P2P storage systems modeling, analysis and evaluation

Abdulhalim Dandoush, Sara Alouf, Philippe Nain

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

P2P storage systems modeling, analysis and evaluation

Abdulhalim Dandoush — Sara Alouf — Philippe Nain

N° ????

Décembre 2007

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--????--FR+ENG

P2P storage systems modeling, analysis and evaluation

Abdulhalim Dandoush , Sara Alouf , Philippe Nain

Thème COM — Systèmes communicants
Projet MAESTRO

Rapport de recherche n° ???? — Décembre 2007 — 30 pages

Abstract: This Report 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 is centralized and relies on a server that recovers multiple losses at once, whereas the second is distributed and recovers one loss at a time. For each scheme, we propose a basic Markovian model where the availability of peers is exponentially distributed, and a more elaborate model where the latter is hyper-exponentially distributed. Our models equally apply to many distributed environments as shown through numerical computations. These allow to assess the impact of each system parameter on the performance. In particular, we provide guidelines on how to tune the system parameters in order to provide desired lifetime and/or availability of data. One important outcome of our analysis is that a simplifying exponential assumption on the peers availability leads to incorrect evaluation of the performance achieved. Thereby, the more elaborate model is necessary to capture the true behavior of peer-to-peer storage systems.

Key-words: Peer-to-Peer systems, performance evaluation, absorbing Markov chain, mean-field approximation

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Modèlisation, analyse et evaluation des systèmes pair-à-pair de stockage de données

Résumé : Ce rapport évalue et compare les performances des systèmes de stockage de données sur des réseaux de pairs en termes de longévité des données et de leur disponibilité. Deux mécanismes de récupération de données perdues sont pris en considèration. Le premier mécanisme est centralisé et repose sur l'utilisation d'un serveur pouvant récupérer plusieurs données à la fois alors que le second mécanisme est distribué. Pour chaque mécanisme, nous avons proposé d'une part un modéle Markovien de base ou la disponibilité des machines sont exponentiellement distribuées, et d'autre part un modéle plus compliqué ou la disponibilité des machines sont hyper-exponentiellement distribuées. Nos modèles s'appliquent dans différents environnements distribués comme montré par les calcules numériques. Ceux-ci permettent d'évaluer l'impact de chaque paramétre de système sur la performance. En particulier, Nous montrons comment nos résultats peuvent être utilisés de sorte à garantir que la qualité de service pré-requise soit pourvue. Un résultat important de notre analyse est qu'une hypothèse exponentielle simple sur la disponibilité de pairs cause une évaluation incorrecte de la performance accomplie. Ainsi, le modèle plus compliqué est nécessaire de capturer le vrai comportement de systèmes pair-à-pair de stockage de données.

Mots-clés : systèmes pair-à-pair, évaluation de performance, chaîne de Markov absorbante, approximation champ moyen

Contents

1 Introduction

Traditional storage solutions rely on robust dedicated servers and magnetic tapes on which data are stored. These equipments are reliable, but expensive and do not scale well. The growth of storage volume, bandwidth, and computational resources has fundamentally changed the way applications are constructed, and has inspired a new class of storage systems that use distributed peer-to-peer (P2P) infrastructures. Some of the recent efforts for building highly available storage system based on the P2P paradigm are Intermemory [12], Freenet [10], OceanStore [22], CFS [11], PAST [16], Farsite [21] and Total Recall [8]. Although scalable and economically attractive compared to traditional systems, these storage systems pose many problems of reliability, confidentiality, availability, routing, 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. Redundancy can be achieved either by replication or by using erasure codes. For the same amount of redundancy, erasure codes provide higher availability of data than replication [18].

However, using redundancy mechanisms without repairing lost data is not efficient, as the level of redundancy decreases when peers leave the system. Consequently, 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 – thanks to its secure agent – recovers only one loss per document. Also, the distributed-recovery mechanism generates more management traffic than the centralized-recovery mechanism. However, the centralized solution can become computationally very heavy and poses the problem of a single-point of failure.

Regardless of the recovery mechanism used, two repair policies can be adopted. In the *eager* policy, when the system detects that one host has left the network, it immediately initiates the reconstruction of the lost data that once recovered will be stored on new peers. Using this policy, data only becomes unavailable when hosts fail more quickly than failures can be detected and repaired. This policy is simple but makes no distinction between permanent departures that need to be recovered, and transient disconnections that do not.

Having in mind that connections may experience temporary failures, one may want a system that defers the repair beyond the detection of first data loss. This alternative policy inherently uses less bandwidth than the eager policy. However, it is obvious that an additional redundancy is necessary to mask and to tolerate host departures for the extended period. This approach is called *lazy* repair because the explicit goal is to delay repair work for as long as possible.

In this paper, we aim at developing mathematical models to characterize fundamental performance metrics (lifetime and availability – see next paragraph) of P2P storage systems. We are interested in evaluating the centralized- and distributed-recovery mechanisms discussed earlier, when either eager or lazy repair policy is enforced. We will focus our study on the quality of service delivered to each block of data. We 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?*

The *lifetime* of data in the P2P system is a random variable; we will investigate its distribution function. *Data availability* metrics refer to the amount of redundant fragments. We will consider two such metrics: the expected number of available redundant fragments, and the fraction of time during which the number of available redundant fragment exceeds a given threshold. For each data recovery implementation (centralized/distributed) we will derive these metrics in closed-form through a Markovian analysis.

In the following, Sect. 2 briefly reviews related work and Sect. 3 introduces the notation and assumptions used throughout the paper. Sections 4 and 5 are dedicated to the modeling of the centralized- and distributed-recovery mechanism, considering two different distributions for peer availability. In Sect. 6, we provide numerical results showing the performance of the centralized and decentralized schemes, under eitehr the eager or the lazy policy. We further discuss some important issues in different contexts using the parameters of four real distributed environments. Section 7 concludes the paper.

2 Related work and background

The literature on the architecture and file system of distributed storage systems is abundant (see [12, 22, 11, 16, 21, 8]; non-exhaustive list) but only a few studies have developed analytical models of distributed storage systems to understand the trade-offs between the availability and lifetime of the files and the redundancy involved in storing the data.

