Global Reliability Evaluation for Cloud Storage Systems with Proactive Fault Tolerance

Jing Li[1], Mingze Li[1], Gang Wang[1(✉)], Xiaoguang Liu[1], Zhongwei Li[1(✉)], and Huijun Tang[2]

[1] Nankai-Baidu Joint Lab,
College of Computer and Control Engineering and College of Software,
Nankai University, Tianjin 300350, China
{lijing,limingze,wgzwp,liuxg,lizhongwei}@nbjl.nankai.edu.cn
[2] Qihoo 360 Technology Company, Beijing 100621, China
tanghuijun@360.cn

Abstract. In addition to the traditional reactive fault-tolerant technologies, such as erasure codes and replication, proactive fault tolerance can be used to improve the system's reliability significantly. To the best of our knowledge, however, there is no previous publications on the reliability of such a cloud storage system except for those on RAID systems. In this paper, several Markov-based models are respectively proposed to evaluate the reliability of the cloud storage systems with/without proactive fault tolerance from the system perspective. Since proactive measure should be coupled with some reactive measure to ensure the systems reliability, the reliability model for such a system will be very intricate. To facilitate model building, we propose the *basic state transition unit* (BSTU), to describe the general pattern of state transition in the proactive cloud storage systems. BSTU serves as the generic "brick" for building the overall reliability model for such a system. Using our models, we demonstrate the benefits that proactive fault tolerance has on a system's reliability, and also estimate the impacts of some system parameters on it.

Keywords: Cloud storage system · Proactive fault tolerance · Reactive fault tolerance · Global reliability model · Rack aware replication

1 Introduction

Replication and erasure codes are the traditional means by which cloud storage systems are made reliable. If some replicas are lost due to node failures, other survived replicas can be used to restore them to maintain the same level of reliability. This is a typical reactive fault-tolerant manner. Recently, some researchers have proposed deploying proactive fault tolerance on storage systems [1–7], which undertakes to predict failures and handle them in advance. The main advantage of proactive fault tolerance is the early head start for the rebuild while the dying drive/node is still alive.

G. Wang et al. (Eds.): ICA3PP 2015, Part IV, LNCS 9531, pp. 189–203, 2015.
DOI: 10.1007/978-3-319-27140-8_14

At present, the research work on the reliability for cloud storage system, has mainly been focused on a single file or an independent peer, and does not consider the rack aware placement strategy which is common in the current systems. Moreover, as said in [8], since proactive fault tolerance can not avoid failure completely, it is also necessary to be coupled with a reactive fault tolerance measure to ensure a system's reliability. Therefore the state transitions of the system are very intricate. So far, there is no relevant research on reliability for such a cloud storage system except for those on RAID-5/6 systems [7,8].

In this work we use Markov chain based methods to respectively present several global reliability evaluation models for the cloud storage systems with/without proactive fault tolerance. To construct the complex and intractable reliability models for systems with proactive fault tolerance, we propose a generic "brick" – basic state transition unit (BSTU), to describe the fundamental laws of state transitions in the model. Using our reliability models, we deduce the actual reliability gaps between systems with different replication factors, demonstrate the effects that proactive fault tolerance has on the reliability of a system given a parameterized sensitivity, and also estimate the impacts of other system parameters.

2 Related Work

Many studies [9,10] were on the distributed file system based on P2P networks, but their focuses were on the reliability of a single file or an independent peer, rather than on the systematic study of data reliability. Based on Markov model, Lian *et al.* [11] provided an analytical framework to reason and quantify the impact of replica placement policy on storage system reliability. However they did not consider the real placement strategy that replicas are random distributed across racks and nodes. And, a few studies had used probabilistic methods [12,13] to estimate the reliability of the system, however, the probability of data loss in these works took into account only the time spent by the system in failure-free state and ignored the rebuild times, which made their results not reflecting a realistic picture of the system's reliability. Moreover, KK Rao *et al.* [14] presented Markov models to determine the reliability of the high-end enterprise storage systems, which were realized through networked storage nodes. Our work is similar to KK Rao's, however we focus on the effect of failure prediction for rack aware replication, which is a common placement policy used in the current systems.

For storage systems with proactive fault tolerance, there are only few studies focusing on their reliability. Eckart *et al.* [8] first used Markov models to rigorously demonstrate the effects that failure prediction has on a system's MTTDL (mean time to data loss). They only devised models for a single hard drive, RAID-1, and RAID-5 systems. In our previous work [7], we extended this study into RAID-6 systems. However, to our best knowledge, there is no study on such cloud storage systems based on replication schemes.

3 Reliability Models for Reactive Cloud Storage Systems

Assuming independent exponential probability distributions for failure and repair of individual storage nodes, we build the Markov reliability evaluation models for traditional reactive cloud storage systems.

