2019

A Data Dependency and Access Threshold Based Replication Strategy for Multi-Cloud Workflow Applications

Fei Xie
University of Wollongong, fx439@uowmail.edu.au

Jun Yan
University of Wollongong, jyan@uow.edu.au

Jun Shen
University of Wollongong, jshen@uow.edu.au

Publication Details

Xie, F., Yan, J. & Shen, J. (2019). A Data Dependency and Access Threshold Based Replication Strategy for Multi-Cloud Workflow Applications. Lecture Notes in Computer Science, 11434 281-293. Hangzhou, China ICSOC 2018: Service-Oriented Computing - ICSOC 2018 Workshops

A Data Dependency and Access Threshold Based Replication Strategy for Multi-Cloud Workflow Applications

Abstract

Data replication is one of the significant sub-areas of data management in cloud based workflows. Data-intensive workflow applications can gain great benefits from cloud environments and usually need data management strategies to manage large amounts of data. At the same time, multi-cloud environments become more and more popular. We propose a cost-effective and threshold-based data replication strategy with the consideration of both data dependency and data access times for data-intensive workflows in the multi-cloud environment. Finally, the simulation results show that our approach can greatly reduce total cost of data-intensive workflow applications by considering both of data dependency and data access times in multi-cloud environments.

Disciplines

Engineering | Science and Technology Studies

Publication Details

Xie, F., Yan, J. & Shen, J. (2019). A Data Dependency and Access Threshold Based Replication Strategy for Multi-Cloud Workflow Applications. Lecture Notes in Computer Science, 11434 281-293. Hangzhou, China ICSOC 2018: Service-Oriented Computing - ICSOC 2018 Workshops

A Data Dependency and Access Threshold Based Replication Strategy for Multi-Cloud Workflow Applications

Fei Xie[1], Jun Yan[1] and Jun Shen[1]

[1] University of Wollongong, Wollongong NSW 2500, Australia
fx439@uowmail.edu.au
jyan@uow.edu.au
jshen@uow.edu.au

Abstract. Data replication is one of the significant sub-areas of data management in cloud based workflows. Data-intensive workflow applications can gain great benefits from cloud environments and usually need data management strategies to manage large amounts of data. At the same time, multi-cloud environments become more and more popular. We propose a cost-effective and threshold-based data replication strategy with the consideration of both data dependency and data access times for data-intensive workflows in the multi-cloud environment. Finally, the simulation results show that our approach can greatly reduce total cost of data-intensive workflow applications by considering both of data dependency and data access times in multi-cloud environments.

Keywords: Multi-cloud, Data Management, Data Replication, Data Dependency, Data Access Times

1 Introduction

In recent years, the increasing amount of data becomes major challenges for all organizations, such as data congestion problems [5,8,16], lower data management cost effectiveness [4] and lower data management efficiency [17]. The emergence of cloud computing technologies constructs a novel paradigm for developing and deploying distributed applications.

Cloud storage is not only the adoption of physical hardware but also a highly integrated system which includes network devices, data storage devices, servers, official applications, common access interfaces, network access and client-side programs. Multi-cloud uses two or more cloud computing services in order to allow users to share the workload across multiple cloud service providers. Multi-cloud is commonly used by several famous applications, such as OpenStack and Microsoft Azure [20]. It allows heterogeneous cloud environments to satisfy the user requirements, and can help users minimize the data loss risks and downtime in order to achieve better cloud computing power and quality of service. It can also help users avoid single vendor lock-in risks to a large extent [20]. Multi-cloud is always used to support global or cross-regional collaborative work because the cloud services in multi-cloud always rely on hardware in multiple locations. By using the multi-cloud environment, it is more agile and scalable than only using a single cloud to perform the tasks and share the data [14].

A data-intensive workflow such as a scientific workflow may consist of hundreds of complex tasks and huge amount of data. Data management in such a scenario is still a difficult research challenge as moving large amount of data can be cost-ineffective [19]. Data-intensive workflow applications may benefit greatly from multi-cloud because a multi-cloud environment satisfies their cross-regional computation and massive data storage requirements better by leveraging computation and storage capacities of many data centers [15].