In [18], Weatherspoon and Kubiatowicz characterize the availability and durability gains provided by an erasure-resilient system. They quantitatively compare replication-based and erasure-coded systems. They show that erasure codes use an order of magnitude less bandwidth and storage than replication for systems with similar durability. Utard and Vernois perform another comparison between the full replication mechanism and erasure codes through a simple stochastic model for node behavior [17]. They observe that simple replication schemes may be more efficient than erasure codes in presence of very low peers availability. A thorough analysis of erasure codes under different scenarios is performed in [13], where Lin, Chiu and Lee consider two key parameters: the peer availability level and the storage overhead.

In [9], Blake and Rodrigues argue that the cost of dynamic membership makes the cooperative storage infeasible in transiently available peer-to-peer environments. In other words, when redundancy, data scale, and dynamics are all high, the needed cross-system bandwidth is unreasonable when clients desire to download files during a reasonable time.

Characterizing machine availability both in local and wide area environments has been the focus of [19]. In this technical report, Nurmi, Brevik and Wolski analyze three sets of data each measuring machine availability in a different setting, and perform goodness-of-fit tests on each data set to assess which out of four distributions best fits the data. They have found that a hyper-exponential model fits more accurately the machine availability durations than the exponential, Pareto, or Weibull distribution. More recently, Ramabhadran and Pasquale analyzed the *All-pairs-ping* data set [24] that reports measures of both uptime and downtime for Planetlab [23] nodes. By plotting the cumulative distribution function of each duration (uptime/downtime), they conjecture in [15] that an exponential distribution is a reasonable fit for both uptime and downtime. This conjecture comes to support one of the key assumptions of the model presented in that paper, namely that "node participation can be modeled by an exponential distribution".

In fact, the main purpose of [15] is the analysis of a storage system using *full replication* for data reliability, so in this aspect, [15] is the closest work to ours even though their model and analysis do not apply for erasure-coded systems (we will see later that our models apply to either replicated or erasure-coded systems). The authors of [15] develop a Markov chain analysis, then derive an

expression for the lifetime of the replicated state and study the impact of bandwidth and storage limits on the system. However – and this is another major difference with the work presented here, transient disconnections are not considered in their model.

3 System description and notation

In the following, we will distinguish the *peers*, which are computers where data is stored and which form a storage system, from the *users* whose objective is to retrieve the data stored in the storage system.

We consider a distributed storage system in which peers randomly join and leave the system. Upon a peer disconnection, all data stored on this peer is no longer available to the users of the storage system and is considered to be lost. In order to improve data availability and increase the reliability of the storage system, it is therefore crucial to add redundancy to the system.

In this paper, we consider a single block of data D, divided into s equally sized fragments to which, using erasure codes (e.g. [6]), 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. We assume that at least s fragments are available at time $t = 0$. Note that this notation can also serve to model systems using replication instead of erasure codes, in which case $s = 1$ and the r redundant fragments will simply be replicas of the unique fragment of the block. This notation – and hence our modeling – is general enough to study both replication-based and erasure codes-based storage systems.

Over time, a peer can be either *connected* to or *disconnected* from the storage system. At reconnection, a peer may still or may not store one fragment. We denote by p the probability that a peer that reconnects still stores one fragment and that this fragment is different from all other fragments available in the system.

We refer to as *on-time* (resp. *off-time*) a time-interval during which a peer is always connected (resp. disconnected). We 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). We further assume that peers behave independently of each other, which implies that on-time and off-time sequences associated with any set of peers are statistically independent.

The off-times are assumed to be exponentially distributed with parameter $\lambda > 0$, in agreement with the analysis of [15]. As for the on-times, we first assume them to be exponentially distributed with parameter $\mu > 0$ (see previous work [7]). The resulting model will be denoted “the basic model”. However, in light of the analyses reported in [19, 15], we realized that different distributed environments may exhibit different on-times distributions. Therefore, we propose and analyze a more elaborate model, called “the improved model”, in which the distribution of on-times durations is hyper-exponential with n phases; the parameters of phase i are $\{p_i, \mu_i\}$. The basic model is in fact a special case of the improved model when $n = 1$.

Data stored on a connected peer is available at once and can be used to retrieve or reconstruct a block of data. Typically, the number of connected peers at any time in a storage system is much larger than the number of fragments associated with a given data D. Therefore, we assume that there

are always at least r connected peers – hereafter referred to as *new* peers – which are ready to receive and to store fragments of D. A peer may store at most one fragment of D.

As discussed in Sect. 1 we 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. In the latter case we must have that $k \leq r$ since D is lost if more than r fragments are not available in the storage system at a given time. Both repair policies can be represented by the threshold parameter $k \in \{1, 2, \ldots, r\}$, where k can take any value in the set $\{2, \ldots, r\}$ in the lazy policy and $k = 1$ in the eager policy.

Let us now describe the fragment recovery mechanism. As mentioned in Sect. 1, we will consider two implementations of the eager and lazy recovery mechanisms, a *centralized* and a *distributed* implementation. Assume that $k \leq r$ fragments are no longer available due to peer disconnections so that lost data have to be restored.

In the centralized implementation, a central authority will: (i) download s fragments from the peers which are connected, (ii) reconstruct at once the k unavailable fragments, and (iii) transmit each of them to a new peer for storage. We will assume that the total time required to perform these tasks is exponentially distributed with rate $\beta_c(k) > 0$ and that successive recoveries are statistically independent.

In the distributed implementation, a secure agent on one new peer is notified of the identity of *one* out of the k unavailable fragments for it to reconstruct it. Upon notification, the secure agent downloads s fragments of D from the peers which are connected, reconstructs the specified fragment and stores it on the peer's disk; the secure agent then discards the s downloaded fragments so as to meet the security constraint that only one fragment of a block of data is held by a peer. This operation iterates until less than k fragments are sensed unavailable.

We will assume that the total time required by a secure agent to perform the download, reconstruct and store a new fragment follows an exponential distribution with rate $\beta_d > 0$; we assume that each recovery is independent of prior recoveries and that concurrent recoveries are also mutually independent rvs.

The exponential distribution for the recovery process has mainly been assumed for the sake of mathematical tractability. We however believe that this is a reasonable assumption due to the unpredictable nature of the peer/user dynamics, and to the variability of network delays and the bandwidth available at peers.

