

REPLICA PLACEMENT STRATEGY BASED ON ANALYTIC HIERARCHY PROCESS IN HETEROGENEOUS CLOUD DATA STORAGE

Mohammed Radi

Computer Science Department, Alaqsa University, Palestine

moh_radi@alaqsa.edu.ps

Abstract. Cloud data storage platforms have attracted increasing attention in data management. Data replication is a well-known technique that reduces access latency, thus improving cloud data storage availability and performance. Nonetheless, replica placement algorithms are among the significant issues in data replication, which affect data availability and access performance considerably. Replica placement algorithms determine where data replicas can be located in the data storage system. Replica placement is a classical Multi-Attribute Decision Making (MADM) problem. The analytic hierarchy process (AHP) is a structured technique for organizing and analyzing complex decisions based on mathematics and psychology. In this study, we present a replica placement algorithm based on AHP(RPAHP) in heterogeneous cloud data storage. Simulation results show that the RPAHP strategy performs better than other replica creation strategies in terms of average response time.

Key Words: cloud data storage, data replication, replica placement algorithms, analytic hierarchy process, CloudAnalyst.

1. Introduction

Cloud computing has become increasingly popular in data management[1, 2]. Within clouds, data replication is widely used to store copies of data sets (files or subsets of a file) across different data centers (DCs). Data replication reduces access latency, increases availability, and improves the performance of applications that are deployed over the cloud[3]. Specifically, this strategy limits access latency by allowing users to access nearby replicas. It enhances availability by providing access to the data even when some of the replicas are unavailable. Finally, it improves cloud system performance by distributing user requests among different DCs. The strategy of assigning replicas to all DCs is unrealistic because of the constraints on storage capacity, cost, and replica consistency.

Nonetheless, two key issues in all data replication algorithms significantly affect data availability and access performance. These issues also influence the number of replicas that should be created and the location for replica storage[4]. Replica number strategies handle the first issue, in which the required number of replicas is determined based on the importance of the data and their popularity. The reliability requirement is high when data are important; thus, the required number of replicas would increase. Increased numbers of user requests are received when data are popular, which in turn generates DC overhead. An increase in the number of replicas can reduce response time and balance the load. Meanwhile, replica creation strategies decide where the replicas should be

stored based on the storage and processing capacity of the DC and on the frequency of user requests.

Replication can be classified into two categories, namely, static and dynamic. In static data replication, the number of replicas to be created and the DCs where replicas should be placed are decided statically during cloud setup, thus, the number and the locations of these replicas do not change once they are determined. The static replication strategies are simple to implement but not frequently used because this strategy does not support data duplication during running time. Meanwhile, dynamic replication algorithms support automatic replica creation and data placement, in which this technique adapts to changes based on user requests, storage capacity, bandwidth, and system performance. Dynamic replication schemes are capable of making intelligent decisions to place data in the cloud storage system based on the current system criteria, such as CPU, Memory, and Bandwidth.

Replica placement is a classical Multi-Attribute Decision Making (MADM) problem[4]. The analytic hierarchy process (AHP)[5] is a structured technique for organizing and analyzing complex decisions, based on mathematics and psychology. In this study we present a replica placement strategy based on AHP (RPAHP) in heterogeneous cloud data storage in order to improve system performance. RPAHP consider 6 attributes(criteria) which characterize the DC and affect its performance. The attributes are, CPU(C), Memory (M), Bandwidth (B), CPU Utilization (CU), Memory Utilization (MU), and Bandwidth Utilization

(BU). The simulation results show that RPAHP strategy performs better than other replica creation strategies in terms of average response time.

The rest of the paper is organized as follows: Section 2 discusses related works; Section 3 describes the (RPAHP); Section 4 presents the experiment results; and Section 5 concludes the paper.

2. Related Work

Replica creation improved the cloud storage system reliability, and system performance. Replica creation contains the static replica creation strategy and the dynamic replica creation strategy.

