9ième Atelier en Evaluation de Performances
Aussois, du 1 au 4 juin 2008

P2P storage systems : Towards a more accurate modeles

Abdulhalim Dandoush

INRIA Sophia Antipolis
MAESTRO Team
2004 Route des Lucioles, B.P. 93
06902 Sophia Antipolis
abdulhalim.dandoush@sophia.inria.fr

Résumé

This paper characterizes the performance of peer-to-peer storage systems in terms of the delivered data lifetime and data availability. Two schemes for recovering lost data are modeled and analyzed : the first scheme is centralized and relies on a server that recovers multiple losses at once, whereas the second scheme is distributed and recovers one loss at a time. For each scheme, I propose a Markovian model where the availability of peers is hyper-exponentially distributed. The proposed models equally apply to many distributed environments as shown through numerical computations. These allow us to assess the impact of each system parameter on the performance. In particular, I provide guidelines on how to tune the system parameters so as to provide desired lifetime and/or availability of data.

1. Introduction

In the current Client/Server model, storage solutions rely on robust dedicated servers. These equipments are reliable and simple to admin, but they are also expensive and do not scale well. Recent alternative solutions try to use the distributed peer-to-peer (P2P) infrastructures. These storage systems are not expensive and scale very well but pose many problems of reliability, confidentiality, availability, etc.

In a P2P network, peers are free to leave and join the system at any time. As a result of the intermittent availability of the peers, ensuring high availability of the stored data is an interesting and challenging problem. To ensure data reliability, redundant data is inserted in the system.

However, using redundancy mechanisms without repairing lost data is not efficient. P2P storage systems need to compensate the loss of data by continuously storing additional redundant data onto new hosts. Systems may rely on a central authority that reconstructs fragments when necessary ; these systems will be referred to as *centralized-recovery systems*. Alternatively, secure agents running on new hosts can reconstruct by themselves the data to be stored on the hosts disks. Such systems will be referred to as *distributed-recovery systems*. A centralized server can recover at once multiple losses of the same document in the centralized-recovery scheme. This is not possible in the distributed case where each new host recovers only one loss per document.

I will focus in this study on the quality of service delivered to each block of data. the work aim at addressing fundamental design issues such as : *how to tune the system parameters so as to maximize data lifetime while keeping a low storage overhead and achievable bandwidth use ?*

Recent work [5], investigate three sets of data, each measuring machine availability in a different setting. It has found that a hyper-exponential model fits more accurately the machine availability durations than the exponential, Pareto, or Weibull distribution.This study supports the key assumption of the models presented in my paper.

2. System description and notation

I consider a single block of data D, divided into s equally sized fragments to which, using erasure codes r redundant fragments are added. These $s + r$ fragments are stored over $s + r$ different peers. Data D is said to be *available* if any s fragments out of the $s + r$ fragments are available and *lost* otherwise.

Over time, a peer can be either *connected* to or *disconnected* from the storage system. At reconnection, a peer may still store its fragments with the probability p.

I refer to as *on-time* (resp. *off-time*) a time-interval during which a peer is always connected (resp. disconnected). I assume that the successive durations of on-times (resp. off-times) of a peer form a sequence of independent and identically distributed (iid) random variables (rvs).

The off-times are assumed to be independent and identically (iud) rvs with a common exponential distribution function with parameter $\lambda > 0$; this assumption is in agreement with the analysis in [4]. However, in light of the analyses reported in [5, 4], I consider that the distribution of on-times durations is hyper-exponential with n phases ; the parameters of phase i are $\{p_i, \mu_i\}$, with p_i the probability that phase i is selected and $1/\mu_i$ the mean duration of phase i. Successive on-times and off-times are assumed to be independent.

I will investigate the performance of two different repair policies : the *eager* and the *lazy* repair policies. In the eager policy a fragment of D is reconstructed as soon as one fragment has become unavailable due to a peer disconnection. In the lazy policy, the repair is delayed until the number of unavailable fragments reaches a given threshold, denoted k. Both repair policies can be represented by the threshold parameter $k \in \{1, 2, \ldots, r\}$, where $k = 1$ in the eager policy and otherwize in lazy. The stat of the markov chain $X^h(t)$ represents the available number of redundant fragments of a given block of data and the chain reaches an absorbing stat a when the available fragments are less than s ; there are no redundant fragments.

