

A Novel Cost-effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres

Wenhao LI^{*}, Yun Yang^{***} (*Corresponding author*), Dong Yuan^{*}

^{*}Faculty of Information and Communication Technologies, Swinburne University of Technology
Hawthorn, Melbourne, VIC 3122, Australia

^{**}School of Computer Science and Technology, Anhui University
Hefei, Anhui 230039, China

E-mail: {wli, yyang, dyuan}@swin.edu.au

Abstract—Nowadays, large-scale Cloud-based applications have put forward higher demand for storage ability of data centres. Data in the Cloud need to be stored with high efficiency and cost effectiveness while meeting the requirement of reliability. While in current Cloud systems, typical 3-replicas data replication strategies are applied for data reliability, in this paper we propose a novel cost-effective dynamic data replication strategy which facilitates an incremental replication method to reduce the storage cost and meet the data reliability requirement at the same time. This replication strategy works very well especially for data which are only used temporarily and/or have a relatively low reliability requirement. The simulation shows that our replication strategy for reliability can reduce the data storage cost in data centres substantially.

Keywords—Data Replication, Data Reliability, Data Centre, Cloud Computing

I. INTRODUCTION

Cloud computing has become one of the most popular paradigms in both academia and industry. As a basic unit of the Cloud [20], data centres contain a large amount of commodity computing and storage facilities which may reach the scale of tens of thousands [2]. These commodity devices are able to provide virtually unlimited computation and storage power in a cost-effective manner to support users' needs.

Data centres have offered the Cloud massive computing and storage capacity, which have greatly promoted the development of high-performance distributed technologies. However, at the same time, the emergence of large-scale Cloud-based applications has also put forward higher demand for the requirement of massive data storage ability in data centres. As more and more Cloud-based applications tend to be data intensive, huge amounts of data are stored in data centres. From application and system providers' point of view, these data may result in huge storage cost. Currently, modern distributed storage systems generally use data replication to support data reliability. For example, data storage systems such as Amazon S3 [3], Google File System [10] and Hadoop Distributed File System [4] all adopt a 3-replicas data replication strategy by default, i.e. store 3 data copies at one time, for the purposes of data reliability. Such a 3-replicas strategy means that to store one gigabyte of data would need three gigabytes of data space and incur two times

additional cost, which has significantly affects the cost effectiveness of the Cloud system. In addition, such huge resource consumption is believed finally passed on to the customers.

Furthermore, we have observed that the typical 3-replicas data replication strategy or any other fixed replica number replication strategy may not be the best solution for data with uncertain reliability duration. Scientific Cloud applications, such as scientific Cloud workflow processes [22] and Cloud MapReduce instances [7, 13], are usually complex and data intensive. They usually contain a large number of tasks. During the execution, large volumes of intermediate data are generated and the amount could be much larger than the original data. However, most of the intermediate data are only aimed for temporary use. For example, in the pulsar searching workflow application presented in [22], all the intermediate data are deleted after having been used, or in the future some of these intermediate data will be stored for later use but it is uncertain for how long the data need to be stored. In this case, fixed number of replicas is unnecessary, and the typical 3-replicas data replication strategy could be an over killer from the reliability perspective. As a matter of fact, in terms of reliability, the storage units in current Cloud storage systems perform well in general. Take Amazon S3 as an example, the average failure rate of a single storage unit (we define "failure" as hard disc fail in this paper) in Amazon S3 is 99.99% (i.e. 4 '9s') over a year, which means that the average annual loss rate of data stored without replication is 0.01%. Such reliability assurance and storage duration is sufficient enough to meet the requirement of most intermediate data in scientific applications without extra data replication. Thus, in order to reduce the storage cost and fully utilise the storage resources in current Cloud systems, a new data replication strategy which is able to adaptively manage the uncertain reliability requirement of data in the Cloud is necessary.