2.1 Static Replica Creation strategy

The static replication strategy is adopted in HDFS[6], in which the number of replicas can be assigned by the client and the default is 3. Two replicas are placed in the different data nodes of the same rack near the client and the third is placed in a random data node of the other rack. If the number of replicas is more than 3, the other replicas are placed in random data node of the cluster. The static replication keeps stationary replica location and replica number, which cannot adapt to the complex environment. This scheme not only guarantees for the reliability and availability of data, but also reduces the read operation of the network aggregate bandwidth. However, the system does not take into account the node space utilization when creating a replica. To improve the efficiency of HDFS[6], a novel block placement strategy has been developed in [7] which focuses on load balancing. The optimal nodes are chosen by the novel strategy based on the remaining utilization of nodes' space, and the load balancing is achieved.

2.2 Dynamic Replica Creation strategy

This strategy is more flexible and replica is no longer static. However, this replica will change with the improvement in the operation of the system. Paper [8] discussed six dynamic replication strategies based on utility and risk, namely MinimizeExpectedUtil; MaximizeTimeDiffUtil; MinimizeMaxRisk; MinimizeMaxAvgRisk; BestClient; and Cascading. The replica placement algorithm using expected utility is demonstrated by considering each node and calculating the expected utility. In MinimizeExpectedUtil, the replica is placed at the node with the lowest expected utility. While in MaximizeTimeDiffUtil, the site with the maximum time difference utility is selected and the replica is placed on the site generating the maximum time difference. Furthermore, in MinimizeMaxRisk algorithm, the replica is placed at the site with the

maximum risk index. In MinimizeMaxAvgRisk algorithm, the replica is placed at the node with the highest average index. In addition, the BestClient algorithm places the replica at the site that has maximum requests for the file. Finally, the Cascading algorithm places the replica on the path of the BestClient.

Literature [9] introduces six kinds of replica creation strategies, which are: no replica strategy, caching strategy, the BestClient strategy, waterfall strategy, waterfall combining with caching strategy, and rapid spread strategy. The caching strategy and BestClient strategy create a replica only based on user visits. This strategy cannot resolve the problem of hot spot data in the next period and are not good in finding a file that needs to create a replica. In the meantime, the waterfall strategy cannot adapt to the complicated and changeable cloud storage system. Moreover, rapid spread strategy can be stored in a node in the system which is only for small size, small visits, and individual files. This is obviously not suitable to solve massive data that accesses a large amount of cloud storage system.

Recently, the replica placement problem has been investigated and replica placement techniques have been introduced. In [10], Vickrey-Clarke-Groves (VCG) mechanism was introduced into replica placement. Mapping from replica placement model to VCG mechanism was established and suitable payment function was designed. A strategy of dynamic replica creation based on acceleration is presented for cloud storage by eliciting information from the concept of acceleration in [11]. For reducing access latency, the strategy predicts the hot files in the way of acceleration based on accessing locality principle. Finally, the performances of the strategy of acceleration replicas creation and the strategy of BestClient replicas creation are simulated and analyzed.

A two-layer geo-cloud based dynamic replica creation strategy called TGstag is proposed[12]. TGstag addresses the issue with twofold strategy: policy constraint heuristic inter-datacenter replication and load aware adaptive intra-datacenter replication. TGstag aims to minimize both cross-datacenter bandwidth consumption and average access time with constraints of policy and commodity node capacity.

A study in [4] proposed a dynamic replica management strategy based on technique for order performance by similarity to ideal solution (TOPSIS) in the cloud storage system. The replica management strategy includes the replica placement algorithm and the

replica number strategy. The replica placement algorithm sorts the nodes by TOPSIS. According to the node performance and load, the replicas are placed in the node with high performance and low load. The replica number strategy design the reliability model and the availability model to compute the number of replicas.

A.Rajalakshmi et al (2014) propose an approach for dynamic data replication in cloud. The proposed work concentrates on designing an algorithm for suitable optimal replica selection and placement to increase availability of data in the cloud[13]. The method consists of two main phases: file application and replication operation. The first phase contains the replica location and creation by using catalog and index, whereas the second phase is used to perform optimization among replicated copies.

Taking the response time of single request into consideration, XiaohuBai et al.[14] propose a dynamic replica management strategy based on response time called RTRM. RTRM strategy consists of replica creation, replica selection, and replica placement mechanisms. The replica selection algorithm chooses the node with highest LPC to replicate the file, where LPC metrics consists of three components: CPU process capability, network transmission capability, and I/O capability of disks.

