Data Life Time for Different Placement Policies in P2P Storage Systems

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

Data Life Time for Different Placement Policies in P2P Storage Systems *

S. Caron[1], F. Giroire[2], D. Mazauric[2], J. Monteiro[2], and S. Pérennes[2]

[1] ENS Paris
[2] MASCOTTE joint team, INRIA, I3S (CNRS, Univ. of Nice)

Abstract. Peer-to-peer systems are foreseen as an efficient solution to achieve reliable data storage at low cost. To deal with common P2P problems such as peer failures or churn, such systems encode the user data into redundant fragments and distribute them among peers. The way they distribute it, known as *placement policy*, has a significant impact on their behavior and reliability. In this paper, we study the impact of different placement policies on the data life time. More precisely, we describe methods to compute and approximate the mean time before the system loses data (*Mean Time to Data Loss*). We compare this metric for three placement policies: two of them *local*, in which the data is stored in logical peer neighborhoods, and one of them *global* in which fragments are parted uniformly at random among the different peers.

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* [14], 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 [8] 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. 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 Sections 3, 4, 5, we describe the analytical methods to compute exact values and approximations of the MTTDL for the three policies. We conclude in Section 6.

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

* This work was partially funded by the ANR project SPREADS and région PACA.

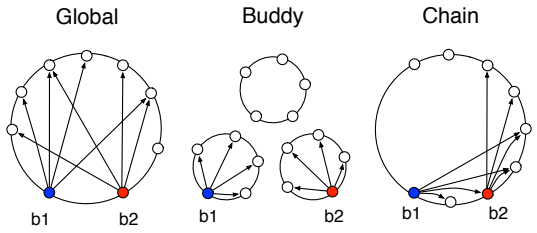

Fig. 1. Placement of two blocks $b1$ and $b2$ in the system using the different policies.

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 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 [2], but we do not discuss them here due to lack of space.

Related Work

The majority of existing or proposed systems, e.g., CFS, Farsite [6], PAST, TotalRecall [1], use a local placement policy. For example, in PAST [13], the authors use the Pastry DHT to store replicas of data into logical neighbors. In the opposite way, some systems use a Global policy, as OceanStore [11] or GFS [7]. GFS spreads chunks of data on any server of the system using a pseudo-random placement. Chun et al. in [3] and Ktari et al. in [10] discuss the impact of data placement. The later 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 [12], 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 [14].

2 Placement Policies

It has been shown that fragment placement has a strong impact on the system performance [8,12]. 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:

• *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

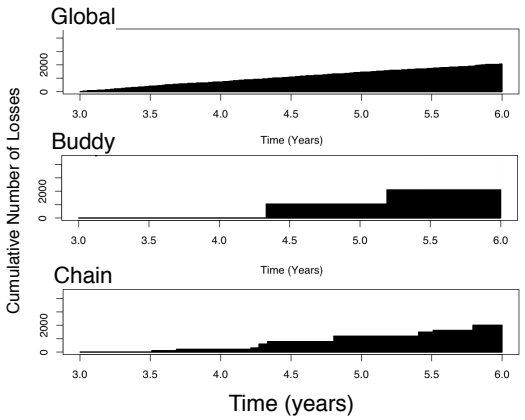

Fig. 2. Illustrative example of the cumulative number of block losses for a period of three years. The number of losses is the same among policies, but its distribution over time is different.

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 [8]. However, the use of Buddy and Chain, namely *local* strategies, brings practical advantages [4,3]. 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.

Mean Time to Data Loss (MTTDL). However, as stated in [8], the measure of the time to the first occurrence of a 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). 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 Buddy Placement Policy

In the next three sections (Section 3, 4 and 5), we present methods to compute exact values and approximations of the MTTDL for the three placement policies. For each policy, we calculate the probabilty $\mathbb{P}_{policy}$ to lose data at any given time step. Then, we deduce $MTTDL_{policy} = 1/\mathbb{P}_{policy}$.

In the Buddy placement policy, the N peers are divided into C clusters of size $s + r$ each. In this strategy, the calculation of the MTTDL_{buddy} is straightforward. Given a cluster, the probability to have a block loss is the probability that the cluster loses at least $r + 1$ peers (i.e., fragments), is given by

$$\mathbb{P}_{cluster} = \sum_{j=r+1}^{s+r} \binom{s+r}{j} \alpha^j (1-\alpha)^{s+r-j}. \tag{1}$$

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 $\mathbb{P}_{buddy} = 1 - (1 - \mathbb{P}_{cluster})^C$.