Consider cloud storage systems with r racks and each rack having n nodes, replicas of each data block are spread across nodes and racks. The systems are block-based storage systems, and maintain two invariants: first, no two replicas of a data block are stored on the same node; and second, replicas of a data block must be found on at least two racks. We use c to denote storage capacity of each node, λ to denote failure rate of a node, and μ to denote rebuild rate of a failure.

3.1 Reactive with Replication Factor Two

In a system with replication factor 2, every user data block is replicated 2 times and the 2 replicas must be stored on two separate racks. We assume that there have been enough data blocks in the system and they are fully dispersed, such that any pair of nodes from different racks share at least one data block. A system with replication factor 2 can tolerate any single node failure and a portion of t node failures for $2 \leq t \leq n$. System data loss occurs if and only if node failures occur on two or more racks.

We build the reliability model for a system with replication factor 2 as shown in Fig. 1. In this model, there are $n+2$ states in total, which fall into three types: (1) S_0 represents a completely healthy state, during which there is no node failure at all; (2) S_i, where $1 \leq i \leq n$, represent the degraded states, during which i nodes have already failed and all the i failures are on the same rack; (3) DL represents the absorbing state at which point true data loss occurs.

The system begins in the healthy state S_0, and will transfer to state S_1 with the rate of $rn\lambda$, when a storage node failure occurs. During the state S_1, the system initiates a rebuild process to repair the failure, then the system can transfer to any one of three states: S_0, with the rate of μ, if the failure has been

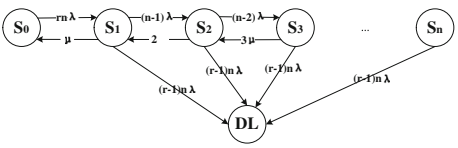

Fig. 1. Markov model for replication factor two without failure prediction.

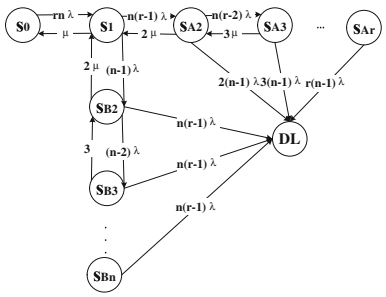

Fig. 2. Markov model for replication factor three without failure prediction.

repaired at rate μ; S_2, with the rate of $(n-1)\lambda$, if another failure occurs on the same rack with the existing one before the rebuild process is finished; or DL, with the rate of $(r-1)n\lambda$, if a new failure occurs on the other rack before the rebuild process is finished, and at this point, since there have happened some node failures on two different racks, the system goes into data loss state.

Similarly, during the state S_i $(1 < i < n)$, the system can transfer to any one of three states: S_{i-1}, with the rate of $i\mu$, if one of the i failures has been repaired; S_{i+1}, with the rate of $(n-i)\lambda$, if a new failure occurs on the same rack with the existing ones; or DL, with the rate of $(r-1)n\lambda$, if a new failure occurs on a previously healthy rack. During the state S_n, when all nodes on some rack have failed, the system is already in the critical condition and can transfer to either of two states: S_{n-1}, with the rate of $n\mu$, if one of the n failures has been repaired; DL, with the rate of $(r-1)n\lambda$, if a new failure occurs.

3.2 Reactive with Replication Factor Three

In a system with replication factor 3, every user data block is replicated 3 times, and then the 3 replicas need to be stored in some 3 nodes out of which 2 nodes are on a same rack and 1 node is on another rack. We assume that there have been enough data blocks in the system and they are fully dispersed, such that any set of 3 nodes, in which 2 nodes are on a single rack and 1 node is on another rack, shares replicas of at least one data block.

The system can tolerate more than 2 failures without data loss in the following two cases: first, all the node failures happen to be on a single rack; or second, all the node failures are on different racks which means that there is up to one node failure on each rack. Therefore, the system can tolerate all the single and double node failures and a portion of t node failures for $3 \leq t \leq max(n, r)$.

We construct the reliability model for systems with replication factor 3 as shown in Fig. 2. In this model, there are $r + n + 2$ states in total, which can be classified into five types: (1) S_0 represents a completely healthy state; (2) S_1 denotes the degraded system state, during which 1 node has already failed; (3) S_{Ai}, where $2 \leq i \leq r$, denote the degraded system states, during which i nodes have already failed and they are separately on i different racks; (4) S_{Bi}, where $2 \leq i \leq n$, denote the degraded system states, during which i nodes have already failed and all of them are on a single rack; (5) DL is the absorbing state at which point data loss occurs.