In [15], the problem of data replica placement in cloud storage system was discussed and a replica placement algorithm was designed. According to the basic principle of queuing theory, the replica placement algorithm based on minimal blocking probability(BPRA) is proposed from the view of resource competition as a dynamic replica placement algorithm[16].The BPRA algorithm calculates the minimal number of replica according to the file available requirement and dynamically adjusts replica factor according to the file access frequency.

3. Replica Placement Algorithms Based On AHP

3.1 Multi-Attribute Decision Making

Multi-Attribute Decision Making (MADM)problem[5] involves the selection of the 'best' alternative from a pool of pre-selected alternatives described in terms of their attributes. The main feature of MADM is that there are usually a limited number of predetermined alternatives, which are associated with a level of the achievement of the attributes. The final decision is made based on the attributes.

Mathematically, a typical MADM problem can be modeled as follows:

$$(MADM) \begin{cases} select : A_1, A_2 \dots A_m \\ s. t. : M_1, M_2 \dots M_n \end{cases}$$

where $A_1, A_2 \dots A_m$ denotes m alternatives, $C=(M_1, M_2 \dots M_n)$ represents n attributes (often called criteria) for characterizing a decision situation. The select here is normally based on maximizing a multi attribute value (or utility) function elicited from the stakeholders.

3.2 AHP

AHP[5] is essentially applied to formulize the intuitive understanding of a complex problem using a hierarchical structure. The core of the AHP is to enable decision makers to structure an MADM problem in the form of an attribute hierarchy. A hierarchy has at least three levels:

- **top level:** the focus or overall goal of the problem.
- **middle level:** multiple attributes (criteria) that define alternatives.
- **bottom level:** competing alternatives.

The AHP method has the following general steps:

- Step 1:** Construct a hierarchy for an MADM problem.
- Step2:** Make the relative importance among the attributes (criteria) by pairwise comparisons in a matrix.
- Step 3:** Make pairwise comparisons of alternatives with respect to attributes (criteria) in a matrix.
- Step 4:** Retrieve the weights of each element in the matrix generated in Steps 2 and 3.
- Step 5:** Compute the contribution of each alternative to the overall goal by aggregating the resulting weights vertically.

3.3 Replica placement as MADM problem

Replica placement is one of the most important issues for cloud storage system. Replica placement strategy determines where to locate the replica dynamically. In addition, optimal placement of data replicas requires knowledge of the current state of the system to select the best DC to locate the replica among different DCs in a system. However, the availability of cloud storage resources as well as the interconnection network performance change constantly. Similarly, data access requests change with each application and each user.

However, selecting the best DC from n DCs is a well-known (MADM) problem, because in cloud data storage, there are m alternative DC where each of them can locate the replica and each one also has m criteria characterizing the DC.

In the cloud storage system, we consider 6 attributes(criteria) which characterize the DC and affect the DC performance. The attributes are: CPU(C), Memory (M), Bandwidth (B), CPU Utilization (CU), Memory Utilization (MU), and Bandwidth Utilization (BU). Furthermore, the replica placement technique selects the best DC among a set n of DCs (DC₁, DC₂...DC_n). The *select* here is normally based on maximizing a multi attribute value.

3.4 Replica placement strategy based on AHP

The following steps illustrate the process of selecting the best DC using AHP method.

Step1: A hierarchy for the MADM replica placement problem is created as in Figure 1.

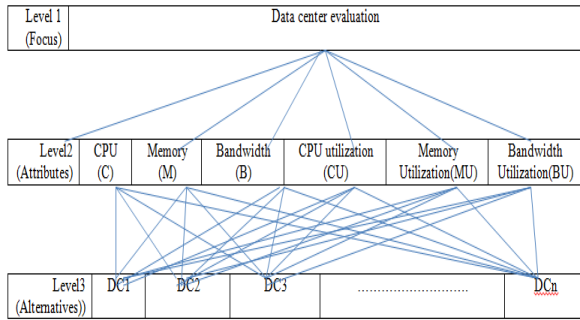


Figure 1: Hierarchy for the MADM replica placement problem

Step 2: Make the relative importance among the attributes (criteria) by pairwise comparisons (PC) in a two dimensional matrix.

