Lecture Notes in Networks and Systems 156

Salim Chikhi · Abdelmalek Amine ·
Allaoua Chaoui ·
Djamel Eddine Saidouni ·
Mohamed Khireddine Kholladi *Editors*

Modelling and Implementation of Complex Systems

Proceedings of the 6th International Symposium, MISC 2020, Batna, Algeria, October 24–26, 2020

Springer

Lecture Notes in Networks and Systems

Volume 156

The series "Lecture Notes in Networks and Systems" publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

**** Indexing: The books of this series are submitted to ISI Proceedings, SCOPUS, Google Scholar and Springerlink ****

More information about this series at http://www.springer.com/series/15179

Salim Chikhi · Abdelmalek Amine ·
Allaoua Chaoui · Djamel Eddine Saidouni ·
Mohamed Khireddine Kholladi
Editors

Modelling and Implementation of Complex Systems

Proceedings of the 6th International Symposium, MISC 2020, Batna, Algeria, October 24–26, 2020

Springer

Editors
Salim Chikhi
Faculty of New Information
and Communication Technologies
University of Constantine 2
Constantine, Algeria

Allaoua Chaoui
Faculty of New Information
and Communication Technologies
University of Constantine 2
Constantine, Algeria

Mohamed Khireddine Kholladi
Faculty of Science and Technology,
Department of Mathematics
and Computer Science
University of El Oued
El-Oued, Algeria

Abdelmalek Amine
Faculty of Technology
University of Saida
Saida, Algeria

Djamel Eddine Saidouni
Faculty of New Information
and Communication Technologies
University of Constantine 2
Constantine, Algeria

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-030-58860-1 ISBN 978-3-030-58861-8 (eBook)
https://doi.org/10.1007/978-3-030-58861-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume contains research papers accepted and presented at the 6th International Symposium on Modelling and Implementation of Complex Systems (MISC 2020), held during October 24–26, 2020, in Batna, Algeria. As the previous editions (MISC 2010 to MISC 2018), this symposium is intended as a tradition offering open forum and meeting space for researchers working in the field of complex systems science. This year, the MISC symposium received 113 submissions from 12 countries: Algeria, Belgium, Finland, France, UK, India, Ireland, Libya, Pakistan, Singapore, Turkey and UAE. In a rigorous reviewing process, the Program Committee selected 21 papers, which represent an acceptance rate of 19%. The PC included 112 researchers and 29 additional reviewers from 11 countries. The accepted papers were organized into sessions as follows: Internet of Things and Smart Systems, Machine Intelligence and Data Science, Cloud Computing and Networking, and Software Technology and Model Transformations.

We would like to thank the co-chairs of the Program Committee and all its members for their effort in the review process and the selection of the papers. We are grateful to the Organizing Committee members from the University of Batna 2 and the University of Constantine 2 for their contribution to the success of the symposium. Our thanks also go to the authors who submitted papers for their interest to our symposium. Enough thanks cannot be expressed to Dr. Nabil Belala for managing EasyChair system for MISC 2020 from submissions to proceedings elaboration and Dr. Ahmed-Chawki Chaouche for managing the symposium Web site.

October 2020

Salim Chikhi
Allaoua Chaoui
Abdelmalek Amine

Organization

The 6th International Symposium on Modelling and Implementation of Complex Systems (MISC 2020) was co-organized by University of Constantine 2–Abdelhamid Mehri and University of Batna 2–Mostefa Ben Boulaïd and took place in Batna, Algeria (October 24–26, 2020).

Honorary Chairs

Hacene Smadi	University of Batna 2, Algeria
Abdelouahab Chemam	University of Constantine 2, Algeria

General Chairs

Salim Chikhi	University of Constantine 2, Algeria
Abdelmalek Amine	University of Saïda, Algeria
Hamouma Moumen	University of Batna 2, Algeria

Steering Committee

Allaoua Chaoui	University of Constantine 2, Algeria
Salim Chikhi	University of Constantine 2, Algeria
Mohamed-Khireddine Kholladi	University of El Oued, Algeria
Djamel Eddine Saïdouni	University of Constantine 2, Algeria
Mohamed Bachir Yagoubi	University of Laghouat, Algeria
Hamouma Moumen	University of Batna 2, Algeria

Organizing Committee Chairs

Djamel Bellala	University of Batna 2, Algeria
Mourad Bouzenada	University of Constantine 2, Algeria

Organizing Committee

Ouahab Kadri	University of Batna 2, Algeria
Souheila Bouam	University of Batna 2, Algeria
Rima Djellab	University of Batna 2, Algeria
Brahimi Mahmoud	University of Batna 2, Algeria
Mounia Laouar	University of Batna 2, Algeria
Titouna Chafik	University of Batna 2, Algeria
Hocine Riadh	University of Batna 2, Algeria
Djeffal El-Amir	University of Batna 2, Algeria
Khamsa Djaroudib	University of Batna 2, Algeria
Alleg Mebarek	University of Batna 2, Algeria
Halaoua Madjid	University of Batna 2, Algeria
Chaabane Mezache	University of Batna 2, Algeria
Amel Behaz	University of Batna 2, Algeria
Ali Bentahar	University of Batna 2, Algeria
Farouk Zireg	University of Batna 2, Algeria
Hafiza Chaouche	University of Batna 2, Algeria

Publication Chairs

Nabil Belala	University of Constantine 2, Algeria
Ahmed-Chawki Chaouche	University of Constantine 2, Algeria

Publicity and Sponsor Chairs

Bedreddine Barguia	University of Batna 2, Algeria
Rafik Nezzar	University of Batna 2, Algeria
Ishak Benmohammed	University of Constantine 2, Algeria
Sami Barkat	University of Batna 2, Algeria