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. In other words, this assumption means that the probability of a peer failure α is small. Since the ratio between two consecutive terms in sum of Equation (1) is $\leq (s + r)\alpha$, we can bound its tail by a geometric series and see that it is of $O((s + r)\alpha)$. We obtain $\mathbb{P}_{cluster} \approx \binom{s+r}{r+1}\alpha^{r+1}$. Then we have

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

4 Global Placement Policy

In the Global policy, block's fragments are parted between $s + r$ peers chosen uniformly at random. First, we present the exact calculation of the MTTDL_{global}. We then present approximated formulas that give an intuition of the system behavior.

4.1 MTTDL calculation

First, we consider i failures happening during one time step. Let F denote the set of the placement classes (i.e., groups of $s + r$ peers) that hold at least $r + 1$ of these i failures; we have:

$$\#F = \sum_{j=r+1}^{i} \binom{i}{j} \binom{N-i}{s+r-j} \tag{3}$$

Then, suppose we insert a new block in the system: his $s+r$ fragments are dispatched randomly in one of the $\binom{N}{s+r}$ placement classes with uniform probability. Thus, the probability $\mathbb{P}_{block}(i)$ for the chosen class to be in F is:

$$\mathbb{P}_{block}(i) := \mathbf{P}\left[\text{placement in } F\right] = \frac{\sum_{j=r+1}^{i}\binom{i}{j}\binom{N-i}{s+r-j}}{\binom{N}{s+r}}$$

As block insertions are *independent*, if we consider our B blocks one after the other, the probability that none of them falls in F is $(1-\mathbb{P}_{block}(i))^B$. We then come back to the global probability to lose data considering different failure scenarii:

$$\begin{aligned}\mathbb{P}_{global} := \mathbf{P}\left[\text{lose data}\right] &= \mathbf{P}\left[\bigcup_{\{i \text{ failures}\}}[\text{failure kills a block}]\right] \\ &= \sum_{i=r+1}^{N}\binom{N}{i}\alpha^i(1-\alpha)^{N-i}\mathbf{P}\left[i \text{ failures kill a block}\right]\end{aligned}$$

Which gives us the MTTDL of the system using the global policy:

$$\text{MTTDL}_{global}^{-1} \approx \sum_{i=r+1}^{N}\binom{N}{i}\alpha^i(1-\alpha)^{N-i}\left[1-\left(1-\frac{\sum_{j=r+1}^{i}\binom{i}{j}\binom{N-i}{s+r-j}}{\binom{N}{s+r}}\right)^B\right] \quad (4)$$

4.2 MTTDL approximation

We provide here approximations for systems with low peer failure rates: *backup systems*. One example is *Brick storage systems* [12]. Each peer is a "brick" dedicated to data storage, that is, a stripped down computer with the fewest possible components: CPU, motherboard, hard drive and network card. In these backup systems, as we want a very high data life time, we have either $\alpha N \ll 1$ or $\alpha N \sim 1$, i.e., we have a not too high mean number of peer failures per time step.

Computations of this complicated sum suggests that only its first terms matter, and especially the very first term when $\alpha N \ll 1$. We can formalize this: let us consider three "zones" for $i \in [\![r+1, N]\!]$: (I) $i \sim s+r$, (II) $s+r \ll i \ll N$ and (III) $i \sim N$. We introduce the following notations:

$$\begin{aligned} A_i &= \textstyle\sum_{j=r+1}^{s+r}\binom{i}{j}\binom{N-i}{s+r-j} \; ; \; C_i = 1-\frac{A_i}{\binom{N}{s+r}} \\ \Gamma_i &= 1-C_i^B \qquad\qquad\qquad ; \; \Delta_i = \binom{N}{i}\alpha^i(1-\alpha)^{N-i}\Gamma_i \end{aligned}$$

Where A_i is nothing but $\#F$ in case i failures happen. In fact, and for the sake of curiosity, we can compute it easily with the following relation.