$$= \begin{bmatrix} PC_{11} & PC_{12} & PC_{13} & PC_{14} & PC_{15} & PC_{16} \\ PC_{21} & PC_{22} & PC_{23} & PC_{24} & PC_{25} & PC_{26} \\ PC_{31} & PC_{32} & PC_{33} & PC_{34} & PC_{35} & PC_{36} \\ PC_{41} & PC_{42} & PC_{43} & PC_{44} & PC_{45} & PC_{46} \\ PC_{51} & PC_{52} & PC_{53} & PC_{54} & PC_{55} & PC_{56} \\ PC_{61} & PC_{62} & PC_{63} & PC_{64} & PC_{65} & PC_{66} \end{bmatrix} \quad (1)$$

The two dimensional matrix for the PC will be as shown in Equation 2.

$$= \begin{bmatrix} 1 & C/M & C/B & C/CU & C/MU & C/BU \\ M/C & 1 & M/B & M/CU & M/MU & M/BU \\ B/M & B/B & 1 & B/CU & B/MU & B/BU \\ CU/C & CU/M & CU/B & 1 & CU/MU & CU/BU \\ MU/C & MU/M & MU/B & MU/CU & 1 & MU/BU \\ BU/C & BU/M & BU/B & BU/CU & BU/MU & 1 \end{bmatrix} \quad (2)$$

Step 3: Pairwise comparisons of alternatives are performed with respect to attributes (criteria) in a matrix. Six matrixes are made for pairwise comparisons of the n DCs with respect to six attributes, as shown in Equations 3, 4, 5, 6, 7, and 8

$$\text{For C} \begin{bmatrix} C_{11} & C_{12} & C_{13} & \dots & C_{1n} \\ C_{21} & C_{22} & C_{23} & \dots & C_{2n} \\ C_{31} & C_{32} & C_{33} & \dots & C_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ C_{n1} & C_{n2} & C_{n3} & \dots & C_{nn} \end{bmatrix} \quad (3)$$

$$\text{For M} \begin{bmatrix} M_{11} & M_{12} & M_{13} & \dots & M_{1n} \\ M_{21} & M_{22} & M_{23} & \dots & M_{2n} \\ M_{31} & M_{32} & M_{33} & \dots & M_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ M_{n1} & M_{n2} & M_{n3} & \dots & M_{nn} \end{bmatrix} \quad (4)$$

$$\text{For B} \begin{bmatrix} B_{11} & B_{12} & B_{13} & \dots & B_{1n} \\ B_{21} & B_{22} & B_{23} & \dots & B_{2n} \\ B_{31} & B_{32} & B_{33} & \dots & B_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ B_{n1} & B_{n2} & B_{n3} & \dots & B_{nn} \end{bmatrix} \quad (5)$$

$$\text{For CU} \begin{bmatrix} CU_{11} & CU_{12} & CU_{13} & \dots & CU_{1n} \\ CU_{21} & CU_{22} & CU_{23} & \dots & CU_{2n} \\ CU_{31} & CU_{32} & CU_{33} & \dots & CU_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ CU_{n1} & CU_{n2} & CU_{n3} & \dots & CU_{nn} \end{bmatrix} \quad (6)$$

$$\text{For MU} \begin{bmatrix} MU_{11} & MU_{12} & MU_{13} & \dots & MU_{1n} \\ MU_{21} & MU_{22} & MU_{23} & \dots & MU_{2n} \\ MU_{31} & MU_{32} & MU_{33} & \dots & MU_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ MU_{n1} & MU_{n2} & MU_{n3} & \dots & MU_{nn} \end{bmatrix} \quad (7)$$

$$\text{For BU} \begin{bmatrix} BU_{11} & BU_{12} & BU_{13} & \dots & BU_{1n} \\ BU_{21} & BU_{22} & BU_{23} & \dots & BU_{2n} \\ BU_{31} & BU_{32} & BU_{33} & \dots & BU_{3n} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ BU_{n1} & BU_{n2} & BU_{n3} & \dots & BU_{nn} \end{bmatrix} \quad (8)$$

Step 4: The weights of each element in the matrix generated in Steps 2 and 3 are retrieved using the two steps of the geometric mean of a row, as follows:

(a) Multiply the n elements in each row, take the n th root, and prepare a new column for the resulting numbers.