Table 1 recapitulates the parameters introduced in this section. We will refer to s, r and k as the *protocol* parameters, p, λ, μ and $\{p_i, \mu_i\}_{i=1,\ldots,n}$ as the *peers* parameters, and $\beta_c(k)$ and β_d as the *network* parameters.

We conclude this section by a word on the notation: a subscript "c" (resp. "d") will indicate that we are considering the centralized (resp. distributed) scheme; in the basic (resp. improved) model, we will add to the rvs a superscript "e" (resp. "h") referring to the assumption on the distribution of peers on-times: exponential in the basic model and hyper-exponential in the improved model. The notation $\mathbf{e}_j^{(i)}$ refers to a *row* vector of dimension j whose entries are null except the i-th entry that is

Table 1: System parameters.

D	Block of data
s	Original number of fragments of a given block
r	Number of redundant fragments
k	Threshold triggering the repair process
p	Persistence probability
λ	Peers arrival rate
μ	Peers failure rate in basic model
$\{p_i, \mu_i\}_{i=1,\ldots,n}$	Parameters of the peers failure process in improved model
$\beta_c(k)$	Recovery rate in centralized implementation
β_d	Recovery rate in distributed implementation

equal to 1; the notation $\mathbf{1}_j$ refers to a *column* vector of dimension j whose each entry is equal to 1. Last, $\mathbb{1}\{A\}$ is the characteristic function of event A.

4 Centralized repair systems

In this section, we address the performance of P2P storage systems using the centralized-recovery mechanism, as described in Sect. 3. We will focus on a single block of data D, and pay attention solely to peers storing fragments of this block.

Recall that we consider two different assumptions on the distribution of peers on-times. The exponential distribution with parameter μ is used in the basic model whose analysis has first appeared in [7]. This model is reviewed in Sect. 4.1 for completeness. The hyper-exponential distribution with n phases is considered in the improved model whose analysis is presented in Sect. 4.2.

4.1 The basic model

Let $X_c^e(t)$ be a $\{a, 0, 1, \ldots, r\}$-valued rv, where $X_c^e(t) = i \in \mathcal{T}^e := \{0, 1, \ldots, r\}$ indicates that $s + i$ fragments of D are available at time t, and $X_c^e(t) = a$ indicates that less than s fragments of D are available at time t. We assume that $X_c^e(0) \in \mathcal{T}^e$ so as to reflect the assumption that at least s fragments are available at $t = 0$.

If at a given time t a peer disconnects from the storage system while $X_c^e(t) = 0$, then there will be strictly less than s fragments of D in the system. Recovering then lost fragments is impossible unless one of the peers having a fragment of D reconnects to the system *and* still has its data. Recall that this happens with a probability p; in other words, recovering D becomes a probabilistic event. The block of data D is available with probability 1 as long as there are at least s fragments of D (implying $X_c^e(t) \geq 0$ but the other way round is not true). Otherwise, we consider the block D to be lost.

Thanks to the assumptions made in Sect. 3, it is easily seen that $\mathbf{X}_c^e := \{X_c^e(t),\, t \geq 0\}$ is an absorbing homogeneous Continuous-Time Markov Chain (CTMC) with transient states $0, 1, \ldots, r$

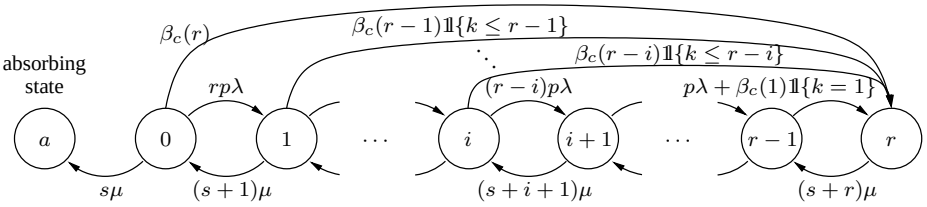


Figure 1: Transition rates of the absorbing Markov chain $\{X_c^e(t), t \geq 0\}$.

and with a single absorbing state a representing the situation when D is lost. Non-zero transition rates of $\{X_c^e(t),\ t \geq 0\}$ are shown in Fig. 1.

4.1.1 Data lifetime

This section is devoted to the analysis of the data lifetime. Let $T_c^e(i) := \inf\{t \geq 0 : X_c^e(t) = a\}$ be the time until absorption in state a starting from $X_c^e(0) = i$, or equivalently the time at which the block of data D is lost given that the initial amount of redundant fragments of D is i. In the following, $T_c^e(i)$ will be referred to as the *conditional block lifetime*.

We are interested in $P(T_c^e(i) \leq x)$ and $\mathrm{E}[T_c^e(i)]$, respectively the probability distribution and expectation of the block lifetime given that $X_c^e(0) = i$ for $i \in \mathcal{T}^e$.

Let $\mathbf{Q}_c^e = [q_c^e(i,j)]_{i,j\in\mathcal{T}^e}$ be a matrix, where for any $i \neq j$, the element $q_c^e(i,j)$ gives the transition rate of the Markov chain $\mathbf{X}_c^e$ from transient state i to transient state j, and $-q_c^e(i,i)$ for any $i \in \mathcal{T}^e$ is the total transition rate out of state i. Non-zero entries of $\mathbf{Q}_c^e$ are

$$\begin{array}{lcll} q_c^e(i,i-1) & = & a_i, & i = 1,2,\ldots,r, \\ q_c^e(i,i+1) & = & b_i + \mathbb{1}\{i = r-1\}c_{r-1}, & i = 0,1,\ldots,r-1, \\ q_c^e(i,r) & = & c_i, & i = 0,1,\ldots,\min\{r-k, r-2\}, \\ q_c^e(i,i) & = & -(a_i+b_i+c_i), & i = 0,1,\ldots,r, \end{array} \tag{1}$$

where $a_i := (s+i)\mu$, $b_i := (r-i)p\lambda$ and $c_i := \beta_c(r-i)\mathbb{1}\{i \leq r-k\}$ for $i \in \mathcal{T}^e$. Note that $\mathbf{Q}_c^e$ is not an infinitesimal generator since entries in its first row ($i = 0$) do not sum up to 0.

