUATION

This section presents an evaluation of the proposed cost model. Our aim is twofold: (i) validating the relevance of the sub-costs used in our cost model and comparing to state-of-the-art models, (ii) showing the flexibility of the cost model through the investigation of the impact of different parameters on the placement cost.

A. Experimental Settings

We used CloudSim simulator [38]. The simulated scenario is composed of a federation of 9 geographically distributed CSPs. Each CSP is composed of one datacenter running 1000 VMs as in [6]. Some of these VMs (set to 20% in our experiments) are devoted to the federation. Each CSP has an internet bandwidth of 1Gbps bought from an Internet service provider (the price was set to 1500\$ per month). We used a standard VM configuration with 8 cores, 8 GB of RAM and a hybrid storage system. Characteristics of the storage system is provided in Table III.

TABLE III: Storage devices specifications

Characteristics	HDD	SSD(\$)
Price (\$)	230	200
Size	1 TB	128 GB
Performance	Seek time: 8.5 ms read/write: 9.5 ms	rr: 10000 IOPS, rw: 40000 IOPS sr: 540 MB/s, sw: 520 MB/s

Each CSP manages a set of databases (DBs) built using TPC-H and TPC-C benchmarks with different sizes and varied workload, we used the configuration given in [16]. For more details about the different DBs and storage system specifications see Table IV. The amount of storage penalty is set to 30 % of the total charges paid by the customer as in Amazon Cloud. The storage price is fixed according to Amazon gp2 price model (0.1\$/GB/month) while the energy cost is set to 0.1\$ per kWh as in [13].

Some of the customers are mobile and ask for a good latency by paying some extra charge. Their home CSP generally migrates their objects to the nearest CSP that meets the latency constraint otherwise the home CSP may undergo a penalty. Mobile customers are supposed to have a 1 week duration mobility. The price of latency is given in Table V [10] and its related penalty in Table VI. The evaluation is conducted over a 1 month with one-hour time period for dynamic resource price update. The insourcing prices are dynamically set by CSPs using equation (eq. 1) and changed each hour (period T). We assume that the outgoing network cost is set to 50% of amazon's which gives 0.09\$/GB.

TABLE IV: Data bases specifications

Data base	Bench.	Reqs. nbr (op/h)	rr(op/h)	rw(op/h)	sr(op/h)	sw(op/h)
DB1(32GB/HDD)	TPC-C	432000	136800	46800	108000	32400
DB2(60GB/SSD)		28800000	601200	104400	5436000	147600
DB3(147GB/HDD)	TPC-H	331200	10800	216000	3600	18000
DB4(34GB/HDD)		335200	32400	216000	7200	10800
DB5(381GB/HDD)		13360	1080	8280	360	1080

TABLE V: Latency price/req.

Latency (ms)	Price by request (\$)
< 200	0.0000001
< 400	0.00000005
< 600	0.00000002
> 600	0

TABLE VI: Penalty/req.

Latency (ms)	Penalty (\$)
< 200	0
< 400	0.00000005
< 600	0.00000008
> 600	0.0000001

B. Evaluation results

1) *Relevance of the sub-costs*: The first experiment concerns the relevance of the used sub-costs. In this scenario, the network latency between each pair of cloud infrastructures is assigned randomly between [200ms, 700ms]. First, we evaluated all different sub-costs for one CSP, then, we calculated the average placement cost of (1) insourced databases see (8), Figure 2 and (2) outsourced databases see (5), Figure 2, and compared the resulting costs with those of some state-of-the-art studies [18], [19], [21], [6], [25]. These costs were chosen as they are the higher ones.

Figure 3a and Figure 3b show the different sub-costs of our cost model (leaf nodes of Figure 2). In this evaluation, geo-replication costs were not considered (only geo-migration has been shown). We observe that all the modeled costs are relevant as each cost is high enough for at least one tested database. We observe that the local placement and penalty costs of internal customers are the highest ones. This is due to the fact that only a small part of customers are mobile (20%) and for small periods (1 week). Also, the database size affects directly the external placement cost because the storage in this case is bought from others CSPs. The database size affects likewise all the costs including geo-migration and back-migration, while the remaining costs are affected by the storage device type and workload patterns.