The past research works have addressed this challenging problem in two directions by using data placement and replication strategies. Parameters such as data dependency and data access times have been used separately from the data perspective to develop different strategies in order to achieve a better data management performance [2,12]. Without the consideration of data dependency, highly-dependent data may be stored in locations distant from one another. This may increase the data access cost and the response time. At the same time, without the consideration of data access times, frequently-accessed data may be stored in a remote location. It may also have a significant influence on the total cost, the response time, and the access delay.

In this paper, we propose a data dependency and access threshold based data replication strategy with the consideration of both data dependency and data access times for data-intensive workflows in the multi-cloud environment. In our approach, the data dependency and data access times of datasets are balanced to dynamically control the creation of data replicas. The simulation shows that our approach is more cost-effective than approaches that consider the data dependency or data access times only. The remainder of the paper is organized as follows. Section 2 reviews the major related work and presents the motivation of our work. Then Section 3 describes our data replication approach in details. Section 4 discusses the simulation results. Finally, Section 5 concludes this paper.

2 Related Work

Cloud computing is known as an emerging and fast growing area of service delivery in information technology aspects. This novel approach is marked as one of the top five emerging technologies that will have a significant improvement on quality of science as well as the society within the next 20 years [1]. In general, cloud technology aims to shift several IT dimensions to remote facilities such as central data storage rather than local processing on capable distant servers instead of stationary or portable devices, integrated data rather than distributed data, and the replacement of dispersion applications by centralized ones [10].

In this paper, we particularly focus on data management challenges in multi-cloud environments by using data replication strategy. Data replication is the strategy of creating multiple data copies and storing the copies in multiple sites [11,18]. Data replication can help users save cost [7] and response time [13] when tasks are being processed, and improve the data availability [3,9,17] and reliability [6].

Several approaches have been proposed for data replication in cloud environments. In [2], authors propose a Latest Access Largest Weight (LALW) strategy in order to select a popular file and calculate a suitable number of copies and grid sites for data replication in data grids by considering access frequency to exhibit the importance for access history in different time intervals. In [12], authors propose a Fair-Share Replication (FSR) strategy that takes both access load and storage load into account to determine the replicas creation. An average access frequency is used to compare with the access frequency of targeted datasets to find the popular file and rank the file. In [3], authors propose a dynamic, cost-aware data replication strategy by identifying the minimum number of replicas in order to satisfy the desired availability, get the maximum value and keep the total weight less than or equal to the peak budget at the same time.

Based on the findings from past research, either data dependency or data access times can significantly influence the data management solution. The attribute of data dependency considers the relationship between two datasets from the perspective of tasks. The attribute of data access times considers the number of access times of a dataset accessed by tasks. We argue that both data dependency and data access times should be considered jointly in order to improve the data management performance.

3 Approaches

By taking both data dependency and data access times into consideration, our approach aims to create replicas for datasets that are both highly dependent and frequently accessed. This also balances the number of the replicas created and the total cost saved. A summary of the notations used in our approach and their definitions is given in Table 1.

Table 1. Notations.

Symbol	Meaning
G	A workflow application
T	The set of tasks in the workflow application G
E	The set of edges in the workflow application G
D	The set of datasets in the workflow application G
$\lvert T(d_i) \rvert$	The number of tasks in T which use the dataset d_i
$Dep\ (d_i, d_j)$	The data dependency between the dataset d_i and d_j
DCD_w	Within-DataCenter Data Dependency
DCD_b	Between-DataCenter Data Dependency
HDD	High-Dependent Dataset
AT_{total}	The sum of all data access times of all datasets
AT_{avg}	The average access times of all datasets

Ø	Threshold value for data access times candidate pool
N_D	The total number of datasets
HAD	Hot-Access Dataset
DC	The set of data centers in the multi-cloud environment
CSP	The set of cloud service providers in the multi-cloud environment
$TCost$	Total cost
$TCost_{max}$	The total cost when there are no replication happened
$TCost_{current}$	The current total cost value when Ø stay at a specific value
$NR_{current}$	The current number of replicas when Ø stay at a specific value
μ	The cost reduction per replica
$Cost_s$	Data storage cost
$time_s$	The storage duration
$Cost_t$	Data transmission cost
DC^*	The set of data centers with all initial datasets and replicas
γ	The data storage rate of the cloud service provider csp

3.1 Prerequisite

Before the start of our data replication strategy, we assume that initial dataset and task placement has been completed by using a data and task placement strategy. Datasets and tasks have been allocated into geographically-dispersed data centers in DC from different cloud service providers in CSP.