From the theory of absorbing Markov chains, we know that (e.g. [3, Lemma 2.2])

$$P(T_c^e(i) \leq x) \quad = \quad 1 - \mathbf{e}_{r+1}^{(i+1)} \cdot \exp\left(x\mathbf{Q}_c^e\right) \cdot \mathbf{1}_{r+1}, \quad x > 0, \quad i \in \mathcal{T}^e. \tag{2}$$

Recall from Sect. 3 that $\mathbf{e}_{r+1}^{(i+1)}$ and $\mathbf{1}_{r+1}$ are vectors of dimension $r+1$; all entries of $\mathbf{e}_{r+1}^{(i+1)}$ are null except the $(i+1)$-th entry (entry i) that is equal to 1, and all entries of $\mathbf{1}_{r+1}$ are equal to 1. The term $\mathbf{e}_{r+1}^{(i+1)} \cdot \exp\left(x\mathbf{Q}_c^e\right) \cdot \mathbf{1}_{r+1}$ is nothing but the summation of all $r+1$ elements in row i of matrix $\exp\left(x\mathbf{Q}_c^e\right)$.

We also know that the expectation of the time until absorption can be written as [3, p. 46],

$$\mathrm{E}\left[T_c^e(i)\right] \quad = \quad -\mathbf{e}_{r+1}^{(i+1)} \cdot \left(\mathbf{Q}_c^e\right)^{-1} \cdot \mathbf{1}_{r+1}, \quad i \in \mathcal{T}^e, \tag{3}$$

where the existence of $(\mathbf{Q}_c^e)^{-1}$ is a consequence of the fact that all states in $\mathcal{T}^e$ are transient [3, p. 45].

Consider now

$$T_c^e(i,j) \quad := \quad \int_0^{T_c^e(i)} \mathbb{1}\{X_c^e(t)=j\}dt$$

that is the total time spent by the CTMC in transient state j given that $X_c^e(0)=i$. It can also be shown that [1]

$$\mathrm{E}\left[T_c^e(i,j)\right] \quad = \quad -\mathbf{e}_{r+1}^{(i+1)} \cdot (\mathbf{Q}_c^e)^{-1} \cdot {}^t\mathbf{e}_{r+1}^{(j+1)}, \quad i,j \in \mathcal{T}^e, \tag{4}$$

where ${}^t\mathbf{e}_{r+1}^{(j+1)}$ is a column vector, transpose of $\mathbf{e}_{r+1}^{(j+1)}$. In other words, $\mathrm{E}[T_c^e(i,j)]$ is the (i,j)-th element of matrix $-(\mathbf{Q}_c^e)^{-1}$.

Even when $\beta_c(0)=\cdots=\beta_c(r)$, an explicit calculation of either $P(T_c^e(i)<x)$, $\mathrm{E}[T_c^e(i)]$ or $\mathrm{E}[T_c^e(i,j)]$ is intractable, for any value of the threshold k in $\{1,2,\ldots,r\}$. Numerical results for $\mathrm{E}[T_c^e(r)]$ and $P(T_c^e(r)>10$ years) are reported in Sect. 6.3 when $\beta_c(0)=\cdots=\beta_c(r)$.

4.1.2 Data availability

In this section we introduce different metrics to quantify the availability of the block of data. The fraction of time spent by the absorbing Markov chain $\{X_c^e(t),\, t\geq 0\}$ in state j starting at time $t=0$ from state i is

$$\mathrm{E}\left[\frac{1}{T_c^e(i)}\int_0^{T_c^e(i)} \mathbb{1}\{X_c^e(t)=j\}dt\right].$$

However, since it is difficult to find a closed-form expression for this quantity, we will instead approximate it by the ratio

$$\frac{\mathrm{E}[T_c^e(i,j)]}{\mathrm{E}[T_c^e(i)]}.$$

Note that we have validated this approximation by simulation, as shown in Fig. 8, Sect. 6.4.

With this in mind, we introduce the first availability metric

$$M_{c,1}^e(i) \quad := \quad \sum_{j=0}^{r} j\frac{\mathrm{E}[T_c^e(i,j)]}{\mathrm{E}[T_c^e(i)]}, \quad i\in\mathcal{T}^e, \tag{5}$$

that we can interpret as the expected number of available redundant fragments during the block lifetime, given that $X_c^e(0)=i\in\mathcal{T}^e$.

A second metric is

$$M_{c,2}^e(i) \quad := \quad \sum_{j=m}^{r} \frac{\mathrm{E}[T_c^e(i,j)]}{\mathrm{E}[T_c^e(i)]}, \quad i\in\mathcal{T}^e, \tag{6}$$

that we can interpret as the fraction of time when there are at least m redundant fragments during the block lifetime, given that $X_c^e(0) = i \in \mathcal{T}^e$.

Both quantities $M_{c,1}^e(i)$ and $M_{c,2}^e(i)$ can be (numerically) computed from (3) and (4). Numerical results for $M_{c,2}^e(r)$ are reported in Sect. 6.4 for $m = r - k$ in (6).

Since it is difficult to come up with an explicit expression for either metric $M_{c,1}^e(i)$ or $M_{c,2}^e(i)$, we make the assumption that parameters k and r have been selected so that the time before absorption is arbitrarily "large". This can be formalized, for instance, by requesting that $P(T_c^e(r) > q) > 1 - \epsilon$, where parameters q and ϵ are set according to the particular storage application(s). Instances are given in Sect. 6.3.

In this setting, one may represent the state of the storage system by a new Markov chain $\tilde{\mathbf{X}}_c^e := \{\tilde{X}_c^e(t),\, t \geq 0\}$, which is irreducible and aperiodic – and therefore ergodic – on the state-space $\mathcal{T}^e$. Let $\tilde{\mathbf{Q}}_c^e = [\tilde{q}_c^e(i,j)]_{i,j \in \mathcal{T}^e}$ be its infinitesimal generator. Matrices $\tilde{\mathbf{Q}}_c^e$ and $\mathbf{Q}_c^e$ – whose non-zero entries are given in (1) – are identical except for $\tilde{q}_c^e(0,0) = -(u_0 + d_0)$. Until the end of this section we assume that $\beta_c(i) = \beta_c$ for $i \in \mathcal{T}^e$.