Figure 4a and Figure 4b show the cost of insourced and

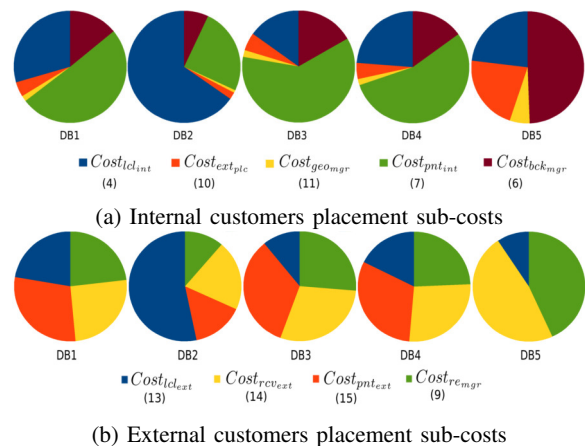


Fig. 3: Sub-costs of the simulated placement cost.

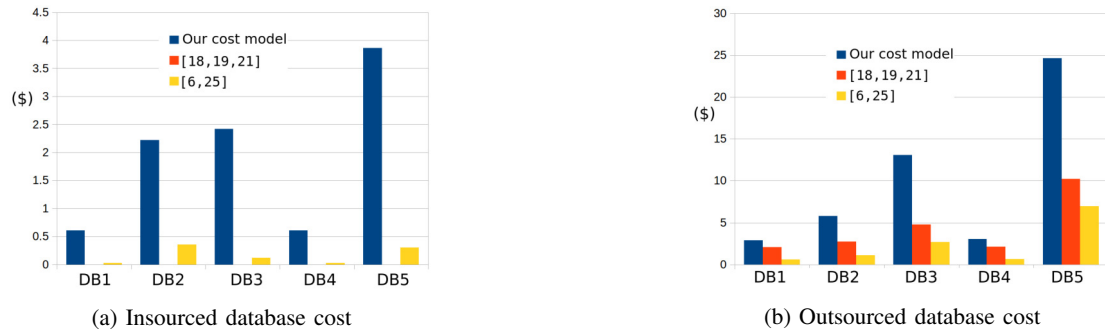


Fig. 4: The average per week placement cost of the insourced and outsourced databases for a given CSP.

outsourced databases for a given CSP for our cost model as compared to state-of-the-art models. A first observation one may draw is that models from [18], [19], [21] do not consider the insourcing cost. In addition, the cost of storage services for outsourced objects is fixed to 1 GB/unit of time. In [6], [25], only the occupation cost is considered with no geo-migration cost which can lead, in the worse case, to a difference with our cost model of 95% for the external customers placement cost and 80% for the outsourcing cost. Furthermore, the penalty cost is ignored in those studies while in our work it can represent up to 61% of the placement cost of internal customers and 32% of that of external customers.

2) *Spare resources pricing model evaluation*: The objective of this part is to show how the proposed cost model makes it possible to easily evaluate the spare resource pricing model used according to the ran workload.

Figure 5.a shows, from one hand the cost of running insourced databases and from the other hand, the price billed to the partner clouds. As one can observe, the financial gain of insourcing objects highly depends on the nature of the ran workload. Insourcing large volumes of data generating a low number of queries (DB5) is of course more interesting.

Figure 5.b shows the outsourced objects cost when buying resources from partner clouds as compared to buying them from an external cloud. This Figure illustrates the ability of the model to highlight the gain obtained for a given CSP knowing its outsourced objects and the used federation pricing model. This may help to optimize the outsourcing decision.

3) *Impact of latency and penalty, Federation vs DCC vs single Cloud*: The designed cost model makes it possible to compare the cost of running a Cloud infrastructure within or out of a Cloud federation. In this part, we compare the average placement cost of mobile customers databases by varying the latency between clouds for three scenarios: (1) Geo-migration using a federation, (2) Geo-migration using a distributed cloud computing platform (DCC) and (3) Single Cloud without geo-migration (see Table 6). The aim of this part is to show the impact of the violation degree and the network latency on the average placement cost for the three configurations.

We notice, from Figure 6, that generally, the cost without outsourcing mobile customers increases with the increase of the network latency. For databases with heavy workloads (e.g.

DB2) it is always interesting to outsource objects. In fact, outsourcing is interesting as long as the amount of outsourced data is small and the workload is heavy. This is not the case for data with small workload and large size. This is because geo and back-migrating these data implies a high network traffic cost (e.g. DB5). For some databases with large sizes and medium workloads, it is not cost-effective to outsource them as long as the violation of the latency is not high. However, when the degradation of network latency reaches a certain level, it becomes interesting to be outsourced. For instance, for DB3 the cost without migration (with penalty) is lower than the cost with migration when latency is $< 400ms$ but it is the opposite when the latency becomes $> 600ms$. Our cost model allows to make a trade-off between all these parameters for optimizing the overall cost of running a Cloud into a federation.