Program Committee Chairs

Allaoua Chaoui	University of Constantine 2, Algeria
Djamel Eddine Saïdouni	University of Constantine 2, Algeria
Chafik Arar	University of Batna 2, Algeria

Program Committee

Sihem Abbassen	University of Constantine 2, Algeria
Abdelkrim Abdelli	USTHB of Algiers, Algeria
Wahabou Abdou	University of Bourgogne, France
Abdelmalek Amine	University of Saida, Algeria
Abdelkrim Amirat	University of Souk Ahras, Algeria
Chafik Arar	University of Batna 2, Algeria
Baghdad Atmani	University of Oran 1, Algeria
Mohamed Chaouki Babahenini	University of Biskra, Algeria
Abdelmalik Bachir	University of Biskra, Algeria
Mohamed Batouche	University of Constantine 2, Algeria
Amel Behaz	University of Batna 2, Algeria
Ali Behloul	University of Batna 2, Algeria
Faiza Belala	University of Constantine 2, Algeria
Nabil Belala	University of Constantine 2, Algeria
Djamel Bellala	University of Batna 2, Algeria
Ghalem Belalam	University Oran 1, Algeria
Hacène Belhadef	University of Constantine 2, Algeria
Fouzia Benchikha	University of Constantine 2, Algeria
Saber Benharzallah	University of Batna 2, Algeria
Mohamed Benmohammed	University of Constantine 2, Algeria
Hammadi Bennoui	University of Biskra, Algeria
Azeddine Bilami	University of Batna 2, Algeria
Salim Bitam	University of Biskra, Algeria
Mustapha Bouakkaz	University of Laghouat, Algeria
Souheila Bouam	University of Batna 2, Algeria
Karim Bouamrane	University of Oran 1, Algeria
Djalal Eddine Boubiche	University of Batna 2, Algeria
Samia Boucherkha	University of Constantine 2, Algeria
Mohamed Amine Boudia	University of Saida, Algeria
Rachid Boudour	University of Annaba, Algeria
Mahmoud Boufaida	University of Constantine 2, Algeria
Zizette Boufaida	University of Constantine 2, Algeria
Kamel Boukhalfa	USTHB of Algiers, Algeria
Abdelkrim Bouramoul	University of Constantine 2, Algeria
Elbey Bourennane	University of Bourgogne, France
Mourad Bouzenada	University of Constantine 2, Algeria
Rachid Chalal	ESI of Algiers, Algeria
Yacine Challal	ESI of Algiers, Algeria
Ahmed-Chawki Chaouche	University of Constantine 2, Algeria
Foudil Cherif	University of Biskra, Algeria
Abdellah Chouarfia	University USTO of Oran, Algeria
Lakhdar Derdouri	University of Oum El Bouaghi, Algeria

Karim Djemame	University of Leeds, UK
Mahieddine Djoudi	University of Poitiers, France
Amer Draa	University of Constantine 2, Algeria
Khalil Drira	LAAS of Toulouse, France
Cédric Eichler	INSA Bourges, France
Zakaria Elberrichi	University of Sidi Bel Abbes, Algeria
Nadir Farah	University of Annaba, Algeria
Mohamed Amine Ferrag	University of Guelma, Algeria
Khadoudja Ghanem	University of Constantine 2, Algeria
Salim Ghanemi	University of Annaba, Algeria
Mohamed Gharzouli	University of Constantine 2, Algeria
Nacira Ghoualmi	University of Annaba, Algeria
Said Ghoul	Philadelphia University, Jordan
Zahia Guessoum	Sorbonne University, France
Gregorio Díaz-Descalzo	Castilla La Mancha University, Spain
Larbi Guezouli	University of Batna 2, Algeria
Lyamine Guezouli	University of Batna 2, Algeria
Djamila Hamdadou	University of Oran, Algeria
Saad Harous	United Arab Emirates University, UAE
Dalila Hocine	University of Tizi Ouzou, Algeria
Riadh Hocine	University of Batna 2, Algeria
Jean-Michel Ilié	Sorbonne University, France
Ouahab Kadri	University of Batna 2, Algeria
Okba Kazar	University of Biskra, Algeria
Tahar Kechadi	UCD School Dublin, Ireland
Elhilali Kerkouche	University of Jijel, Algeria
Kamel Khoualdi	University of King Abdulaziz of Jeddah, Saudi Arabia
Mohamed Nadjib Kouahla	University of Guelma, Algeria
Mohamed Tahar Kimour	University of Annaba, Algeria
Said Labed	University of Constantine 2, Algeria
Zakaria Laboudi	University of Oum El Bouaghi, Algeria
Yacine Lafifi	University of Guelma, Algeria
Ali Lemouari	University of Jijel, Algeria
Ramdane Maamri	University of Constantine 2, Algeria
Mimoun Malki	University of Sidi Bel Abbes, Algeria
Mourad Maouche	Philadelphia University, Jordan
Smaine Mazouzi	University of Skikda, Algeria
Djamila Mechta	University of Setif 1, Algeria
Kamel Eddine Melkemi	University of Batna 2, Algeria
Salah Merniz	University of Constantine 2, Algeria
Hayet-Farida Merouani	University of Annaba, Algeria
Mohammed A. Merzoug	University of Batna 2, Algeria
Souham Meshoul	University of Constantine 2, Algeria
Kamel Messaoudi	University of Souk Ahras, Algeria