3.2 Workflow application model

A workflow application $G = (T, E)$ is modelled as a Directed Acyclic Graph (DAG), where T is the set of vertices as tasks and E is a set of edges as the control dependencies between the tasks. In the workflow application G, the child task can only start after its parent tasks have finished and the associated control dependencies have been transferred to the child task.

3.3 Data dependency model

The data dependency represents the data relationship between each two datasets in D. The data dependency between datasets d_i and d_j is defined as the ratio of the number of tasks that use both d_i and d_j to the total number of workflow tasks T [19]. Therefore, the data dependency can be calculated as follows in equation 1.

$$Dep\ (d_i, d_j) = \frac{|(T(d_i) \cap T(d_j))|}{|T|} \tag{1}$$

In multi-cloud environments, we define Within-DataCenter Data Dependency (DCD_w) and Between-DataCenter Data Dependency (DCD_b). DCD_w is the data dependency between the dataset d_i and all other datasets within the same location of d_i. DCD_b is the data dependency between the dataset d_i and all other datasets within the different locations of d_i. DCD_w and DCD_b are both represented as a 2-tuple (dc, d). A $DCD(dc, d)$ function is used to calculate DCD_w and DCD_b for each dataset d in each data center dc. For each dataset d_i in D, we calculate their DCD_w and DCD_b based on its location dc in DC as follows in equation 2 and 3.

$$DCD_w(dc, d_i) = \sum_{j=1}^{n} Dep\ (d_i, d_j),\ i \neq j,\ (d_i \text{ and } d_j \text{ store in the same location}) \quad (2)$$

$$DCD_b(dc, d_i) = \sum_{j=1}^{n} Dep\ (d_i, d_j),\ i \neq j,\ (d_i \text{ and } d_j \text{ store in different locations}) \quad (3)$$

For a dataset d placed in the data center dc, if its $DCD_b(dc, d) > DCD_w(dc, d)$, we partition the dataset d into a new set of datasets called High-Dependent Dataset HDD. A $DepCompare()$ function is used to compare DCD_w and DCD_b for each dataset d in D in order to partition the datasets into High-Dependent Dataset HDD.

3.4 Data access times model

Data access times is the number of times of a dataset accessed by all tasks in a single execution of the workflow. We count data access times AT for each dataset d in D during workflow execution period by the function $AT(d)$. Then we calculate the sum of all data access times of all datasets AT_{total} as follows in equation 4 and set the threshold Ø. A $ATCalculation()$ function is used to calculate the value of AT_{total} and AT_{avg}.

$$AT_{total} = \sum_{i=1}^{n} AT(d_i),\ d_i \in D \quad (4)$$

Then we calculate the average data access times of all datasets ET_{avg} with the total number of datasets N_D as follows in equation 5.

$$AT_{avg} = \frac{AT_{total}}{N_D} \quad (5)$$

If $AT(d) >$ Ø $* AT_{avg}$ then, we partition the dataset d into a new set of datasets called Hot-Access Dataset HAD. The threshold value Ø can be dynamically changed from 0 to N_D in order to optimize the total cost and the number of replicas. The $ATCompare()$ function is designed to compare the value between $AT(d)$ and Ø $* AT_{avg}$ in order to determine if a dataset d in D should be categorized into HAD.

3.5 Eligible replicated dataset candidate pool

We compare HDD and HAD in order to identify the eligible dataset candidates for replication, which are the overlapping elements in both HDD and HAD. These eligible dataset candidates are both highly dependent and highly accessed. Replicas of these datasets should be created and placed into appropriate data centers using our replica placement strategy.

3.6 Multi-cloud environment model

Multi-cloud is the use of two or more cloud computing services in order to allow users to share the workload across multiple cloud service providers. A multi-cloud environment is represented as a 2-tuple $MC = (DC, CSP)$, where

- DC: $\{dc_1, dc_2, dc_3, \dots, dc_p\}$ is the set of data centers in the multi-cloud environment.
- CSP: $\{csp_1, csp_2, csp_3, \dots, csp_u\}$ is the set of cloud service providers in the multi-cloud environment.
- Each dc has only one csp, while one csp may have multiple dc.