VI. CONCLUSION

A CSP can meet its customers QoS and minimize the cost of data placement by using either local or partner resources in a Federation. In this work, we have proposed a cost model for data placement on hybrid storage systems in a Cloud federation. Our model extends state-of-the-art work by considering geo-migration, penalty, back-migration, and geo-replication costs.

For future work, we will first investigate placement optimization strategies to find the optimal internal and/or external placement for CSP objects. We will also design pricing strategies for resources in a Federation.

The performed evaluations proved the relevance of the considered costs. It also pointed out that outsourcing and insourcing is a complex task that requires taking into account a large number of parameters. Even though our model is storage oriented, it can be used and integrated in a broader cost model considering other resources (e.g. CPU and memory).

REFERENCES

- [1] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. M. Sadjadi, and M. Parashar, "Cloud federation in a layered service model," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330–1344, 2012, <http://dx.doi.org/10.1016/j.jcss.2011.12.017>.
- [2] H. Li, C. Wu, Z. Li, and F. C. Lau, "Profit-maximizing virtual machine trading in a federation of selfish clouds," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 25–29, <http://dx.doi.org/10.1109/infcom.2013.6566728>.

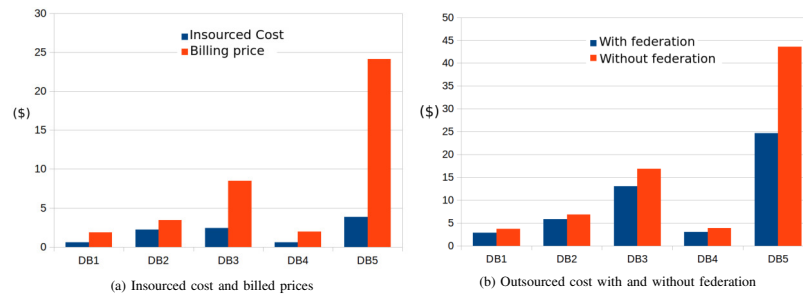


Fig. 5: Federation impact on the cost

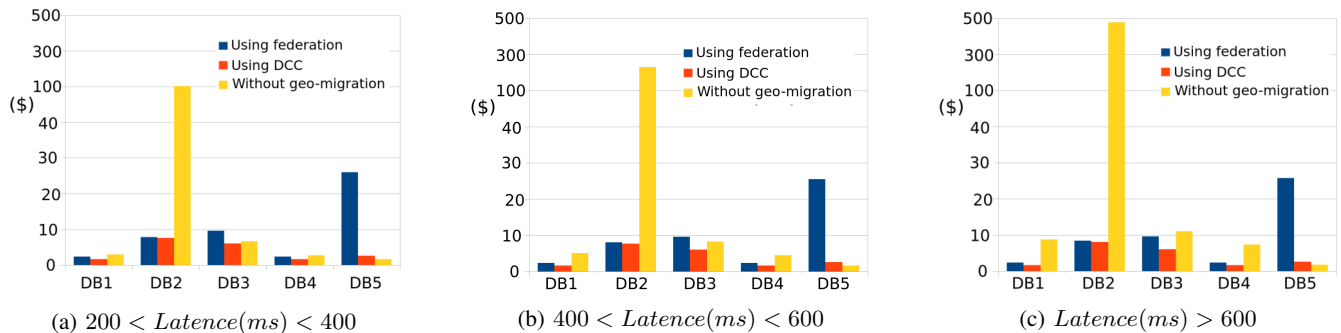


Fig. 6: Average cost with migration in federated cloud and distributed cloud VS average cost without migration

