# A QoS-enhanced data replication service in virtualised cloud environments

Tienan Zhang

School of Computer and Communication,
Hunan Institute of Engineering,
Hunan Province, China
Email: ztnan1974@126.com

**Abstract:** In recent cloud systems, data-intensive applications have been widely deployed for solving non-trivial problems, which often involve large amount of dataset. As a standard service for offering data availability and reliability, data replication service plays an important role and is proven to be effective in many real-world cloud environments. However, many existing replication services are designed for improving the system-oriented metrics instead of user-oriented metrics. In this paper, we first formulate the quality-of-service (QoS) constrained replication placing problem in different manners and then a novel replication service is proposed which uses multi-objective technique to solve the replica placement problem via a time-efficient approach. A set of experiments are conducted to evaluate the effectiveness and efficiency of the proposed service and the results indicate it can significantly reduce the average response time metric comparing with other existing replication services and therefore improve the QoS for user applications. In addition, it also exhibits a good robustness when the cloud storage system is facing intensive replication workloads.

**Keywords:** data replication; quality-of-service; QoS; response time; cloud computing.

**Reference** to this paper should be made as follows: Zhang, T. (2020) ‘A QoS-enhanced data replication service in virtualised cloud environments’, *Int. J. Networking and Virtual Organisations*, Vol. 22, No. 1, pp.1–16.

**Biographical notes:** Tienan Zhang received his Master’s in the Xiangtan University at 2008. Currently, he is a Lecturer of the Hunan Institute of Engineering and works as the Dean of the Computer Department. His research directions include cloud computing, distributed computing and storage, distributed intelligent and green computing in datacentres. Also, he is a member of the ACM Society and IEEE Computer Society.

## 1 Introduction

In the past decade, cloud computing has become a promising distributed computing paradigm for various large-scale applications (Zhang et al., 2013; Grozev and Buyya, 2014). In many real-world cloud systems, data-intensive applications have been widely deployed for solving non-trivial problems, which often involve large amount of dataset (Shamsi et al., 2013; Szabo et al., 2014; Su, 2014). As a standard service for offering data availability and reliability, data replication service plays an important role and is proven to be effective in many real-world cloud environments (Amjad et al., 2012; Mokadem and Hameurlain, 2015). Generally speaking, data replication service is to store copies of data in different locations so that the data can be easily recovered if one copy at one location is lost or unavailable (Preis and Seitz, 2012; Tos et al., 2015). As cloud users are paying more and more attentions on quality-of-service (QoS), an effective data replication service should reduce access latency and bandwidth consumption, also be capable of achieving load balancing and improves reliability by creating multiple data copies (Victer Paul et al., 2012; Inacio and Dantas, 2014; Mansouri, 2016). Unfortunately, due to the dramatic increasing size of datasets, many traditional data replication services become more and more inefficient, which is especially true when the Big Data age is coming (Balasangameshwara and Raju, 2013; Mokadem and Hameurlain, 2015; Savi et al., 2015).

In current cloud platforms, replication service is generally deployed on the storage infrastructure which can be constructed via IaaS model (Laatikainen et al., 2016; Song et al., 2016). In such an IaaS model, storage resource can be flexibly provisioned based on pre-defined resource management strategies, such as high dependability, low energy consumption, etc. (Hu et al., 2010; Aisyah and Cockcroft, 2014; Ren et al., 2015). As a result, the efficiency of data transferring in cloud-based storage platforms becomes every unstable since available bandwidth will be changed with the available storage nodes (Zeng and Veeravalli, 2014; Boru et al., 2015), which in turn adversely impact the QoS performance of data replication service (Thompson and Lehaney, 2008; Gill and Singh, 2016). Although increasing the number of replicas can reduce the data transferring latency, it also leads to other kinds of performance degradation (Choi and Youn, 2012). From the perspective of cloud providers, a desired replication service should be able to improve the user QoS satisfactory without causing too many extra costs (Abawajy and Deris, 2014). So, many effective replication services aiming at cloud systems have been proposed in recent years. However, many existing replication services are designed for improving the system-oriented metrics instead of user-oriented metrics.

In this study, we proposed a QoS-enhanced data replication service which uses multi-objective technique to solve the replica placement problem via a time-efficient approach. To do this, we first formulate the QoS constrained replication placing problem with different objectives, including minimising storage nodes, minimising the maximal response time and minimising average response time (ART). Specifically, the first objective is to select a storage node set as less as possible which can cover all the replication requests from the current network site; the second objective is to select the candidate sites to host replica by minimising the maximum response time; the third objective is to optimise the request-weighted ART, which is the time elapsed to transfer a file from the nearest replication site. By individually analysing the above constrained optimisation problems, we find an approach to jointly solving them in a time-efficient manner. Extensive experiments are conducted in a well-designed test-bed and the results indicate that the proposed replication service can significantly reduce the ART metric comparing with other existing replication services and therefore improve the QoS for user applications. In addition, it also exhibits a good robustness when the cloud storage system is facing intensive replication workloads.

The rest of this paper is organised as follows. Section 2 presents the related work. In Section 3, we describe the replication placement problems in formal manner. In Section 4, we present the implementation of the proposed replication algorithm. In

Section 5, experiments are conducted to investigate the effectiveness of the proposed approach. Finally, Section 6 concludes the paper with a brief discussion of the future work.