Let $\pi_c^e(i)$ be the stationary probability that $\tilde{\mathbf{X}}_c^e$ is in state i. Our objective is to compute $\mathrm{E}[\tilde{X}_c^e] = \sum_{i=0}^r i\pi_c^e(i)$, the (stationary) expected number of available redundant fragments. To this end, let us introduce $f_c^e(z) = \sum_{i=0}^r z^i \pi_c^e(i)$, the generating function of the stationary probabilities $\pi_c^e = (\pi_c^e(0), \pi_c^e(1), \ldots, \pi_c^e(r))$.

Starting from the Kolmogorov balance equations $\pi_c^e \cdot \tilde{\mathbf{Q}}_c^e = 0$, and using the normalizing equation $\pi_c^e \cdot \mathbf{1}_{r+1} = 1$, standard algebra yields

$$\begin{aligned}(\mu + p\lambda\, z)\frac{df_c^e(z)}{dz} &= rp\lambda f_c^e(z) - s\mu\frac{f_c^e(z) - \pi_c^e(0)}{z} + \beta_c\,\frac{f_c^e(z) - z^r}{1-z} \\ &\quad -\beta_c \sum_{i=r-k+1}^{r} \frac{z^i - z^r}{1-z}\pi_c^e(i).\end{aligned}$$

Letting $z = 1$ and using the identities $f_c^e(1) = 1$ and $df_c^e(z)/dz|_{z=1} = \mathrm{E}[\tilde{X}_c^e]$, we find

$$\mathrm{E}[\tilde{X}_c^e] = \frac{r(p\lambda + \beta_c) - s\mu(1 - \pi_c^e(0)) - \beta_c \sum_{i=0}^{k-1} i\pi_c^e(r-i)}{\mu + p\lambda + \beta_c}. \tag{7}$$

Unfortunately, it is not possible to find an explicit expression for $\mathrm{E}[\tilde{X}_c^e]$ since this quantity depends on the probabilities $\pi_c^e(0), \pi_c^e(r-(k-1)), \pi_c^e(r-(k-2)), \ldots, \pi_c^e(r)$, which cannot be computed in explicit form. If $k = 1$ then

$$\mathrm{E}[\tilde{X}_c^e] = \frac{r(p\lambda + \beta_c) - s\mu(1 - \pi_c^e(0))}{\mu + p\lambda + \beta_c}, \tag{8}$$

which still depends on the unknown probability $\pi_c^e(0)$.

Below, we use a mean field approximation to develop an approximation formula for $\mathrm{E}[\tilde{X}_c^e]$ for $k = 1$, in the case where the maximum number of redundant fragments r is large. Until the end of this section we assume that $k = 1$. Using [5, Thm. 3.1] we know that, when r is large, the expected

number of available redundant fragments at time t, $\mathrm{E}[\tilde{X}_c^e(t)]$, is solution of the following first-order differential (ODE) equation

$$\dot{y}(t) \quad = \quad -(\mu + p\lambda + \beta_c)y(t) - s\mu + r(p\lambda + \beta_c).$$

The equilibrium point of the above ODE is reached when time goes to infinity, which suggests to approximate $\mathrm{E}[\tilde{X}_c^e]$, when r is large, by

$$\mathrm{E}[\tilde{X}_c^e] \quad \approx \quad y(\infty) = \frac{r(p\lambda + \beta_c) - s\mu}{\mu + p\lambda + \beta_c}. \tag{9}$$

Observe that this simply amounts to neglect of the probability $\pi_c^e(0)$ in (8) for large r.

4.2 The improved model

We consider in this section that peers on-times are hyper-exponentially distributed, having n phases; phase i has parameter $\mu_i > 0$ and occurs with probability p_i for $i = 1, \ldots, n$. We naturally have $\sum_{i=1}^n p_i = 1$. Recall that the hyper-exponential distribution is a mixture or weighted sum of exponentials and its density function is given by $\sum_{i=0}^n p_i \mu_i \exp(-\mu_i x)$. This model is hence a generalization of the basic model, since setting $n = 1$ (then $p_1 = 1$) and $\mu_1 = \mu$ returns the basic model.

The easiest way to think about hyper-exponential distribution in our context is to suppose that there are n types of peers, [4, p. 266], where peers of type i have on-times distributed exponentially with parameter μ_i, and $0 \leq p_i \leq 1$ is the proportion of peers that are of type i. The system state-space, unlike the previous model, will have to include knowledge of the peers types to be able to incorporate hyper-exponential distribution into a Markov chain model, where the Markov property must hold. It is no longer sufficient to consider solely the number of available *redundant* fragments of D. Not only we need to keep trace of all available fragments but also on which type of peers are they stored.

For later use, introduce an n-tuple $\mathbf{i} = (i_1, \ldots, i_n)$ with $i_l \in \{0, \ldots, s + r\}$ and a function $S(\mathbf{i}) := \sum_{l=1}^n i_l$. It will be convenient to introduce sets $\mathcal{E}_I := \{\mathbf{i} \in \{0, \ldots, s + r\}^n, S(\mathbf{i}) = I\}$ for $I = s, \ldots, s + r$. The set $\mathcal{E}_I$ consists of all system states in which the number of fragments of D currently available is equal to I. For any I, the cardinal of $\mathcal{E}_I$ is $\binom{I+n-1}{n-1}$ (think of the possible selections of $n - 1$ boxes in a row of $I + n - 1$ boxes, so as to delimit n groups of boxes summing up to I).

Let $X_c^h(t)$ represent the system state at time t. The rv $X_c^h(t)$ takes value in $\{a\} \cup \mathcal{T}^h$ where $\mathcal{T}^h := \bigcup_{I=s}^{s+r} \mathcal{E}_I$. $X_c^h(t) = a$ indicates that less than s fragments of D are available at time t, and $X_c^h(t) = \mathbf{i} = (i_1, \ldots, i_n)$ indicates that $i_l \in \{0, \ldots, s + r\}$ fragments of D are stored on a peer of type l for $l \in \{1, \ldots, n\}$, such that the total number of available fragments $S(\mathbf{i})$ lays between s and $s + r$.

Thanks to the assumptions made in Sect. 3, the process $\mathbf{X}_c^h := \{X_c^h(t), t \geq 0\}$ is an absorbing Markov chain, with one single absorbing state a and $|\mathcal{T}^h| = \sum_{I=s}^{s+r} \binom{I+n-1}{n-1}$ transient states.