3.7 Cost model for multi-cloud

The total cost $TCost$ is defined as the sum of the data storage cost $Cost_s$ and the data transmission cost $Cost_t$, as follows in equation 6.

$$TCost = \sum Cost_s + \sum Cost_t \tag{6}$$

The data storage cost $Cost_s$ is dependent on the data storage rate of the cloud service provider γ, the size of the dataset $Size\ (d)$, and the storage duration $time_s$. As each cloud service provider has its own data storage pricing model, it is necessary and indispensable to consider the data storage cost rates γ of different dc in DC. Data storage cost $Cost_s$ for the dataset d can be presented as follows in equation 7.

$$Cost_s = \sum_{dc=1}^{p} \gamma * Size\ (d) * time_s \tag{7}$$

The data transfer cost $Cost_t$ is dependent on the transfer cost ratio α , the size of the dataset $Size(d)$, and the data access times of the dataset $AT(d)$. Therefore, data transfer cost $Cost_t$ for the dataset d can be presented as follows in equation 8.

$$Cost_t = \alpha * Size(d) * AT(d) \tag{8}$$

3.8 Recommend value of Ø′

A recommend value of Ø′ will return when the result of following equation 8 (μ) is optimal, where $TCost_{max}$ denotes the total cost when there are no replication happened, and $TCost_{current}$ and $NR_{current}$ denotes the current total cost value and the current number of replicas respectively when Ø stay at a specific value. We insert an evaluation parameter μ to evaluate cost reduction per replica in equation 9. Therefore when μ stays at a maximum value at a specific value of Ø, it means the cost reduction per replica is optimal and this value of Ø can be returned as the recommend value Ø′.

$$\mu = \frac{TCost_{max} - TCost_{current}}{NR_{current}} \tag{9}$$

3.9 Algorithms

Our data replication algorithms include two sub-algorithms as follows.

Algorithm 1. Data replication loop

Input: $DC, D, CSP, \emptyset$

Output: DC^{*}: set of data centers with all initial datasets and replicas

$\emptyset'$: A recommended value of $\emptyset$

1. **begin**
2. Insert workflow G
3. Dynamically change threshold parameter $\emptyset$ from 0 to N_D by step 0.01
4. **start** Algorithm 2
5. List all eligible datasets
6. Place all eligible datasets to related task locations
7. Account the number of replicas $NR_{current}$
8. Calculate $TCost_{current}$ based on the placed location for all replicas
9. Account the $TCost_{max}$ when there are no replication happened
10. Calculate each value of evaluation parameter μ at different value of $\emptyset$
11. **end** Algorithm 2 after $\emptyset$ reach N_D
12. Find the best value of μ
13. **return** $\emptyset'$ and DC^{*}
14. **end**

Algorithm 2. Eligible replicated dataset creation

Input: $DC, D, CSP, \emptyset$

Output: eligible replicated datasets

1. **begin**
2. **for** (each dataset d, $d \in D$) **do**
3. Locate the location of all datasets
4. Calculate all data dependencies for each dataset
5. **for** (each data center dc, $dc \in DC$) **do**
6. Calculate DCD_w and DCD_b by function $DCD(dc, d)$

```
            Compare DCD_w and DCD_b for each dataset d in D by function DepCompare()
                While (DCD_b(d) > DCD_w(d)) do
                    Generate HDD candidate pool
                end while
    Continue
    Calculate all data access times for each d
        ATCalculation()
            ATCompare()
                While (AT(d) > Ø * AT_avg) do
                    Generate HAD candidate pool
                end while
      if  d ∈ {HAD ∩ HDD}
          then d is a eligible replicated dataset
      end if
      end for
        return all datasets and eligible replicated datasets
end for
end
```