To tackle this issue, in this paper we propose a novel cost-effective dynamic data replication strategy named Cost-effective Incremental Replication (CIR). CIR is a data reliability strategy for Cloud-based applications in which the main focus is for cost-effectively managing the data reliability issue in a data centre. In CIR, an incremental replication approach is proposed for calculating the replica creation time point, which indicates the storage duration that the reliability requirement can be met. By predicting when an

additional replica is needed to ensure the reliability requirement, CIR dynamically increases the number of replicas. In this way, the number of replicas can be minimised, so that the cost-effective dynamic data replication management goal can be reached. To better illustrate our work, we have evaluated the effectiveness of our strategy by comparing it with the typical 3-replica data replication strategy. The result of the evaluation indicates that CIR strategy can substantially reduce the number of replicas in a data centre whilst meeting the reliability requirement, thus the storage cost of the entire storage system can be significantly reduced.

The rest of this paper is organised as follows. The analysis of the problem is stated in Section II. The data storage reliability model for our replication strategy is described in Section III and the details of the replication strategy are presented in Section IV. The evaluation of our strategy comparing with the typical 3-replica data storage strategy is illustrated in Section V. The related works on data reliability and data replication technologies in data management systems are addressed in Section VI. Some further discussions are mentioned in Section VII. Finally, conclusions and future work are summarised in Section VIII.

II. PROBLEM ANALYSIS

Modern data centres contain a large amount of storage units. These storage units all have certain life spans. The probability of hardware failure, hence data loss, increases according to the storage duration of data. In classical theories [17, 21], the relationship between failure rate and storage duration follows the exponential distribution with failure rate λ , which equals to the expected number of failures of a storage unit in a certain time: $Reliability = 1 - F(x) = e^{-\lambda T}$, where $F(x)$ is the exponential cumulative distribution function.

According to Amazon, the average failure rate of storage units in the S3 storage system is 0.01% over one year. By substituting this failure rate into the equation above, we can estimate that one replica can be stored with higher reliability assurance of 99.999% (i.e. 5 '9s') for about 36 days (more than 1 month) and 99.9999% (i.e. 6 '9s') for about 3 days. This indicates that for a considerable amount of data which only needs short storage duration, the reliability assurance from a single storage unit is expected to be high enough to meet the reliability requirement, and extra replicas are not necessary and wasteful.

Now we apply this principle into a more general scenario: for data with uncertain data storage duration, a minimum number of replicas are available to meet the reliability requirement of the data for a certain period of time, and additional replicas can be created dynamically when necessary. Such a principle can be recursively applied to all the data, and the reliability requirement can always be met.

Therefore, there are two issues which need to be addressed when we put the idea into practice:

- First, given a data centre is composed of various kinds of storage units with each storage unit at a different age,

naturally, each storage unit has a different reliability property, which indicates that the probability of hardware failure and data loss is different. Due to the heterogeneous nature of these storage units, we need to build a reliability model for storage units in a data centre, which can accurately indicate the differences in the reliability aspect among these storage units.

- Second, as time goes by, the reliability assurance of replicas decreases, the probability of hardware failure and data loss becomes higher and higher. At a specific time point, current replicas cannot assure the reliability requirement any longer so that more replicas should be created. An efficient method based on the reliability model for identifying the replica creation time points is needed.

In this paper, the issues presented above are addressed in Sections III and IV respectively.

III. DATA STORAGE RELIABILITY MODEL

We propose a data storage reliability model in order to solve the first issue stated in Section II. Assume that the number of storage units be m . The storage component of the whole data centre can be expressed as storage set $SS = \{u_1, u_2, u_3, \dots, u_m\}$.

Based on the storage exponential distribution theory, the reliability of all m storage units in SS can be described by a failure rate set $FRS = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m\}$. By applying FRS , the reliability requirements of all storage units in a data centre can be accurately expressed, in which a lower failure rate represents a higher reliability.

The data storage reliability model demonstrating the relationship between the lower reliability bound X , the number of replicas and the storage duration can be expressed as equation below:

$$X = 1 - \prod_{i=1}^k (1 - e^{-\lambda_i T_k}) \quad (1)$$

In this equation, X is the lower bound of the reliability requirement. k is the number of replicas. T_k is the longest storage duration. The right-hand side of this equation describes the probability that no failure happens during the storage duration of T_k when k data replicas are stored in storage units with failure rates of $\lambda_1, \lambda_2, \dots, \lambda_k$ respectively. By using this equation, our aim is to derive T_k , which indicates the storage duration that k replicas can assure with the reliability requirement X . This reliability model is applicable to various kinds of data storage scenarios that require one or more replicas. However, the number of replicas in real Cloud storage systems is limited. In this paper, we only use a subversion of the reliability model, in which the number of replicas is limited. According to some modern large-scale storage systems [3, 4, 10], in which the number of replicas is set to 3 by default, we only illustrate the situation that the number of replicas is no more than 3.