Djamel Meslati	University of Annaba, Algeria
Chaker Mezioud	University of Constantine 2, Algeria
Abdelouaheb Moussaoui	University of Setif, Algeria
Mohamed Elhadi Rahmani	University of Saida, Algeria
Mathieu Roche	CIRAD of Montpellier, France
Zaidi Sahnoun	University of Constantine 2, Algeria
Maamar Sedrati	University of Batna 2, Algeria
Rachid Seghir	University of Batna 2, Algeria
Larbi Sekhri	University Oran 1, Algeria
Abdelhak-Djamel Seriai	University of Montpellier 2, France
Hamid Seridi	University of Guelma, Algeria
Hassina Seridi	University of Annaba, Algeria
Yahia Slimani	Faculty of Science of Tunis, Tunisia
Hichem Talbi	University of Constantine 2, Algeria
Noria Taghezout	University Oran 1, Algeria
Sadek Labib Terissa	University of Biskra, Algeria
Chouki Tibermacine	University of Montpelier 2, France
Chafik Titouna	University of Batna 2, Algeria
Faiza Titouna	University of Batna 2, Algeria
Belabbas Yagoubi	University of Oran 1, Algeria
Nacereddine Zarour	University of Constantine 2, Algeria
Nadia Zeghib	University of Constantine 2, Algeria
Mounira Zerari	University of Constantine 2, Algeria
Amer Zerek	Zawia University, Libya
Noureddine Zerhouni	ENSMM of Besançon, France
Abdelhafid Zitouni	University of Constantine 2, Algeria

Co-editors

Abdelmalek Amine	University of Saïda, Algeria
Allaoua Chaoui	University of Constantine 2, Algeria
Salim Chikhi	University of Constantine 2, Algeria
Mohamed Khireddine Kholladi	University of El Oued, Algeria
Djamel Eddine Saïdouni	University of Constantine 2, Algeria

Additional Reviewers

Barka, Kamel
Belayachi, Naima
Bensalem, Imene
Benzadri, Zakaria
Boucebsi, Rachida
Boukelia, Abdelbasset

Boulemden, Ahmed
Brahimi, Mohamed
Brahimi, Said
Djellal, Asma
Farou, Brahim
Guessoum, Meriem
Heba, Ismail
Kerdoudi, Mohamed Lamine
Kouahla, Zineddine
Mami, Mohammed Amine
Mansoul, Abdelhak
Miles, Badr-Eddine
Rhayem, Ahlem
Rahab, Hichem
Rouabhi, Miloud
Sellami, Samir
Taibi, Aissa
Yachba, Khadidja
Zarour, Karim
Zernadji, Tarek
Zitouni, Farouq

Contents

Cloud Computing, Networking and IoT

Dynamic Replication Based on a Data Classification Model in Cloud Computing

Imad Eddine Miloudi, Belabbas Yagoubi(✉), and Fatima Zohra Bellounar

Laboratoire d'Informatique d'Oran - LIO, University Oran 1, Oran, Algeria
miloudi.imad@gmail.com, byagoubi@gmail.com, fbellounar@yahoo.fr

Abstract. Cloud Computing provides on demand resources for customers and enterprises to outsource their online activities efficiently and less expensively. However, the cloud environment is heterogeneous and very dynamic, storage node failures and increasing demands on data can lead to data unavailability situations leading to a decrease in quality of service. Cloud service providers face the challenge of ensuring maximum data availability and reliability. Replication of data to different nodes in the cloud has become the most common solution for achieving good performance in terms of load balancing, response time and availability. In this article, we propose a new dynamic replication strategy based on a data classification model that would adapt the replication process according to user behavior towards data. This strategy dynamically and adaptively creates the replicas necessary in order to obtain the desired performance such as, reduced response time and improved system availability while ensuring the quality of service. The solution also attempts to meet customer requirements by respecting the SLA contract. The CloudSim simulator was used to evaluate the proposed strategy and compare it to other strategies. The results obtained showed an improvement in the criteria studied in a satisfactory manner.

Keywords: Cloud computing · Dynamic replication · Classification · SLA · Cloudsim

1 Introduction

Cloud computing is a universal and practical model that provides services on demand [1, 2]. Where the customer only pays for what he consumes [3]. This paradigm provides hardware infrastructure and software as services to enterprises and users [4]. As a result, cloud computing frees the end user from the problems of finding computing and data storage capacity [5]. With this model, the client can choose the services he needs. All cloud service providers must provide the right quality of service (QoS) metrics to speed up data access, increase data availability, provide better fault tolerance and data recovery, distribute workload and minimize bandwidth consumption [6, 7]. So the Cloud faces several challenges which notably affect the quality of service. Several data management systems have been proposed, to satisfy these metrics, such as the Google File System

S. Chikhi et al. (Eds.): MISC 2020, LNNS 156, pp. 3–17, 2021.
https://doi.org/10.1007/978-3-030-58861-8_1

[8] and the Hadoop distributed file system [9], all these systems adopt a key technique which is data replication.

Replication is the process that creates different replicas of the same service on different nodes. With this technique, a data intensive application or system can achieve high availability, better fault tolerance and data recovery as well as high performance [6, 10]. Data replication is indispensable for large scale systems, especially clouds. However, like each technique, replication has many problems that must be taken into account before being able to benefit from all the advantages offered [11]. These problems generate several questions such as: What to replicate?, How many to replicate?, Where to place replicas?, When to replicate?.

Our contribution consists in proposing a model for the management of data replication in Cloud Computing. This solution aims to optimize the performance required by users in terms of availability and especially in terms of reduction of response time. The proposed solution is based on a data classification model that would adapt the replication process according to user behavior towards data. The solution also attempts to meet the replication objectives mentioned above. It has been implemented and simulated using the CloudSim simulator [12] as well as the results have been compared with other existing strategies.

The rest of the article is organized as follows: Sect. 2 cites related work. Section 3 is reserved for a detailed description of the proposed strategy. Section 4 illustrates the experimental results and the last section concludes this work and exposes future work.