- [3] M. R. Assis and L. F. Bittencourt, "A survey on cloud federation architectures: identifying functional and non-functional properties," *Journal of Network and Computer Applications*, vol. 72, pp. 51–71, 2016, <http://dx.doi.org/10.1016/j.jnca.2016.06.014>.
- [4] R. Moreno-Vozmediano, E. Huedo, I. M. Llorente, R. S. Montero, P. Massonet, M. Villari, G. Merlino, A. Celesti, A. Levin, L. Schour *et al.*, "Beacon: a cloud network federation framework," in *Communications in Computer and Information Science*. Springer, 2016, pp. 325–337, http://dx.doi.org/10.1007/978-3-319-33313-7_25.
- [5] M. Amiri and L. Mohammad-Khanli, "Survey on prediction models of applications for resources provisioning in cloud," *Journal of Network and Computer Applications*, vol. 82, pp. 93–113, 2017, <http://dx.doi.org/10.1016/j.jnca.2017.01.016>.
- [6] A. N. Toosi, R. N. Calheiros, R. K. Thulasiram, and R. Buyya, "Resource provisioning policies to increase iaas provider's profit in a federated cloud environment," in *High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on*. IEEE, 2011, pp. 279–287, <http://dx.doi.org/10.1109/hpcc.2011.44>.
- [7] Y. Gu, D. Wang, and C. Liu, "Dr-cloud: Multi-cloud based disaster recovery service," *Tsinghua Science and Technology*, vol. 19, no. 1, pp. 13–23, 2014, <http://dx.doi.org/10.1109/tst.2014.6733204>.
- [8] E. Shriver, "Performance modeling for realistic storage devices," 1997, <https://dl.acm.org/citation.cfm?id=269078>.
- [9] M. A. Sharaf, P. K. Chrysanthis, A. Labrinidis, and C. Amza, "Optimizing i/o-intensive transactions in highly interactive applications," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009, pp. 785–798, <http://dx.doi.org/10.1145/1559845.1559927>.
- [10] D. B. Terry, V. Prabhakaran, R. Kotla, M. Balakrishnan, M. K. Aguilera, and H. Abu-Libdeh, "Consistency-based service level agreements for cloud storage," in *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*. ACM, 2013, pp. 309–324, <http://dx.doi.org/10.1145/2517349.2522731>.
- [11] Y. Cheng, M. S. Iqbal, A. Gupta, A. R. Butt, and V. Tech, "Pricing games for hybrid object stores in the cloud: Provider vs. tenant," in *HotStorage*, 2015, <https://www.usenix.org/conference/hotcloud15/workshop-program/presentation/cheng>.
- [12] L. Lin, Y. Zhu, J. Yue, Z. Cai, and B. Segee, "Hot random off-loading: A hybrid storage system with dynamic data migration," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. IEEE, 2011, pp. 318–325, <http://dx.doi.org/10.1109/mascots.2011.41>.
- [13] Y. Kim, A. Gupta, B. Urgaonkar, P. Berman, and A. Sivasubramanian, "Hybridstore: A cost-efficient, high-performance storage system combining ssds and hdds," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. IEEE, 2011, pp. 227–236, <http://dx.doi.org/10.1109/mascots.2011.64>.
- [14] N. Zhang, J. Tatemura, J. M. Patel, and H. Hacigümüş, "Towards cost-effective storage provisioning for dbmss," *Proceedings of the VLDB Endowment*, vol. 5, no. 4, pp. 274–285, 2011, <http://dx.doi.org/10.14778/2095686.2095687>.
- [15] H. Ouarnoughi, J. Boukhobza, F. Singhoff, and S. Rubini, "A cost model for virtual machine storage in cloud iaas context," in *Parallel, Distributed, and Network-Based Processing (PDP), 2016 24th Euromicro International Conference on*. IEEE, 2016, pp. 664–671, <http://dx.doi.org/10.1109/pdp.2016.119>.
- [16] D. Boukhelef, J. Boukhobza, and K. Boukhalfa, "A cost model for dbaas storage," in *International Conference on Database and Expert Systems Applications*. Springer, 2016, pp. 223–239, http://dx.doi.org/10.1007/978-3-319-44403-1_14.
- [17] M. Hadji and D. Zeghlache, "Mathematical programming approach for revenue maximization in cloud federations," *IEEE transactions on cloud computing*, vol. 5, no. 1, pp. 99–111, 2017, <http://dx.doi.org/10.1109/tcc.2015.2402674>.
- [18] Z. Wen, J. Cała, P. Watson, and A. Romanovsky, "Cost effective, reliable and secure workflow deployment over federated clouds," *IEEE Transactions on Services Computing*, vol. 10, no. 6, pp. 929–941, 2017, <http://dx.doi.org/10.1109/tsc.2016.2543719>.
- [19] L. Zhang, C. Wu, Z. Li, C. Guo, M. Chen, and F. C. Lau, "Moving big data to the cloud: An online cost-minimizing approach," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2710–2721, 2013, <http://dx.doi.org/10.1109/jsac.2013.131211>.
- [20] W. Xiao, W. Bao, X. Zhu, and L. Liu, "Cost-aware big data processing across geo-distributed datacenters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3114–3127, 2017, <http://dx.doi.org/10.1109/tpds.2017.2708120>.
- [21] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for

dynamic replication and migration of data in cloud data centers,” *IEEE Transactions on Cloud Computing*, 2017, <http://dx.doi.org/10.1109/tcc.2017.2659728>.