$$\text{When } k=1, \quad X = e^{-\lambda_1 T_1}$$

$$\text{When } k=2, \quad X = 1 - (1 - e^{-\lambda_1 T_2})(1 - e^{-\lambda_2 T_2})$$

When $k=3$, $X = 1 - (1 - e^{-\lambda_1 T_3})(1 - e^{-\lambda_2 T_3})(1 - e^{-\lambda_3 T_3})$

In order to facilitate the solving process, these equations are transformed to functions as follows.

$$\text{When } k=1, f(T_1) = e^{-\lambda_1 T_1} / X \quad (2)$$

$$\text{When } k=2, f(T_2) = X a^{\lambda_1 + \lambda_2} - a^{\lambda_1} - a^{\lambda_2} + 1, \text{ where } a = e^{T_2} \quad (3)$$

$$\text{When } k=3, f(T_3) = X b^{(\lambda_1 + \lambda_2 + \lambda_3)} - b^{(\lambda_1 + \lambda_2)} - b^{(\lambda_1 + \lambda_3)} - b^{(\lambda_2 + \lambda_3)} + b^{\lambda_1} + b^{\lambda_2} + b^{\lambda_3} - 1, \text{ where } b = e^{T_3} \quad (4)$$

IV. COST-EFFECTIVE INCREMENTAL REPLICATION (CIR) STRATEGY

As mentioned in Section II, in this section we introduce the details of CIR based on the reliability model. The idea of

CIR is to use the minimum number of replicas while meeting the data reliability requirement.

Due to the uncertainty of the data storage durations, it needs to decide how many replicas are sufficient to meet the reliability requirement. Initially, the minimum data replica number is bound to 1 by default, i.e. only the original data will be stored and no extra replicas will be made at the beginning of a data storage instance. When time goes by, more replicas need to be incrementally created at certain time points to maintain the reliability assurance. Based on the reliability model in Section III, by solving reliability functions (2), (3) and (4) separately, the time points for replica creation can be determined, which indicate when the current number of replicas cannot assure the data reliability requirement any longer and a new replica should be created. At the beginning of each data storage instance or when the latest replica creation time point reaches, a process maintained by the storage system for calculating the replica creation time points is activated.

Cost-effective Incremental Replication strategy:

Input: FRS : failure rate set $FRS = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m\}$

```

1. Start{
2.    $RCTP <- 0$ ; // initialise the record of next replica creation time point
3.   while (a data storage instance be launched || the replica creation time point be reached){
4.      $D <-$  get the data which activated this process
5.      $X <-$  get the reliability requirement of  $D$ 
6.     if ( $D.k=1$ ) { // if new instance
7.        $SU <-$  get the storage unit for  $D$  //  $D$  has only one original replica now
8.        $\lambda_1 <-$  get the failure rate of  $SU$  from  $FRS$ ;
9.       solve function  $f(T_1) = e^{-\lambda_1 T_1} / X$ ; // function (2)
10.       $T_1 <-$  find the positive real root of  $f(T_1)$ ; // obtain the storage duration
11.       $RCTP <-$  current time +  $T_1$ ; // the first replica creation time point
12.     }
13.     if ( $D.k=2$ ) { // if first replica creation time point reaches
14.        $SU <-$  get the storage units for  $D$  // two storage units that store replicas of  $D$ 
15.        $\lambda_1, \lambda_2 <-$  get the failure rate of  $SU$  from  $FRS$ ;
16.       solve function  $f(T_2) = X a^{\lambda_1 + \lambda_2} - a^{\lambda_1} - a^{\lambda_2} + 1$ ; // function (3)
17.        $a <-$  find the positive real root of  $f(a)$ ;
18.        $T_2 <- \ln a$ ; // solve the function in two steps and obtain the storage duration
19.        $RCTP <-$  current time +  $T_2$ ; // the second replica creation time point
20.     }
21.     if ( $D.k=3$ ) { // if second replica creation time point reaches
22.        $SU <-$  get the storage units for  $D$  // three storage units that store replicas of  $D$ 
23.        $\lambda_1, \lambda_2, \lambda_3 <-$  get the failure rate of  $SU$  from  $FRS$ ;
24.       solve function  $f(T_3) = X b^{(\lambda_1 + \lambda_2 + \lambda_3)} - b^{(\lambda_1 + \lambda_2)} - b^{(\lambda_1 + \lambda_3)} - b^{(\lambda_2 + \lambda_3)} + b^{\lambda_1} + b^{\lambda_2} + b^{\lambda_3} - 1$ ;
25.        $b <-$  find the positive real root of  $f(b)$ ; // function (4)
26.        $T_3 <- \ln b$ ;
27.        $RCTP <-$  current time +  $T_3$ ; // the time point that the storage instance ends
28.     }
29.     if ( $k \neq 3$ ) { // create one more replica for the data if current replica number is lower than 3
30.        $U <-$  get a new storage unit for  $D$ ;
31.       create a new replica of  $D$  in  $U$ ;
32.        $D.k <- D.k + 1$ ; // update the replica counter
33.        $D.RCTP <- RCTP$ ; // update the replica creation time point of  $D$ 
34.     }
35.   }
36. }

