P2P Storage Systems: Data Life Time for Different Placement Policies

Stéphane Caron, Frédéric Giroire, Dorian Mazauric, Julian Monteiro, Stéphane Pérennes

P2P Storage Systems: Data Life Time for Different Placement Policies †

S. Caron and F. Giroire and D. Mazauric and J. Monteiro and S. Pérennes

MASCOTTE joint project team, INRIA, I3S, CNRS, Univ. Nice Sophia - B.P. 93, F-06902 Sophia Antipolis, France.

Les systèmes pair-à-pair à grande échelle représentent un moyen fiable pour stocker des données à faible coût. Afin d'assurer la pérennité des données des utilisateurs, il est nécessaire d'ajouter de la redondance. Ainsi à partir de s fragments initiaux composant un bloc de données, $s+r$ fragments sont générés et répartis entre les pairs du réseau. Nous étudions dans ce papier l'impact des différentes politiques de placement sur la durée de vie des données. Plus particulièrement nous décrivons des méthodes pour calculer et approximer le temps moyen avant que le système perde une donnée (Mean Time to Data Loss). Nous comparons cette métrique pour trois politiques de placement: deux sont *locales*, distribuant les fragments sur des voisins logiques, et la troisième est *globale*.

Keywords: P2P storage system, data placement, performance, data durability, Markov chain model

1 Introduction and System Description

The key concept of Peer-to-Peer storage systems is to distribute redundant data among peers to achieve high reliability and fault tolerance at low cost. The addition of redundant data could be done by *Erasure Codes* [6], such as Reed Solomon, as used by some RAID schemes. When using Erasure Codes, the original user data (e.g. files, raw data, etc.) is cut into *blocks* that are in turn divided into s initial *fragments*. The encoding scheme produces $s+r$ fragments that can tolerate r failures. In other words, the original block can be recovered from any s of the $s+r$ encoded fragments. In a P2P storage system, these fragments are then placed on $s+r$ different peers of the network according to a placement policy, which is the main subject of this paper. In [3] we studied placement policies by simulations, and we presented the amount of resource (bandwidth and storage space) required to maintain redundancy and to ensure a given level of reliability. In this paper, we present an analytical method to compute the metric Mean Time to Data Loss (MTTDL) for three different placement policies. An extended version of this work can be found in [1]. The remainder of this paper is organized as follows: first we briefly present the characteristics of the studied P2P storage systems, followed by the related work. In Section 2, we describe the studied placement policies. Then, in Section 3 we describe the analytical methods to compute exact values and approximations of the MTTDL for the three policies. We conclude in Section 4.

Peer Failures. It is assumed that the peers stay connected almost all the time into the system. Indeed, in our model a peer failure represents a disk crash or a peer that definitively leaves the system. In both cases, it is assumed that all the data on the peer's disk are lost. Following most works on P2P storage systems, peers get faulty independently according to a memoryless process. For a given peer, the probability to fail at a given time step is α.

Reconstruction Strategy. To ensure a durable long-term storage despite disk failures, the system needs to continuously monitor the number of fragments of each block and maintain a minimum number of redundancy fragments available in the network.

In this work, we study the case where the reconstruction starts as soon as one of its fragments is lost, namely *eager* reconstruction strategy. In addition, the blocks are reconstructed in one time step, i.e., there is

†This work was partially funded by the ANR projects SPREADS, DIMAGREEN and région PACA.

enough bandwidth to process the reconstruction quickly. After the reconstruction, the regenerated missing fragments are spread among different peers. Hence, after each time step, the system is fully reconstructed. We also studied systems with other reconstruction processes in [1], but we do not discuss them here due to lack of space.

Related Work

The majority of existing or proposed systems, e.g., Intermemory, CFS, Farsite, PAST, TotalRecall, Glacier, use a local placement policy. In [4] the authors disuss the impact of data placement. They do a practical study of a large number of placement policies for a system with high churn. They exhibit differences of performance in terms of delay, control overhead, success rate, and overlay route length. In the work closer to ours [5], the authors study the impact of data placement on the Mean Time to Data Loss (MTTDL) metric. All these studies consider the case of systems using replication. In this paper, we address the more complex case of Erasure Codes which are usually more efficient for the same storage overhead [6].