## 2 Related works

### 2.1 Replication services in cluster and grid

In early studies, data replication services are mainly implemented in high-performance cluster or grid systems. For instance, in Chang and Chang (2008), the authors proposed a dynamic data replication mechanism called latest access largest weight (LALW), which selects a popular file for replication and calculates a suitable number of copies and grid sites for replication. By associating a different weight to each historical data access record, the importance of each record is differentiated. A more recent data access record has a larger weight. In Doğan (2009), the author analysed the performance of eight dynamic replication algorithms under various data grid settings and their results indicated that replication algorithm should be workload-aware. In Khanli et al. (2011), the authors proposed a dynamic replication method in a multi-tier data grid called predictive hierarchical fast spread (PHFS) which is an extended version of fast spread. PHFS tries to predict future needs and pre-replicates them in hierarchal manner to increase locality in accesses and consequently improves performance. In Nukarapu et al. (2011), the authors proposed a data replication algorithm that not only has a provable theoretical performance guarantee, but also can be implemented in a distributed and practical manner. Specifically, they designed a polynomial time centralised replication algorithm that reduces the total data file access delay by at least half of that reduced by the optimal replication solution. Based on this centralised algorithm, they also design a distributed caching algorithm, which can be easily adopted in a distributed environment such as data grids. In Bsoul et al. (2012), a dynamic replication strategy based on fast spread but superior to it in terms of total response time and total bandwidth consumption is proposed, which is achieved by storing only the important replicas on the storage of the node. The main idea of this strategy is using a threshold to determine if the requested replica needs to be copied to the node. In Wang et al. (2012), two dynamic data replication strategies were designed. The first one employs historical access records which are useful for picking up a file to replicate. The second one is a proactive deletion method, which is applied to control the replica number to reach an optimal balance between the read access time and the write update overhead. In Abawajy and Deris (2014), the authors addressed the problem of data consistency when using replication service in distributed environments and proposed a quorum-based data replication protocol with the objectives of minimising the data update cost, providing high availability and data consistency.

### 2.2 Replication service in cloud

In recent years, as cloud computing is becoming more and more popular, integrating data replication service in cloud system has become a hot topic. For example, Sun et al. (2012) comprehensively analysed the relationship between resource availability and the number of replicas in clouds and then evaluated the model of triggering a replication operation when the popularity data passes a dynamic threshold. In Long et al. (2014), the authors proposed a multi-objective offline optimisation approach for replica management, in which various factors influencing replication decisions such as mean file unavailability, mean service time, load variance, energy consumption and mean access latency are individually considered as optimisation objectives and artificial immune algorithm was applied to solve the constrained optimisation problems of replication placement. In Zeng and Veeravalli (2014), the authors attempted to determine the optimal number of replicas so as to achieve the minimum mean response time of all the metadata requests. The target optimal constrained function was formulated and a novel metadata request balancing algorithm based on request arrival rates was proposed, which can find near-optimal solutions by a theoretical proof. In Boru et al. (2015), the authors considered both energy efficiency and bandwidth consumption of the cloud system and designed an energy-aware replication scheme, which can achieve certain tradeoffs between performance and energy consumption when performing massive data replication operations in cloud environments. In Gill and Singh (2016), a cost-aware data replication strategy was proposed which identifies the minimum number of replicas required to ensure the desired availability. The concept of knapsack was used to optimise the cost of replication and to re-replicate the replicas from higher-cost data centres to lower-cost datacentres without compromising the data availability. In Nagarajan and Mohamed (2017), an *intelligent replica manager* (IRM) was designed and incorporated in the middleware of clouds for scheduling data-intensive applications. IRM uses a multi-criteria-based replication algorithm which considers multiple parameters like storage capacity, bandwidth and communication cost of the neighbouring sites before taking decisions for the selection and placement of replica.

## 3 Problem model and definition

For the convenience of analysis, we first give some notations which will be used in the following sections. Consider there are $n$ storage sites in a cloud system, which can be used as potential hosting locations for all data replicas. A replication cover scheme can be noted as a $n \times n$ matrix **S**, in which if the element $s_{i,j} = 1$ then it means site $j$ can cover the replica requesting from site $i$, otherwise $s_{i,j} = 0$. A replication assigning scheme is also denoted as $n \times n$ matrix **R**, in which if a replica requesting from site $i$ is assigned to the replica at site $j$, then the element $r_{i,j} = 1$, otherwise $r_{i,j} = 0$. For a given data replica, we use the Boolean variance $x_i$ to denote that whether site $i$ is selected to host this replica.

In a data replication service, an important issue is to select the most suitable candidate location where data replicas can be placed. Based on the above definitions and notations, if the objective of the data replication service is to locate minimum number of sites to place a replica to cover all the requesting sites, the candidate sites that host replica will be at least covering time distance away from the requesting sites. This problem can be formulated as a classic *set covering problem* as following:

$$\begin{cases} \min \sum_{i=1}^{n} x_i \\ \text{s.t.} \sum_{i=1}^{n} x_j s_{i,j} \geq 1 \end{cases} \tag{1}$$

As shown in equation (1), the goal is to minimise the number of sites that are selected and the constraint is to ensure that that each requesting site must be covered by at least one replication site.

In many scenarios, the data replication service is expected to address the problem of minimising the maximum response time using a given number of replication sites that host a data replica. Let the maximum response time be denoted as $T_{\max}$, the given number of replication sites be noted as $k$, the response time between site $i$ and site $j$ is noted as $d_{i,j}$. Then, we can formulate this problem as following:

$$\begin{cases} \min T_{\max} \\ \text{s.t.} \sum_{j=1}^{n} r_{i,j} = 1 \\ \sum_{i=1}^{k} x_i = k \\ \forall i, j \left( r_{i,j} - x_j \right) \le 0 \\ \sum_{j=1}^{n} d_{i,j} y_{i,j} \le T_{\max} \end{cases} \tag{2}$$

As shown in equation (2), the first constraint indicates that each requesting site should be allocated exactly one replication site from which it can fetch the replica; the second constraint states that exactly $k$ sites to be located to place the replica; the third constraint ensures that the requests at site $i$ can only be assigned at replication site $j$ if a replica is placed at site $j$; the final constraint means that the response time between any requesting site and the nearest replication site should be smaller than $T_{\max}$. If we use an individual weight $w_i$ to indicate the priority of the requests from site $i$, then the maximum response time $T_{\max}$ can be replaced by the request-weighted response time $T_{avg} = \sum_{i=1}^{n} \sum_{j=1}^{n} \left( w_i d_{i,j} r_{i,j} \right)$ and the problem shown in equation (2) can be rewritten as

$$\begin{cases} \min \sum_{i=1}^{n} \sum_{j=1}^{n} \left( w_i d_{i,j} r_{i,j} \right) \\ \text{s.t.} \sum_{j=1}^{n} r_{i,j} = 1 \\ \sum_{i=1}^{k} x_i = k \\ \forall i, j \left( r_{i,j} - x_j \right) \le 0 \\ \sum_{j=1}^{n} d_{i,j} y_{i,j} \le T_{\max} \end{cases} \tag{3}$$

As the constraints in equations (2) and (3) are the same, the two problems in fact can be jointly considered. In the next section, we will design a method to solve these two problems.