3. The Metrics of the models

I'll define the metrics used to evaluate the storage system which are used in the Numerical results section. I'll put the difinition of these metrics for the centralized-repair system, refered by the subscript "c". It is the same difinition for the distributed-repair system but with the subscript "d".

3.1. The data lifetime

$T^h_c(\mathcal{E}_I) := \inf\{t \geq 0 : X^h_c(t) = a | X^h_c(0) \in \mathcal{E}_I\}$, the time until absorption in state a given that the initial number of fragments of D available in the system is equal to I. I derived its probability distribution and its expectation value.

3.2. Data availability

The data availability are quantified by the following two metrics (for $s \leq I \leq s + r$).

$$M^h_{c,1}(\mathcal{E}_I), \quad M^h_{c,2}(\mathcal{E}_I) \tag{1}$$

The first availability metric can be interpreted as the expected number of available fragments during the block lifetime, given that the initial number of fragments at time $t = 0$ is I. The second metric can be interpreted as the fraction of time when there are at least m fragments during the block lifetime, given that the initial number of fragments at time $t = 0$ is I. Both quantities can be numerically computed.

4. Numerical results

In this section, I will show some of the numerical results. For more information about the values of the parameters, you can see the following research report [2]

4.1. Data sets

The *LMG* set has been collected by Long, Muir and Golding [3]. The sets *CSIL* and *Condor* have been collected by Nurmi, Brevik and Wolski [5]. The three data sets analyzed in [5] report different flavors of peer "availability", but all are best fit by a hyper-exponential distribution. An exponential distribution is found to "reasonably" fit the *All-pairs-ping* data set in [4].

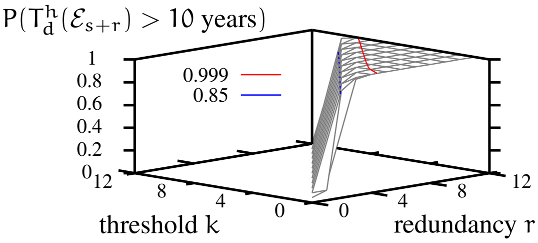

FIG. 1 – CCDF of data lifetime versus r and k using *CSIL* data.

4.2. The conditional block lifetime

I have computed the expectation and the complementary cumulative distribution function (CCDF) of the data lifetime given that all $s + r$ fragments of D are initially available, namely $T^h_c(\mathcal{E}_{s+r})$ and $T^h_d(\mathcal{E}_{s+r})$. It appears that, whichever the scenario and the recovery mechanism considered, the expected data lifetime increases roughly exponentially with r and decreases with an increasing k.

The CCDF of the data lifetime, given that r redundant fragments are available at time $t = 0$, is evaluated at points $q = 1$ and $q = 10$ years. The CCDF appears to depend on r and k in the same way regardless of the recovery scheme implemented in all scenarios where the system is not so dynamic. In the *Condor* scenario, the shape of the 3D curve is different from that of the other scenarios, but here again it is not affected by whether the recovery mechanim is centralized or distributed because the system is highly dynamic.

4.3. The availability metrics

I have computed the first and the second availability metric. Some of the results are reported in Fig. 2 and in Fig. 3.

We see from Fig. 2 that metrics $M^h_{c,1}(\mathcal{E}_{s+r})$ and $M^h_{d,1}(\mathcal{E}_{s+r})$ are differently affected by the parameters r and k. In the centralized implementation, changing the peers failure rate alters the effect of k on the performance for large r and as one could expect, the centralized scheme achieves higher availability than the distributed scheme.

4.4. Engineering the system

Using my theoretical framework it is easy to tune the system parameters for fulfilling predefined requirements. As an illustration, I consider here only the *Condor* data.