2 Related Work

There are many works which study the management of replicas in the Cloud and which are based on different models, we can cite.

The authors of [13] study availability and workload problems in cloud computing, by proposing a mathematical model that calculates the minimum number of replicas in order to satisfy the availability requirements of a file and estimates the probability of blocking the nodes in a heterogeneous environment to place the replicas. Results demonstrate that the strategy meets availability requirements, access latency has been improved, load balancing and system stability have been observed.

In this article [14], the authors focused on the problem of cost in could computing, through the proposal of a new strategy called Cost effective Incremental Replication (CIR) in datacenters. This strategy applies an incremental replication approach to minimize the number of replicas while meeting reliability requirements in order to achieve cost effectiveness. In the evaluation simulation, they have demonstrated that the strategy can significantly reduce the cost of data storage, especially when the data is stored for a short period of time or is not very reliable.

In order to optimize energy consumption and communication delay, a new replication solution has been proposed in [15]. This solution used a three tiered topology: core, aggregation and access which are interconnected. This data replication technique improves communication delay and network bandwidth between geographically distributed datacenters as well as the interior of each datacenter. The performance evaluation was carried out using GreenCloud.

The author in [16] proposed a multi-objective dynamic replication strategy to find the placement and replacement of replicas. In this strategy, users can define weightings according to their own needs such as setting a higher value on expected performance. The results demonstrate that strategy can improve performance in terms of average response time, efficient use of the network, load balancing, frequency of replication and storage usage.

The authors [17] proposed an algorithm that selects the best site according to the context of the network and the data nodes which can reduce the average request time of the file. This algorithm is based on two phases, the first reduces the search in the catalog and the second proposes a selection model which envisages an overview of the defined criteria to make the best selection of the replica site.

In [18] the authors presented a dynamic data replication strategy used exponential smoothing prediction method, This strategy analyzes the access history of each file according to the number of accesses in order to predict the optimal number of replicas. This allows them to reduce the response time and the additional cost of the cloud storage system.

The authors of [19] presented a dynamic replication strategy that responds to customer needs while taking into account the benefit to the provider. This strategy is based on estimates of expenses and revenues during the execution of a query. As a result, replicas are distributed based on this cost. In addition, the placement is done with a load balancing between the regions. The results indicate that strategy can significantly increase the availability and performance of the cloud system, while taking into account the profit of the provider.

3 The Proposed Replication Strategy

Given the large volume of data stored in the Cloud as well as the unpredictable behavior of users who consult this data at widely varying frequencies, an adaptive data replication model is required. We found that there are periods when data are consulted much more than other periods, such as the summer product sales period, football championship season, occasions, national and religious holidays, …etc. Consequently, we thought of integrating into the replication model a data classification module which will replicate the data predictively.

Our contribution therefore consists in proposing a replication model based on user behavior using data classification. This model will improve the response time to requests, availability, storage capacity of nodes as well as the management of the replicas in order to satisfy the needs of the users according to the SLA contract.

To better understand the proposed approach, we will present the diagram below which gives a global view of the functionalities of the application (Fig. 1):

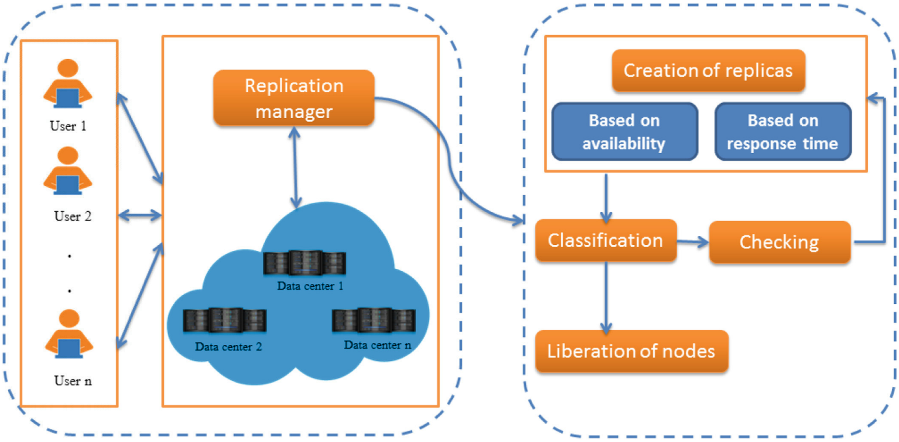

Fig. 1. Functional diagram of the proposed strategy

3.1 Creation

3.1.1 Creating Replicas Based on Availability

Initially, the proposed strategy creates replicas according to the availability required by the user. Data replication is necessary if the availability required in the SLA contract is higher than the current availability of a storage node. To calculate the minimum number of replicas that must be added to the system in order to guarantee the required availability of the file we use the following formula 1 [20, 21]:

$$A_{file\ i} = 1 - (1 - A_{nodej})^n \quad (1)$$

$$\text{Then} \quad n = \frac{\ln\left(1 - A_{file\ i}\right)}{\ln\left(1 - A_{nodej}\right)} \quad (2)$$

Where n is the number of replicas meeting the availability for a file, $A_{file\ i}$ is the availability required in SLA for a file and A_{nodej} is the availability of a storage node.

3.1.2 Creating Replicas Based on Response Time

After creating the necessary replicas to ensure availability, we will regularly test if these replicas still satisfy the SLA response time. Otherwise, this module will be triggered.

