

# Understanding the Session Durability in Peer-to-Peer Storage System\*

Jing Tian, Yafei Dai, Hao Wang, and Mao Yang

Department of Computer Science, Peking University, Beijing, China  
{tianjing, dyf, wanghao, ym}@net.pku.edu.cn  
<http://net.pku.edu.cn>

**Abstract.** This paper emphasizes that instead of long-term availability and reliability, the short-term session durability analysis will greatly impact the design of the real large-scale Peer-to-Peer storage system. In this paper, we use a Markov chain to model the session durability, and then derive the session durability probability distribution. Subsequently, we show the difference between our analysis and the traditional *Mean Time to Failure* (MTTF) analysis, from which we conclude that the misuse of MTTF analysis will greatly mislead our understanding of the session durability. We further show the impact of session durability analysis on the real system design. To our best knowledge, this is the first time ever to discuss the effects of session durability in large-scale Peer-to-Peer storage system.

## 1 Introduction

Peer-to-Peer storage system is shown to be promising distributed storage architecture for its scalability. However, at the same time these systems suffer from unit failures for the existing of a large number of storage units. As a result, how to improve the availability and the reliability has become a critical and heated issue in the system design. Some approaches, such as TotalRecall[1] and OceanStore[2], have been proposed to improve the reliability as well as availability, and some analytical works have been done, for instance, [3], [4] and [5].

The availability is defined as at any given time  $t$ , the probability that data is accessible. The reliability is defined as the probability that data is not irretrievably lost at time  $t$ , and is usually measured by MTTF. From the definition, we can see that some transient failures of the storage units do not affect the reliability, but they do decrease the availability. Consequently, the reliability is a relative long-term system property. Furthermore, the availability is not a time relative property, and it only captures the probability in the long-term steady state. As a result, we argue that the short-term *session durability* property, defined as the probability that the data is always accessible before time  $t$ , is more important and practical than the long-term availability and reliability for some Peer-to-Peer storage systems. In fact the storage units are not likely to fail immediately after a data is stored. For example, consider that we store a

---

\* This work is supported by National Grand Fundamental Research 973 program of China under Grant No.2004CB318204, National Natural Science Foundation of China under Grant No.90412008.

data object on a storage node  $n$ , whose lifetime and recovery time are both exponential distribution with a mean time of 10 days, then the long-term availability is only 0.5 while the probability that the data has a 24 hours continuous accessible session, is over 0.9. In contrast, if the storage node has a mean lifetime of 24 hours and a mean recovery time of 2.4 hours, the availability is more than 0.9 while the probability of a 24 hours continuous session is only about 0.37. Thus, it is clear that there is little correlation between the session durability and the availability. Consider a streaming service for hot new movies building on a Peer-to-Peer storage, for instance, we may care more about the probability that a newly added movie can be continuous accessible throughout the first 3 days rather than the long-term availability, for a transient interruption in playing will be annoying. Here, the session durability analysis can give a great help.

The session durability probability seems somewhat similar to the reliability, but it takes the transient failure into account. The transient failure makes the session durability calculation much far from the reliability calculation. As we will show in section 4, the misuse of reliability calculation can greatly mislead our understanding of the session durability in Peer-to-Peer storage system.

This paper makes the following contributions: First, we address the session durability analysis that captures the short-term data availability. Second, we present a Markov chain to model the session durability, and demonstrate how to resolve the session durability. Third, we analyze the difference between session durability calculation and reliability calculation, and conclude that MTTF calculation can not be applied in a high dynamic environment. Fourth, we show the great impact of session durability analysis on real system design.

## 2 Background and Motivation

**Erasure Code.** In the face of frequent component failures, the designers usually employ a replication or an erasure code[6] technology to achieve high data durability. Replication is a straightforward scheme which makes replicas to tolerate the failures. Erasure code provides redundancy by encoding the data into fragments and restoring data from a subset of all fragments. First of all, erasure code divides a data object into  $m$  fragments, and encodes them into  $n$  fragments, where all the fragments are in the same size and  $n > m$ , so the redundancy rate  $r$  is  $n/m$ . According to the erasure code theory, the original data object can be reconstructed from any  $m$  fragments out of the  $n$  fragments. In fact, we can consider the replication scheme as a subset of erasure code by making  $m$  to 1, so we use the term “*erasure code*” to refer to both redundancy strategies in the following discussions.