Lemma 1. *For $i \geq r+1$, $A_{i+1} = A_i + \binom{i}{r}\binom{N-(i+1)}{s-1}$.*

Proof. F is the set of placement classes with at least $r+1$ of them falling into a given "failure" set of size i. Let us see what happens when we increment the size of this failure set. We denote by S_i the initial failure set of F and $S_{i+1} = S_i \cup \{x\}$. A placement class falls in S_{i+1} iff it has at least $r+1$ peers in it, which is equivalent to either (a) having

more that $r+1$ peers in S_i or (b) containing x and *exactly* r peers in S_i (cases where there are more than $r+1$ peers in S_{i+1}, including x, are already counted in (a)). From this we conclude that: $A_{i+1} = A_i + \binom{i}{r}\binom{N-(i+1)}{s-1}$.

The ratio between two consecutive terms of sum (4) is:

$$\rho := \frac{\Delta_{i+1}}{\Delta_i} \;=\; \frac{\alpha}{1-\alpha}\frac{N-i+1}{i+1}\frac{\Gamma_{i+1}}{\Gamma_i} \;\approx\; \alpha N \cdot \frac{\Gamma_{i+1}}{i\Gamma_i} \tag{5}$$

In zones (II) and (III), we can show this ratio is low enough so we can bound the tail of our sum by a geometric series of common ration $\rho \ll 1$.

Lemma 2. *In zone (I), under the assumption* $\frac{N}{(s+r)^2} \gg 1$*,*

$$\Delta_i \approx B\binom{s+r}{r+1}(\alpha N)^{i-(r+1)}\alpha^{r+1}(1-\alpha)^{N-i} \tag{6}$$

Proof. When $i \sim s+r$, we usually (read: in practice) have $A \Big/ \binom{N}{s+r} \ll 1$. Under our (strong) assumption, which is also verified in practice, we indeed have the simple bound $A \Big/ \binom{N}{s+r} \leq \left(\frac{(s+r)^2}{N}\right)^{r+1} \frac{s}{(r+1)!} \ll \frac{1}{B}$. Thus, Γ_i is almost proportional to C_i in zone (I), which implies $\Delta_i \approx B\alpha^i(1-\alpha)^{N-i}A\binom{N}{i} \Big/ \binom{N}{s+r}$. But simple combinatorics show that $A\binom{N}{i} = \sum_{j=r+1}^{s+r}\binom{s+r}{j}\binom{N-(s+r)}{i-j}\binom{N}{s+r}$, leading us to equation (6).

Lemma 3. *In zone (II),* $\rho \approx \frac{\alpha N}{i}$*.*

Proof. When $s+r \ll i \ll N$, we have

$$\begin{aligned}
A_i &\approx \sum_{j=r+1}^{s+r} \frac{i^j}{j!}\frac{(N-i)^{s+r-j}}{(s+r-j)!} \\
C_i &\approx \left(1-\frac{i}{N}\right)^{s+r}\sum_{j=0}^{r}\binom{s+r}{j}\left(\frac{i}{N-i}\right)^j \\
&\approx \sum_{j=0}^{r}\binom{s+r}{j}\left(\frac{i}{N}\right)^j \sum_{l=0}^{s+r-j}(-1)^l\left(\frac{i}{N}\right)^l
\end{aligned}$$

Taylor expansion to second order in $\frac{i}{N}$ leads us to $\Gamma_i \approx B\left[2(s+r)-3\right]\left(\frac{i}{N}\right)^2$. Hence we see that $\frac{\Gamma_{i+1}}{\Gamma_i} \approx \left(1+\frac{1}{i}\right)^2 \approx 1$, equation (5) leading us to $\rho \approx \alpha N/i$.

Lemma 4. *In zone (III),* $\rho \leq \frac{\alpha N}{i}$*.*

Proof. Let $\epsilon_i = 1 - \frac{i}{N}$: when $i \sim N$, we have $C_i \approx \sum_{j=0}^{r}\left(\frac{i}{N}\right)^j \epsilon_i^{s+r-j}\binom{s+r}{j} \approx \epsilon_i^s\binom{s+r}{r}$. Hence, $C_{i+1} - C_i \approx \frac{1}{N^s}\left(\epsilon_{i+1}^{s-1} + \cdots + \epsilon_i^{s-1}\right)\binom{s+r}{r} \leq \frac{1}{N^s}s\epsilon_i^{s-1}\binom{s+r}{r} \ll 1$. Then, Taylor expansion of the convex function $f(x) = 1 - x^B$ leads us to ($f'' < 0$):