## 4 Data replication service

### *4.1 Replication placement algorithm*

To solve the problems shown in equations (2) and (3), the common approach is using the Lagrangian relaxation (LR) technique which relaxes some constraints of the original model and adds those constraints, multiplied by Lagrange multiplier to the objective function. In fact, we can easily to obtain the optimal solution of problem (2) or (3) by using the LR technique. Unfortunately, the LR technique is very inefficient if not impossible when multi-objective optimisation is required. So, we need to find an efficient method to jointly solve the problem with two optimisation objective.

Considering the problem (2), a feasible method to approximately solve it is by using the upper and lower bounds of the objective function. Specifically, let $T_{up}$ and $T_{low}$ be the current upper and lower bounds of the problem (2), then the middle value of $T_{up}$ and $T_{low}$ can be used as *covering time distance* which can be considered as a feasible (but not necessarily optimal) solution to the *set covering problem* as shown in equation (1). If the number of sites found by problem (1) is equal to $k$, we stop the algorithm otherwise we modify the upper and lower bound and solve the problem (1) again. As the purpose, this procedure is to minimise the median objective without keeping any requesting site too far from a candidate replication site, the current solution will not exceed a certain level when problem (2) is considered. Based on this observation, our multi-objective replica placement problem can be solved by the following algorithm

**RPMO algorithm: replication placement with multi-objective optimisation**

**Begin**

1 Initialise the problem (1) by the upper and lower bounds of $T_{\max}$.

2 Solving the problem (1) and find a minimum site set $k^*$.

3 Using $k^*$ to solve problem (1) and problem(2)

4 Calculating the values of objective functions found by problem (1) and problem (2), noted them as $T_1$ and $T_2$;

5 **for** $t := 1$ to $m$ **do**

6 $$L(t) = T_1 + \frac{t}{m-t} \cdot (T_2 - T_1);$$

7   Using $L(t)$ to solve the problem (2) and (3);

8   Calculating the $k_1$ and $k_2$ when problem (2) and (3) are solved.

9   **if** $k_1$ and $k_2$ **then**

10     **return** the current solutions for problem (2) and (3);

11   **end if**

12 **end for**

**End.**

In Step 5–Step 12, the algorithm generate a set of non-inferior solutions, which are defined as feasible solutions if there is no other feasible solution that will yield an improvement in one objective without causing degradation at least in one objective. When using LR technique to solve problem (2) or (3), we use sub-gradient optimisation method to update the Lagrangian multipliers, which is effective to maximise the lower bound of the solution to satisfy the relaxed constraint that is relaxed. Specifically, we first define the sub-gradient in each iteration as following:

$$G_i(t) = \sum_{j=1}^{n} \left(1 - r_{i,j}(t)\right) \tag{4}$$

Then, the step size in each iteration is defined as

$$\Delta\delta = \frac{T_{\max} - T_{\min}}{\sum_{i=1}^{n} G_i^2(t)} \tag{5}$$

Therefore, Lagrangian multiplier $\lambda_i(t)$ in each iteration can be update by following:

$$\lambda_i(t+1) = \max\left\{0, \lambda_i(t) + \Delta\delta \cdot G_i(t)\right\} \tag{6}$$

### *4.2 Minimise data transferring cost for replication service*

The replication placement algorithm can ensure that the optimal or near-optimal solution based on the user requests and network characteristics for the current period. However, the candidate sites that hold replicas currently may not be the best sites to fetch replica if the user requests and network latency changes. Therefore, relocation needs to be considered if the performance is to be maintained. Unfortunately, due to dramatic increasing size of dataset in many cloud storage systems, the costs of data replica relocation are typically very higher. Meanwhile, frequently performing relocation operations will lead to short-term bandwidth starvation, which significantly reduce the QoS from the perspective the cloud users. Therefore, we must take into account the data transferring costs when managing the current replication service.

In a replication service, the costs of data transferring will be increased when more and more data replicas are placed in the current system. To determine the performance degradation, we must determine the optimal cumulative ART if reallocation is permitted while accounting for the transfer costs. In current cloud systems, the performance monitoring is often done by a meta-scheduler or resource broker. To remove a data from a site, the resource broker must send a message, a small overhead message to initiate the much larger data transfer. Assume the replication service has been executed for $m$ time periods, $c_i(t)$ is the basic cost function when performing data-transferring from site $i$ at time period $t$. Thus, the problem of minimising data transferring cost in replication service can be formulated as following:

$$\begin{cases} \min \sum_{t=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(w_i(t) \cdot d_{i,j}(t) \cdot r_{i,j}(t)\right) + \sum_{t=1}^{m} \sum_{j=1}^{n} \left(c_j(t) \cdot s_{i,j}(t)\right) \\ \text{s.t.} \sum_{j=1}^{n} r_{i,j} = 1 \\ \sum_{i=1}^{k} x_i = k \\ \forall i, j \left(r_{i,j} - x_j\right) \le 0 \end{cases} \tag{7}$$