4.2.1 Data lifetime

Introduce $T_c^h(\mathcal{E}_I) := \inf\{t \geq 0 : X_c^h(t) = a | X_c^h(0) \in \mathcal{E}_I\}$, the time until absorption in state a given that the initial number of fragments of D available in the system is equal to I. In this section, we will derive the probability distribution and the expectation of $T_c^h(\mathcal{E}_I)$.

Let $\mathbf{Q}_c^h = [q_c^h(\mathbf{i},\mathbf{j})]_{\mathbf{i},\mathbf{j}\in\mathcal{T}^h}$ be a matrix where $q_c^h(\mathbf{i},\mathbf{j})$ gives the transition rate of $\mathbf{X}_c^h$ from transient state $\mathbf{i}$ to transient state $\mathbf{j}$ for $\mathbf{i} \neq \mathbf{j}$, and $-q_c^h(\mathbf{i},\mathbf{i})$ gives the total transition rate out of state $\mathbf{i}$. Introduce for $\mathbf{i},\mathbf{j} \in \mathcal{T}^h$ and $l = 1,\ldots,n$

$$
\begin{aligned}
A_l &:= i_l \mu_l \mathbb{1}\{1 \leq i_l \leq s+r\} \\
B_{\mathbf{i},l} &:= p_l (s+r-S(\mathbf{i})) p \lambda \mathbb{1}\{0 \leq i_l \leq s+r-1\} \\
C_{\mathbf{i},\mathbf{j}} &:= \beta_c (s+r-S(\mathbf{i})) \mathbb{1}\{S(\mathbf{i}) \leq s+r-k\} \binom{S(\mathbf{j}-\mathbf{i})}{j_1-i_1, j_2-i_2, \ldots, j_n-i_n} \prod_{l=1}^{n} p_l^{j_l-i_l}
\end{aligned}
$$

where the multinomial coefficient is used in the expression of $C_{\mathbf{i},\mathbf{j}}$. Non-zero elements of $\mathbf{Q}_c^h$ are

$$
\left.
\begin{array}{ll}
q_c^h\left(\mathbf{i}, \mathbf{i} - \mathbf{e}_n^{(l)}\right) = A_l, & s+1 \leq S(\mathbf{i}) \leq s+r, \\
 & 1 \leq i_l \leq s+r, \\
q_c^h\left(\mathbf{i}, \mathbf{i} + \mathbf{e}_n^{(l)}\right) = B_{\mathbf{i},l}, & s \leq S(\mathbf{i}) \leq s+r-2, \\
q_c^h\left(\mathbf{i}, \mathbf{i} + \mathbf{e}_n^{(l)}\right) = B_{\mathbf{i},l} + C_{\mathbf{i},\mathbf{i}+\mathbf{e}_n^{(l)}}, & S(\mathbf{i}) = s+r-1,
\end{array}
\right\} \text{ for } l = 1,\ldots,n,
$$

$$
\begin{array}{ll}
q_c^h(\mathbf{i},\mathbf{j}) = C_{\mathbf{i},\mathbf{j}}, & s \leq S(\mathbf{i}) \leq \min\{s+r-k, s+r-2\}, \\
 & S(\mathbf{j}) = s+r, \\
 & j_l \geq i_l \text{ for } l = 1,\ldots,n,
\end{array}
$$

$$
q_c^h(\mathbf{i},\mathbf{i}) = -\sum_{l=1}^{n}(A_l + B_{\mathbf{i},l}) - \sum_{\mathbf{j}\in\mathcal{T}^h, S(\mathbf{j})=s+r} C_{\mathbf{i},\mathbf{j}}, \qquad s \leq S(\mathbf{i}) \leq s+r. \tag{10}
$$

Similarly to (2), we can write

$$
P\left(T_c^h(\{\mathbf{i}\}) \leq x\right) \quad = \quad 1 - \mathbf{e}_{|\mathcal{T}^h|}^{(\mathrm{ind}(\mathbf{i}))} \cdot \exp\left(x\mathbf{Q}_c^h\right) \cdot \mathbf{1}_{|\mathcal{T}^h|}, \; x > 0, \; \mathbf{i} \in \mathcal{T}^h, \tag{11}
$$

where $\mathrm{ind}(\mathbf{i})$ refers to the index of state $\mathbf{i}$ in matrix $\mathbf{Q}_c^h$ and $T_c^h(\{\mathbf{i}\})$ is the time until absorption in state a given that the system initiates in state $\mathbf{i}$. Let $\pi_{\mathbf{i}}$ denote the probability that the system initiates in state $\mathbf{i} \in \mathcal{E}_I$ given that $X_c^h(0) \in \mathcal{E}_I$. We can write

$$
\pi_{\mathbf{i}} \quad := \quad P\left(X_c^h(0) = \mathbf{i} \in \mathcal{E}_I | X_c^h(0) \in \mathcal{E}_I\right) = \binom{I}{i_1, i_2, \ldots, i_n} \prod_{l=1}^{n} p_l^{i_l}. \tag{12}
$$

We naturally have that $\sum_{\mathbf{i}\in\mathcal{E}_I}\pi_{\mathbf{i}}=1$ whatever $I=s,\ldots,s+r$. Using (11) and (12) and the total probability theorem yields

$$P\left(T_c^h(\mathcal{E}_I)\le x\right) = \sum_{\mathbf{i}\in\mathcal{E}_I} P\left(T_c^h(\{\mathbf{i}\})\le x\right)\pi_{\mathbf{i}} \quad (13)$$

$$= 1-\sum_{\mathbf{i}\in\mathcal{E}_I}\pi_{\mathbf{i}}\mathbf{e}_{|\mathcal{T}^h|}^{(\mathrm{ind}(\mathbf{i}))}\cdot\exp\left(x\mathbf{Q}_c^h\right)\cdot\mathbf{1}_{|\mathcal{T}^h|},\ x>0,\ \mathbf{i}\in\mathcal{T}^h. \quad (14)$$