```

Figure 1. Pseudo code of CIR strategy

The pseudo code of CIR is shown in Figure 1. We take an example to illustrate how CIR works. Consider the most common case, a set of data D is newly uploaded or

generated, the reliability requirement is initialised to a certain value, e.g. 99.999% (i.e. 5 '9s'). At the beginning of the storage instance, only the original replica of the data

stores in the system, the storage duration of this 1-replica state and the first replica creation time point is derived by solving function (2) in Section III ((Lines 6-11): Assume that the storage unit u_1 has a failure rate of λ_1 , by solving function (2), the first replica creation time point is T_1 from the current time. When the first replica creation time point is reached, the process for calculating the second time point will be invoked. Similarly, by solving function (3) in Section III (Lines 12-18), T_2 can be obtained and the second replica creation time point is T_2 from the current time. This process continues until the data has been changed or the maximum number of replicas is reached or data loss happens. At this stage of the process, there are at maximum of three replicas stored in the system. Then, at the end, by solving function (4) in Section III (Lines 19-25), the storage duration of the “3-replica stage” can be derived, In which T_3 is the storage duration of the data with 3 replicas, after T_3 from current time, the reliability requirement X cannot be met any longer.

In the case of data loss which is another important part in data management area, the corresponding research is beyond the scope of this paper.

V. EVALUATION

In this section we describe the simulation for evaluating CIR in detail. We have implemented CIR in an experimental environment and compared CIR with the typical 3-replica data storage strategy from the aspects of storage efficiency and storage cost. Through the comparison of two replication strategies, the advantages of CIR can be clearly seen.

In the implementation of the simulation, several parameters are needed:

- Instance number: it is the number of data instances in the simulation. In the simulation we consider a new data uploading or updating event to be a data storage instance. The number of instances indicates the accuracy of the result, and also provides the upper bound for the number of replicas.
- Storage unit number m : it is the size of the failure rate set FRS . In the simulation we need this parameter to generate the failure rate set.
- Reliability: it is the lower bound of the reliability requirement. The storage system should ensure the reliability of data not lower than this bound.
- Storage duration: it is the storage duration of data from the time it is generated or uploaded to the time it is removed (or no longer needed).
- Price: it is the price for storing the unit amount of data for the unit time without replication.

In order to obtain simulation result that reflect the reality, we have investigated the pricing model and reliability indicators of Amazon S3 storage system and the parameters of the simulation are set accordingly. Table 1 shows the parameter settings of the simulation. First, in Amazon S3, the reliability assurances provided by the system are 99.99% (i.e. 4 ‘9s’) over a year by reduce redundancy storage and

99.99999999% (i.e. 11 ‘9s’) over a year by standard storage. CIR is able to provide more flexible reliability assurances between the two. We divide such reliability into 8 discrete values, which can be identified as 4 ‘9s’ to 11 ‘9s’. As mentioned in Section II, the average failure rate of storage units in Amazon S3 is 0.01% over a year. Therefore, we assume that the failure rates of storage units follow the normal distribution with the mean value of 0.01%.

