Economic data replication management in the cloud

Abdenour Lazeb[1], Riad Mokadem[2] and Ghalem Belalem[3]

[1] Université Oran1, Ahmed Ben Bella , Algérie
lazeb.abdenour@edu.univ-oran1.dz
[2] Institut de Recherche en Informatique de Toulouse (IRIT), Paul Sabatier University, Toulouse, France
Riad.Mokadem@irit.fr
[3] Université Oran1, Ahmed Ben Bella , Algérie
ghalem1dz@gmail.com

Abstract. The applications produce huge volumes of data that are distributed on remote and heterogeneous sites. This generates problems related to access and sharing of such data. As a result, managing data in large-scale environments is a real challenge. In this context, large-scale data management systems often use data replication, a well-known technique that treats generated problems by storing multiple copies of data, called replicas, across multiple nodes. Most of the replication strategies in these environments are difficult to adapt to cloud environments. They aim to achieve the best performance of the system without meeting the important objectives of the cloud provider. Our proposed approach generates the optimal replication strategy. In theory, we show that our algorithm significantly improves provider gain over a wide range of cloud and SLA-conditions without neglecting customer satisfaction.

Keywords: Data Management, Cloud Systems, SLA, Provider, Data Replication, Cost Model, Business Model, Performance.

1 Introduction

Cloud Computing could be common term utilized to portray a modern lesson of organizing based computing that takes put over the Web.

In addition, the stage gives on-demand services, that are continuously on, anytime and any place. Pay for utilizing and as required, elastic(scale up and down in capacity and functionalities). The equipment and software services are accessible to common public, undertakings, organizations and businesses markets But, what commitments does the cloud provider that you have chosen? How long will it take to restart your solution in case of a problem? Can he lose your data? These are classic questions that I am regularly asked when I talk about cloud computing the answer is in the SLA established between a cloud provider and its tenants, i.e., consumers. which includes the service level objectives (SLO) of the tenant, for example, availability and performance, which must be met by the provider.

For that reason, It is very important to focus on replication strategies for efficient and fast exploitation data in cloud. These strategies address classic problems such as: (i) which data to replicate? (ii) when to replicate these data? (iii) where to replicate these data but also to specific issues of the cloud environment such as (iv) determine the number of necessary replicas such as the objectives of the tenant will be satisfied while ensuring a profit for the cloud provider.

Some solutions can be brought to this problem:

i. The proposal of a cost model allowing replication only if it is necessary.
ii. Effective placement of data replicas.
iii. An elastic management of the number of replicas.
iv. The proposition of an economic model for the cloud provider such as information replication is advantageous. Usually conditioned by a minimization of the punishments paid by the provider which makes it possible to extend its economic profit.

To guarantee failure tolerance, a capacity advertising copies data among different copies. These copies store the same set of information, so in case any of copies is lost, information may still be gotten and recouped from the other replicas.

In this paper, we will propose an algorithm that mixes all these solutions for good replication management.

This paper is organized as follows: Section 2 tackle Related work, Section 3 explains our approach aspects ; Positioning of our approach is presented in Section 4 And at the last section contains the conclusions and future work.

2 Related Work

Several types of research have been dedicated to the field of dynamic replication. we find:

Fei Xie et al.[1] set three threshold parameters for dataset conditions among datasets, get to frequencies of datasets, and the storage capacity of information centers. Dataset reliance among datasets and get to recurrence for each dataset are calculated as limitations of the dataset. They utilize the limit esteem of capacity space to restrain information replication to maintain a strategic distance from flood issues and guarantee full errand completion in the corresponding area. They moreover classify information sorts into three categories, settled dataset, free-flexible dataset and constrained-flexible dataset, to develop a mapping between datasets and each information center. By receiving their methodology, they endeavor to assist diminish information development and information exchange cost. Their work I find it a little expensive compared to ours and does not treat current state each time.

Tadeusz et al. [2] propose a strategy for reproducing NoSQL information. The calculation is called Lorq. The most highlights of Lorq are (a) information replication is realized by implies of reproducing logs putting away upgrade operations, and so-called pulse operation sent by the pioneer; (b) the preparing and replication methodologies ensure that inevitably all operations in each replica are executed within the same arrange and no operation is misplaced. An uncommon consideration is paid to distinctive sorts of consistency, which can be ensured by the framework. they propose a strategy based on data put away by client administrations to ensure diverse consistency levels, in this manner actualizing SLA usefulness But replicated data types specification, verification and optimality [3] are neglecting.