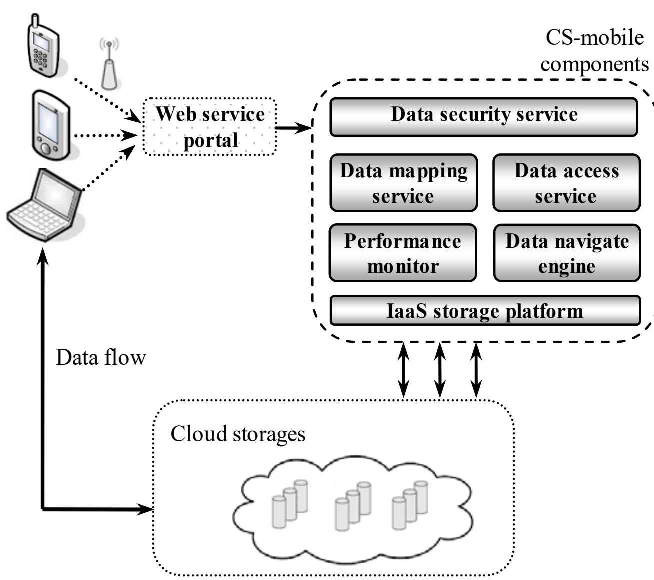
Problem (7) also can be efficiently solved by using LR technique and periodically executed when cloud administrator find that the current performance of replication service has been degraded below a pre-define threshold.

## 5 Experiment and performance

### 5.1 Experiment settings

In order to examine the performance of the proposed replication service, we deploy its implementation in the DCSP-MC system (Peng and Guofeng, 2013), which is a cloud-based storage infrastructure designed for offering reliable storage service for mobile applications. In Figure 1, we demonstrate the framework of the DCSP-MC system and the data replication service is currently integrated into the data mapping service component. In this test-bed DCSP-MC, the underlying physical storage nodes come from an IaaS storage infrastructure which can be flexibly scaled up or down the number of available storage nodes. In our experiments, we use the DCSP-MC's service to configure the ranges of available storage numbers, including [50, 100], [100, 150], [150, 200], [200, 250], [250, 300] and [300, 350]. By changing the available storage nodes, we hope to examine the scalability of our RPMO replication algorithm. Meanwhile, we also select four existing replication policies and compare their performance with the proposed RPMO service in terms of various metrics. The selected replication policies include maximising availability data replication (MADR), threshold-based dynamic data replication (TDDR), performance-prediction-based replication (PPR) and life-cycle-based data replication (LCDR). These replication policies are proposed for different goals and have been applied in many real-world storage platforms. For example, TDDR is aiming at maintaining the runtime performance of storage platform (i.e., I/O throughput and admission) when being in presence of various workloads; as to PPR, it is mainly used to help job schedulers to improve the execution efficiency of tasks.

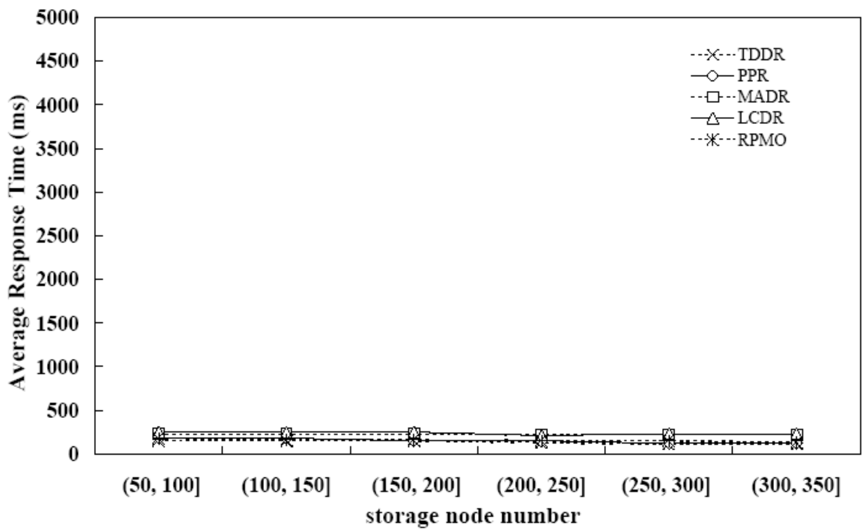
**Figure 1** Framework of DCSP-MC test-bed



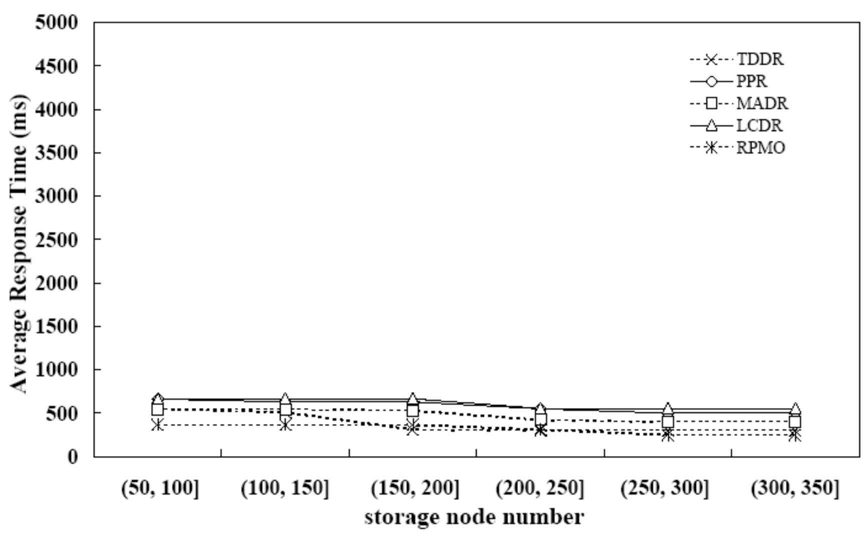
## *5.2 Settings of experiments*

The first performance metric we examined is the *ART*, which plays a critical role when user QoS satisfactory is concerned. To fully take the size of replicas, we configure the size of data items from 100 KB to 10 MB and conduct the experiment four times. The results are shown in Figures 2(a)–2(d).