**MTTF Calculation.** In stochastic terminology, the reliability is defined as: for any target life time  $t$ , as the probability that an individual system survives for time  $t$  given that it is initially operational[7]. In a RAID system, Gibson points out that the reliability function  $R(t)$ , is well-approximated by an exponential function  $e^{-t/MTTF}$ , with the same MTTF. As a result, in stead of the reliability function, MTTF is used as the reliability metric for the convenience of computation and expression with the implicit exponential distribution assumption of system lifetime, though MTTF the mean value itself, can tell us nothing about the lifetime distribution.

### 3 Session Durability Model and Calculation

#### 3.1 Exponential and Independent Unit Lifetime

Gibson [7] has shown the exponential lifetime distribution of one disk in a traditional RAID system. However, when investigating the lifetime of a storage unit in a large-scale dynamic Peer-to-Peer environment, we should take some other aspects into account besides the storage device failures, for instance, the network failures and the power problems. By analyzing the traces, Praveen et al.[8] show that the unit lifetime of PlanetLab and the web servers follows an exponential distribution. Web servers are intended to be representative of public-access machines maintained by different administrative domains, while PlanetLab potentially describes the behavior of a centrally administered distributed system. In a Peer-to-Peer storage system formed by end users, the arrival and departure behavior of an end user is unpredictable in the long run, so the behavior is best modeled by a memoryless exponential distribution[9]. Though there is no strong evidence of exponential lifetime in P2P system, some previous studies[10, 11, 12] adopt exponential lifetime in simulations or analyses. In this paper, we take the exponential lifetime assumption for a peer’s lifetime.

In a large-scale Peer-to-Peer system, the storage units (servers or peers) locate in a very wide area, so there is little chance that different units fail dependently. Bhagwan et al.[13] also point out it highly unlikely to have lifetime dependency in P2P systems. In this paper, we assume that the failures are independent.

#### 3.2 The Markov Analysis Model

By assuming all the storage units have the same independent exponential lifetime, we use a continuous-time Markov chain to model the session durability of a Peer-to-Peer storage system with erasure code. Given a system, if the erasure code’s parameters are  $n$  and  $m$ , a data object will be encoded into  $n$  fragments and stored in  $n$  different storage units, called a redundancy group. Then there will be  $n-m+2$  states for a data object illustrated in Figure 1. State  $0$  is the initial state with all storage units alive, and state  $k$  is the state with  $k$  storage units failed, so obviously state  $n-m+1$  is the absorbing state in which we can not reconstruct the data object.

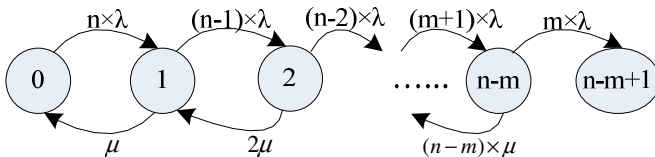


Fig. 1. Markov Model for Session Durability Analysis

The system can transit from one state to another when a failure or recovery happens. For all the storage units have the same mean lifetime  $t_i$ , the failure rate of a unit is  $\lambda = 1/t_i$ . In state  $k$ , all the  $n-k$  live storage units potentially fail, so the failure rate of

state  $k$  is  $(n-k) \times \lambda$ . At the same time, the failed storage units can recover from the transient failure. By assuming that all the failed storage units have the same independent mean recovery time  $t_r$ , we derive that the recovery rate of a failed unit is  $u$ , where  $u$  is the reciprocal of  $t_r$ . Consequently, the recovery rate of state  $k$  is  $k \times u$  since all the  $k$  failed units potentially recover.