The expectation of the block lifetime when there are initially I fragments available in the system is given by

$$\mathrm{E}\left[T_c^h(\mathcal{E}_I)\right] = -\sum_{\mathbf{i}\in\mathcal{E}_I}\pi_{\mathbf{i}}\mathbf{e}_{|\mathcal{T}^h|}^{(\mathrm{ind}(\mathbf{i}))}\cdot\left(\mathbf{Q}_c^h\right)^{-1}\cdot\mathbf{1}_{|\mathcal{T}^h|},\ I=s,\ldots,s+r. \quad (15)$$

Similarly to (4), we can compute

$$\mathrm{E}\left[T_c^h(\mathcal{E}_I,\mathcal{E}_J)\right] = \sum_{\mathbf{j}\in\mathcal{E}_J}\mathrm{E}\left[T_c^h(\mathcal{E}_I,\{\mathbf{j}\})\right]$$

$$= -\sum_{\mathbf{i}\in\mathcal{E}_I}\sum_{\mathbf{j}\in\mathcal{E}_J}\pi_{\mathbf{i}}\mathbf{e}_{|\mathcal{T}^h|}^{(\mathrm{ind}(\mathbf{i}))}\cdot\left(\mathbf{Q}_c^h\right)^{-1}\cdot\mathbf{e}_{|\mathcal{T}^h|}^{(\mathrm{ind}(\mathbf{j}))},\ I,J=s,\ldots,s+r. \quad (16)$$

where $T_c^h(\mathcal{E}_I,\mathcal{E}_J)$ is the total time spent in transient states $\mathbf{j}\in\mathcal{E}_J$ given that $X_c^h(0)\in\mathcal{E}_I$. In other words, $T_c^h(\mathcal{E}_I,\mathcal{E}_J)$ represents the time during which J fragments of D are available given that the system had initially I fragments of D.

Numerical results are reported in Sect. 6.

4.2.2 Data availability

The data availability are quantified, as motivated in Sect. 4.1.2, by the following two metrics for $s\le I\le s+r$.

$$M_{c,1}^h(\mathcal{E}_I):=\sum_{J=s}^{s+r}J\frac{\mathrm{E}\left[T_c^h(\mathcal{E}_I,\mathcal{E}_J)\right]}{\mathrm{E}\left[T_c^h(\mathcal{E}_I)\right]},\quad M_{c,2}^h(\mathcal{E}_I):=\sum_{J=m}^{s+r}\frac{\mathrm{E}\left[T_c^h(\mathcal{E}_I,\mathcal{E}_J)\right]}{\mathrm{E}\left[T_c^h(\mathcal{E}_I)\right]}. \quad (17)$$

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.

Again similar to what was done in Sect. 4.1.2, we will assume that the parameters r and k are tuned such that the time before absorption in state a is arbitrarily long. Neglecting then the absorbing state a, we may represent the state of the storage system by an irreducible, aperiodic Markov chain on

the state-space $\mathcal{T}^h$, denoted $\tilde{\mathbf{X}}_c^h := \{\tilde{X}_c^h(t),\, t \geq 0\}$. Let $\tilde{\mathbf{Q}}_c^h = [\tilde{q}_c^h(\mathbf{i},\mathbf{j})]_{\mathbf{i},\mathbf{j}\in\mathcal{T}^h}$ be its infinitesimal generator. Matrices $\tilde{\mathbf{Q}}_c^h$ and $\mathbf{Q}_c^h$, whose non-zero entries are given in (10), are identical except for

$$\tilde{q}_c^h(\mathbf{i},\mathbf{i}) \;=\; -\sum_{l=1}^{n} B_{\mathbf{i},l} - \sum_{\mathbf{j}\in\mathcal{T}^h, S(\mathbf{j})=s+r} C_{\mathbf{i},\mathbf{j}}$$

for all states $\mathbf{i} \in \mathcal{E}_s$ (i.e. $S(\mathbf{i}) = s$).

Let $\pi_c^h(\mathbf{i})$ be the stationary probability that $\tilde{\mathbf{X}}_c^h$ is in state $\mathbf{i}$. The (stationary) expected number of available fragments can be computed from

$$\begin{aligned}\mathrm{E}[S(\tilde{X}_c^h)] &= \sum_{\mathbf{i}\in\mathcal{T}^h} S(\mathbf{i})\pi_c^h(\mathbf{i}) \\ &= \sum_{I=s}^{s+r} I \sum_{\mathbf{i}\in\mathcal{E}_I} \pi_c^h(\mathbf{i}).\end{aligned}$$

As the data lifetime becomes sufficiently long, the first availability metric given in (17) converges to $\mathrm{E}[S(\tilde{X}_c^h)]$.

5 Distributed repair systems

In this section, we address the performance of P2P storage systems that use the distributed-recovery mechanism, as described in Sect. 3. Alike in Sect. 4, we will start by reviewing the basic model that appeared in [7] and then study the improved model. Since the analysis is very similar to the analysis in Sect. 4 we will only sketch it.

5.1 The basic model

We assume, as in Sect. 4.1, that the successive durations of on-times of a peer form a sequence of iid rvs, with an exponential distribution with parameter $\mu > 0$.

5.1.1 Data lifetime

Alike the basic model in the centralized implementation, the state of the system can be represented by an absorbing Markov chain $\mathbf{X}_d^e := \{X_d^e(t),\, t \geq 0\}$, taking values in the set $\{a\} \cup \mathcal{T}^e$ (recall that $\mathcal{T}^e = \{0, 1, \ldots, r\}$). State a is the absorbing state indicating that the block of data is lost (less than s fragments of D available), and state $i \in \mathcal{T}^e$ gives the number of available *redundant* fragments. The non-zero transition rates of this absorbing Markov chain are displayed in Fig. 2.