$$\begin{aligned}
\Gamma_{i+1} - \Gamma_i &\leq (C_{i+1} - C_i)f'(C_i) \\
&\leq \frac{1}{N^s}s\epsilon_i^{s-1}\binom{s+r}{r}BC_i^{B-1} \\
\frac{\Gamma_{i+1}}{\Gamma_i} &\leq 1 + \frac{B\epsilon_i^{s-1}s\binom{s+r}{r}}{N^s}\frac{C_i^{B-1}}{1-C_i^B}
\end{aligned}$$

Since in practice we have $B \ll N^s$, this upper bound is close to 1 and we conclude – as usual – with equation (5) giving $\rho \leq \alpha N/i$.

Lemmas 3 and 4 tell us that, when $i \gg s + r$, our big sum is bounded by a geometric series of common ratio $\leq \frac{\alpha N}{i} \ll 1$, so only the terms before zones (II) and (III) numerically matter.

Lemma 2 can provide us with a stronger result. Equation (6) leads to $\rho \approx \alpha N$ in zone (I). Hence, if we also have $\alpha N \ll 1$, that is, mean number of failures per time step is really low (or, equivalently, time step is short enough), then only the first term of the sum matters. If we simplify it further, we find:

$$\mathrm{MTTDL}_{global} \approx \frac{1}{B\binom{s+r}{r+1}\alpha^{r+1}} \tag{7}$$

5 Chain Placement Policy

For the Chain policy, the computation of MTTDL_{chain} is more difficult than the two previous ones, mainly because the chains are not independent of each other. 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.

We present in this paper two approaches to compute or approximate the MTTDL for the Chain policy. We first describe computations using Markov chains techniques, and we then describe an analytical approximation value assuming that α is small enough.

5.1 Markov Chain Approach

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, $i \leq j < i+s+r$. Peer numbered $N+k$ is really the peer numbered k. Remark that the binary-vector of S_{i+1} is $(b_{i+1}, \ldots, b_{i+s+r})$.

As an example, consider a system composed of $N = 10$ peers with the values $s = 3$ and $r = 2$. The first sequence S_1 of peers is associated with the vector $(b_1, \ldots, b_5)$. If $\sum_{i=1}^{5} b_i \geq 3$, then it means that there is a data loss. Otherwise we have for example the vector $(0, 0, 1, 0, 0)$. Thus we now look at the vector $(b_2, \ldots, b_6)$ associated with the second sequence S_2 of peers. To get this new vector, we remove the first bit b_1 of the previous vector and we add the new bit b_6 at the end. We get for example $(0, 1, 0, 0, 1)$ if $b_6 = 1$. Two peer failures appear in the sequence S_2, and so we do not have a data loss. If for example $b_7 = 1$, then the vector associated with S_3 is $(1, 0, 0, 1, 1)$. In that case a data loss is found.

We now want to compute the probability to find at least one "bad" sequence S_i containing at least $r+1$ bits 1 in its vector. 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

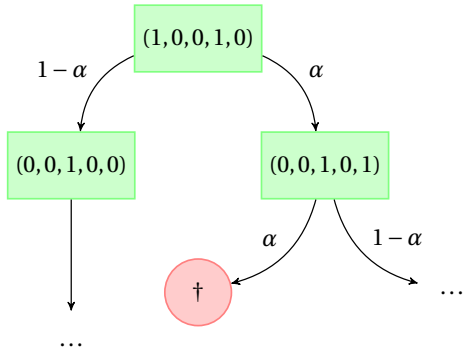

Fig. 3. Sample part of the Markov chain for $s + r = 5$ and $r + 1 = 3$.

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 [5].

Consider the previous example with $s = 3$ and $r = 2$. Figure 3 describes the two possible transitions from the state $(1, 0, 0, 1, 0)$ (corresponding to the current sequence S_i): the last peer of the next sequence S_{i+1} is failed with probability α, and it is not failed with probability $1 - \alpha$. The two possible states are $(0, 0, 1, 0, 1)$ and $(0, 0, 1, 0, 0)$, respectively. Furthermore from state $(0, 0, 1, 0, 1)$, it is possible to transit to state v_{dead} because with probability α the vector of the next sequence is $(0, 1, 0, 1, 1)$.