The system begins in the healthy state S_0, and can transfer to state S_1 with the rate of $rn\lambda$, when a node failure occurs in the system. During the state S_1, the system can transfer to any one of three states: S_0, with the rate of μ, if the failure has been repaired; state S_{A2}, with the rate of $n(r-1)\lambda$, if a new failure occurs on the other rack; or S_{B2}, with the rate of $(n-1)\lambda$, if a new failure occurs on the same rack with the existing one.

During the state S_{Ai}, the system can transfer to any of three states: state $S_{A(i+1)}$, with the rate of $n(r-i)\lambda$, if a new failure occurs on a completely healthy rack; state $S_{A(i-1)}$, with the rate of $i\mu$, if one of the i failures has been repaired;

or DL, with the rate of $i(n-1)\lambda$, if any one of the i racks, which have already a node failure on them, generates a new failure.

During the state S_{Ar}, when each rack in the system has had one failure on it, the system is already in the critical condition and can transfer to either of two states: $S_{A(r-1)}$, with the rate of $r\mu$, if one of the r failures has been repaired; or DL, with the rate of $r(n-1)\lambda$, if a new failure happens.

During the state S_{Bi}, the system can transfer to either of two states: $S_{B(i+1)}$, with the rate of $(n-i)\lambda$, if a new node failure occurs on the same rack with the existing ones; $S_{B(i-1)}$, with the rate of $i\mu$, if one of the i failures has been repaired; or DL, with the rate of $n(r-1)\lambda$, if a new failure occurs on a previously healthy rack.

During the state S_{Bn}, when all nodes on some rack have failed, the system is already in the critical condition and can transfer to either of two states: $S_{B(n-1)}$, with the rate of $n\mu$, if one of the n failures has been repaired; or DL, with the rate of $n(r-1)\lambda$, if a new failure happens.

4 Reliability Models for Systems with Proactive Fault Tolerance

In addition to the assumption of independent exponential probability distributions for failure and repair of individual storage nodes, we also assume independent exponential probability distributions for node warnings and their repairs, and then construct the Markov reliability models for the proactive cloud storage systems.

4.1 Basic State Transition Unit (BSTU)

Since no node failure predictor can guarantee 100 % accuracy, it is necessary to combine the proactive and reactive fault-tolerant measures to ensure reliability. Therefore, the state transitions in the reliability model for such a system are very intricate. One could not describe the model easily in the traditional way. Therefore, we design a generic "brick" – basic state transition unit (BSTU), by composing which we can derive the complex and intractable reliability models for cloud storage systems with proactive fault tolerance.

Using the BSTU shown in Fig. 3, we want to describe the fundamental laws of state transitions in the Markov models. The symbol p represents the failure detection rate of predictors deployed in the systems, γ represents the failure rate of a node warning, and μ represents the rebuild rate of a node failure/warning.

We use P_{ij} to represent the degraded states during which i nodes have actually failed and j nodes are currently predicted imminent failures. The symbol A_{ij} is used to represent the probability that the system can tolerate the $(i+1)$-th node failure without data loss and the new failure occurs on a previously completely healthy node. We use B_{ij} to represent the probability that the system can survive from the $(i+1)$-th node failure and the new failure evolves out of a warning. And C_{ij} is used to represent the probability that the $(i+1)$-th failure

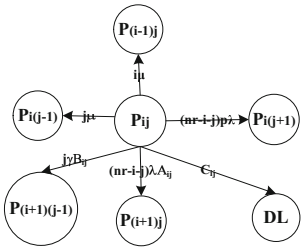

Fig. 3. Basic state transition unit (BSTU) for cloud storage systems with proactive fault tolerance.

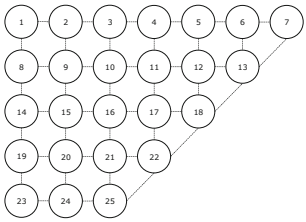

Fig. 4. A reliability model composed by BSTUs for a proactive cloud storage system.

induces system data loss. For simplicity sake, we only draw the outgoing edges and the corresponding transition rates for state P_{ij}, and its incoming edges and their transition rates can be drawn similarly. We can calculate the values of A_{ij}, B_{ij} and C_{ij} for each P_{ij} in some system using combinatorial analyses.

In addition to the basic internal states P_{ij}, there are some boundary states, the transitions of which are similar to the BSTU except for missing some incoming and outgoing edges. For cloud storage systems with proactive fault tolerance, we can construct their reliability models by combining a large number of internal and boundary BSTUs. For example, Fig. 4 is a simple diagram for a reliability model composed by 25 BSTUs for a proactive system, in which the state of data loss is omitted.