From Equation (1) in step 2, we obtain Equation 9:

$$\begin{bmatrix} WPC_1 \\ WPC_2 \\ WPC_3 \\ WPC_4 \\ WPC_5 \\ WPC_6 \end{bmatrix} = \begin{bmatrix} \sqrt[n]{PC_{11} \times PC_{12} \times PC_{13} \times PC_{14} \times PC_{15} \times PC_{16}} \\ \sqrt[n]{PC_{21} \times PC_{22} \times PC_{23} \times PC_{24} \times PC_{25} \times PC_{26}} \\ \sqrt[n]{PC_{31} \times PC_{32} \times PC_{33} \times PC_{34} \times PC_{35} \times PC_{36}} \\ \sqrt[n]{PC_{41} \times PC_{42} \times PC_{43} \times PC_{44} \times PC_{45} \times PC_{46}} \\ \sqrt[n]{PC_{51} \times PC_{52} \times PC_{53} \times PC_{54} \times PC_{55} \times PC_{56}} \\ \sqrt[n]{PC_{61} \times PC_{62} \times PC_{63} \times PC_{64} \times PC_{65} \times PC_{66}} \end{bmatrix} \quad (9)$$

Additionally, from Equations 3,4,5,6,7, and 8, we obtain Equation (10)

$$\begin{bmatrix} WPC_{C1} & WPC_{M1} & WPC_{B1} & WPC_{CU1} & WPC_{MU1} & WPC_{BU1} \\ WPC_{C2} & WPC_{M2} & WPC_{B2} & WPC_{CU2} & WPC_{MU2} & WPC_{BU2} \\ WPC_{C3} & WPC_{M3} & WPC_{B3} & WPC_{CU3} & WPC_{MU3} & WPC_{BU3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ WPC_{Cn} & WPC_{Mn} & WPC_{Bn} & WPC_{CUn} & WPC_{MUn} & WPC_{BUUn} \end{bmatrix} \quad (10)$$

(b) The new column(i.e., divide each number by the sum of all numbers)is normalized to obtain the normalized matrix, as stated in Equation 11 and 12.

$$\begin{bmatrix} RWPC_1 \\ RWPC_2 \\ RWPC_3 \\ RWPC_4 \\ RWPC_5 \\ RWPC_6 \end{bmatrix} = \begin{bmatrix} WPC_1 / \sum_{i=1}^n WPC_i \\ WPC_2 / \sum_{i=1}^n WPC_i \\ WPC_3 / \sum_{i=1}^n WPC_i \\ WPC_4 / \sum_{i=1}^n WPC_i \\ WPC_5 / \sum_{i=1}^n WPC_i \\ WPC_6 / \sum_{i=1}^n WPC_i \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} RWPC_1 & RWPC_2 & RWPC_3 & RWPC_4 & RWPC_5 & RWPC_6 \\ RWPC_1 & RWPC_2 & RWPC_3 & RWPC_4 & RWPC_5 & RWPC_6 \\ RWPC_1 & RWPC_2 & RWPC_3 & RWPC_4 & RWPC_5 & RWPC_6 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ RWPC_n & RWPC_n & RWPC_n & RWPC_n & RWPC_n & RWPC_n \end{bmatrix} \quad (12)$$

Step 5: The contribution of each alternative to the overall goal is calculated by aggregating the resulting weights vertically for each DC as stated in Equation 13:

$$\begin{bmatrix} DC_1 \\ DC_2 \\ DC_3 \\ \vdots \\ DC_n \end{bmatrix} = \begin{bmatrix} RWPC_1 \times RWPC_1 + RWPC_2 \times RWPC_1 + \dots \\ RWPC_1 \times RWPC_2 + RWPC_2 \times RWPC_2 + \dots \\ RWPC_1 \times RWPC_3 + RWPC_2 \times RWPC_3 + \dots \\ \vdots \\ RWPC_1 \times RWPC_n + RWPC_n \times RWPC_n + \dots \end{bmatrix} \quad (13)$$

The result shows each DC with its weight, the DC with highest weights will be selected to locate the replica, and so on.