First, we assume that the N peers are ordered in a *line* instead of a *ring*. In other words we do not take into cosideration such vectors of sequences: $(\ldots, b_N, b_1, \ldots)$. In that case we look at $N - (s + r) + 1$ sequences. We 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 $\mathbb{P}_{line}$ is $\pi(v_{dead})$.

To get the value $\mathbb{P}_{chain}$, we have to carefully take into consideration sequences containing peers on both borders of the network (becoming a ring again). The concerned sequences admit vectors $(\ldots, b_N, b_1, \ldots)$. 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. Lemma 5 proves that we can reduce this number of states showing some properties.

Lemma 5. *There exists a Markov chain having the same $\pi(v_{dead})$ such that:*

$$|V| = 1 + \sum_{i=0}^{r} \binom{s+r}{i} - \sum_{k=1}^{r} \sum_{j=0}^{k-1} \binom{s+k-1}{j} \tag{8}$$

Proof. One of the peer failures in the chain is meaningful if and only if it can be present in some following chain containing at least $r + 1$ failures. For example, in the state $(1, 0, \ldots, 0)$, the first dead is not meaningful because, even if we have r dead peers

following, it will be too far away to make a chain with $r+1$ peer failures. In this sense, states $(0,0,\ldots,0)$ and $(1,0,\ldots,0)$ are equivalent and we can merge them.

More generally suppose we have k peer failures in the current state (current sequence of peers): we miss $r+1-k$ peer failures to make a data loss; hence, a peer failure in the current sequence will have incidence if and only if it is one of the last $s+k-1$ peers of the chain: otherwise, even if the next $r+1-k$ peers are dead, they won't fit with our k deads in a frame of size $s+r$.

Thus, among all the states with k peer failures, only those where all failures are in the tail of size $s+k-1$ are meaningful. As to the others, the first failures do not matter and we can forget them. This merging algorithm leads us to state space size (8): in a nutshell, we forget all states with k failures and less than k peer failures in the tail of size $s+k-1$.

We presented a method to compute the exact value of $\mathbb{P}_{chain}$ ($\text{MTTDL}_{chain} = 1/\mathbb{P}_{chain}$). We now propose a simple method to approximate the MTTDL using Absorbing Markov chains techniques. We first consider that the number of peers is infinite. In fact peers numbered i, $i+N$, $i+2N$, ..., $i+kN$, ... represent the same peer numbered i but at different time steps. 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 $MTTDL_{chain} \approx \lfloor t_{abs}/N \rfloor$. Indeed let P and Q denote the transition matrices of respectively the complete chain (described before) and the sub-chain where we removed the absorbing state and all its incident transitions. Then the fundamental matrix $R = (I-Q)^{-1}$ gives us the time to absorption t_{abs} starting from any state (see [9] for details). t_{abs} is not exactly the MTTDL since $N-(s+r)$ steps correspond to one time step (we survey the whole ring). Hence, $\lfloor t_{abs}/N \rfloor$ gives us the expected number of *time* steps before we reach the absorbing state, which is, this time, the MTTDL we are looking for.

5.2 Analytical Approximation

In the rest of this section, a *syndrome* is a sequence of $s+r$ consecutive peers containing at least $r+1$ peer failures. Under the assumption that α is "small enough" (we will see how much), we can derive an analytical expression of the MTTDL.

$$MTTDL_{chain} \approx \frac{1}{N\frac{r+1}{s+r}\binom{s+r}{r+1}\alpha^{r+1}}. \tag{9}$$

Let us begin with two lemmas.