TABLE I. PARAMETER SETTINGS IN THE SIMULATION

Instance Number		10000
Storage Unit Number		1000
Reliability Assurance		From 99.99% to 99.99999999%
Price	<i>One replica</i>	\$0.0183 per GB*month
	<i>Two replicas</i>	\$0.0366 per GB*month
	<i>Three replicas</i>	\$0.055 per GB*month

Second, in reality, the pricing of storage service is subject to many factors including storage location, number of replicas, storage duration and size of data, etc. In addition, due to commercial in confidence, the detailed information about the storage cost is not easy to obtain. However, we can deduce the price of storage service from real storage systems. The lowest price of US-standard storage in Amazon S3 is \$0.055 per GB*month. Assuming that the Amazon S3 price be derived by using the typical 3-replica strategy and generated only from the storage cost, the pricing of this simulation can then be derived, Namely, the price for using only one replica is set to \$0.0183 per GB*month (i.e. one-third) and the price for using two replicas is set to \$0.0366 per GB*month (i.e. two-thirds).

To obtain a representative result without losing generality, the replication process in this simulation applies random selection for a new storage unit. This reasonable assumption is to avoid the bias of storage unit selection on the result. By applying random selection, CIR stores the newly generated replica in a randomly selected storage unit.

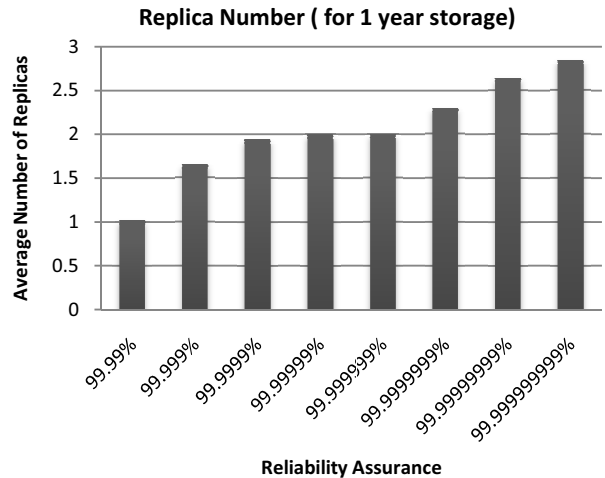


Figure 2. The average replica number for saving data for one year

TABLE II. EXPECTED STORAGE DURATION

Reliability Assurance	Expected Storage Duration (year)
99.99% (4 '9s')	625.187
99.999% (5 '9s')	266.237
99.9999% (6 '9s')	119.871
99.99999% (7 '9s')	54.046
99.999999% (8 '9s')	23.699
99.9999999% (9 '9s')	10.683
99.99999999% (10 '9s')	5.240
99.999999999% (11 '9s')	2.444

The simulation was conducted under different operating conditions with reliability requirement and storage duration. In the simulation, 10000 storage instances were executed over one year on the storage system of 1000 storage units, and the maximum replica number constraint is set to three. Fig. 2-4 have shown the result of the simulation comparing CIR with the typical 3-replica strategy.

Fig. 2 and Table 2 show some of the statistical outcome of the CIR simulation. Fig. 2 shows the average replica numbers for storing data for one year. The result shows that the average number of replica ranges from around 1.02 to 2.85 and this number increases with the growth in reliability assurance. While the 3-replica strategy stores 3 replicas from the beginning of a data instance for all types of data, CIR reduces the average replica number substantially, especially when the reliability assurance is lower. Table 2 shows the expected data storage duration with certain reliability assurance. The result shows that the storage duration varies widely according to the reliability assurance. It ranges from several hundred years to several years and drops quickly while the reliability requirement increases. Providing the highest reliability assurance, the expected

storage duration of data is only about 2.4 years, which seems somewhat lower than the demand of data with high reliability requirement. However, the result still seems good as other storage durations are much longer. From the results of Fig. 2 and Table 2, we can claim that CIR can provide good storage duration and lower storage resource consumption whilst maintaining the satisfactory reliability.