Initially, the number of replicas to satisfy the SLA response time is calculated using the following formula 3 [16]:

$$NBr_i = \frac{\sum_{J=1}^{n} \frac{size_i}{Bj} * AF(i,j)}{RT_i SLA} \quad (3)$$

Where NBr_i represents the number of replicas estimated for file i by responding to the response time specified in SLA, $size_i$ indicates the size of file i, Bj indicates the bandwidth of node j, $AF(i, j)$ presents the access frequency of file i on node j, RT_iSLA is the response time specified in the SLA contract for file i.

After calculating this number, we are going to calculate the difference between the number of replicas, NBr_i and the number of replicas meeting availability n which is calculated by the formula 4.

$$Th = NBr - n \tag{4}$$

If Th is more than zero, replication is necessary, we will then save the data with the dates of access and the number Th. These dates will be used to classify the data. The operation of creating response time based replicas will be repeated on a regular basis in order to avoid violation of the SLA contract. This will enrich the histories and will be used for the classification of data according to the dates of consultation.

3.2 Classification

This module consists to make a classification of data according to the behavior of the users. It represents a large part of our contribution. We are leaning on the step of classification because it will help us to create and remove replicas in a dynamic and adaptive manner. This will have like effect a profit in storage space and service on the side of customers and providers.

In this module, we have classified the data according to the time constraint. Data is classified according to the period when the number of accesses is very high. The principal objective of this module is to determine the period of time when the data does not have too much access, in order to determine its lifetime and be able to trigger the liberation module if necessary. Another objective is the assurance of response time for viral data.

3.2.1 Classified Data Types

- **Data relating to the occasion:** the data is consulted in a period less than or equal to 15 days.
- **Instant data:** the data was consulted in only one moment
- **Viral data:** the data is always consulted with a very high rate
- **Unclassified data:** the data has been consulted in different periods compared to previous years.
- **Half yearly data:** the data has been accessed in a six month period.
- **Periodic data:** the data has been consulted in a period of nine months

3.2.2 The Process to Follow

The classification of data types will be determined by following several steps:

1. **Union and intersection of dates by years:** after creating the replicas responding to the response time, we have as output the consultation dates by different years which are registered with the number of replicas. The union and the intersection of these dates give us the intervals used in the calculations below.

2. **Calculation of the number of union (Nu) and intersection days (Ni) as well as the start and end date of each:** this step will help us to determine the data membership class, specifying on the one hand, the number of days of common consultation dates for the intersection and on the other hand the number of days of all dates for the union.
3. **Calculation of NES (Number of days between end and start date of the intersection):** the NES represents the difference between the start and the end date of the intersection interval. Through this parameter, we will determine if the dates are near or not. To determine if the data will be classified or not, the P parameter of derogation has been proposed.
4. **Calculation of the derogation parameter (P)**: it is a parameter that represents the number of additional days tolerated for a data to be classified. (See the following example)

The parameter "P" is expressed by the following formula:

$$P = \frac{(I[1]-U[1]) + (U[n] - I[n])}{Nu - Ni} \times 10 \tag{5}$$

With $I[1]$ is the first date of the intersection. $U[1]$ indicates the first date of the union. $U[n]$ indicates the last date of the union. $I[n]$ indicates the last date of the intersection. Ni indicates the number of days of the intersection of a data. Nu indicates the number of days of the union of a data.

To better understand the usefulness of the NES and P parameters, we will present the following example:

Tables 1 and 2 represent the dates (days) of consultation of a file fi for two years.

Table 1. The dates of consultation of file fi during year 1.

Year 1								
D1	D2	D3	D5	D7	D8	D9	D10	D60

Table 2. The dates of consultation of file fi during year 2.

Year 2							
D3	D4	D5	D7	D9	D10	D11	D60

The following two tables represent the union and the intersection of the consultation dates for file fi during two years (Tables 3 and 4).

In order to be able to classify the data «file fi», we will first calculate the parameter NES:

NES = D60 − D3 = 57 days.

This parameter indicates to us that the dates of consultation of this file are not near, so this file may not be classified. Calculation of the derogation parameter P, can determine whether this data will be classified or not. For this we will do the following calculation:

Table 3. The intersection between the dates of consultation of file fi

Year 1∩Year 2					
D3	D5	D7	D9	D10	D60

Table 4. The union between the dates of consultation of file fi

Year 1 U Year 2										
D1	D2	D3	D4	D5	D7	D8	D9	D10	D11	D60

The number of union (Nu) is eleven and the number of intersection (Ni) is six.

$$P = \frac{(\mathbf{D}3 - \mathbf{D}1) + (\mathbf{D}60 - \mathbf{D}60)}{11 - 6} * 10 = 4\,\text{days}$$

The value 4 represents the number of days (dates) tolerated which are nearer to intersection interval for the data to be classified. Or D60 − D10 = 50 days. D60 is not near the interval of the intersection, thus we can note that this data does not belong to a category of data consulted in a certain time, which prevents it from being classified.

3.2.3 Flow Chart of the Classification Method

In what follows, we will use the following notations:

- U: the union between the years of a data i
- I: the intersection between the years of a data i
- Ni: number of intersection of a data i
- Nu: number of union of a data i
- NES: number of days between end and start date of the intersection
- P: derogation parameter for data i (Fig. 2).

3.3 Viral Data Checking Module

Viral data is data whose access frequency is always high, this will cause a significant increase in response time. To avoid this, we used the viral data verification module in order to create new replicas if necessary.

This module is based on an algorithm that will be executed on a regular basis. its principle is summed up by the checking of response times in order not to violate the SLA contract. In this case, the module will increase the availability which will have a direct effect on the response time.