4. Simulation and results

In this section, we compare RPAHP replica placement mechanism with the other four replica placement strategies using CloudAnalyst simulator[17]. We compare RPAHP with MinimizeExpectedUtil, BestClient, RTRM, and random. We use CloudAnalyst simulator to evaluate the proposed algorithm[17]. In the experiments, the replica Degree is defined to represent the number of replicas in the system, and use average response time to represent the average response time of all nodes in the system. We compare the value of average response time of the five strategies with different replica Degree (1, 2,...,5).

We run the simulators in different scenarios, varying the number of replica Degree. In all experiments, the user-based configuration, the DC configuration and the other simulation parameters are indicated in Tables 1, 2, and 3, respectively.

Table 1:User-basedconfiguration

UB Name	Region	Requests per users per hr	Peak hrs (GMT)	Avg. users	Avg. off-peak users
UB1	0	60	3:00—9:00	1000	100
UB2	1	60	3:00—9:00	1000	100
UB3	2	60	3:00—9:00	1000	100
UB4	3	60	3:00—9:00	1000	100
UB5	4	60	3:00—9:00	1000	100
UB6	5	60	3:00—9:00	1000	100

Table 2: DC configuration

Name	Region	#V Ms	Cost/V M (\$/Hr)	Data transfer cost (\$/Gb)	Number of processors	Speed (MIPS)
DC1	0	5	0.1	0.1	4	10000
DC2	1	5	0.1	0.1	4	11000
DC3	2	5	0.1	0.1	4	12000
DC4	3	5	0.1	0.1	4	13000
DC5	4	5	0.1	0.1	4	14000
DC6	5	5	0.1	0.1	4	15000

Table 3: Other simulation parameters

Parameter	
User grouping factor in user base	1000
Request grouping factor	10
Executable instruction length/request	500
Load balancing policy	Round-robin
Replica selection algorithm	Closest data center
Simulation duration	24 Hr
VM image size	10000
VM memory	512 Mb
VM bandwidth	1000
DCarchitecture	X86
DC processor/machine	4
DC OS	Linux

The Internet characteristics, which include the Internet latency and bandwidth, are configured as in Table 4 and Table 5, respectively.

Table 4: the Internet latency

Region/ Region	0	1	2	3	4	6
0	25.0	100.0	150.0	250.0	250.0	100.0
1	100.0	25.0	250.0	500.0	350.0	200.0
2	150.0	250.0	25.0	150.0	150.0	200.0
3	250.0	500.0	150.0	25.0	500.0	500.0
4	250.0	350.0	150.0	500.0	25.0	500.0
5	100.0	200.0	200.0	500.0	500.0	25.0

Table 5: Internet bandwidth

Region/ Region	0	1	2	3	4	6
0	2000.0	1000.0	1000.0	1000.0	1000.0	1000.0
1	1000.0	800.0	1000.0	1000.0	1000.0	1000.0
2	1000.0	1000.0	2500.0	1000.0	1000.0	1000.0
3	1000.0	1000.0	1000.0	1500.0	1000.0	1000.0
4	1000.0	1000.0	1000.0	1000.0	500.0	1000.0
5	1000.0	1000.0	1000.0	1000.0	1000.0	2000.0

When the replica degree is 1, the replica can be located at one of the 6 regions. The experiment results shown in Figure 2 indicate that the system provides the best average response time using the RPAHP algorithm, where the replica is located at DC1. By contrast, the worst average response time is generated when the replica is located at DC6 which is obtained by running the random algorithm.