### 3.3 Session Durability Probability Calculations

In stochastic terminology, we define the session durability probability  $R(t)$  as:

$$R(t) = \text{Prob}(\text{lifetime} > t \mid \text{initially all storage units alive})$$

In Figure 1,  $R(t)$  can be expressed as the probability that the system is not in the absorbing state  $n-m+1$  at time  $t$ . In this subsection, we explore and demonstrate several methods to get the  $R(t)$  function by resolving the Markov model. First of all, we give the transition matrix of the model

$$Q = \begin{pmatrix} -n \times \lambda & n \times \lambda & 0 & 0 & \dots & 0 \\ \mu & -\mu - (n-1) \times \lambda & (n-1) \times \lambda & 0 & \dots & 0 \\ 0 & 2\mu & -2\mu - (n-2) \times \lambda & (n-2) \times \lambda & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -(n-m) \times \mu - m \times \lambda & m \times \lambda \\ 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

Let  $P(t) = (p_{i,j}(t))$  be the transition matrix function, where  $p_{i,j}(t)$  is the probability that the system transits from state  $i$  to state  $j$  through time  $t$ . Then we have

$$R(t) = p_{0,0}(t) + p_{0,1}(t) + \dots + p_{0,n-m}(t) = 1 - p_{0,n-m+1}(t) \tag{1}$$

Hence, our goal is to get  $p_{0,n-m+1}(t)$ . According to the forward Kolmogorov equation [14],  $P(t)$  is determined by linear differential equations as follows

$$P'(t) = P(t)Q \tag{2}$$

Taking the Laplace transform of both sides of (2) yields

$$sP^*(s) - P(0) = P^*(s)Q$$

Under the condition  $P(0)=I$ , we have

$$P^*(s) = (P_{i,j}^*(s)) = (sI - Q)^{-1} \tag{3}$$

Consequently, we can get  $p_{0,n-m+1}(t)$  by the inverse Laplace transform of  $p_{0,n-m+1}^*(s)$ , then we get  $R(t)$  from (1). Alternatively, we can directly use the unique solution [14] to (2) under the initial condition  $P(0)=I$  as follows

$$P(t) = \exp\{Qt\} = \sum_{n=0}^{\infty} \frac{Q^n t^n}{n!} \tag{4}$$

### 3.4 Review MTTF Calculation

The session durability is somewhat similar to the reliability in definition, so here we review the calculation of reliability for comparison. For the reliability analysis, we can still use the Markov model illustrated in Figure 1. However, there is only permanent failure but no transient failure in reliability analysis. As a result, the failure rate of a single unit is  $\lambda = 1/MTTF_{unit}$ , where  $MTTF_{unit}$  is the mean time to permanent failure. Subsequently, there is only data repair but no recovery, and the repair rate of a single unit is  $u = 1/MTTR_{unit}$ , where  $MTTR_{unit}$  is mean time to repair.

In [7], Gibson presents an iterative method to get the system MTTF, while Thomas gets the same results in[3] via the Laplace transform. The solution can be expressed as follows

$$MTTF = -(1, 0, \dots, 0) \cdot A^{-1} \cdot \vec{e} \tag{5}$$

$A$  is the submatrix of transition matrix  $Q$  ignoring the absorbing state. By applying the exponential distribution assumption, we can identify an approximate reliability function (6) by the system MTTF. We use *exponential approximation* to refer to this approximate calculation.

$$R_{MTTF}(t) = e^{-t/MTTF} \tag{6}$$

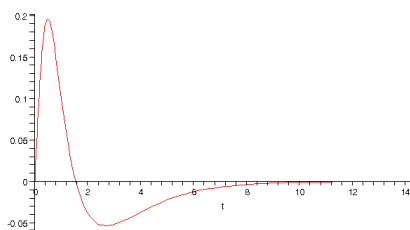
## 4 Difference of Session Durability Calculation and MTTF Calculation

Because the session durability and the reliability share the similar analysis model, one may argue that we can use the traditional MTTF calculation as a substitute for the complicated session durability resolving. Unfortunately, we show in this section, that the MTTF calculation can not be applied in Peer-to-Peer environment, because the exponential function assumption does not hold in a high dynamic environment. We first give a qualitative analysis to get a preliminary understanding of the difference between two calculations, and then give a quantitative insight into the difference influenced by dynamic system parameters.