**Figure 2** ART comparisons under different storage nodes, (a) data size = 100 KB (b) data size = 1 MB (c) data size = 10 MB (d) data size = 20 MB

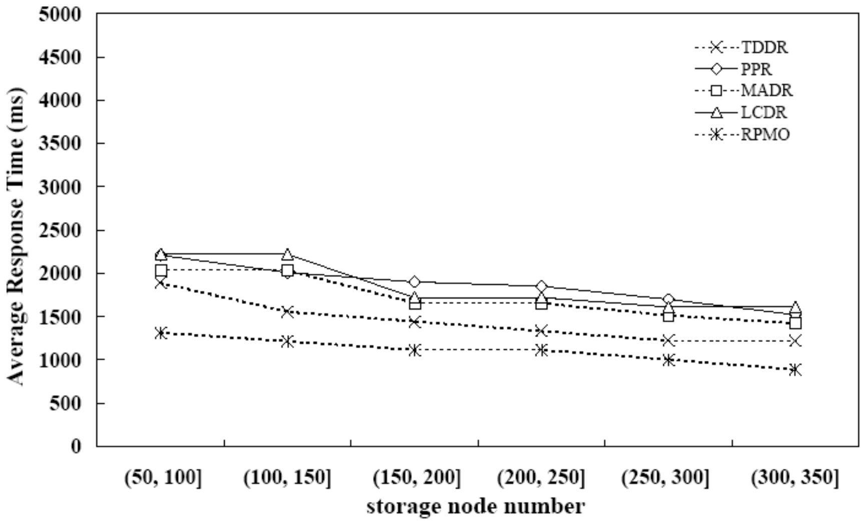



(a)

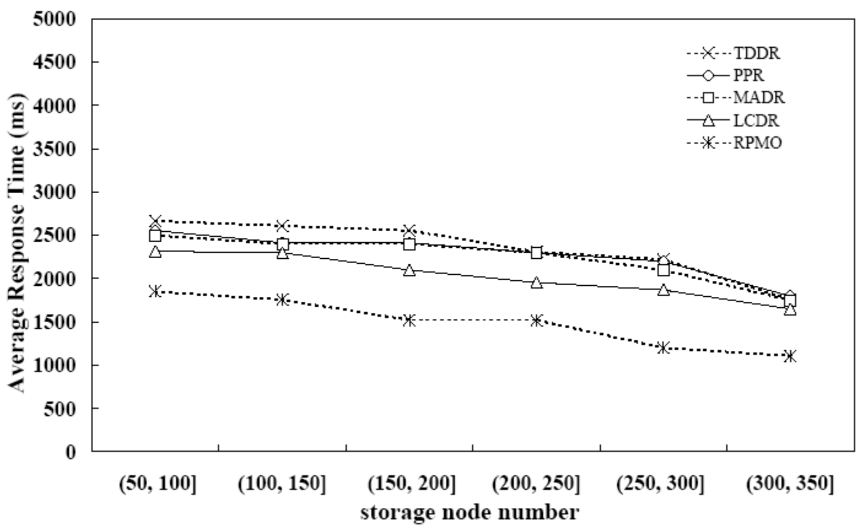



(b)

**Figure 2** ART comparisons under different storage nodes, (a) data size = 100 KB (b) data size = 1 MB (c) data size = 10 MB (d) data size = 20 MB (continued)




(c)




(d)

As shown in Figure 2(a), we notice that when using different replication services, the ART metric always maintains in a very low level and their differences almost can be ignored. This indicates that the performance of different replication services is very close when requested data size is very small. According to our experimental results, we find that TDDR outperforms other replication services by about 5%–8%, while the PPR performs worst among the five tested services. Also, we notice that the ART metric will be slightly decreased with the increasing of storage node number. This is because more available storage nodes means the requested replica can be retrieved from more hosted locations.

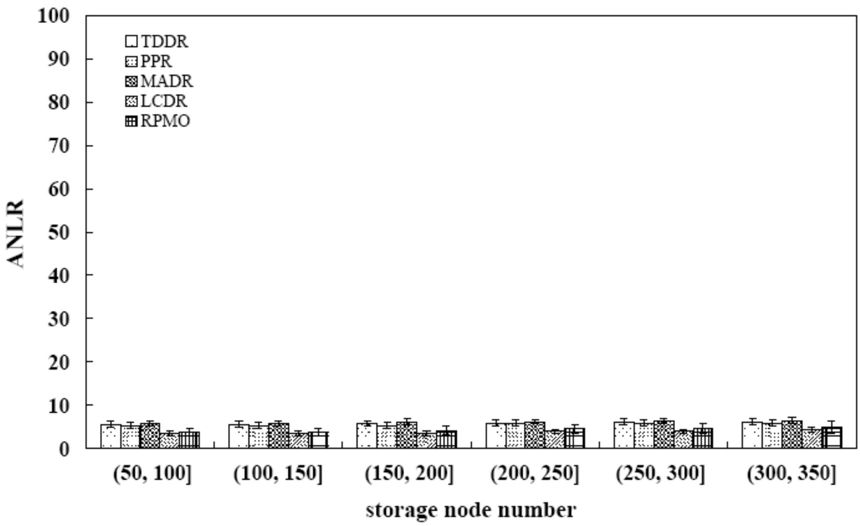
When the data size is increased to 1 MB as shown in Figure 2(b), it is clear that the ART metrics in all experiments are generally increased by about 210%–440% depending on the used replication service and the available storage nodes. In this experimental scenario, the PPR service still performs worst while the proposed RPMO service performs best among the five replication services. Specifically, RPMO outperforms other four services by about 35%–77% when the available storage nodes is less than 200; when the storage node number is increased more than 200, we find that the performance of RPMO and TDDR is very close. In this experimental scenario, we notice that the utilisation rate of underlying storage nodes maintains in low level (about 22%), which means that there are sufficient storage nodes to accommodate the demands from upper applications. So, we further increase the size of replicated data items to 10 MB and 20 MB and the experimental results are shown in Figures 2(c) and 2(d). In these scenarios, we find the utilisation rate of underlying storage nodes are increased to about 35% and 58%, which are similar to the real-world usage according to our former experience. The results clear show that the ART metric obtained by using RPMO service is significantly lower than other tested replication services by about 54%–112% in the tested cases. More importantly, we can see that our RPMO service exhibits a good robustness when the system is facing very intensive replication workloads. As to the other services, we find that TDDR still performs well when the size of replicas is 10 MB, however its performance dramatically degrades when the size of replicas is 20 MB. By examining the experimental logs, we notice that due to admission controlling policy used in TDDR, many replication requests have to be distributed to some nodes with lower storage utilisation, which typically have longer average transferring distances.