2 Placement Policies

It has been shown that fragment placement has a strong impact on the system performance [3, 5]. We study here three different strategies to place the $s+r$ fragments of a block, as explained in the following and depicted in Figure 1:

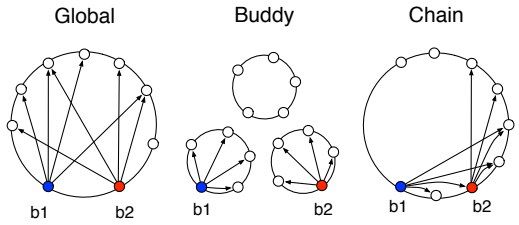

Figure 1: Placement of two blocks $b1$ and $b2$ in the system using different policies.

• *Global Policy:* fragments are sent to peers chosen uniformly at random among all the N peers.

• *Buddy Policy:* peers are grouped into C independent clusters of size exactly $s+r$ each. The fragments are then sent to a cluster chosen uniformly at random among the clusters. In this situation, all peers of a cluster store fragments of the same set of blocks. It could be seen as a collection of local RAID like storage.

• *Chain Policy:* the network is seen as a directed ring of N peers and the fragments are sent to $s+r$ consecutive peers choosen uniformly at random. This policy corresponds to what is done in most distributed systems implementing a DHT.

The use of the Global policy allows the system to distribute more uniformly the load among peers, leading to a faster reconstruction and a smoother operation of the system [3]. However, the use of Buddy and Chain, namely *local* strategies, brings practical advantages [2]. For example, the management traffic and the amount of meta-information to be stored by the peers are kept low.

Data Loss Rate. A data loss occurs when at least one block is lost. A block is considered lost if it loses at least $r+1$ fragments during one time step, otherwise, recall that all the $s+r$ fragments are fully reconstructed at next time step. The data loss rate for a given block comes straightforward. This loss rate does not depend on the placement policy (as soon as it is assured that all fragments are stored on different peers). Hence, we have the same expected number of lost blocks for the three placement policies.

However, as stated in [3], the measure of the time to the first occurrence of data loss shows that the three policies have very distinct behaviors. It is shown by simulations that the average quantity of data loss per year is the same, but the distribution across time of these losses is very different (see

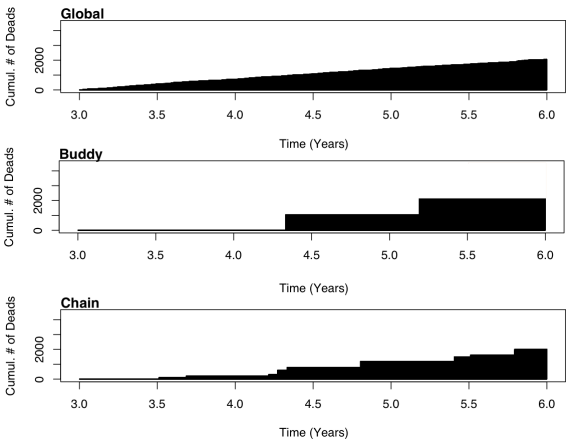

Figure 2: Illustrative example of the cumulative number of dead blocks for a period of three years.

Figure 2). In the Global policy the losses occurs regularly. Conversely, they occur very rarely for the Buddy placement, but, when they occur, they affect a large batch of data. Basically, all the blocks of a Buddy cluster are lost at the same time. The behavior of the Chain policy is somewhere in the middle of both. In the next section we propose analytical methods to compute these variations through the metric MTTDL.

3 Mean Time to Data Loss (MTTDL)

In this section we present methods to compute exact values and approximations of the MTTDL for the three placement policies (see [1] for more details and proofs). For each policy, we calculate the probabilty $\mathbb{P}_{policy}$ to lose data at any given time step. Then, we deduce $MTTDL_{policy} = 1/\mathbb{P}_{policy}$.