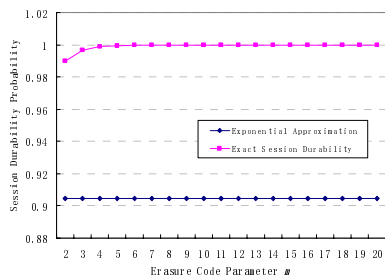
### 4.1 A Qualitative Analysis

There are two main conditions which the traditional exponential distribution assumption is based on: First, the failure rate is much larger than repair rate, i.e.  $\lambda \gg u$ . Second, the number of system states is not very big, e.g. 3 in traditional RAID analysis[7]. In fact, these two conditions do not exist at all in dynamic Peer-to-Peer environment. In Peer-to-Peer environment, the unit failure caused by node leaving may be very often, and the recovery (node rejoining) usually takes a long time, so  $\lambda$  is comparable with  $u$ . The system designers must use more redundancy or more erasure code fragments (e.g.  $n=128$  and  $m=64$ ) to make the data more available and reliable, which enlarges the number of system states very much.

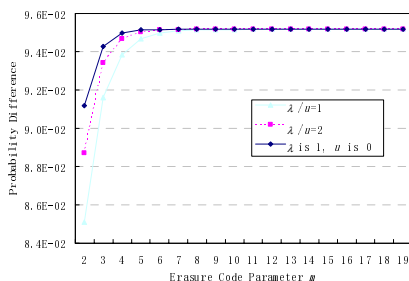
By exploring a large number of sample erasure-coded systems with randomly generalized parameters  $m$ ,  $n$ ,  $\lambda$  and  $u$ , we find that the patterns of the difference functions of the exact session durability calculation and the exponential approximation calculation are all alike in shape. In Figure 2, we plot a sample difference function to show the pattern. It is clear that the exponential approximation calculation first underestimates the session durability probability and then overestimates it a little bit. The underestimation is because the real system can not move too fast from the initial state to the failure state in the very beginning while it is relatively easy to fail in the two states exponential approximation. Since both the exact session durability function and the exponential approximation have the same mean session time  $MST$ , the integral of the difference should be 0. Consequently, exponential approximation will definitely overestimate the session durability after the underestimation.



**Fig. 2.** The difference function of the exact session durability and the exponential approximation,  $m=4$ ,  $n=8$ ,  $\lambda=1$ ,  $\mu=1$



**Fig. 3a.** Exact session durability and the exponential approximation



**Fig. 3b.** Differences between the exact session probabilities and the exponential approximations

### 4.2 Quantitative Insight into the Difference

To gain a deeper understanding of the difference between exact session durability and the exponential approximation influenced by the number of the states, we fix the failure rates and repair rates to investigate the difference versus the erasure code parameters  $m$  and  $n$ . By fixing the redundancy rate  $r=n/m=2$ , we plot the exact session durability probability as well as the exponential approximation probability at  $MST/10$

versus parameter  $m$  in Figure 3a. From the figure, we can see that the difference becomes larger when the number of states grows, and that the approximation greatly underestimates the session durability from a probability near 1 to a probability about 0.9. Furthermore, we investigate the difference trend by increase the ratio of failure rate  $\lambda$  to the recovery rate  $u$ . In Figure 3b, we plot the probability differences of three pairs of parameters  $\lambda$  and  $u$  at  $MST/10$ . What we find is that the differences increase when the ratio of  $\lambda$  to  $u$  increases.

We conclude the findings in this subsection that the dynamic feature of Peer-to-Peer network and the fact of using more redundancy fragments make it dangerous to use MTTF calculation as a substitute for session durability, and the misuse may greatly mislead our understanding of the session durability.

## 5 Impact of Session Durability on System Design

This subsection uses several cases to demonstrate the impact of our session durability analysis on real system design.

**Streaming Service.** Consider we are building a streaming service for new hot movies on the PlanetLab. The newly added movie is to be a hotspot in the first several days, and we do not want the annoying transient failures to interrupt the playing. Therefore, the system will require a long continuous accessible session time in first several days rather than a high availability in long-term. According to the PlanetLab trace used in[8], we find most of the nodes have a mean lifetime of  $10^6$  seconds, while many of them have a mean recovery time of  $5 \times 10^6$  seconds. Assume that we use an erasure code scheme with parameters  $m=4$  and  $n=8$ , then we get that the one day session durability is 99.99% and the three days session durability is 99%. If we use the MTTF's exponential assumption for calculation, we can only get 96.56% and 90% for one day and three days session durability respectively. As to the availability, the analysis gives us an availability of 91.24%. According to results of the availability analysis and the reliability like calculation, we may abandon the idea of building the service on a Peer-to-Peer environment, or use more redundancy data to enhance the durability. However, the fact is that the session durability is high enough for the requirement of the service according to our session durability analysis.