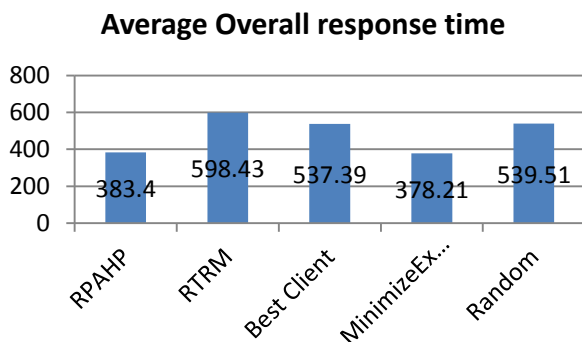


Figure 2: Average overall response time with replication degree:1

When the replica Degree is 2, the replica can be located at two of the 6 regions. The experiment results shown in Figure 3 indicate that the system provides the best average response time when the replica is located at DC2 and DC4 using the RPAHP algorithm. However, the worst average response time is generated when running RTRM algorithm which locate the replica at DC5 and DC6.

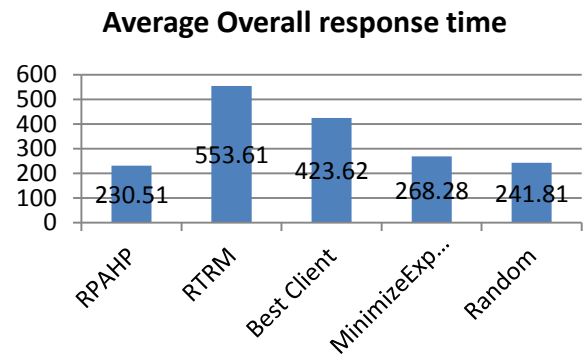


Figure 3: Average overall response time with replication degree: 2

When the replica degree is 3, the replica can be located at three of the 6 regions. The experiment results shown in Figure 4 indicate that the system provides the best average response time when the replica is located at DC2, DC3, and DC4 using the RPAHP algorithm. And the worst average response time is generated when running BestClient algorithm which located the replica at DC3, DC4, and DC5.

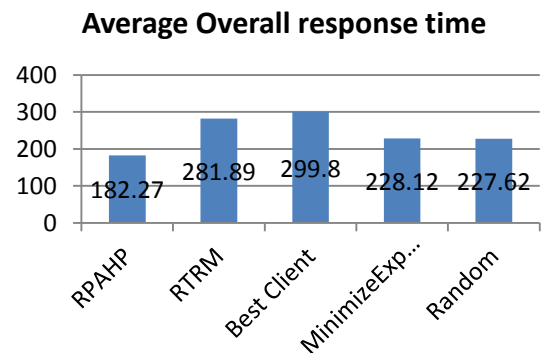


Figure 4: Average overall response time with replication degree: 3

When the replica degree is 4, the replica can be located at four of the 6 regions. The experiment results shown in Figure 5 indicate that the system provides the best average response time when the replica is located at DC1, DC2, DC3, and DC4 using the RPAHP algorithm. On the other hand, the worst average

response time is generated when running RTRM algorithm which located the replica at DC3, DC4, D5, and DC6.

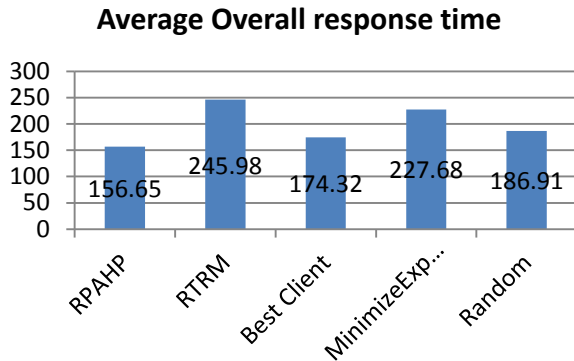


Figure 5: Average overall response time with replication degree: 4

When the replica degree is 5, the replica can be located at four of the 6 regions. The experiment results shown in Figure 6 indicate that the system provides the best average response time when the replica is located at DC1, DC2, DC3, DC4, and DC5 using the RPAHP algorithm. However, the worst average response time is generated when running MinimizeExpectedUtil algorithm which located the replica at DC1, DC2, DC3, D5, and DC6.