Xiuguo et al. [5] went towards accomplishing the least taken a toll copies dispersion benchmark in a down to earth way, they propose a reproductions arrangements procedure show, counting the way to distinguish the need of making reproduction, and plan an calculation for copies situations that can effectively decrease the whole taking a toll within the cloud and proposing information sets administration fetched models, including capacity fetched and exchange taken a toll; showing a novel worldwide information set copies arrangements methodology from cost-effective see named MCRP, which is an inexact minimum-cost arrangement. They proposed a cost-effective information replication methodology with a thought of get to recurrence and the average response time to decide whether the dataset should be imitated or not in cloud environment.

Sathiya et al. [6] examine changes on a consistency convention called LibRe, which acts as an in-between consistency technique between the default inevitable consistency and the solid consistency choices determined from the crossing point property. The initial LibRe convention utilized a registry, which records the list of replica nodes containing the foremost later form of the information things. Consequently, alluding to the registry amid examined time makes a difference to forward the studied demands to a reproduction hub holding the foremost later form of the required information thing. For the other side, protocol would encounter brief inconsistency.

Tos et al. [8] propose Execution and Benefit Arranged Information Replication Methodology (PEPR) that guarantees SLA ensures, e.g. accessibility and execution, to the occupant whereas maximizing the financial advantage of the cloud provider. For the degree of execution, they consider reaction time ensure as a fundamental portion of the SLA. In PEPR, when assessing an inquiry, in the event that an assessed reaction time esteem is more prominent than the SLO reaction time limit, this implies that a replication prepare may be activated. At that time, economic benefit, i.e. profitability, of the cloud provider is additionally estimated. Replication choice is made as it were when both the reaction time and financial advantage of the provider are satisfied. The number of copies is powerfully balanced taking after whether the SLA objectives are fulfilled over time. Additionally, the least number of copies are continuously kept to guarantee least availability.Response time estimation for inhabitant inquiries are calculated when the queries are arrive at the cloud. If the estimation show that a alluring execution cannot be fulfilled, information replication is performed, but as it were when it is financially attainable for the provider.

Yaser et al. [9] think about is propelled by these pioneer considers as none of them can at the same time reply around arrangements and relocation times of objects. To address these questions, they make the taking after key commitments: To begin with, by misusing energetic programming, they define offline taken a toll optimization issue in which the ideal fetched of capacity, Get, Put, and movement is calculated where the precise future workload is assumed to be known a priori. Moment, they propose two online calculations to discover near-optimal taken a toll.

3 The Proposed Strategy

We propose a replication strategy which contains dependent and independent modules in the architecture shown in **Fig** 1. Each module plays a role in the Work Process.

.

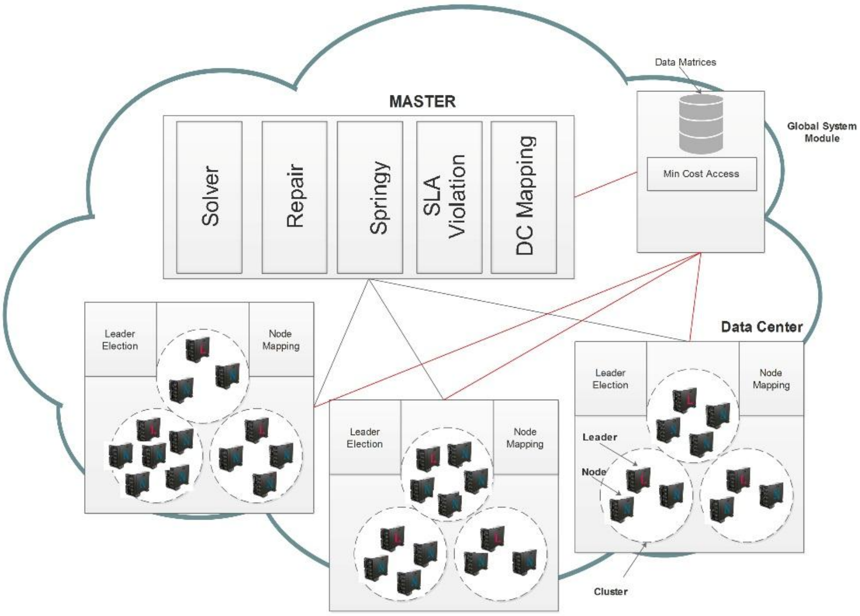

Fig. 1. Architecture of Proposed approach

3.1 Model System

For our model, we have a Master, Global System Module and data-centers. Master composed of five modules: Solver, Repair, Springy, SLA-Violation and DC Mapping.