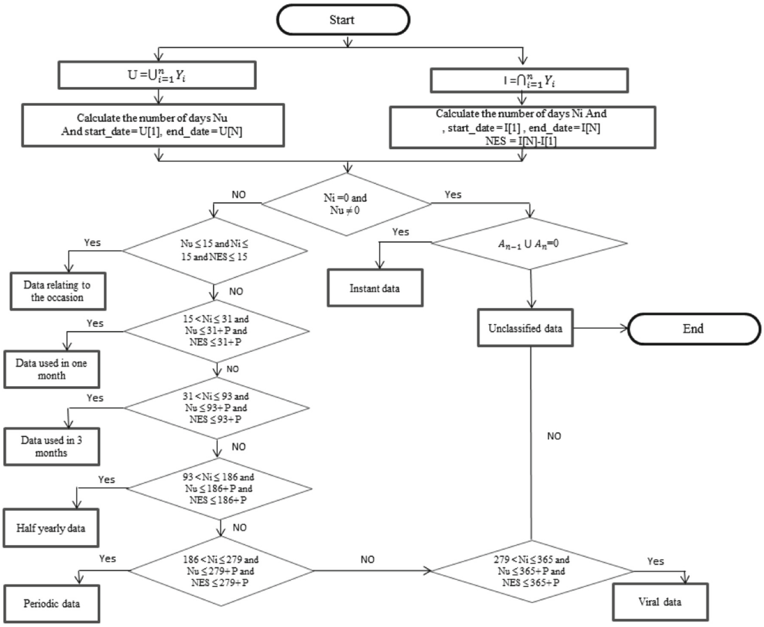

Fig. 2. File classification flow chart

Algorithm 1: Response time check for viral data

Input: List of file classes, Current response time list for each file RT, SLA parameters
Output: Replica creation.
BEGIN
while (Data is viral) **do**
  for each node i contains the data **do**
    if (Rt[i] > Tsla) **then**
    // Creation of replicas to meet response time
    end if
  end for
end while
END

3.4 Liberation

This module consists in having the created replicas deleted responding to the response time of a data which does not belong to the unclassified and viral classes. This process will result in an optimization of the storage space as well as a gain on the customer and provider side. The principle consists in calculating a liberation factor for each node having the replica in order to choose which one should delete the replicas. This factor is calculated by the following formula:

$$LF_i = AS_j \tag{6}$$

With AS_j is the available storage capacity for a node j and LF_i indicates the liberation factor.

The deletion operation is done as follows:

1. Calculation of the liberation factor for each node storing all the replicas of the Fi file. Then sort them increasingly.
2. Suppression of replicas responding to the response time in the nodes having the smallest liberation factors.

Algorithm 2: Liberation

Input: Matrix result of the classification algorithm, Lists of nodes concerning the data.
Output: Suppression of replicas**.**
BEGIN
While (The data is not viral and unclassified) **do**
 if (the end date of the union = current date +1) **then**
  for each node j **do**
   // Calculate available storage space
   $AS_j = \sum_{i=1}^{n} F_j$
   // Calculate liberation factor
   $LF_i = AS_j$
  end for
  // Sort ascending (LF) ;
 // Delete the files in the nodes with the smallest LF according to the number of replicas already created by S.
 end if
end while
END

---

4 Implementation and Experimentation

4.1 Experimental Study

In order to validate and evaluate the behavior of the proposed replication solution, we carried out a series of experiments using the CloudSim simulator [12] which is a good environment for testing dynamic replication strategies.

In these experiments, we are interested in the following metrics: response time, number of replicas created, number of replicas deleted, and free storage space.

In our experiments, the comparison of the results is mainly carried out between four replication strategies implemented which are: the proposed strategy, CDRM [13], ADRS [16], without using a data replication.

4.2 Simulation Parameters

To carry out the various experiments, we used a certain number of simulation parameters whose values are defined in Table 5. These parameters are the same for all of the experiments presented.

Table 5. Simulation parameters

Parameter	Value
Number of data center	6
Number of storage nodes	120
Storage capacity	[30 .. 50] GB
Bandwidth of storage nodes	[90 .. 130] MB/s
MIPS of processing	1000 MIPS
Number of files	100
File size	[1 .. 3] GB
Total number of Cloudlets	[300, 600, 900, 1200, 1500, 1800]
Length of Cloudlet	[700 1000] KB

4.3 Experimental Results and Analysis

4.3.1 Response Time to Requests

In this experiment, we measure the average response time obtained with our replication approach as well as the strategies implemented. Figure 3 shows the impact of replication on request response time. These results show that with the increase in the number of requests, the response time of the four strategies also increases.

The response time of the strategy without replication is high because each file is stored in a single node, so all requests that request this file are sent to this node which causes a high load. As a result, requests are waiting in queues and this causes increased response time. On the other hand, CDRM, ADRS and the proposed strategies increase the availability of replicas which improves the response time.

From Fig. 3, we notice a significant decrease in response time by applying the proposed strategy. A gain of 66.2% is observed compared to the approach without replication, 52.4% compared to the CDRM strategy and 33.9% compared to the ADRS strategy.

Our strategy gives a better response time, because it regularly replicates files that do not meet the response time of the SLA contract, this by determining the minimum number of replicas that must be added to the system to ensure the required response time. On the other hand, in the CDRM strategy, the number of replicas created that respond to the response time is determined statically according to a threshold and does not adapt to the behavior of cloud users. Concerning ADRS, the problem of creating replicas is not treated, the creation in this strategy is done in a random way.