4 Simulations

4.1 Simulation settings

Our simulations are conducted on CloudSim. We performed three scientific workflows, 25 nodes Montage workflow, 30 nodes CyberShake workflow and 30 nodes LIGO Inspiral workflow in order to simulate the effectiveness of our strategy. The data items of Montage workflow includes d_1 to d_{18} which are accessed by tasks {1, 45, 45, 45, 45, 45, 107, 107, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1} times respectively and has the data size from d_1 to d_{18} {0.29, 4000, 4000, 4000, 4000, 4000, 0.26, 270, 7.2, 2.3, 2.8, 21, 12, 7.2, 165430, 165430, 6600, 320} respectively. The data items of CyberShake workflow includes d_1 to d_5 which are accessed by tasks {90, 572, 574, 200, 1} times respectively and has the data size from d_1 to d_5 {220, 5500, 0.3, 2000, 2100} respectively. The data items of LIGO Inspiral workflow includes d_1 to d_8 which are accessed by tasks {42, 84, 42, 14, 79, 14, 35, 42} times respectively and has the data size from d_1 to d_8 {800, 150, 8600, 230, 300, 320, 940, 1200} respectively. The pricing model of four adopted cloud service providers (Amazon, Microsoft, AT&T and Google) is shown in Table 2. Besides, we set the storage duration $time_s$ as 1 for the cost calculation convenience in order to make the consistence of each data storage time in every different CSP.

Table 2. The pricing model of adopted multi-cloud service providers

Cloud service provider	Storage service	Storage Price (per data unit)
Amazon	Amazon S3	0.025
Microsoft	Microsoft Azure	0.034
AT&T	AT&T Cloud Storage	0.040
Google	Google Cloud Storage	0.026
Data Transfer Cost	0.070 per data unit	

After eligible datasets are determined, we create replicas for them and distribute the replicas to all task locations which require these replicas as input datasets and have enough available storage space. The reason of this placement operation is that replicas are frequently required by tasks which require these replicas as input datasets. Therefore, replicas may store as near as task locations for reducing the data movement cost.

4.2 Simulation results

In the first simulation, we tested four scenarios on all three scientific workflow applications. As shown in Figure 1, it is obvious that our strategy can significantly decrease the total cost compared with other three approaches in all three data-intensive workflows. Our strategy has a 94.12%, 99.10%, and 69.91% decrease respectively in Montage, CyberShake and LIGO Inspiral workflow to compare with the no replication scenario of those three workflows. Besides, our strategy has a 40.11% and 92.49% reduction respectively in Montage and CyberShake workflow to compare with the data dependency adoption only scenario of those two workflows. Apart from that, our strategy has a 31.41%, 92.80% and 67.32% decrease respectively in Montage, CyberShake and LIGO Inspiral workflow to compare with the data access times adoption only scenario of those three workflows.

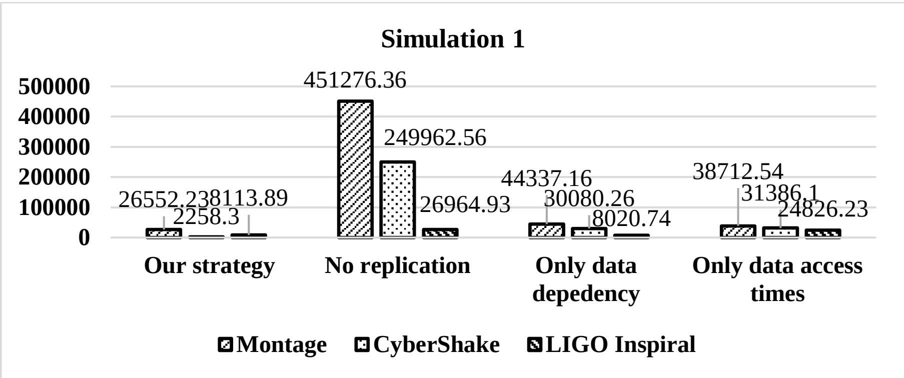

Fig. 1. The result of simulation 1

In the second simulation, we change the threshold Ø to dynamically adjust HAD in order to view the impact on the number of replica created and the total cost saving.