Global System Module contains Min Cost Access and data matrices that are used by all the system.

In the end, we have a set of data-centers where: $DC_i = \{DC_1,\ DC_2,\ DC_3,\ DC_4,\ DC_5,\ldots,DC_N\}$ Each DC_i have two modules Leader Election and Node-Mapping and many Group where:

$G_{i.h} = \{G_{i.1},\ G_{i.2},\ G_{i.3},\ G_{i.4},\ldots,G_{i.H}\}$ Each Group have Leader $L_{i.h}$ So $L_{i.h} = \{L_{i.1}, L_{i.2}, L_{i.3}, L_{i.4},\ldots L_{i.H}\}$ Without forgetting that each group contain Lot of Node $N_{i.h.k} = \{N_{i.h.1},\ N_{i.h.2},\ N_{i.h.3},\ N_{i.h.4},\ldots N_{i.h.K}\}$. And of curse , we have

File for each Node in Group for Data center.
We mentioned that $F_{i.h,k,j} = \{ F_{i.h.k.1},\ F_{i.h.k.2},\ F_{i.h.k.3}, \dots\ F_{i.h.k.M} \}$.

For example, $F_{5.7.2.3}$ means file 3 of node 2 of group 7 of data center 5.

3.2 Description

our architecture consists of three module described as follows:

Global System Module: it's a component contains three elements that are used globally throughout the system

Min Access Cost: for all transactions, it is necessary that the requests follow the shortest way to arrive at the desired destination (Djikstra algorithm) without neglecting the cost of access for each Node traversed.

$$Min \sum_{j=1}^{destination} C_{acc_j} + \sum_{i=1}^{destination} C_{transfer_i} \tag{1}$$

C_{acc_j} : cost of access Node j. $C_{transfer_i}$: Cost of transfer by the link i.

Data Matrices: As his name means, it accommodate three matrices:
Matrix of Popularity P_{ihkj} , which refers to the access frequency for each replica .
Matrix of Capacity-Node S_{ihk} for the storage of Host (Node) .
Matrix of Size-Dataset V_j for storage space of Dataset .
Matrix of Threshold $Td_{u,j}$ is compromise between quality of service, maximum budget and minimum response time for each customer ($Td_{u,j} = \alpha_{u,j} + \beta_{u,j} + \varphi_{u,j}$) where alpha beta gama are mark Level (for exemple $\alpha_{u,j} = 1$, any response time with $\beta_{u,j} = 3$, average quality and $\varphi_{u,j} = 5$ a very high budget) all these parameters are established in a contract SLA-Conditions.

Master Module: We can call it the brain since it has several components we start with:

Solver: Since our problem is to determine the number of necessary replicas such as the objectives of the tenant will be satisfied while ensuring a profit for the cloud provider we think about this Mathematical statement of the problem:

$$\begin{aligned} &Minimize \quad \sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}x_{ij} \\ &\text{subject to} \quad \sum_{j=1}^{m} v_j x_{ij} \le Cap_i, \quad i = 1,\dots n \\ &\qquad \sum_{i=1}^{n} a_j x_{ij} \ge Td_{u,j},\ j = 1,\dots m, u = 1,\dots r \\ &\qquad x_{ij} \in N, \quad i = 1,\dots n,\ j = 1,\dots m \end{aligned} \quad (2)$$

Where:

C_{ij} : replication cost and allocate Dataset j space in the datacenter i ($F_{i.h.k.j}$) // whatever the node or the cluster

X_{ij}: number replicas of dataset j in the datacenter i ($F_{i.h.k.j}$)

V_j: storage space (file size) of Dataset j ($F_{i.h.k.j}$) (file size)

Cap_i : Storage capacity Of DC_i where

$$Cap_i = \sum_{k=1}^{K}\sum_{h=1}^{H} S_{ihk} \quad (3)$$

a_j: the Coefficient importance of Dataset j ($F_{i.h.k.j}$)

$$a_j = \frac{1}{\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{h=1}^{H} P_{ihkj}} \quad (4)$$

$Td_{u,j}$: Threshold SLA fixed by provider and consumer u for the Dataset j ($F_{i.h.k.j}$)

Such as the resolution is done by the simplex. At the end, we replicate and delete to reach the optimal number of replicas.

Repair: We want the system to take into consideration consistency and fault tolerance so it is preferable to resolve them by the principle of quorum we start to launch a verification request for all the replicas to obtain the correct value by the majority and we make the update for all the data with errors and false value.