Based on Fig. 4, we want to explain some concepts used in the following. The BSTUs numbered 1–7 make up the upper boundary of the model. The ones numbered $1, 8, 14, 19, 23$ make up the left boundary. The ones numbered $7, 13, 18, 22, 25$ make up the right boundary. The ones numbered 23–25 make up the lower boundary. For the proactive systems with replication factor two, the ones numbered 9–12, 15–17, 20, 21 are the internal states. For the proactive systems with replication factor three, the ones numbered 8–13 make up the second layer of the model, and the ones numbered 15–17, 20, 21 are the internal states.

The systems with proactive fault tolerance have the same fault tolerance and characteristics as the traditional reactive systems except for having the ability of failure prediction and pre-warning treatment.

4.2 Proactive with Replication Factor Two

There are five types of state transition units at all, by combining which we can construct the overall reliability model for the proactive storage system with replication factor two.

The model is composed mainly of intact BSTUs, namely internal states. During an internal state P_{ij} ($0 < i < n$ and $0 < j < nr - i$), the system can

transfer to any one of six states: (1) state $P_{(i-1)j}$ with the rate of $i\mu$, if one of the i node failures has been repaired; (2) state $P_{i(j-1)}$ with the rate of $j\mu$, if one of the j node warnings has been repaired; (3) state $P_{(i+1)j}$ with the rate of $(nr-i-j)(1-p)A_{ij}\lambda$, if a healthy node fails, and fortunately it does not induce system data loss; (4) state $P_{i(j+1)}$ with the rate of $(nr-i-j)p\lambda$, if a healthy node is predicted to be facing an impending failure; (5) state $P_{(i+1)(j-1)}$ with the rate of $jB_{ij}\gamma$, if a warning node does not be repaired timely and actually fails at last, however it does not induce system data loss fortunately; (6) state DL with the rate of C_{ij}, if a new failure does induce system data loss.

The value of A_{ij} can be calculated by counting the proportion of healthy-to-fail transitions occurring on the faulty rack. We use x to denote the number of warnings which are on the same rack with the i failures. For some x, $(n-i-x)C_{n-i}^{x}C_{n(r-1)}^{j-x}$ is the number of cases for which one healthy node on the faulty rack fails. The formula $(nr-i-j)C_{nr-i}^{j}$ is the number of cases for which a healthy node in the system fails. So the value of A_{ij} can be calculated as:

$$A_{ij}=\frac{\sum_{x=0}^{min(n-i-1,j)}(n-i-x)C_{n-i}^{x}C_{n(r-1)}^{j-x}}{(nr-i-j)C_{nr-i}^{j}}. \tag{1}$$

We calculate B_{ij} by counting the proportion of warning-to-fail transitions occurring on the faulty rack. We use x to denote the similar thing as in (1). For some x, $xC_{n-i}^{x}C_{n(r-1)}^{j-x}$ is the number of cases for which a previously warning node on the faulty rack fails. And, jC_{nr-i}^{j} is the number of all cases for which a previously warning node in the system fails. So the value of B_{ij} can be calculated as:

$$B_{ij}=\frac{\sum_{x=1}^{min(n-i,j)}xC_{n-i}^{x}C_{n(r-1)}^{j-x}}{jC_{nr-i}^{j}}. \tag{2}$$

The C_{ij} represent the probability that the new failure is not on the faulty rack and incurs system data loss. The formula $(1-p)\lambda(nr-i-j)(1-A_{ij})$ denotes the probability that a failure occurring on a previously healthy node is not predicted by the predictor and induces the system data loss. So the value of C_{ij} can be calculated as:

$$C_{ij}=(1-p)\lambda(nr-i-j)(1-A_{ij})+j\gamma(1-B_{ij}). \tag{3}$$

Besides the internal states, there are other four types of boundary states. They are similar to the intact BSTUs, except for missing some transitions:

(1) when $i=0$ and $0<j<rn$: states P_{ij} make up the upper boundary of the model, during which no node failure happens and just j warnings are predicted by the predictor. Since $i=0$, there is no state $P_{(i-1)j}$ and transition $P_{ij}\rightarrow P_{(i-1)j}$; since the system can tolerate any single-failure, there is no transition $P_{ij}\rightarrow DL$.