- [22] A. Khosravi, L. L. Andrew, and R. Buyya, “Dynamic vm placement method for minimizing energy and carbon cost in geographically distributed cloud data centers,” *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 183–196, 2017, <http://dx.doi.org/10.1109/tsusc.2017.2709980>.
- [23] K. Le, R. Bianchini, J. Zhang, Y. Jaluria, J. Meng, and T. D. Nguyen, “Reducing electricity cost through virtual machine placement in high performance computing clouds,” in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2011, p. 22, <http://dx.doi.org/10.1145/2063384.2063413>.
- [24] H. Yuan, J. Bi, W. Tan, and B. H. Li, “Cawsac: Cost-aware workload scheduling and admission control for distributed cloud data centers,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 976–985, 2016, <http://dx.doi.org/10.1109/tase.2015.2427234>.
- [25] S. Rebai, M. Hadji, and D. Zeghlache, “Improving profit through cloud federation,” in *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*. IEEE, 2015, pp. 732–739, <http://dx.doi.org/10.1109/ccnc.2015.7158069>.
- [26] M. Hadji, B. Aupetit, and D. Zeghlache, “Cost-efficient algorithms for critical resource allocation in cloud federations,” in *Cloud Networking (Cloudnet), 2016 5th IEEE International Conference on*. IEEE, 2016, pp. 1–6, <http://dx.doi.org/10.1109/cloudnet.2016.11>.
- [27] J. Boukhobza, “Flashing in the cloud: Shedding some light on nand flash memory storage systems,” in *Data Intensive Storage Services for Cloud Environments*. IGI Global, 2013, pp. 241–266, <http://dx.doi.org/10.4018/978-1-4666-3934-8.ch015>.
- [28] J. Boukhobza and P. Olivier, *Flash Memory Integration: Performance and Energy Issues*. Elsevier, 2017, <https://www.elsevier.com/books/flash-memory-integration/boukhobza/978-1-78548-124-6>.
- [29] D. Lee, C. Min, and Y. I. Eom, “Effective flash-based ssd caching for high performance home cloud server,” *IEEE Transactions on Consumer Electronics*, vol. 61, no. 2, pp. 215–221, 2015, <http://dx.doi.org/10.1109/tce.2015.7150596>.
- [30] C. Wu and R. Buyya, *Cloud Data Centers and Cost Modeling: A complete guide to planning, designing and building a cloud data center*. Morgan Kaufmann, 2015, <https://www.elsevier.com/books/cloud-data-centers-and-cost-modeling/wu/978-0-12-801413-4>.
- [31] J. Tai, B. Sheng, Y. Yao, and N. Mi, “Live data migration for reducing sla violations in multi-tiered storage systems,” in *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. IEEE, 2014, pp. 361–366, <http://dx.doi.org/10.1109/ic2e.2014.8>.
- [32] W. Xiao, X. Lei, R. Li, N. Park, and D. J. Lilja, “Pass: a hybrid storage system for performance-synchronization tradeoffs using ssds,” in *Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on*. IEEE, 2012, pp. 403–410, <http://dx.doi.org/10.1109/ispa.2012.59>.
- [33] Square kilometre array. <https://www.skatelescope.org/>.
- [34] A. Simonet, A. Lebre, and A.-C. Orgerie, “Deploying distributed cloud infrastructures: Who and at what cost?” in *Cloud Engineering Workshop (IC2EW), 2016 IEEE International Conference on*. IEEE, 2016, pp. 178–183, <http://dx.doi.org/10.1109/ic2ew.2016.48>.
- [35] N. Grozev and R. Buyya, “Inter-cloud architectures and application brokering: taxonomy and survey,” *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014, <http://dx.doi.org/10.1002/spe.2168>.
- [36] A. N. Toosi, R. N. Calheiros, and R. Buyya, “Interconnected cloud computing environments: Challenges, taxonomy, and survey,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 7, 2014, <http://dx.doi.org/10.1145/2593512>.
- [37] N. K. Gill and S. Singh, “A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers,” *Future Generation Computer Systems*, vol. 65, pp. 10–32, 2016, <http://dx.doi.org/10.1016/j.future.2016.05.016>.
- [38] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. 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, 2011, <http://dx.doi.org/10.1002/spe.995>.