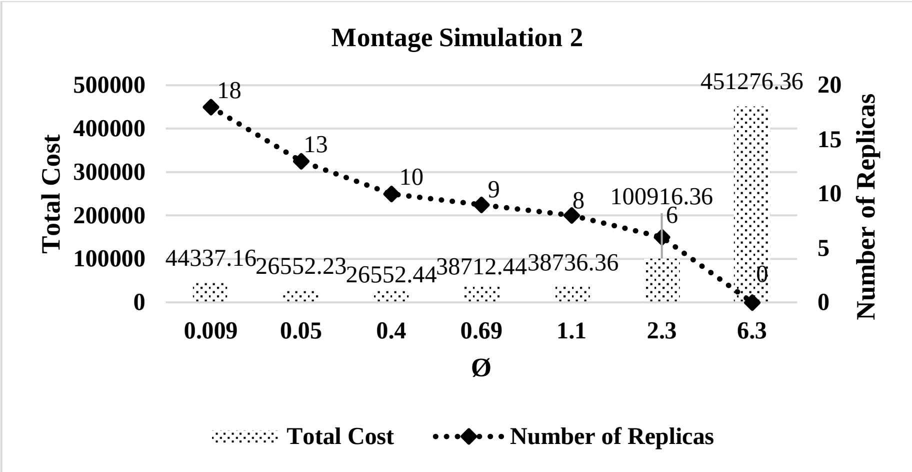

Fig. 2. The result of Montage workflow in simulation 2

As shown in Figure 2, there is an obvious fluctuation on the total cost and the number of replicas when the value of Ø dynamically increase from 0 to N_D in the Montage workflow. It is recommended that the cost reduction per replica remains at a maximum level when Ø stays at 2.3 in the Montage workflow. Similarly, we can find the results of CyberShake and LIGO Inspiral workflow in our simulation 2 as follows in Figure 3 and 4 as follows. It is recommend that the total cost and the number of replicas exist in an acceptable level when Ø stays in the range from 0.79 to 1.79 in the CyberShake workflow, and when Ø stays at 0.95 in the LIGO Inspiral workflow.

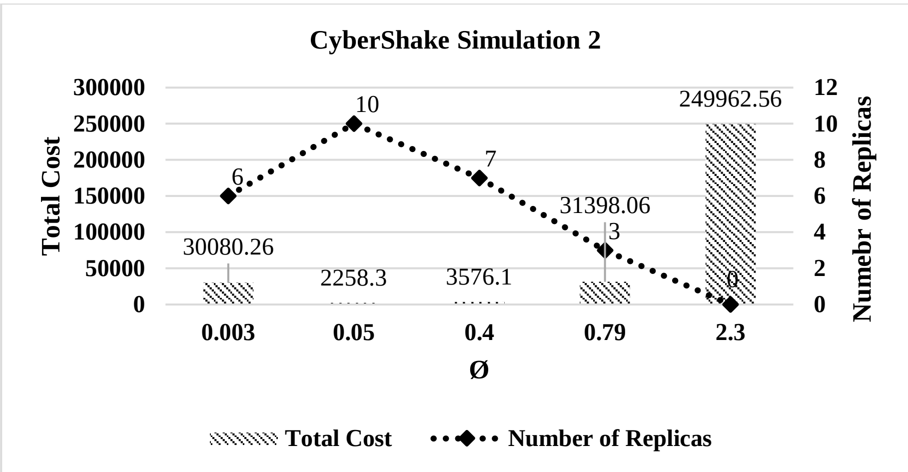

Fig. 3. The result of CyberShake workflow in simulation 2

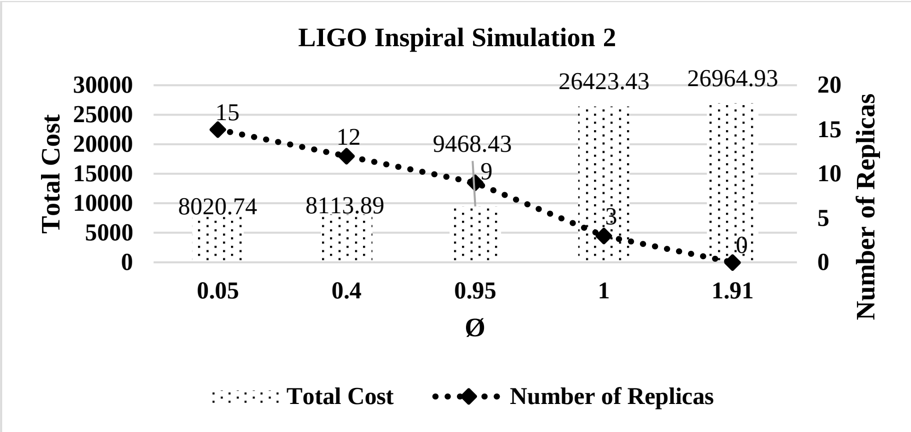

Fig. 4. The result of LIGO Inspiral workflow in simulation 2

5 Conclusions

To conclude, data replication is commonly used to decrease access latency, improve data availability, and reduce data transfer cost by creating data replicas to geographically-distributed data centers. In this paper, we propose a data dependency and access threshold based data replication strategy with the consideration of both data dependency and data access times jointly for data-intensive workflows in the multi-cloud environment. The simulation results shows that our data replication strategy can greatly reduce the total cost of data-intensive workflow execution and suggest a recommended value of Ø in order to find the optimal performance by using our strategy.