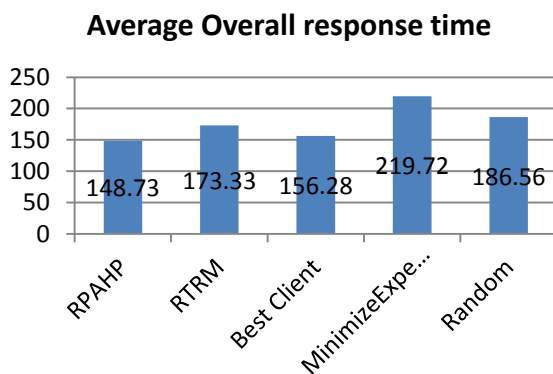


Figure 6: Average overall response time with replication degree: 5

From the above results, we can conclude that our replica placement strategy RPAHP has the smallest average response time compared to the four algorithms even when we have different replication degree. Ultimately, we highly recommend the use of this strategy especially for heterogeneous cloud data storage.

5. Conclusion

In this paper, we presented RPAHP in heterogeneous cloud data storage. RPAHP considered six attributes(criteria), namely, CPU, Memory, Bandwidth, CPU Utilization, Memory Utilization, and Bandwidth Utilization, which characterized the DC and affected the DC performance. We used CloudAnalyst simulator to evaluate the proposed algorithm. The simulation results showed that RPAHP strategy performed better than other replica creation strategies in terms of average response time.

6. References

1. Elzeiny, A., A.A. Elfetouh, and A. Riad, *Cloud Storage: A Survey*. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS) 2013. 2(4): p. 342-349.
2. Radi, M., *Improved Aggressive Update Propagation Technique in Cloud Data Storage*. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2014. 3(2): p. 102-106.
3. Wu, J., et al. *Cloud storage as the infrastructure of cloud computing*. in *Intelligent Computing and Cognitive Informatics (ICICCI)*, 2010 International Conference on. 2010. IEEE.
4. Zhiyong, Y., et al., *DRMST: A Novel Dynamic Replica Management Strategy Based on TOPSIS in Cloud Storage*. Journal of Software Engineering, 2014. 8(4).
5. Lu, J., et al., *Multi-Objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Techniques (With CD-ROM)*. Vol. 6. 2007: World Scientific.
6. Shvachko, K., et al. *The hadoop distributed file system*. in *Mass Storage Systems and Technologies (MSST)*, 2010 IEEE 26th Symposium on. 2010. IEEE.
7. Ye, X., et al. *A Novel Blocks Placement Strategy for Hadoop*. in *ACIS-ICIS*. 2012.
8. Rahman, R.M., K. Barker, and R. Alhajj. *Replica placement in data grid: Considering utility and risk*. in *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*. 2005. IEEE.
9. Kavitha, R. and I. Foster. *Design and evaluation of replication strategies for a high performance data grid*. in *Computing and High Energy and Nuclear Physics*. 2001.
10. Hongxia, W., *Application of VCG in Replica Placement Strategy of Cloud Storage*.

- International Journal of Grid and Distributed Computing, 2016. **9**(4): p. 27-40.
11. Mengxing, H., et al. *A Strategy of Dynamic Replica Creation in Cloud Storage*. in *1st International Workshop on Cloud Computing and Information Security*. 2013. Atlantis Press.
12. Ye, Z., S. Li, and J. Zhou, *A Two-layer Geo-cloud based Dynamic Replica Creation Strategy*. Appl. Math, 2014. **8**(1): p. 431-440.
13. Rajalakshmi, A., D. Vijayakumar, and K. Srinivasagan, *An Improved Dynamic Data Replica Selection and Placement in Hybrid Cloud*. Networking and Communication Engineering, 2014. **6**(3): p. 112-117.
14. Bai, X., et al., *RTRM: A Response Time-Based Replica Management Strategy for Cloud Storage System*, in *Grid and Pervasive Computing 2013*, Springer. p. 124-133.
15. Tao, Z. *Data Replica Placement in Cloud Storage System*. in *International Workshop on Cloud Computing and Information Security (CCIS 2013)*. 2013.
16. Xue, M., J. Shen, and X. Guo, *Replica Placement in Cloud Storage based on Minimal Blocking Probability*. 2015.
17. Wickremasinghe, B., R.N. Calheiros, and R. Buyya. *Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications*. in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*. 2010. IEEE.