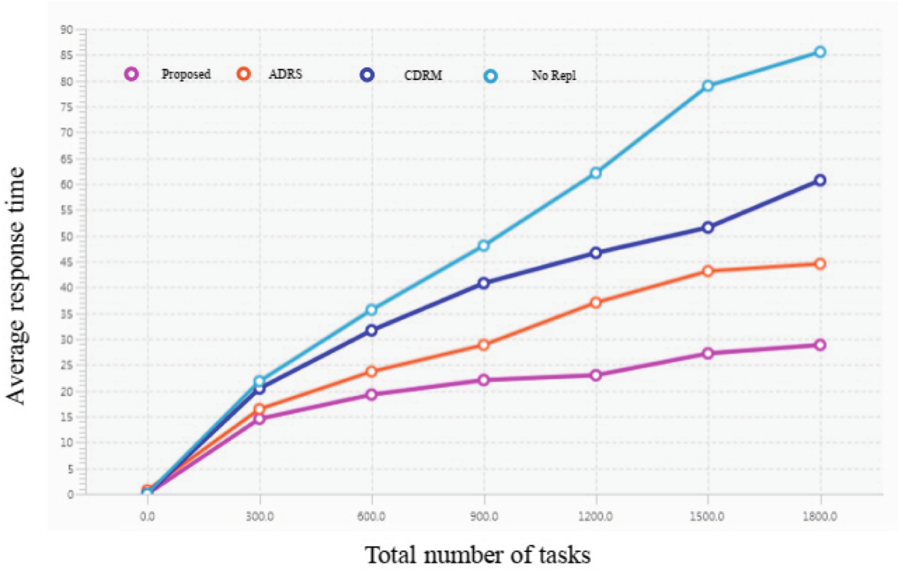

Fig. 3. Average response time based on change in number of requests.

4.3.2 Number of Replicas Created

In this experiment, we are studying the contribution of our strategy regarding the number of replicas created. We compared only three strategies because the strategy without replication does not replicate data dynamically. Figure 4 shows that the ADRS strategy

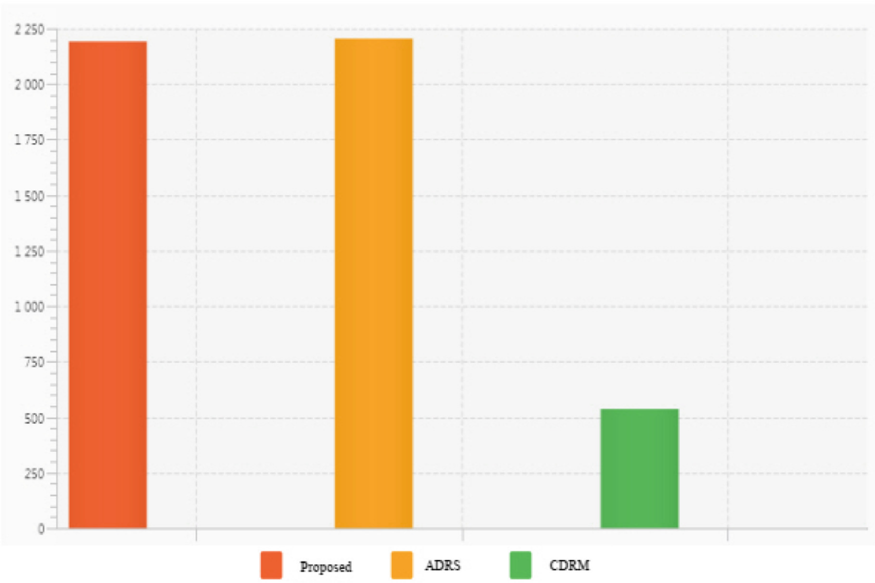

Fig. 4. Representation the number of replicas created.

creates the highest number of replicas. One would think that the proposed approach creates many more replicas because this creation is done according to two criteria namely availability and response time. However, the results show that the number of replicas created does not exceed that of the ADRS method and better meets the other QoS criteria. The number of replicas created by the CDRM method was predictable because although it is dynamic, this method does not adapt to user behavior.

4.3.3 Number of Replicas Deleted

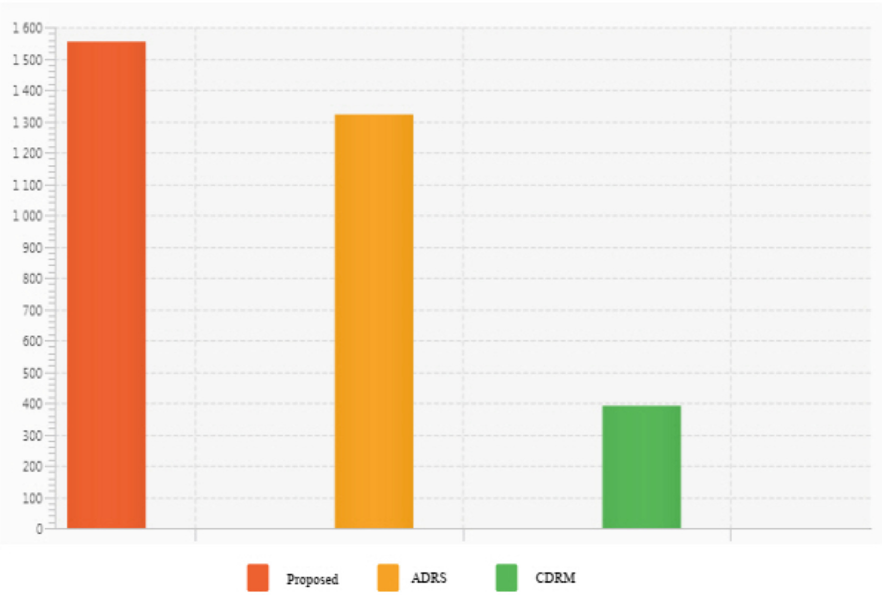

Fig. 5. Representation the number of replicas deleted

Figure 5 represents the number of replicas deleted for the three algorithms during the whole simulation. In the proposed strategy, we note that the number of replicas deleted is the highest, because it uses a liberation module which deletes the replicas when their consultation intervals have expired. Concerning ADRS, it only uses a replacement module that is triggered when the storage space is insufficient. For CDRM, it is completely logical that the number of deleted replicas should be the smallest because it depends on the number of replicas created.