6 References

1. Buyya, R., Broberg, J., Gościński, A.: Cloud computing: principles and paradigms. Hoboken, NJ: John Wiley & Sons Publications.
2. Chang, R-S., Chang, H-P., Wang, Y-T.: A dynamic weighted data replication strategy in data grids. In: Computer Systems and Applications, 2008. AICCSA 2008. IEEE/ACS International Conference on, pp. 414-421. (2008).
3. Gill, N.K., Singh, S.: A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers. In: Future Generation Computer Systems 65, 10-32 (2016).
4. Janpet, J.,Wen, Y-F.: Reliable and available data replication planning for cloud storage. In: Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on, pp.678-685. (2013).
5. Khalajzadeh, H., Yuan, D., Grundy, J., Yang, Y.: Improving Cloud-based Online Social Network Data Placement and Replication. In: Cloud Computing (CLOUD), 2016 IEEE 9th International Conference on, pp. 678-685. (2016).
6. Li, W., Yang, Y., Yuan, D.: Ensuring Cloud data reliability with minimum replication by proactive replica checking. In: IEEE Transactions on Computers 65(5), 1494-1506 (2016).

7. Lin, J-W., Chen, C-H., Chang, JM.: QoS-aware data replication for data-intensive applications in cloud computing systems. IEEE Transactions on Cloud Computing 1(1), 101-115 (2013).
8. Liu, G., Shen, H., Chandler, H.: Selective data replication for online social networks with distributed datacenters. IEEE Transactions on Parallel and Distributed Systems 27(8), 2377-2393 (2016).
9. Long, S-Q., Zhao, Y-L., Chen, W.: MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster. Journal of Systems Architecture 60(2), 234-244 (2014).
10. Marinescu, D.C.: Cloud computing: theory and practice. Boston: Elsevier/Morgan Kaufmann, Morgan Kaufmann is an imprint of Elsevier (2013).
11. Milani, B.A., Navimipour, N.J.: A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions. Journal of Network and Computer Applications 64, 229-238 (2016).
12. Rasool, Q., Li, J., Oreku, G.S., Zhang, S., Yang, D.: A load balancing replica placement strategy in Data Grid. In: Digital Information Management, 2008. ICDIM 2008. Third International Conference on, pp. 751-756. (2008).
13. Tos, U., Mokadem, R., Hameurlain, A., Ayav, T., Bora, S.: A performance and profit oriented data replication strategy for cloud systems. In: Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences, pp. 780-787. (2016).
14. Wang, C., Lu, Z., Wu, Z., Wu, J., Huang, S.: Optimizing Multi-Cloud CDN Deployment and Scheduling Strategies Using Big Data Analysis. In: Services Computing (SCC), 2017 IEEE International Conference on, pp. 273-280. (2017).
15. Wang, T., Yao, S., Xu, Z., Jia, S.: DCCP: an effective data placement strategy for data-intensive computations in distributed cloud computing systems. The Journal of Supercomputing 72(7), 2537-2564 (2016).
16. Wu, X.: Data Sets Replicas Placements Strategy from Cost-Effective View in the Cloud. Scientific Programming 2016, (2016).
17. Ye, Z., Li, S., Zhou, J.: A two-layer geo-cloud based dynamic replica creation strategy. Applied Mathematics & Information Sciences 8(1), 431-439 (2014).
18. Yuan, D., Cui, L., Liu, X.: Cloud data management for scientific workflows: Research issues, methodologies, and state-of-the-art. In: Semantics, Knowledge and Grids (SKG), 2014 10th International Conference on, pp. 21-28. (2014).
19. Yuan, D., Yang, Y., Liu, X., Chen, J.: A data placement strategy in scientific cloud workflows. Future Generation Computer Systems 26(8), 1200-1214 (2010).
20. Zhang, Q., Li, S., Li, Z., Xing, Y., Yang, Z., Dai, Y.: CHARM: A cost-efficient multi-cloud data hosting scheme with high availability. IEEE Transactions on Cloud Computing 3(3), 372-386 (2015).