APPENDIX A: NOTATION TABLE

Symbol	Meaning
Federation	
F	Federation of a set of CPs
CP^d	cloud provider d
$Cap^{max, res}, Cap^{id, res}, P_{rsc}$	The total and the idle capacities of storage resource res and its price.
F_{res}^d	The insourcing price of storage resource res of CP^d
Customers	
U_{int}, u_k	The set of internal customers and k th customer
U_{ext}, u_k^d	The set of external customers and k 'th external customer of CP^d (internal customer of the cloud CP^d)
w_k	the workload of u_k
$iops_{sla,k}, ltc_{sla,k}$	The requested performance in terms of IOPS and latency
$pn_k, pn_{iops,k}, pn_{lte,k}$	The penalty function, and its parts iops penalty and latency penalty
$iops_{offered}(u_k)$	The IOPS offered to customer u_k
$lte_{offered}(u_k)$	The latency offered to customer u_k
Storage	
SC, sc_j	The set of storage classes, The j th storage class
$sc_{j,int}, sc_{j,ext}$	The internal and external customers storage parts
$c_{sc_j}, p_{sc_j}, w_{sc_j}$	The capacity, price and wear out of sc_j
Bandwidth	
bw, pbw	The bandwidth and its purchased cost
bw_{int}, bw_{ext}	The internal and external customers bandwidths
Objects	
$o_{i,k}, s_{o_{i,k}}$	The i th object of customer u_k and its size
$req_{op,o_{i,k}}$	The average IOPS of type op issued to the object $o_{i,k}$
$O_{m,int}, O_{m,ext}$	The set of internal (external, respectively) customers objects to migrate between internal storage classes
$A[i,j], B[i,j]$	the internal and external customers objects placement matrices
General	
T	Period of time
T_{sp}	The internet subscription period
Costs	
$Cost_{plc,T}$	The total placement cost
$Cost_{plc_{int},T}$	The internal customers objects placement cost
$Cost_{plc_{ext},T}$	The external customers objects placement cost
$Cost_{lcl_{int},T}$	The local placement costs of the internal customers
$Cost_{lcl_{ext},T}$	The local placement costs of the external customers
$Cost_{pnt_{int},T}$	The penalty costs of the internal customers
$Cost_{pnt_{ext},T}$	The penalty costs of the external customers
$Cost_{pnt_{iops},T}$	IOPS penalty costs
$Cost_{pnt_{lte},T}$	Latency penalty costs
$Cost_{stg_{int},T}$	The storage cost of the internal customers objects
$Cost_{stg_{ext},T}$	The storage cost of the external customers objects
$Cost_{migr_{int},T}$	The inner migration costs of the internal customers objects
$Cost_{migr_{ext},T}$	The inner migration costs of the external customers objects
$Cost_{out_{src},T}$	The outsourcing cost
$Cost_{in_{src},T}$	The insourcing cost
$Cost_{geo_{migr},T}$	The geo-migration cost
$Cost_{bck_{migr},T}$	The back-migration cost
$Cost_{re_{migr},T}$	The re-migration cost
$Cost_{geo_{rpl},T}$	The geo-replication cost
$Cost_{add_{rep},T}$	The cost of adding a replica
$Cost_{syn_{rep},T}$	The cost of synchronizing replicas
$Cost_{rd,T}$	The local read cost
$Cost_{rd_{enrg},T}$	The read energy cost
$Cost_{rd_{edr},T}$	The read wear out cost
$Cost_{wr,T}$	The local write cost
$Cost_{wr_{enrg},T}$	The write energy cost
$Cost_{wr_{edr},T}$	The write wear out cost
$Cost_{bw,T}$	The local consumed internet bandwidth cost
$Cost_{bw_{amz},1}$	The amortized internet bandwidth over one unit of time
$Cost_{ntw_{out},T}$	The external outgoing network cost
$Cost_{ext_{occ},T}$	The external occupation cost
$Cost_{ext_{wld},T}$	The external workload cost
$Cost_{ext_{pnt},T}$	The penalty cost