After analyzing the results, we found that the frequency of creation/deletion in the CDRM method is the highest. Other methods handle data better (See Table 6).

Table 6. Number of replicas created and deleted.

Strategy	Number of replicas created	Number of replicas deleted
Proposed strategy	2193	1553
ADRS	2203	1321
CDRM	539	392

4.3.4 Storage Usage

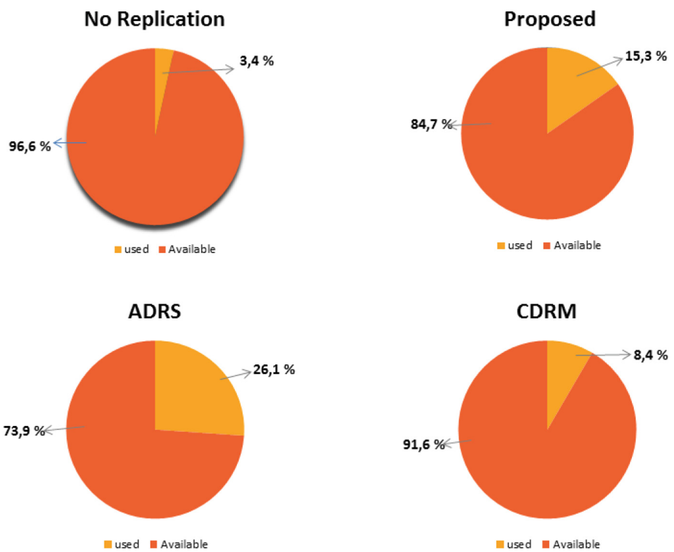

Fig. 6. Storage resource usage

Figure 6 shows the rate of space used in relation to total space. Note that the percentage of storage space used is directly related to the number and size of replicas created and deleted during the simulation, because it is not a question of replicating any request to meet the quality of service. In the proposed strategy, the variation of the data classes influences the number of replicas deleted, therefore directly on the storage space used.

The storage space used in the proposed approach is less than the ADRS approach with a difference of 10.8% and higher than the CDRM approach with only a difference of 6.9%. We can conclude that the proposed approach appropriately uses the management of replica storage in the nodes, while improving response time compared to other strategies.

5 Conclusion

Cloud Computing offers customers the most adaptable services to their needs, it is based on an economic model which allows to users to be invoiced only on the share of resource use. However, the Cloud faces several challenges that notably affect the quality of service or the reliability of data. Replica management is one of the most important issues in the Cloud, which greatly affects the performance of the cloud system. In this article, we have proposed a new replication strategy that adapts to unpredictable user behavior, by integrating a data classification module that will replicate data in an adaptive manner. It consists of replicating the data dynamically in the nodes according to an availability required by the user, in order to improve availability and reduce response time. The proposed replication solution has four main phases which are: creation of replicas, data classification, release of storage space and verification of compliance with the SLA contract. To experiment and validate our work, we implemented and simulated our strategy under Cloudsim, with the goal of comparing the results with two existing strategies (CDRM and ADRS). The results obtained from our experiences have shown that the proposed strategy has satisfactorily improved the criteria studied.

This work has opened up some interesting perspectives which we summarize in the following points: i) integrate a replica consistency management service, ii) validate our proposal in a real cloud, iii) use the classification module for the prediction.

References

1. Mell, P., Grance, T.: The NIST definition of cloud computing recommendations of the national institute of standards and technology. J. NIST Spec. Publ. **145**, 7 (2011)
2. Senyo, P.K., Addae, E., Boateng, R.: Cloud computing research: a review of research themes, frameworks, methods and future research directions. Int. J. Inf. Manag. **38**(1), 128–139 (2018)
3. Li, K.: Power and performance management for parallel computations in clouds and data centers. J. Comput. Syst. Sci. **82**(2), 174–190 (2016)
4. Bokhari, M.U., Makki, Q., Tamandani, Y.K.: A survey on cloud computing. In: Aggarwal, V.B., Bhatnagar, V., Mishra, D.K. (eds.) Big Data Analytics. AISC, vol. 654, pp. 149–164. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-6620-7_16
5. Kofahi, N.A., Al-Rabad, A.: Identifying the top threats in cloud computing and its suggested solutions: a survey. Adv. Netw. **6**(1), 1–13 (2018)
6. Malik, S.U.R., et al.: Performance analysis of data intensive cloud systems based on data management and replication: a survey. J. Distrib. Parallel Databases **34**(2), 179–215 (2016). https://doi.org/10.1007/s10619-015-7173-2
7. Tos, U., Mokadem, R., Hameurlain, A., Ayav, T., Bora, S.: Ensuring performance and provider profit through data replication in cloud systems. Cluster Comput. **21**(3), 1479–1492 (2017). https://doi.org/10.1007/s10586-017-1507-y
8. Ghemawat, S., Gobioff, H., Leung, S.-T.: The Google file system. In: Proceedings of SOSP, pp. 29–43 (2003)
9. Borthakur, D.: The Hadoop Distributed File System: Architecture and Design. Hadoop Project Website (2007)
10. Mansouri, Y., Buyya, R.: Dynamic replication and migration of data objects with hot-spot and cold-spot statuses across storage data centers. J. Parallel Distrib. Comput. **126**, 121–133 (2019)
11. Mokadem, R., Hameurlain, A.: A data replication strategy with tenant performance and provider economic profit guarantees in cloud data centers. J. Syst. Softw. **159**, 110447 (2020)
12. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw.: Pract. Exp. **41**(1), 23–50 (2011)