### *5.3 Average number of living replicas*

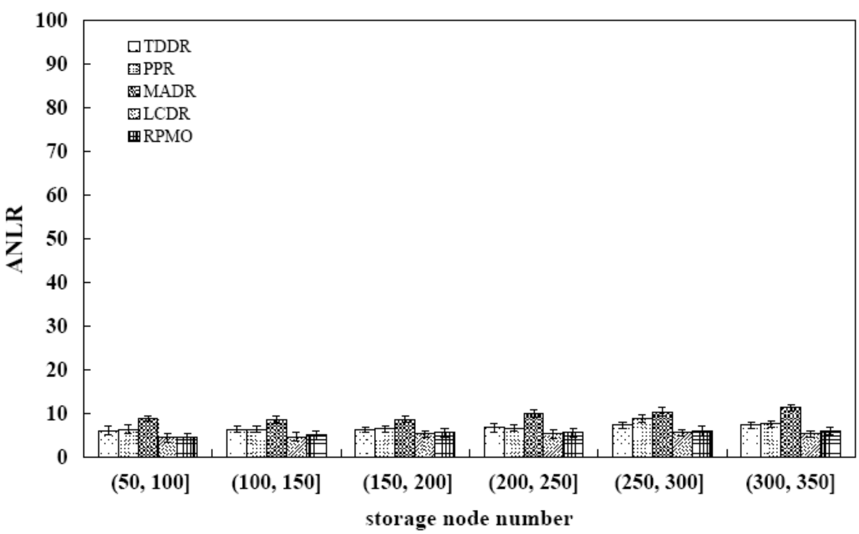
For a replication service, one of the major objectives is to maintain a desirable level of data availability. However, improving data availability typically needs to increase the number of replicas of data items, which means lower effective utilisation of storage resources. Therefore, better tradeoffs between data availability and utilisation of storage resources are of significant importance when designing an effective replication scheme. In this set of experiments, we first define different *levels of data availability* (LDA) from 80% to 95%, under which we investigate the *average number of living replicas* (ANLR) metric when using different replication services. In this way, we can evaluate the cost we have to pay when a specific LDA is required. To keep the workload in a high level, we configure that the average size of replicas is 10 MB and the experimental results are shown in Figures 3(a)–3(d).

According to the results shown in Figure 3, we can see that the ANLR metric only slightly increases with increasing number of underlying storage nodes. In Figure 3(a), we see that the ANLR metrics obtained by using TDDR, PPR and MADR are very similar, while the ANLR metrics of using LCDR and the proposed RPMO is less than them by about 25%–31%. As mentioned in Section 5.1, the LCDR service is designed for reduce the replicas based on the life-cycle estimation method, which means that it will periodically delete some of the replicas when these replicas have an unexpected longer existing time. In this way, LCDR service is very effective to improve the effective utilisation of underlying storage resources. As to the RPMO, we do not enforce any enforce any replica controlling or management as described in Section 4. However, we can see that the ANLR metrics of using RPMO service is only higher than that of using LCDR service by 3%–7% when the LDA is defined as 80%. This result means that to maintain the same level of data availability, the proposed RPMO service only needs to pay 3%–7% extra storage spaces comparing with the optimal result obtained by using LCDR.

**Figure 3** ANLR comparisons under different storage nodes, (a) LDA = 80% (b) LDA = 85% (c) LDA = 90% (d) LDA = 95%

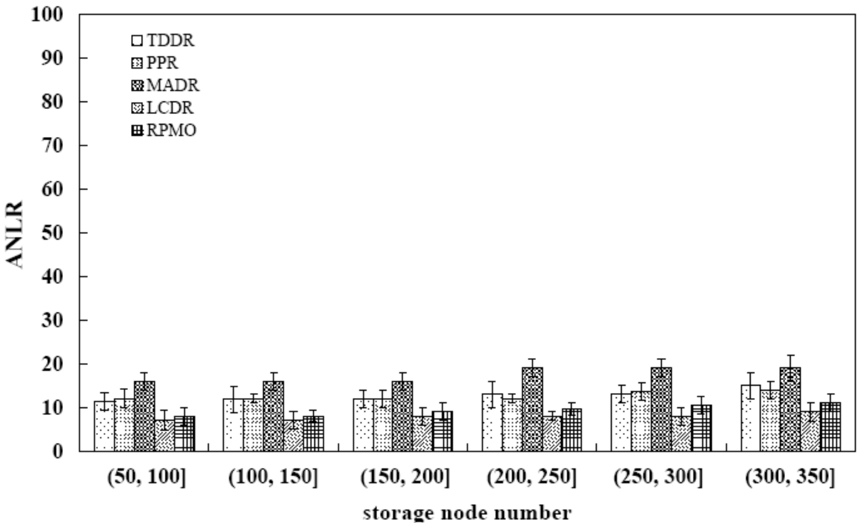



(a)