(2) when $j = 0$ and $0 < i < n$: states P_{ij} make up the left boundary of the model, during which i node failures have happened and no warning is predicted. Accordingly, there is neither $P_{ij} \rightarrow P_{i(j-1)}$ nor $P_{ij} \rightarrow P_{(i+1)(j-1)}$ transition.
(3) When $i = n$ and $0 < j < (r-1)n$: states P_{ij} make up the lower boundary of the model, during which all nodes on some rack have failed already and j warnings are also predicted. Since the system can tolerate at most n failures, there is neither $P_{ij} \rightarrow P_{(i+1)j}$ nor $P_{ij} \rightarrow P_{(i+1)(j-1)}$ transition.
(4) when $j = nr - i$ and $0 < i < n$: states P_{ij} make up the right boundary of the model, during which i node failures have happened and all the rest nodes in system are also predicted to be soon-to-fail. Since there is no healthy node left, neither $P_{ij} \rightarrow P_{i(j+1)}$ nor $P_{ij} \rightarrow P_{(i+1)j}$ exists.

4.3 Proactive with Replication Factor Three

There are six types of state transition units in the reliability model for the proactive cloud storage systems with replication factor three.

Similarly, most states are intact BSTUs, namely internal states P_{ij}, where $1 < i < max(n, r)$ and $0 < j < nr - i$. They are basically the same as those in the model for replication factor two, except for the values of A_{ij}, B_{ij} and C_{ij}. Accordingly, we only discuss how to calculate the three probabilities here.

There are two cases for which the system survives from i node failures, where $2 < i \leq min(n, r)$: first, all the i node failures are on a single rack; or second, the i node failures are respectively on i different racks. We use q_i to represent the proportion of the first case in the both, and the value of q_i can be calculated as:

$$q_i = \frac{rC_n^i}{(rC_n^i + n^i C_r^i)}. \tag{4}$$

The proportion of the second case is $1 - q_i$. When $min(n, r) < i \leq max(n, r)$, the value of q_i is equal to either 0 or 1.

For the case that the system tolerates the (i+1)-th failure which occurs on a previously healthy node, we use N_s to denote the number of cases for which the new failure is on the same rack with the existing i failures, and use N_d to denote the number of cases for which the new failure is on a previously healthy rack. The value of N_s can be calculated as:

$$N_s = \sum_{x=0}^{min(n-i-1,j)} (n - i - x) C_{n-i}^x C_{n(r-1)}^{j-x} \tag{5}$$

where x denotes the number of warnings which are on the same rack with the i failures.

The value of N_d can be calculated as:

$$N_d = \sum_{x=0}^{min(n(r-i)-1,j)} (n(r-i) - x) C_{n(r-i)}^x C_{i(n-1)}^{j-x}) \tag{6}$$

where x denotes the number of warnings which are on the healthy racks.

Then the value of A_{ij} can be calculated as:

$$A_{ij} = \frac{(q_i N_s + (1 - q_i)N_d)}{(nr - i - j)C_{nr-i}^{j}} \tag{7}$$

Similarly, for the case that the system tolerates the (i+1)-th failure which evolves out of a warning, we use N'_s to denote the number of cases for which a warning on the same rack with the existing failures fails, and use N'_d to denote the number of cases for which a warning on a previously healthy rack fails. The value of N'_s can be calculated as:

$$N'_s = \sum_{x=1}^{min(n-i,j)} xC_{n-i}^{x} C_{n(r-1)}^{j-x} \tag{8}$$

where x denotes the similar thing as in (5).

The value of N'_d can be calculated as:

$$N'_d = \sum_{x=1}^{min(n(r-i),j)} xC_{n(r-i)}^{x} C_{i(n-1)}^{j-x} \tag{9}$$

where x denotes the similar thing as in (6).

Then the value of B_{ij} can be calculated as:

$$B_{ij} = \frac{(q_i N'_s + (1 - q_i)N'_d)}{jC_{nr-i}^{j}}. \tag{10}$$

And, the value of C_{ij} can be calculated as:

$$C_{ij} = (1 - p)\lambda(nr - i - j)(1 - A_{ij}) + j\gamma(1 - B_{ij}). \tag{11}$$

Besides the internal states, there are other five types of boundary states. They are similar to the intact BSTUs, except for missing some transitions:

(1) when $i = 0$ and $0 < j < rn$: states P_{ij} make up the upper boundary of the model, during which no node failure happens and just j warnings are predicted by the predictor. Since $i = 0$, there is no state $P_{(i-1)j}$ and transition $P_{ij} \rightarrow P_{(i-1)j}$; since the system with replication factor 3 can tolerate any double-failure, there is no transition $P_{ij} \rightarrow DL$.
(2) when $i = 1$ and $0 < j < rn - 1$: states P_{ij} make up the second layer of the model, during which only 1 node failure happens and j warnings are predicted by the predictor. There is no transition $P_{ij} \rightarrow DL$ either.
(3) when $j = 0$ and $0 < i < max(n, r)$: states P_{ij} make up the left boundary of the model, during which i node failures have happened and no warning is predicted. Accordingly, there is neither $P_{ij} \rightarrow P_{i(j-1)}$ nor $P_{ij} \rightarrow P_{(i+1)(j-1)}$ transition.