Consider point B in Fig. 4 which corresponds to $r = 17$ and $k = 9$ (recall $s = 8$). Selecting this point as the operating point of the storage system will ensure that $P(T^h_c(\mathcal{E}_{s+r}) > 1) = 0.84$ and $M^h_{c,2}(\mathcal{E}_{s+r}) = 0.94$.

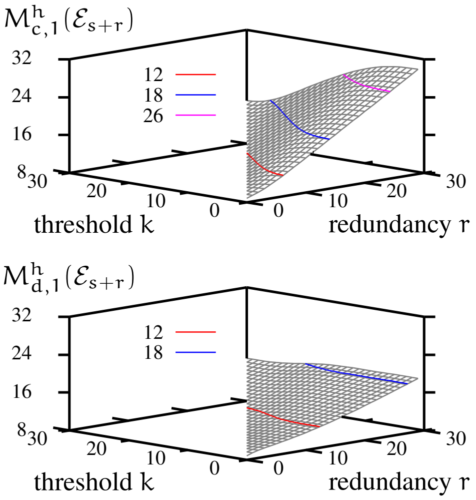

FIG. 2 – Availability metrics versus r and k in *Condor* scenario.

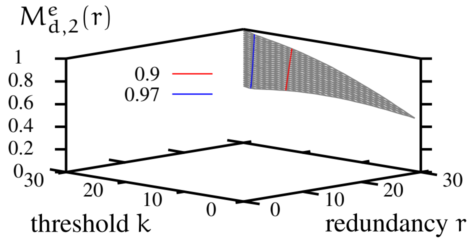

FIG. 3 – Availability metrics $M^e_{c,2}(r)$ and $M^e_{d,2}(r)$ for $m = r - k$ assuming a distributed scheme using the *All-pairs-ping* data.

In other words, when $r = 17$ and $k = 9$, only 16% of the stored blocks would be lost after one year and for 94% of a block lifetime there will be 8 ($= r - k$) or more redundant fragments from the block available in the system. Observe that the storage overhead, usually defined as r/s, will be equal to 2.125.

5. Conclusion

I have proposed analytical models for evaluating the performance of two approaches for recovering lost data in distributed storage systems. I have analyzed the lifetime and the availability of data achieved by both centralized- and distributed-repair systems through Markovian analysis. Using the theoretical framework in this work, it is easy to tune and optimize the system parameters for fulfilling predefined requirements.

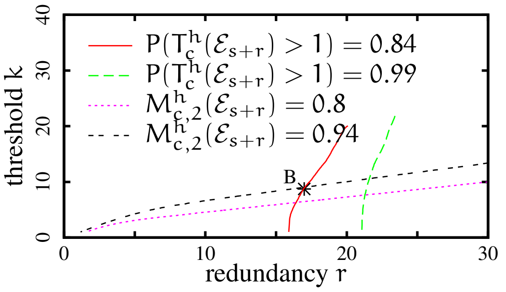

FIG. 4 – Selection of r and k according to predefined requirements assuming a centralized-recovery scheme using the *Condor* data.

Bibliographie

1. S. Alouf, A. Dandoush, P. Nain, Performance analysis of peer-to-peer storage systems, in : Proc. of 20th ITC, Ottawa, Canada, Vol. 4516 of Lecture Notes in Computer Science, 2007, pp. 642–653.
2. A. Dandoush, S. Alouf, P. Nain, P2P storage systems modeling, analysis and evaluation, INRIA research report Number 6392, Sophia Antipolis, France, december 2007.
3. D. Long, A. Muir, R. Golding, A longitudinal survey of internet host reliability, in : Proc. of 14th Symposium on Reliable Distributed Systems, Ben Neuenahr, Germany, 1995, pp. 2–9.
4. S. Ramabhadran, J. Pasquale, Analysis of long-running replicated systems, in : Proc. of IEEE Infocom '06, Barcelona, Spain, 2006.
5. D. Nurmi, J. Brevik, R. Wolski, Modeling machine availability in enterprise and wide-area distributed computing environments, Tech. Rep. CS2003-28, University of California Santa Barbara (2003).