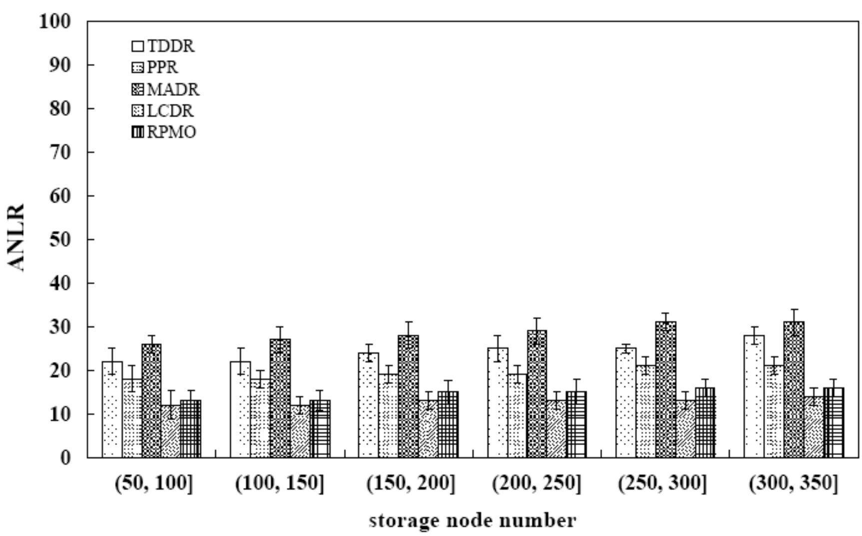



(b)

**Figure 3** ANLR comparisons under different storage nodes, (a) LDA = 80% (b) LDA = 85% (c) LDA = 90% (d) LDA = 95% (continued)




(c)




(d)

Then, we further increase the LDA to 85% and 90% as shown in Figures 3(b) and 3(c). It is clear the trends of ANLR metric are very similar to the results shown in Figure 3(a), except that the ANLR metric obtained by using MADR service increase more quickly than other services. This is because that MADR service is designed for maximising the data availability, which typically means more replicas are required for a given LDA value. The MADR service is very effective for those systems where the data availability or reliability is the primary designing goal. As the price for achieving this goal, MADR has to spend more storage resource on replicated data items. Based on our experimental results, we find that the increasing ANLR metric of using MADR is not linear to the increasing of LDA. Such a result implies that the scalability of MADR service is not very well and its expected performance in large-scale platforms will be significantly reduced as well. For TDDR and PPR, we find that their performance difference in terms of ANLR metric be clearly only when the LDA is set as 95%. In this case, PPR outperforms TDDR by about 25%–37%. As to the proposed RPMO service, its corresponding ANLR metrics is higher than that of using LCDR by about 5%–13%, which is slighted increased comparing with the cases in Figures 3(a) and 3(b) where LDA is lower than 90%. Even so, we still see such a price is worthy comparing with the benefits obtained by RPMO. More important, we know that user QoS satisfactory has become more and more important in nowadays cloud systems, which also directly decides the final revenue of a cloud provider. So, improving ART metric has more advantages than improving the storage resource utilisation.

## 6 Summary and future works

As data-intensive applications have been widely deployed for solving non-trivial problems, data replication service is playing a more and more important role in many real-world cloud environments. However, many existing replication services are designed for improving the system-oriented metrics instead of user-oriented metrics. In this paper, we formulate the QoS constrained replication placing problem in different manners and then a novel replication service is proposed which uses multi-objective technique to solve the replica placement problem via a time-efficient approach. Extensive experiments are conducted to evaluate the effectiveness and efficiency of the proposed service and the results indicate it can significantly reduce the ART metric comparing with other existing replication services and therefore improve the QoS for user applications.

Currently, the proposed replication service is implemented as sub-module in our existing cloud storage framework. So, the next work of this study is to integrated it into some well-known cloud middleware (i.e., OpenStack). In addition, based on the observations in our experiments, we also make some researching plans to improve our current work, including:

1. We are planning to integrate a redundancy management policy in our replication service with aiming at improving its resource utilisation rate.
2. Realising certain load-balancing mechanism is another objective in our future study.
3. In the future, we are planning to take more QoS metrics into account, such as resource price, real-time demand, energy-efficiency and etc, after all the cloud computing is expected to accommodate more and more applications with various QoS requirements.

## Acknowledgements

This work is a project supported by the Scientific Research Fund of the Hunan Provincial Education Department (No. 16C0397).

## References


Abawajy, J.H. and Deris, M.M. (2014) 'Data replication approach with consistency guarantee for data grid', *IEEE Transactions on Computers*, Vol. 63, No. 12, pp.2975–2987.

Aisyah, M. and Cockcroft, S. (2014) 'A snapshot of data quality issues in Indonesian community health', *International Journal of Networking and Virtual Organisations*, Vol. 14, No. 3, pp.280–297.

Amjad, T., Sher, M. et al. (2012) 'A survey of dynamic replication strategies for improving data availability in data grids', *Future Generation Computer Systems*, Vol. 28, No. 2, pp.337–349.

Balasangameshwara, J. and Raju, N. (2013) 'Performance-driven load balancing with a primary-backup approach for computational grids with low communication cost and replication cost', *IEEE Transactions on Computers*, Vol. 62, No. 5, pp.990–1003.

Boru, D., Kliazovich, D. et al. (2015) 'Energy-efficient data replication in cloud computing datacenters', *Cluster Computing*, Vol. 18, No. 1, pp.385–402.

Bsoul, M., Al-Khasawneh, A. et al. (2012) 'A threshold-based dynamic data replication strategy', *Journal of Supercomputing*, Vol. 60, No. 3, pp.301–310.

Chang, R-S. and Chang, H-P. (2008) 'A dynamic data replication strategy using access-weights in data grids', *Journal of Supercomputing*, Vol. 45, No. 3, pp.277–295.