3.1 Global Placement Policy

In the Global policy, given $i \geq r+1$ failures, the probability to lose a given block is

$$\mathbb{P}_{block}(i) = \mathbf{P}\left[\text{block loss} \mid i \text{ failures}\right] = \frac{\sum_{j=r+1}^{s+r} \binom{i}{j}\binom{N-i}{s+r-j}}{\binom{N}{s+r}},$$

that is, the enumeration of the possible ways of placement that kill this block in the case of i failures, over all the possible combinations $\binom{N}{s+r}$ of placement. For a system with B blocks, the probability that none of them is lost is $(1-\mathbb{P}_{block}(i))^B$. We then consider all the different failure scenarios, which gives the probability to have at least one block loss $\mathbb{P}_{global} = \sum_{i=r+1}^{N} \mathbf{P}\left[i \text{ failures}\right] \cdot (1-(1-\mathbb{P}_{block}(i))^B)$, where $\mathbf{P}\left[i \text{ failures}\right] = \binom{N}{i}\alpha^i(1-\alpha)^{N-i}$. Then comes directly $\text{MTTDL}_{global} = 1/\mathbb{P}_{global}$. Assuming that $\alpha N \ll 1$, we obtain the following approximation

$$\text{MTTDL}_{global} \approx \frac{1}{B \cdot \binom{s+r}{r+1}\alpha^{r+1}} \tag{1}$$

3.2 Buddy Placement Policy

In the Buddy placement policy, given a cluster, the probability to have a block loss is the probability that the cluster loses at least $r+1$ peers, given by $\mathbb{P}_{cluster} = \sum_{j=r+1}^{s+r} \binom{s+r}{j}\alpha^j(1-\alpha)^{s+r-j}$. In fact, when that happens all the data stored on that cluster is lost. Remember that α is the probability of a given peer to fail at one time step. Since all the C clusters are independent, the probability to have a data loss is given by $\mathbb{P}_{buddy} = 1-(1-\mathbb{P}_{cluster})^C$, and thus $\text{MTTDL}_{buddy} = 1/\mathbb{P}_{buddy}$.

If the average number of cluster failures per time step $C \cdot \mathbb{P}_{cluster} \ll 1$, as expected in a real system (i.e., the probabilty of simultaneous cluster failures is small), then we have $\mathbb{P}_{buddy} \approx C \cdot \mathbb{P}_{cluster}$, and so $\text{MTTDL}_{buddy} \approx 1/(C \cdot \mathbb{P}_{cluster})$.

If $(s+r)\alpha \ll 1$, we can approximate even more and obtain $\mathbb{P}_{cluster} \approx \binom{s+r}{r+1}\alpha^{r+1}$. In other words, this assumption means that the probability of a peer failure α is small. We obtain

$$\text{MTTDL}_{buddy} \approx \frac{1}{\frac{N}{s+r} \cdot \binom{s+r}{r+1}\alpha^{r+1}}. \tag{2}$$

3.3 Chain Placement Policy

For the Chain policy, the computation of MTTDL_{chain} is more difficult than the two previous ones. From the definition of the Chain policy, a data loss occurs only when $r+1$ (or more) peer failures are located at $s+r$ consecutive peers.

The idea is to survey the N sequences $S_1, S_2, \ldots, S_N$ of $s+r$ consecutive peers. First, we define a binary-vector $(b_i, b_{i+1}, \ldots, b_{i+s+r-1})$ for each S_i, where the elements of this vector represent the state of peers of S_i: $b_j = 1$ if the peer numbered j is failed, $b_j = 0$ otherwise, $i \leq j < i+s+r$. Remark that the binary-vector of S_{i+1} is $(b_{i+1}, \ldots, b_{i+s+r})$.

Then, we use a discrete time discrete space Markov chain to represent the transitions between sequences. Indeed, the set of states V of such Markov chain is the set of all possible binary-vectors of size $s+r$ such that

the sum of its elements is at most r, plus an absorbing state namely v_{dead} (containing all other binary-vectors of size $s+r$ in which the sum of its elements is greater than r). For a binary-vector $(b_i, b_{i+1}, \ldots, b_{i+s+r-1})$, we have two possible transitions: $(b_{i+1}, \ldots, b_{i+s+r-1}, 1)$ with probability α and $(b_{i+1}, \ldots, b_{i+s+r-1}, 0)$ with probability $1-\alpha$. One of these vectors (states) could belong to v_{dead}. Remark that we can see this Markov chain as a De Bruijn graph.