Springy: An elasticity of the resources it can be an increase or decrease according to the popularity of each data, it means duplicate a data F_{ihkj} if its popularity (frequency of access to this data) is greater than a given threshold $T_{replication}$ and delete a data F_{ihkj} if its popularity (frequency of access to this data) is lower than a given threshold $T_{erasure}$.

The popularity of each file is calculated by the following simple formula:

$$P_{ihkj} = \frac{number\ of\ request\ for\ the\ F_{ihkj}}{total\ number\ of\ request} \quad (5)$$

SLA Violation: SLA Violation Use quality of service, maximum budget for each customer and minimum response time as a constraint in a cost minimization based algorithm. in cases of very strict violation, the Replication mechanism triggers an increase in resources.

DC Mapping: a module made to orientate the request to the Concerned data center since it contains an information structure about all data centers (fast indexing).

Data-Center Module: We see that the data centers contains cluster each cluster contains nodes and a leader as well as two module Node mapping and leader election..

Leader Election: Sometimes the completion of a task requires the involvement of multiple instances of the same cloud service. If the service consumer invoking the cloud service instances does not have the necessary logic to coordinate them, runtime exceptions can occur leading to data corruption and failure to complete the task.

These are the most available nodes in each Group in Data Center by following Algo of selection (sorted list of the availability of the nodes and takes the first one).

Node Mapping: a module made to orientate the request to the Concerned data center since it contains an information structure about all Node in the Data Center (fast indexing) . It is linked with the Leaders.

3.3 Functioning of system

As we already said before, each module has a mechanism that works in it. In the following section, we will describe them one by one:

Solver: a sequence of operations to be executed:

First, Master Ask for data matrices from Global System Module to solve the problem of constraints. Then, Master sends an execution command according to the result obtained by the solver to Datacenters. Next, Leaders ask for concerned Node and receive Result. Finally, they Execute Commands (Replicate or Delete).

Response and Repair: For this part, we distinguish three cases:
- 1st Case: we have: user request node for data and Get response immediately.
if the response not founded we switch to the second case

- 2nd Case: we have: First, user request node for data .then request transferred to one leader of data-center. This last one ask for concerned Node and transfer request to him. Finally, user receive response.
If the response not founded locally (in the data center) we switch to the third case

- 3rd Case we have: First, user request node for data .then request transferred to one leader of data-center. This last one ask for concerned Node and receive “Not founded”. So , he Transfer request to Master who also transfer request to leaders (one for each datacenter).then Leader request for the value of the data after applying the majority quorum (we keep the true value and we correct the rest).Then, they send the value to the master who creates a copy at the nearest customer node (bring data closer to the client -best client- reduce response time, cost of access and replicate to increase availability).Finally , Response will be transferred to the user.

Springy: the procedure starts with : First , Master ask for Popularity Matrix from Global System Module sending to solve algorithm (already mentioned in description section).

Next, Master sends an execution command according to the result obtained by the Springy Modules to Datacenters (Leaders).

Leaders ask for concerned Node and receive Result. Finally, they Execute Commands (Replicate or Delete).

SLA Violation: the success of operations begins with :

2) User triggers a violation alert. Then, Master send a command to increase resources.So Leaders take the data for replicated in the closest node to the client.

4 Positioning Of Our Approach

We will try to position our approach in relation to other articles in the state of the art by taking into consideration several characteristics described as follows:

Fig. 2 Comparison table of different approaches

	Cost	latency	Availability	throughput / bandwidth	Placement	Size replica	Consistency	Fault tolerance	Popularity	Leader Election
Fei-Xie(2017) [1]	Yes	Yes	Yes		Yes	Yes			Yes	
Hussam Abu-Libdeh(2013) [12]							Yes	Yes		
Guy-Laden-(2011) [10]		Yes		Yes						
Jayalakshmi-D.S 2015 [14]		Yes	Yes	Yes					Yes	
Ismaeel-AlRidhawi 2015 [13]	Yes	Yes	Yes	Yes	Yes					
Yaser Mansouri 2017 [9]	Yes	Yes			Yes					
Zhendong Cheng 2012 [4]			Yes	Yes	Yes					
Najme Mansouri 2015 [11]	Yes	Yes	Yes		Yes	Yes		Yes		
Mohammad Bsoul 2013 [7]		Yes	Yes	Yes					Yes	
Ilir-Fetai 2017[15]	Yes	Yes					Yes			
Farouk Bouharaouaa 2017 [16]	Yes	Yes	Yes				Yes			Yes
Our Approach	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Fig 2 shown Comparison table of different approaches compared to the features and services studied in this experiment and before like Cost, latency, Availability, throughput / bandwidth, Placement, Size,replica, Consistency, Fault tolerance, Popularity, Leader Election.