(4) when $i = max(n, r)$ and $0 < j < rn - i$: states P_{ij} make up the lower boundary of the model, during which the system is already in the critical condition and j warnings are also predicted. Since the system can tolerate at most $max(n, r)$ failures, there is neither $P_{ij} \rightarrow P_{(i+1)j}$ nor $P_{ij} \rightarrow P_{(i+1)(j-1)}$ transition.
(5) when $j = nr - i$ and $0 < i < max(n, r)$: states P_{ij} make up the right boundary of the model, during which i node failures have happened and all the rest nodes in system are predicted to happen impending failure. Since there is no healthy node left, neither $P_{ij} \rightarrow P_{i(j+1)}$ nor $P_{ij} \rightarrow P_{(i+1)j}$ transition exists.

5 Experiment and Analyses

Table 1 shows the values of parameters used in our experiments. Typical values for practical systems are used for all parameters, except the sensitivity of node failure predictors, which have not been really used in systems. In our previous work [7], our classification tree prediction models can achieve the failure detection rate (FDR) of 95 % with the mean time in advance (TIA) of near 360 h, and can maintain a FDR above 90 % for the long-term use and for both drive families. In this paper, we have chosen a relatively conservative prediction sensitivity (the FDR of 80 % and the TIA of 360 h) for the failure predictors. Unless otherwise stated, we keep these settings unchanged in the following experiments.

Limited by the memory and computing ability, it is difficult to use Markov models to obtain the reliability values at a large data center scale, even using the best matrix algorithm. Therefore, to obtain the reliability of large scale storage systems (Sect. 5.2), we run simulations of systems using Monte Carlo methods, in which the chronological behavior of a system is simulated [15]. For each setting, we run simulation 100 times, and finally the bootstrap 95 % confidence interval for the time to data loss is computed. However, since the error margins are nearly 10 times less than the average values, to make the compared results more clearly, we only draw the average values of simulations on the figures.

We compare systems with the same effective storage space (excluding the space for redundant data). And, We transform the MTTDL to a useful measure – the

Table 1. The value of system parameters used in our experiments.

Parameter	Meaning	Range
c	storage capacity of each node	12 TB
n	the number of nodes on each rack	15
$1/\lambda$	mean time to failure of storage nodes	100000 h
$1/\mu$	mean time to repair a failure/warning	24 h
p	failure detection rate of failure predictors	80 %
$1/\gamma$	mean time in advance of a node warning	360 h

expected number of data lost event per usable petabyte within one year, by which ones can better understand the reliability gaps between different cloud storage systems.

5.1 Sensitivity of Node Reliability

We assume that there are 40 racks in the systems with replication factor 2, and 60 racks in the systems with replication factor 3, which means that the systems can respectively store 3600 TB user data. Unless otherwise stated, we keep this assumption in the following experiments.

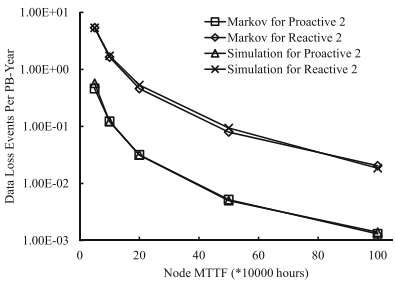

Fig. 5. The change of the reliability of systems with replication factor 2 according to the node reliability.

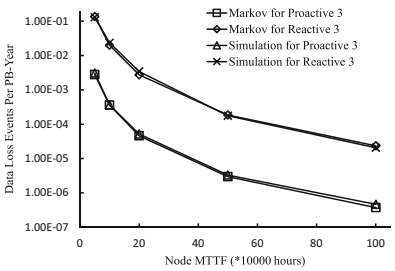

Fig. 6. The change of the reliability of systems with replication factor 3 according to the node reliability.

We can see from Figs. 5 and 6, with the different node MTTF, Markov-based results have a perfect match with the simulation-based values, which verifies the accuracy of our reliability models and the Monte Carlo simulations. In addition, we can learn about the following: first, all the four systems' reliability have be improved hardly with the enhanced node MTTF, which demonstrates the great value of redundancy distributed within nodes; second, the systems with replication factor 3 yield larger gains than the ones with replication factor 2 with the enhanced node MTTF. Specifically, when the node MTTF is doubled with other parameters constant, the reliability of systems with replication factor 2 will decline by three-quarters and the reliability of systems with replication factor 3 will decline by seven-eighths; and third, having the predictor with the FDR of 80 %, both the systems with replication factor 2 and 3 can reduce redundancy within their nodes to decrease the node MTTF nearly by three-quarters, while the reliability of them remain the same.