Lemma 6. *The probability to have two distinct syndromes is negligible compared to the probability to have only one and bounded by*

$$\mathbf{P}\left[\exists \text{ two distinct syndromes } / \exists \text{ a syndrome}\right] < \frac{\alpha N(s+r)\cdot(\alpha(s+r))^{r-1}}{r!} \tag{10}$$

Proof. The probability for a syndrome to begin at a given peer (the beginning of a syndrome being considered as his first peer failure) is given by $p = \alpha\sum_{i=r}^{s+r-1}\binom{s+r-1}{i}\alpha^i(1-$

$\alpha)^{s+r-1-i}$. Meanwhile, we have

$$\mathbf{P}\left[\exists \text{ 2 distinct syndromes}\right] = \mathbf{P}\left[\cup_{|i-j|\geq s+r} \exists \text{ 2 syndromes beginning at peers } i \text{ and } j\right],$$

which is $\leq \binom{N}{2}p^2 < (pN)^2$. Normalizing by pN gives us the probability to have two syndromes knowing that there is at least one:

$$\mathbf{P}\left[\exists \text{ two distinct syndromes} \mid \exists \text{ a syndrome}\right] < pN.$$

Hence, we would like to show that pN is negligible. An upper bound on p is easy to figure out: given that $\alpha(s+r) \ll 1$, we have $p \approx \binom{s+r-1}{r}\alpha^r(1-\alpha)^{s-1} \leq (\alpha(s+r))^r/r!$, and so $pN \leq (\alpha N(s+r))(\alpha(s+r))^{r-1}/r!$. Hence, assuming $\alpha N(s+r) \ll 1$ (or otherwise $r \geq \log N$) suffices to conclude.

Lemma 7. *The probability to have more than $r+1$ dead peers in a given syndrome is negligible and bounded by*

$$\mathbf{P}\left[\exists > r+1 \textit{ dead peers} \mid \exists \geq r+1 \textit{ peers}\right] < \alpha(s+r) \tag{11}$$

Proof. Since we are working in a syndrome, the probability we want to bound is, in a given chain:

$$\begin{aligned}\mathbf{P}\left[\exists > \text{r+1 dead peers} \mid \exists \geq \text{r+1 dead peers}\right] &= \frac{\sum_{r+2}^{s+r}\binom{s+r}{i}\alpha^i(1-\alpha)^{s+r-i}}{\sum_{r+1}^{s+r}\binom{s+r}{i}\alpha^i(1-\alpha)^{s+r-i}} \\ &\leq \frac{\sum_{r+2}^{s+r}\binom{s+r}{i}\alpha^i(1-\alpha)^{s+r-i}}{\binom{s+r}{r+1}\alpha^{r+1}(1-\alpha)^{s-1}}\end{aligned}$$

Since the ratio between a term of the binomial series and its predecessor is $\frac{\alpha}{1-\alpha}\cdot\frac{s+r-i}{i+1}$, we can bound the tail of the binomial sum by a geometric series of common ratio $q = \frac{\alpha}{1-\alpha}\cdot\frac{s-1}{s+r} \ll 1$. Thus we have:

$$\mathbf{P}\left[\exists > \text{r+1 dead peers} \mid \exists \geq \text{r+1 dead peers}\right] < \frac{\alpha}{1-\alpha}\cdot\frac{s-1}{r+2}\cdot\frac{1}{1-q} < \alpha(s+r) \ll 1.\square$$

Therefore, if we only look for a single syndrome with exactly $r+1$ dead peers, we get a close approximation of the MTTDL.

$$\begin{aligned}\mathbb{P}_{chain} &= \mathbf{P}\left[\exists \text{ one syndrome}\right] \\ &= \mathbf{P}\left[\cup_i \exists \text{ one syndrome beginning at peer } i\right] \\ &= (N-(s+r))p\end{aligned}$$

Indeed, since there is only one syndrome, the events [syndrome begins at peer i] are exclusives. Here p is the probability for the syndrome to begin at a given peer, which we saw in proof of lemma 6. Given lemma 7, we can approximate it by $\binom{s+r-1}{r}\alpha^{r+1}(1-\alpha)^{s-1}$, which leads us too:

$$\text{MTTDL}'_{chain} \approx \frac{1}{N\binom{s+r-1}{r}\alpha^{r+1}} \tag{12}$$

One may notice that this is the same formula as (2) in the Buddy case with $c = N\frac{r+1}{s+r}$.