Fig. 3 shows the ratio of storage cost generated by CIR and the typical 3-replica strategy over different storage durations. This figure is obtained by comparing the cumulative storage space used by two strategies. Fig. 3 shows a similar consistent result with Fig. 2 that CIR consumes less storage resources when the reliability requirement of data is lower. Furthermore, it also indicates the storage space utilisation differences between the two strategies. The result shows that CIR can reduce the storage cost and increase the resource utilisation in data centres meeting all reliability requirements, and the outcome is much better especially when the data have lower reliability requirements and are only stored for shorter durations. As CIR incrementally creates replicas one after another only when current replicas cannot assure the reliability requirement, it always demands less storage space than the typical 3-replica strategy. Additionally, CIR occupies less storage space when the storage duration is shorter. It could save up to two thirds of the storage cost at the most, comparing with its counterpart.

Fig. 4 shows the storage prices of two strategies. In this figure the storage durations have 3 categories, which are less than 1 month, 6 months and 12 months. The reason is that we wish to show the changes in price under different storage durations more clearly. The result shows that CIR is able to provide a competitive price for data storage, which is as low as 1.8 cents/GB*month, while at the same time Amazon S3 provide the service at the price of 5.5 cents/GB*month. At the same time, CIR can provide flexible prices according to the reliability requirement and storage duration, which is also an advantage comparing with the fixed 3-replica strategy. The result of Fig. 4 that short