5.2 Sensitivity of System Size

Figure 7 shows the change of systems reliability, as the effective storage space is varied. We can adjust the storage space size by changing the number of racks in

system. Three observations are evident from Fig. 7: first, for all the four systems, the reliability will be reduced by less than a half for each doubling in capacity and these decrease taper off as systems grow larger; second, having a predictor with the FDR of 80 %, the reliability of system with replication factor 2 can be improved by more than one order of magnitude and the reliability of system with replication factor 3 can be improved by nearly two orders of magnitude; and third, providing the same effective storage space, the system with replication factor 3 requires 50 % more storage nodes and achieves the reliability two orders of magnitude higher than that with replication factor 2.

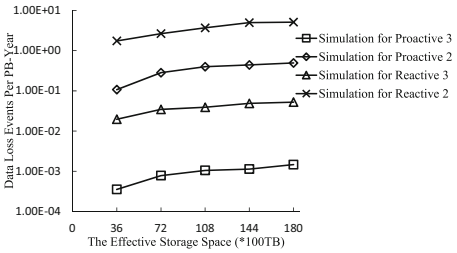

Fig. 7. The change of systems reliability according to the effective space size.

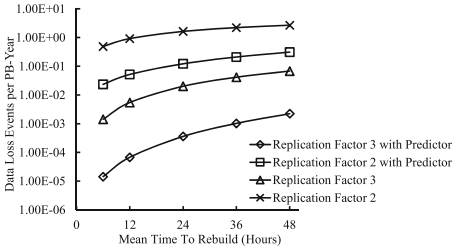

Fig. 8. The change of systems reliability according to the rebuild time.

5.3 Sensitivity of Rebuild Time

Figure 8 shows how the reliability of systems changes according to the rebuild time. We can also learn about that: first, the systems with replication factor 3 are more sensitivity to the rebuild time than the systems with replication factor 2. Specifically, when the rebuild time for a failure/warning is doubled with other parameters constant, the reliability of systems with replication factor 2 will decline by half and the reliability of systems with replication factor 3 will decline by three-quarters; second, having a predictor with the FDR of 80 %, both the systems can reduce the bandwidth available for rebuild process to extend the rebuild time seven times longer (by which systems can reserve more bandwidth to serve user requests), while the reliability of them remain the same.

5.4 Sensitivity of Failure Detection Rate (FDR)

We compare the reliability as the FDR is varied. The results are shown in Fig. 9. We can learn about the following: first, it's clear that the more accurate are the predictors the more reliable are the systems; second, for the system with replication factor 2, when the FDR of predictor is higher than 70 %, its reliability is more sensitivity to the capability of failure prediction and can be improved by 1 $\sim$ 2 orders of magnitude with other system parameters constant; third,

for the system with replication factor 3, when the FDR is higher than 60 %, its reliability is more sensitivity to the capability of failure prediction and can be improved by $1 \sim 3$ orders of magnitude with other system parameters constant; and third, when the FDR achieves 97 %, the proactive system with replication factor 2 can achieve the same level of reliability as the reactive system with replication factor 3, which demonstrates the great value of the proactive fault tolerance mechanism.

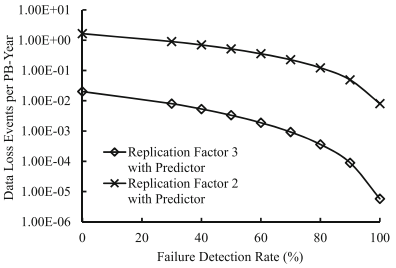

Fig. 9. The change of proactive systems reliability according to the failure detection rate of predictors.

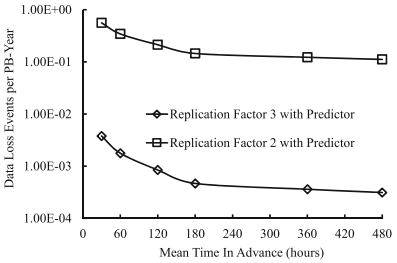

Fig. 10. The change of proactive systems reliability according to the mean time in advance of predictors.

5.5 Sensitivity of Time in Advance

Figure 10 shows the effect of TIA on the reliability of proactive systems. The systems reliability is significantly improved as the increase of TIA. Beyond that, both the curves are relatively flat after the TIA of 180 h, which denotes that the TIA of 180 h is the borderline between prominent and non-prominent influence on the reliability of systems. This observation can guide designers to build a appropriate predictor for cloud storage systems. Note that, this borderline is drawn given the rebuild time of 24 h for a node warning, and a longer rebuild time may induce a higher borderline.