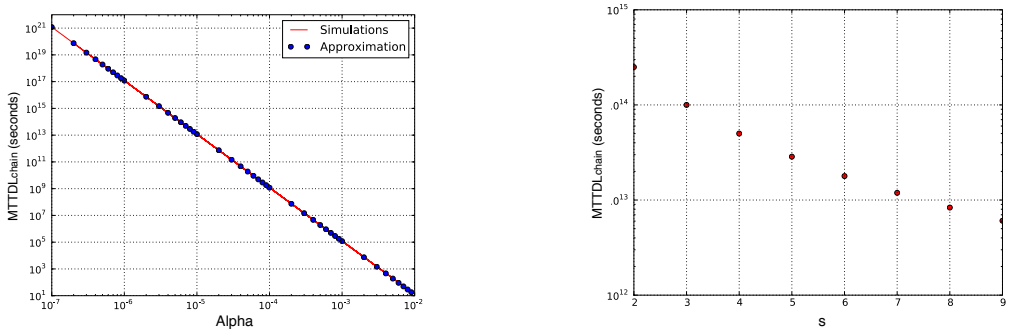

Fig. 4. Behavior of the MTTDL_{chain} when varying α (left) and s (right).

Behavior of the MTTDL Simulations led with common values of the parameters ($\alpha = 10^{-5}$, $s = 7$, $r = 3$) suggest that approximation (12) succeeds in describing the behavior of the MTTDL, as e.g. depicted by Figure 4.

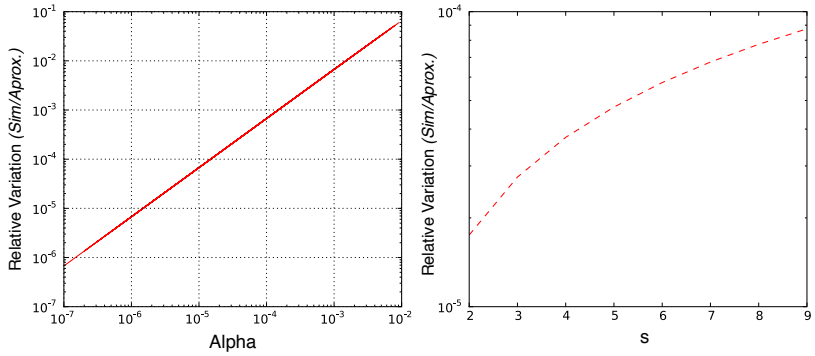

Fig. 5. Impact of α and s on the relative variation between simulation and approximation of the MTTDL_{chain}.

Validity of the approximation We have been able to compare the approximation with the exact results given by the MCM in cases where space size (8) was low enough (roughly $s < 15$ and $r < 5$), see Figure 5 for sample values. Numerical results suggested formula (12) was a good approximation for $\alpha < 10^{-3}$, s having little influence (and r almost none) on the relative variation between simulation and approximation.

6 Discussion and Conclusion

The approximations given by the Equations (2), (7), and (9) 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 $B(s+r)/N$.

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

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. R. Bhagwan, K. Tati, Y. chung Cheng, S. Savage, and G. M. Voelker. Total recall: System support for automated availability management. In *Proc. of NSDI*, pages 337–350, 2004.
2. 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/.
3. B.-G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, M. F. Kaashoek, J. Kubiatowicz, and R. Morris. Efficient replica maintenance for distributed storage systems. In *Proc. of the NSDI'06*, pages 45–58, Berkeley, CA, USA, 2006. USENIX Association.
4. 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, USA, 2004.
5. N. De Bruijn. A combinatorial problem. *Kibern. Sb., Nov. Ser.*, 6:33–40, 1969.
6. J. R. Douceur and R. P. Wattenhofer. Large-scale simulation of replica placement algorithms for a serverless distributed file system. In *Proc. of MASCOTS*, pages 311–319, 2001.
7. S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. *19th ACM Symposium on Operating Systems Principles*, October 2003.
8. 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.
9. C. M. Grinstead and L. J. Snell. *Grinstead and Snell's Introduction to Probability*. American Mathematical Society, version dated 4 july 2006 edition, 2006.
10. 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, USA, 2007. ACM.
11. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, C. Wells, et al. OceanStore: an architecture for global-scale persistent storage. *ACM SIGARCH Computer Architecture News*, 28(5):190–201, 2000.
12. 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.
13. A. Rowstron and P. Druschel. Storage management and caching in past, a large-scale, persistent peer-to-peer storage utility. In *Proc. ACM SOSP*, pages 188–201, 2001.
14. H. Weatherspoon and J. Kubiatowicz. Erasure coding vs. replication: A quantitative comparison. In *Proc. of IPTPS*, volume 2, pages 328–338. Springer, 2002.