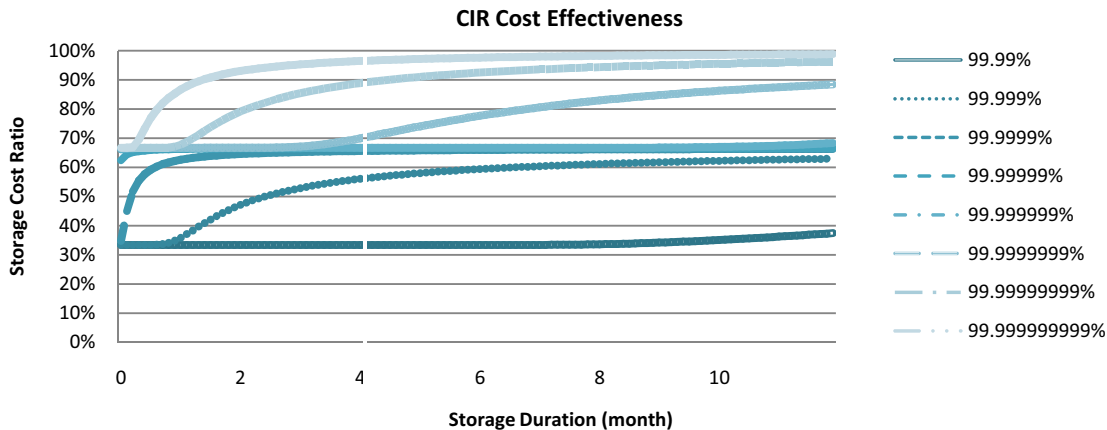


Figure 3. Ratio of storage cost of two strategies

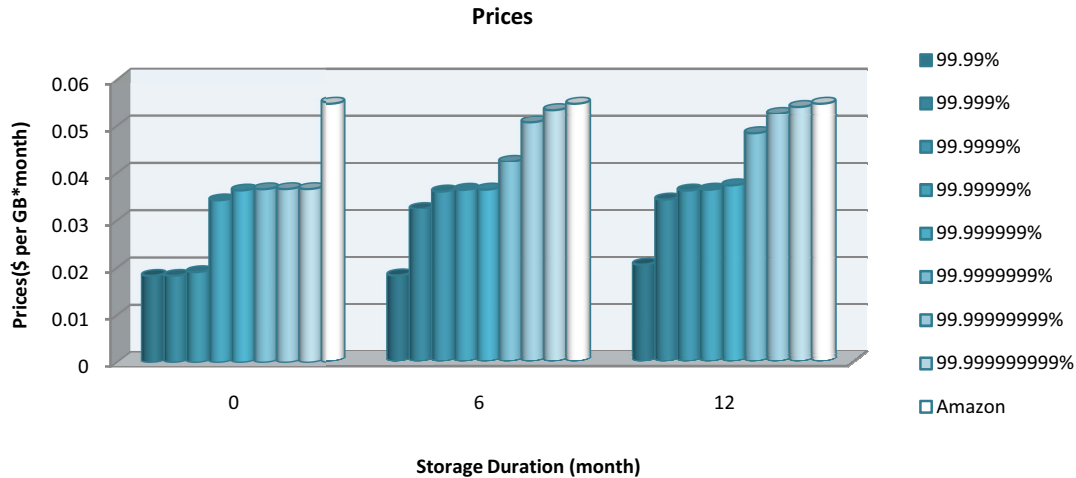


Figure 4. Service prices of two strategies

VI. RELATED WORK

Data reliability is one of the key research issues for today's distributed storage systems. It is defined as "a state that exists when data is sufficiently complete and error free to be convincing for its purpose and context" [15]. In order to ensure the reliable processing and storage of data, much research has been done in different aspects, such as fault detection and recovery [9], resource replication [1], data replication [13, 16, 19], etc. Among these researches, data replication has emerged as one of the most important topics and data replication technologies have been widely used in industry [3, 4, 10]. Data replication has been a research topic of long standing in distributed systems [5]. As early as 1980s, a study on redundant array of independent disks (RAID) was proposed [16]. More recently, the emergence of large-scale Web applications such as social network services (SNS) and large-scale distributed systems such as Grid and Cloud has made data replication becoming a research hot pot once again. Some summaries on this general topic are available in [5, 18]. Aiming at providing high data reliability, [11] proposes a RAID-like stackable file system, which can be mounted on top of any combination of other file systems including network, distributed file systems, etc. Some data replication approaches in Grid computing systems have been studied in [8, 14], while some other approaches for Cloud computing systems have been studied in [12, 23].

Although some of existing researches on data reliability and replication technologies have considered the cost factor in their algorithms, none of them has considered the impact caused by the varied data storage duration and the duration uncertainty. As a matter of fact, different storage durations of data would greatly affect corresponding reliability requirements and thus the number of replicas needed. In the case of scientific applications in the Cloud, large amounts of intermediate data are primarily generated only for short-term purpose. The minimisation of replicas for this kind of data can greatly reduce the storage cost. The CIR strategy

proposed in this paper has taken the factors of data storage variation and the duration uncertainty into account. This strategy is able to balance the trade-off of reliability requirement and storage cost, minimise the replicas while meeting the reliability requirement, and thus manage massive data in data centres more cost effectively.

VII. DISCUSSION

In the simulation of CIR, we applied the reliability parameters and pricing model of Amazon S3. However, there are many other storage systems, which have different reliability indicators in these areas due to their different application environments. For example, a Google report published in 2006 [6] showed that Google cluster experiences approximately ten disk outages out of 10000 disks per day under the heavy workload of running MapReduce processes. We can deduce from this information that the average failure rate of disks approximately equal to 0.001 over a day, which is much higher than the failure rate of Amazon S3 storage units. Under such a high failure rate, current CIR strategy (with the maximum 3 replica number constraint) could only provide storage duration of less than 2 months with the reliability assurance of 99.99% (i.e. 4 '9s'). Such performance may not meet the requirement of some applications in the Cloud. In order to solve this problem, more than 3 replicas are probably needed to provide high reliability and enough storage duration. However, CIR is generic and would still have a better performance than any fixed n-replica storage strategy.

VIII. CONCLUSION AND FUTURE WORK

In this paper we proposed a novel cost-effective dynamic data replication strategy, named Cost-effective Incremental Replication (CIR) in Cloud data centres. This strategy applies an incremental replication approach to minimise the number of replicas while meeting the reliability requirement in order to reach the cost-effective

data replication management goal. In the simulation for evaluation, we demonstrate that our strategy can reduce the data storage cost substantially, especially when the data are only stored for a short duration or have a lower reliability requirement. Our study has provided an alternative option for the Cloud storage system, which treats the cost as the top priority. However, there are still some works not well studied in this paper. First, how to model and simulate storage unit aging process is very critical for the implementation of CIR in real systems. Second, while meeting the data reliability requirement, the trade-off between cost and performance is always an issue worth studying. These issues will be done in the near future.