First, we assume that the N peers are ordered in a line instead of a ring. We can compute the distribution of probability π after N steps as follows: $\pi = v_0 M^N$ where $v_0 = (0,0,\ldots,0)$ is the state without peer failures and M is the transition matrix of our Markov chain. In that case $\mathbb{P}_{chain}$ is $\pi(v_{dead})$. To get the right value of $\mathbb{P}_{chain}$, we have to carefully take into consideration sequences containing peers on both borders of the network (becoming a ring again). We get $\pi = \sum_{v \in V} P(v)(v_0 M_{b_{i_1}} \ldots M_{b_{i_{s+r}}} M^{N-(s+r)} M_{b_{i_1}} \ldots M_{b_{i_{s+r-1}}})$ with $P(v)$ the probability to have v as initial state, and M_k, $k \in \{0,1\}$, the transition matrix replacing α by k.

The number of states of the previously described Markov chain is $|V| = 1 + \sum_{i=0}^{r} \binom{s+r}{i}$ states. We can reduce it in order to have $|V| = 1 + \sum_{i=0}^{r} \binom{s+r}{i} - \sum_{k=1}^{r} \sum_{j=0}^{k-1} \binom{s+k-1}{j}$ seeing some properties (see [1] for details and proofs).

We also propose two other methods to compute an approximation of the MTTDL. The first one is based on classical approaches for absorbing Markov chains. Indeed if we consider that the number of peers is infinite, then the corresponding fundamental matrix gives us the average time t_{abs} to absorption, that is the average number of consecutive sequences of peers to find a data loss. Thus $MTTDL_{chain} \approx \lfloor t_{abs}/N \rfloor$. The second one assumes that α is small enough to derive an analytical expression:

$$MTTDL_{chain} \approx \frac{1}{N \frac{r+1}{s+r} \binom{s+r}{r+1} \alpha^{r+1}}. \tag{3}$$

4 Discussion and Conclusion

The approximations given by the Equations (1), (2), and (3) give an interesting insight on the relation between the placement policies. For instance, note that the ratio between MTTDL_{buddy} and MTTDL_{chain} does not depend of N, nor B, nor s. When $B \ll \binom{N}{r+1}$, the ratio between MTTDL_{buddy} and MTTDL_{global} depends on the number of fragments per disk $B(s+r)/N$.

$$\frac{\text{MTTDL}_{buddy}}{\text{MTTDL}_{chain}} \approx r+1, \qquad \frac{\text{MTTDL}_{buddy}}{\text{MTTDL}_{global}} \approx \frac{B(s+r)}{N}, \qquad \frac{\text{MTTDL}_{chain}}{\text{MTTDL}_{global}} \approx \frac{B(s+r)}{N(r+1)}.$$

We succeeded in quantifying the MTTDL of the three policies. The Buddy policy has the advantage of having a larger MTTDL than the Chain and the Global. However, when a failure occurs a large number of reconstructions start. When the bandwidth available for reconstruction is low, the reconstructions are delayed which may lead to an increased failure rate. This trade-off has still to be investigated.

References

[1] S. Caron, F. Giroire, D. Mazauric, J. Monteiro, and S. Pérennes. P2P Storage Systems: Data Life Time for Different Placement Policies. Research Report RR-7209, INRIA, Feb 2010. http://hal.inria.fr/inria-00458190/en/.

[2] F. Dabek, J. Li, E. Sit, J. Robertson, M. F. Kaashoek, and R. Morris. Designing a DHT for low latency and high throughput. In *Proc. of NSDI*, pages 85–98, San Francisco, California, 2004.

[3] F. Giroire, J. Monteiro, and S. Pérennes. P2p storage systems: How much locality can they tolerate? In *Proc. of LCN'09*, pages 320–323, Oct 2009.

[4] S. Ktari, M. Zoubert, A. Hecker, and H. Labiod. Performance evaluation of replication strategies in dhts under churn. In *MUM '07*, pages 90–97, New York, NY, USA, 2007. ACM.

[5] Q. Lian, W. Chen, and Z. Zhang. On the impact of replica placement to the reliability of distributed brick storage systems. In *Proc. of ICDCS'05*, volume 0, pages 187–196, 2005.

[6] H. Weatherspoon and J. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *Proc. of IPTPS*, volume 2, pages 328–338. Springer, 2002.