For example, if we compare article of Mohammed Bsoul (2013) [7] with our article we find that he studied latency, avaibility and popularity but he has neglected the cost and the consistency thus other parameters that one included them in our approach.

In the other side, we find that Ilir Fetai (2017)[14] omit Avaibility and placement of replica . These two parameter are important for our studies.

On the other hand, the approach of Najme MANSOURI (2015) [11] has been very interesting since she has studied several criteria but she has neglected the popularity of the files which has an impact on the when and how much to replicate and delete.

Finally, no way to compare with Hussam Abu-Libdeh(2013) [12] because it focus on two parameters are consistency and tolerance to failure on ten parameters to study in our approach .

5 Conclusion and Perspectives

In this investigate work; replication of datasets has been presented with existing information arrangement procedure. It is incomprehensible to fulfill all the conditions to put the datasets at fitting position where all assignments can get to the information with the least information exchange cost and fulfillment of SLA goals despite the fact of using the majority of the parameters like cost, latency, response time, popularity with different method like simplex and quorum. In the future, we try to apply this method in a simulation environment as cloudsim to better deflate the results of this proposed approach.

References

1. Fei Xie ; Jun Yan ; Jun Shen, "Towards Cost Reduction in Cloud-Based Workflow Management," Fifth International Conference on Advanced Cloud and Big Data, (2017).
2. T. Pankowski, "Lorq: A System for Replicated NoSQL Data," Evaluation of Novel Approaches to Software Engineering. ENASE 2015, pp. 62-79, (2016).
3. Sebastian Burckhardt, "Replicated data types: specification, verification, optimality," ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, (2014).
4. Zhendong Cheng ; Zhongzhi Luan ; You Meng ; Yijing Xu ; Depei Qian ; Alain Roy ; Ning Zhang ; Gang Guan, "ERMS : An Elastic Replication Management System for HDFS," IEEE International Conference on Cluster Computing Workshops, pp. 32-40, (2012).
5. Xiuguo Wu, "Data Sets Replicas Placements Strategy from Cost-Effective View in the Cloud," Hindawi Publishing Corporation, pp. 1-14, (2016).

6. Sathiya Prabhu Kumar , Sylvain Lefebvre ,"CaLibRe: A better Consistency-Latency Tradeoff for Quorum based Replication systems," Database and Expert Systems Applications. Globe (2015), DEXA 2015. Lecture Notes in Computer Science, pp. 2-14,(2015).
7. Mohammad Bsoul ; Alaa E. Abdallah ; Khaled Almakadmeh ; "A Round-based Data Replication Strategy," IEEE Transactions on Parallel and Distributed Systems, pp. 2-18, (2013).
8. Uras Tos ; Riad Mokadem ; Abdelkader Hameurlain ; Tolga Ayav ; Sebnem Bora , "A Performance and Profit Oriented Data Replication Strategy for Cloud Systems," IEEE Conferences on Ubiquitous Intelligence & Computing, (2016).
9. Yaser Mansouri ; Adel Nadjaran Toosi ; Rajkumar Buyya, "Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers," IEEE Transactions on Cloud Computing, pp. 1-16, (2017).
10. Guy Laden, Roie Melamed , Ymir Vigfusson "Adaptive and Dynamic Funnel Replication in Clouds," ACM SIGOPS Operating Systems Review, vol. 46, pp. 40-46, (2011).
11. N. MANSOURI, "Adaptive data replication strategy in cloud computing for performance improvement," Frontiers of Computer Science., pp. 1-11,(2015).
12. H. Abu-Libdeh, "Elastic Replication for Scalable Consistent Services" (2012).
13. I.Al Ridhawi, Nour Mostafa, Wassim Masri «Location-Aware Data Replication in Cloud Computing Systems," Eight International Workshop on Selected Topics in Mobile and Wireless Computing, (2015).
14. Jayalakshmi D. S. ; Rashmi Ranjana T. P. ; R. Srinivasan, "Dynamic Data Replication Strategy in Cloud Environments" Fifth International Conference on Advances in Computing and Communications, (2015).
15. Ilir Fetai ; Alexander Stiemer ; Heiko Schuldt , "QuAD: A quorum protocol for adaptive data management in the cloud",Big Data (Big Data), IEEE International Conference on, 405-414, (2017) .
16. Bouharaoua, Farouk, Belalem, Ghalem "A quorum-based intelligent replicas management in data grids to improve performances",Multiagent and Grid Systems, 13, 143--161,(2017).