6 Conclusion

In this paper, we present several Markov-based reliability models for cloud storage systems with/without proactive fault tolerance respectively, by which one can systematically analyze the reliability of systems. To describe and compute the intricate reliability models for proactive cloud storage systems, we propose a generic "brick" – basic state transition unit (BSTU), to describe the general pattern of state transitions in the reliability models. Using our models, we evaluate the influences of some system parameters such as the sensitivity of predictor, node MTTF, rebuild time, and system size on the reliability of cloud storage systems. We wish that our models could serve as a guideline for system designers

and administrators, who are not reliability experts, to decide system parameters when building cloud storage systems. For example, to ensure the system availability, users will want to take up as less bandwidth as possible for the rebuild process, which induces a long time to repair the failure or warning. However, as shown in Fig. 8, the increasing of the rebuild time will significantly decrease the reliability. Therefore, using our models, users can choose the proper rebuild bandwidth to coordinate the availability and reliability.

However, the Markov models are build by the assumptions that node failures and rebuild time follow an exponential distribution, which have been contested by recent empirical studies of real world storage systems. Therefore in the future work, we want to extend our methods to support non-exponential distributions. Moreover, it is an important work of further investigation to take into account the behavior of correlated failures to understand cloud storage system's reliability.

Acknowledgments. This work is partially supported by NSF of China (61373018, 11301288, 11450110409), Program for New Century Excellent Talents in University (NCET130301), the Fundamental Research Funds for the Central Universities (65141021) and the Ph.D. Candidate Research Innovation Fund of Nankai University.

References

1. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Machine learning methods for predicting failures in hard drives: a multiple-instance application. J. Mach. Learn. Res. **6**, 783–816 (2005)
2. Hamerly, G., Elkan, C.: Bayesian approaches to failure prediction for disk drives. In: Proceedings of the 18th International Conference on Machine Learning, pp. 202–209 (2001)
3. Hughes, G.F., Murray, J.F., Kreutz-Delgado, K., Elkan, C.: Improved disk-drive failure warnings. IEEE Trans. Reliab. **51**(3), 350–357 (2002)
4. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Hard drive failure prediction using non-parametric statistical methods. In: Proceedings of the International Conference on Artificial Neural Networks (2003)
5. Zhao, Y., Liu, X., Gan, S., Zheng, W.: Predicting disk failures with HMM- and HSMM-based approaches. In: Proceedings of the 10th Industrial Conference on Advances in Data Mining: Applications and Theoretical Aspects, pp. 390–404 (2010)
6. Zhu, B., Wang, G., Liu, X., Hu, D., Lin, S., Ma, J.: Proactive drive failure prediction for large scale storage systems. In: Proceedings of 29th IEEE Conference on Massive Storage Systems and Technologies (MSST), pp. 1–5. Long Beach, CA (2013)
7. Li, J., Ji, X., Jia, Y., Zhu, B., Wang, G., Li, Z., Liu, X.: Hard drive failure prediction using classification and regression trees. In: Proceedings of 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 383–394 (2014)
8. Eckart, B., Chen, X., He, X., Scott, S.L.: Failure prediction models for proactive fault tolerance within storage systems. In: Proceedings of IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems, pp. 1–8 (2008)

9. Zhang, Z., Lian, Q.: Reperasure: replication protocol using erasure-code in peer-to-peer storage network. In: Proceedings of 21st IEEE Symposium on Reliable Distributed Systems, pp. 330–335 (2002)
10. Peric, D., Bocek, T., Hecht, F.V., Hausheer, D., Stiller, B.: The design and evaluation of a distributed reliable file system. In: Proceedings of International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 348–353 (2009)
11. Chen, M., Chen, W., Liu, L., Zhang, Z.: An analytical framework and its applications for studying brick storage reliability. In: Proceedings of the 26th IEEE International Symposium on Reliable Distributed Systems, pp. 242–252. IEEE (2007)
12. Leslie, M., Davies, J., Huffman, T.: A comparison of replication strategies for reliable decentralised storage. J. Netw. **1**(6), 36–44 (2006)
13. Venkatesan, V., Iliadis, I.: A general reliability model for data storage systems. In: Proceedings of Ninth International Conference on Quantitative Evaluation of Systems Quantitative Evaluation of Systems (QEST), pp. 209–219 (2012)
14. Rao, K., Hafner, J.L., Golding, R.A.: Reliability for networked storage nodes. In: DSN 2006 International Conference on Dependable Systems and Networks, pp. 237–248 (2006)
15. Borges, C.L., Falcão, D.M., Mello, J.C.O., Melo, A.C.: Composite reliability evaluation by sequential monte carlo simulation on parallel and distributed processing environments. IEEE Trans. Power Syst. **16**(2), 203–209 (2001)