**OpenDHT's Storage.** OpenDHT[15] requires a definite time-to-live(TTL) along with a storage request for the fairness of storage allocation. As a result, the designer can only concentrate on how to improve session durability within the specified *TTL*, but not think about the availability and reliability. Since the availability analysis and reliability analysis give very low underestimations, the designer can use less system resource to guarantee a good enough session durability within *TTL* by using session durability analysis.

**Fixed Repair Epoch for Large-Scale Distributed Storage.** Large-scale distributed storage systems usually use a fixed repair epoch for the simplicity of repair mechanism. For example, [4] assumes a system with fixed repair epoch, and the system employs an erasure code with  $m=32$  and  $n=64$ . The designer should get the knowledge about the probability that the data can survive a single epoch. Though it may not be a high dynamic system, the calculation under the exponential assumption will greatly mislead us, because there is no repair within an epoch. Assume a five years

$MTTF_{\text{unit}}$ , under the exponential assumption we calculate the probabilities that a data can survive a four months epoch and a 12 months epoch respectively, and get 91.1% and 75.6%. However the real probability is greater than  $1-10^{16}$  for four months, and greater than  $1-10^8$  for 12 months.

## 6 Conclusions

In this paper, we first addressed the new metric, session durability, for a Peer-to-Peer storage system. Subsequently, we presented the analysis model, and demonstrated how to resolve the session durability from the model. Our experiments have shown strong evidence that MTTF calculation can not be applied in high dynamic environment, and session durability is far from the reliability analysis. We further showed the impact of session durability analysis on real system design.

## References

1. R. Bhagwan, K. Tati, Y. Cheng, S. Savage, and G. M. Voelker. Total recall: System support for automated availability management. In *Proc. of the First ACM/Usenix Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.
2. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: An Architecture for Global-Scale Persistent Storage. In *ACM ASPLOS*, November 2000.
3. T. Schwarz. Generalized Reed Solomon codes for erasure correction in SDDS. In *Workshop on Distributed Data and Structures (WDAS 2002)*, Paris, France, Mar. 2002.
4. H. Weatherspoon and J. D. Kubiatowicz. Erasure Coding vs. Replication: A Quantitative Comparison, In *Proc. of IPTPS '02*, March 2002.
5. R. Rodrigues and B. Liskov. High Availability in DHTs: Erasure Coding vs. Replication. In *Proc. of IPTPS'05*, February 2005.
6. J. Plank. A tutorial on Reed-Solomon coding for fault-tolerance in raid-like systems. *Software Practice and Experience*, 27(9):995-1012, September 1997
7. G. A. Gibson. Redundant Disk Arrays: Reliable, Parallel Secondary Storage. *PhD thesis, U. C. Berkeley*, April 1991.
8. P. Yalagandula, S. Nath, H. Yu, P. B. Gibbons and S. Seshan. Beyond Availability: Towards a Deeper Understanding of Machine Failure Characteristics in Large Distributed Systems. In *Proc. of the 1st Workshop on Real, Large Distributed Systems*, 2004.
9. G. Pandurangan, P. Raghavan and E. Upfal. Building low-diameter P2P networks. In *Proc. of the 42nd Annual IEEE Symposium on the Foundations of Computer Science*, Oct. 2001.
10. D. Liben-Nowell, H. Balakrishnan and D. Karger. Analysis of the evolution of Peer-to-Peer systems. In *Proc. of the 21st ACM Symposium on Principles of Distributed Computing*. Monterey, CA, USA: ACP Press, 2002
11. Y. Zhao. Decentralized Object Location and Routing: A New Networking Paradigm. *U.C. Berkeley PhD Dissertation*, August 2004
12. S. Giesecke, T. Warns and W. Hasselbring. Availability Simulation of Peer-to-Peer Architectural Styles. In *Proc. of ICSE 2005 WADS*.
13. R. Bhagwan, S. Savage and G. Voelker. Understanding availability. In *proc. of International Workshop on Peer-to-Peer Systems (IPTPS03)*, February 2003.
14. M Kijima. Markov Processes for Stochastic Modeling. *Chapman and Hall, London*, 1997.
15. <http://www.opendht.org/>