ACKNOWLEDGEMENTS

The research work reported in this paper is partly supported by Australian Research Council under Linkage Project LP0990393 and Discovery Project DP110101340, as well as State Administration of Foreign Experts Affairs of China Project P201100001.

REFERENCES

- [1] J. Abawajy, "Fault-tolerant scheduling policy for grid computing systems," in International Parallel and Distributed Processing Symposium, pp. 238–244, 2004.
- [2] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *Computer Communication Review*, vol. 38, pp. 63-74, 2008.
- [3] Amazon. (2008). Amazon simple storage service (Amazon S3). Available: <http://aws.amazon.com/s3>
- [4] D. Borthakur. (2007). The Hadoop distributed file system: Architecture and design. Available: http://hadoop.apache.org/common/docs/r0.18.3/hdfs_design.html
- [5] C. Dabrowski, "Reliability in grid computing systems," *Concurrency and Computation: Practice and Experience*, vol. 21, pp. 927-959, 2009.
- [6] J. Dean, "Experiences with MapReduce, an abstraction for large-scale computation," in International Conference on Parallel Architectures and Compilation Techniques pp. 1 - 1, 2006.
- [7] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, pp. 107-113, 2008.
- [8] M. Deris, J. Abawajy, and H. Suzuri, "An efficient replicated data access approach for large-scale distributed systems," in IEEE International Symposium on Cluster Computing and the Grid, pp. 588–594, 2004.
- [9] R. Duan, R. Prodan, and T. Fahringer, "Data mining-based fault prediction and detection on the Grid," in IEEE International Symposium on High Performance Distributed Computing, pp. 305–308, 2006.
- [10] S. Ghemawat, H. Gobioff, and S. Leung, "The Google file system," in ACM Symposium on Operating Systems Principles pp. 29 - 43, 2003.
- [11] N. Joukov, A. Rai, and E. Zadok, "Increasing distributed storage survivability with a stackable RAID-like file system," in IEEE International Symposium on Cluster Computing and the Grid, pp. 82–89, 2005.
- [12] S. Y. Ko, I. Hoque, B. Cho, and I. Gupta, "On availability of intermediate data in cloud computations," in Conference on Hot Topics in Operating Systems, pp. 6-6, 2009.
- [13] S. Y. Ko, I. Hoque, B. Cho, and G. I., "Making cloud intermediate data fault-tolerant," in ACM Symposium on Cloud computing, pp. 181-192, 2010.
- [14] M. Lei, S. Vrbsky, and Q. Zijie, "Online grid replication optimizers to improve system reliability," in IEEE International Symposium on Parallel and Distributed Processing, pp. 1–8, 2007.
- [15] S. L. Morgan and C. Waring, G. (2004). Guidance on testing data reliability. Available: <http://www.ci.austin.tx.us/auditor/downloads/testing.pdf>
- [16] D. Patterson, G. Gibson, and R. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in ACM SIGMOD International Conference on the Management of Data, pp. 109–116, 1988.
- [17] E. B. Richard, "A bayes explanation of an apparent failure rate paradox," *IEEE Transactions on Reliability*, vol. 34, pp. 107-108 1985.
- [18] Y. Saito and M. Shapiro, "Optimistic replication," *ACM Computing Surveys*, vol. 37, pp. 42-81, 2005.
- [19] S. Takizawa, Y. Takamiya, H. Nakada, and S. Matsuoka, "A scalable multi-replication framework for data grid," in Symposium on Applications and the Internet Workshops, pp. 310–315, 2005.
- [20] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, pp. 50-55, 2009.
- [21] J. W. Young, "A first order approximation to the optimal checkpoint Interval," *Communications of the ACM*, vol. 17, pp. 530-531, 1974.
- [22] D. Yuan, Y. Yang, X. Liu, G. Zhang, and J. Chen, "A data dependency based strategy for intermediate data storage in scientific cloud workflow systems," *Concurrency and Computation: Practice and Experience*, 2010.
- [23] Q. Zheng, "Improving mapreduce fault tolerance in the cloud," in IEEE International Symposium on Parallel & Distributed Processing, Workshops and PhD Forum, pp. 1 - 6, 2010.