Choi, S.C. and Youn, H.Y. (2012) 'Dynamic hybrid replication effectively combining tree and grid topology', *Journal of Supercomputing*, Vol. 59, No. 3, pp.1289–1311.

Doğan, A. (2009) 'A study on performance of dynamic file replication algorithms for real-time file access in data grids', *Future Generation Computer Systems*, Vol. 25, No. 8, pp.829–839.

Gill, N.K. and Singh, S. (2016) 'A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers', *Future Generation Computer Systems*, Vol. 65, No. 1, pp.10–32.

Grozev, N. and Buyya, R. (2014) 'Inter-cloud architectures and application brokering: taxonomy and survey', *Software-Practice & Experience*, Vol. 44, No. 3, pp.369–390.

Hu, D., Cao, Q. et al. (2010) 'The study on schedule policy of data transferring in storage network', *International Journal of High Performance Computing and Networking*, Vol. 6, Nos. 3–4, pp.208–212.

Inacio, E.C. and Dantas, M.A.R. (2014) 'A survey into performance and energy efficiency in HPC, cloud and big data environments', *International Journal of Networking and Virtual Organisations*, Vol. 14, No. 4, pp.299–318.

Khanli, L.M., Isazadeh, A. et al. (2011) 'PHFS: a dynamic replication method, to decrease access latency in the multi-tier data grid', *Future Generation Computer Systems*, Vol. 27, No. 3, pp.233–244.

Laatikainen, G., Mazhelis, O. et al. (2016) 'Cost benefits of flexible hybrid cloud storage: mitigating volume variation with shorter acquisition cycle', *Journal of Systems and Software*, Vol. 122, No. 1, pp.180–201.

Long, S-Q., Zhao, Y-L. et al. (2014) 'MORM: a multi-objective optimized replication management strategy for cloud storage cluster', *Journal of Systems Architecture*, Vol. 60, No. 2, pp.234–244.

Mansouri, N. (2016) 'QDR: a QoS-aware data replication algorithm for data grids considering security factors', *Cluster Computing*, Vol. 19, No. 3, pp.1071–1087.

Mokadem, R. and Hameurlain, A. (2015) 'Data replication strategies with performance objective in data grid systems: a survey', *International Journal of Grid and Utility Computing*, Vol. 6, No. 1, pp.30–46.

Nagarajan, V. and Mohamed, M.A.M. (2017) 'A prediction-based dynamic replication strategy for data-intensive applications', *Computers and Electrical Engineering*, Vol. 57, No. 2, pp.281–293.

Nukarapu, D.T., Tang, B. et al. (2011) 'Data replication in data intensive scientific applications with performance guarantee', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 8, pp.1299–1306.

Peng, X. and Guofeng, Y. (2013) 'DCSP-MC: dependable cloud-based storage platform for mobile computing', *International Journal of Networking and Virtual Organisations*, Vol. 12, No. 2, pp.133–148.

Preis, M. and Seitz, J. (2012) 'Challenges and conflicts integrating heterogeneous data warehouses in virtual organisations', *International Journal of Networking and Virtual Organisations*, Vol. 11, Nos. 3–4, pp.329–344.

Ren, Y., Shen, J. et al. (2015) 'Mutual verifiable provable data auditing in public cloud storage', *Journal of Internet Technology*, Vol. 16, No. 2, pp.317–323.

Savi, M., Fratini, R. et al. (2015) 'Performance evaluation of video server replication in metro/access networks', *Computer Networks*, Vol. 93, No. 1, pp.96–110.

Shamsi, J., Khojaye, M.A. et al. (2013) 'Data-intensive cloud computing: requirements, expectations, challenges, and solutions', *Journal of Grid Computing*, Vol. 11, No. 2, pp.281–310.

Song, L., Wei, J. et al. (2016) 'Identity-based storage management and integrity verify protocol for secure outsourcing in multi-cloud', *Concurrency Computation*, Vol. 28, No. 6, pp.1930–1945.

Su, Q. (2014) 'Research of instrument to measure customer perceived service quality of mobile data services', *International Journal of Networking and Virtual Organisations*, Vol. 14, Nos. 1–2, pp.176–196.

Sun, D-W., Chang, G-R. et al. (2012) 'Modeling a dynamic data replication strategy to increase system availability in cloud computing environments', *Journal of Computer Science and Technology*, Vol. 27, No. 2, pp.256–272.

Szabo, C., Sheng, Q.Z. et al. (2014) 'Science in the cloud: allocation and execution of data-intensive scientific workflows', *Journal of Grid Computing*, Vol. 12, No. 2, pp.245–264.

Thompson, P. and Lehaney, B. (2008) 'The development and testing of a prototype to link peer-to-peer computing with databases of different types', *International Journal of Networking and Virtual Organisations*, Vol. 5, No. 2, pp.190–203.

Tos, U., Mokadem, R. et al. (2015) 'Dynamic replication strategies in data grid systems: a survey', *Journal of Supercomputing*, Vol. 71, No. 11, pp.4116–4140.

Victer Paul, P., Saravanan, N. et al. (2012) 'QoS enhancements for global replication management in peer to peer networks', *Future Generation Computer Systems*, Vol. 28, No. 3, pp.573–582.

Wang, Z., Li, T. et al. (2012) 'A novel dynamic network data replication scheme based on historical access record and proactive deletion', *Journal of Supercomputing*, Vol. 62, No. 1, pp.227–250.

Zeng, Z. and Veeravalli, B. (2014) 'Optimal metadata replications and request balancing strategy on cloud data centers', *Journal of Parallel and Distributed Computing*, Vol. 74, No. 10, pp.2934–2940.

Zhang, Z., Wu, C. et al. (2013) 'A survey on cloud interoperability: taxonomies, standards, and practice', *Performance Evaluation Review*, Vol. 40, No. 4, pp.13–22.