Non-zero entries of the matrix $\mathbf{Q}_d^e = [q_d^e(i,j)]_{i,j\in\mathcal{T}^e}$ associated with the absorbing Markov chain $\mathbf{X}_d^e$ are given by

$$\begin{array}{ll} q_d^e(i,i-1) = a_i, & i = 1,2,\ldots,r, \\ q_d^e(i,i+1) = b_i + d_i, & i = 0,1,\ldots,r-1, \\ q_d^e(i,i) = -(a_i+b_i+d_i), & i = 0,1,\ldots,r, \end{array}$$

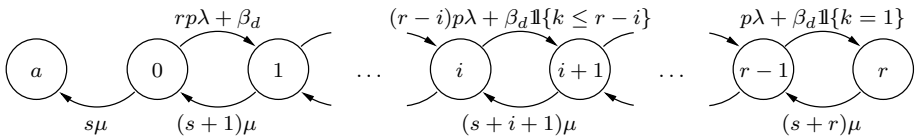


Figure 2: Transition rates of the absorbing Markov chain $\{X_d^e(t), t \ge 0\}$.

with $d_i := \beta_d \mathbb{1}\{i \le r-k\}$ for $i = 0, 1, \ldots, r$, where a_i and b_i are defined in Sect. 4.1.1. Introduce $T_d^e(i) := \inf\{t \ge 0 : X_d^e(t) = a\}$ the time until absorption in state a given that $X_d^e(0) = i$, and let $T_d^e(i,j)$ be the total time spent in transient state j starting at time $t = 0$ in transient state i. The probability distribution $P(T_d^e(i) \le x)$, $\mathrm{E}[T_d^e(i)]$ and $\mathrm{E}[T_d^e(i,j)]$ are given by (2), (3) and (4), respectively, after replacing the matrix $\mathbf{Q}_c^e$ with the matrix $\mathbf{Q}_d^e$. Alike for $\mathbf{Q}_c^e$ it is not tractable to explicitly invert $\mathbf{Q}_d^e$. Numerical results for $\mathrm{E}[T_d^e(r)]$ and $P(T_d^e(r) > 1 \text{ year})$ are reported in Sect. 6.3.

5.1.2 Data availability

As motivated in Sect. 4.1.2 the two availability metrics are given by

$$M_{d,1}^e(i) := \sum_{j=0}^{r} j \frac{\mathrm{E}[T_d^e(i,j)]}{\mathrm{E}[T_d^e(i)]}, \quad M_{d,2}^e(i) := \sum_{j=m}^{r} \frac{\mathrm{E}[T_d^e(i,j)]}{\mathrm{E}[T_d^e(i)]}, \tag{18}$$

Numerical results are given in Sect. 6.4. Similar to what was done in Sect. 4.1.2, let us assume that parameters r and k have been tuned so that the time before absorption is "long". If so, then as an approximation one can consider that absorbing state a can no longer be reached. The Markov chain $\mathbf{X}_d^e$ becomes an irreducible, aperiodic Markov chain on the set $\mathcal{T}^e$, denoted $\tilde{\mathbf{X}}_d^e$. More precisely, it becomes a birth and death process (see Fig. 2). Let $\pi_d^e(i)$ be the stationary probability that $\tilde{\mathbf{X}}_d^e$ is in state i, then (e.g. [2])

$$\pi_d^e(i) \;=\; \left[1 + \sum_{i=1}^{r} \prod_{j=0}^{i-1} \frac{b_j + d_j}{a_{j+1}}\right]^{-1} \cdot \prod_{j=0}^{i-1} \frac{b_j + d_j}{a_{j+1}}, \quad i \in \mathcal{T}^e. \tag{19}$$

From (19) we can derive the expected number of available redundant fragments through the formula $\mathrm{E}[\tilde{X}_d^e] = \sum_{i=0}^{r} i\pi_d^e(i)$. Numerical results for $\mathrm{E}[\tilde{X}_d^e]$, or more precisely, for its deviation from $M_{d,1}^e(r)$ are reported in Sect. 6.4.

5.2 The improved model

Consider now that peers on-times are hyper-exponentially distributed, with n phases and same parameters as in Sect. 4.2. Again, the basic model can be retrieved by setting $n = 1$ (then $p_1 = 1$)

and $\mu_1 = \mu$. As discussed in Sect. 4.2, the state of the system can be represented by an absorbing Markov chain $\mathbf{X}_d^h := \{X_d^h(t), t \geq 0\}$, taking values in $\{a\} \cup \mathcal{T}^h$. Non-zero entries of matrix $\mathbf{Q}_d^h := [q_d^h(\mathbf{i},\mathbf{j})]_{\mathbf{i},\mathbf{j}\in\mathcal{T}^h}$ associated with the absorbing Markov chain $\mathbf{X}_d^h$ are given by

$$\left.\begin{array}{ll} q_d^h\left(\mathbf{i}, \mathbf{i} - \mathbf{e}_n^{(l)}\right) = A_l, & s+1 \leq S(\mathbf{i}) \leq s+r, \\ & 1 \leq i_l \leq s+r, \\ q_d^h\left(\mathbf{i}, \mathbf{i} + \mathbf{e}_n^{(l)}\right) = B_{\mathbf{i},l} + D_{\mathbf{i},l}, & s \leq S(\mathbf{i}) \leq s+r-1, \end{array}\right\} \text{ for } l = 1, \ldots, n,$$

$$q_d^h(\mathbf{i},\mathbf{i}) = -\sum_{l=1}^{n}(A_l + B_{\mathbf{i},l} + D_{\mathbf{i},l}), \qquad s \leq S(\mathbf{i}) \leq s+r, \tag{20}$$

where $D_{\mathbf{i},l} := p_l \beta_d \mathbb{1}\{S(\mathbf{i}) \leq s+r-k\}$ and A_l and $B_{\mathbf{i},l}$ have been defined in Sect. 4.2.1.

$P\left(T_d^h(\mathcal{E}_I) \leq x\right)$, $\mathrm{E}\left[T_d^h(\mathcal{E}_I)\right]$, $\mathrm{E}\left[T_d^h(\mathcal{E}_I, \mathcal{E}_J)\right]$, $M_{d,1}^h(\mathcal{E}_I)$ and $M_{d,2}^h(\mathcal{E}_I)$ are given by (14), (15), (16) and (17) respectively, after replacing the matrix $\mathbf{Q}_c^h$ with the matrix $\mathbf{Q}_d^h$.

Observe that neglect of the absorption in state a yields a quasi birth-death process, where all states in $\mathcal{E}_I$ are grouped together for $I = s, \ldots, s+r$.

Numerical results for these metrics are reported in the next section.

6 Numerical results

In this section, we will first assess whether the basic model is robust against violation of the exponential assumption on peers on-times. Applying then the appropriate model to four different scenarios, we characterize the performance metrics defined in the paper against the system parameters. Last, we illustrate how our models can be used to engineer storage systems. Throughout the numerical computations, we consider both centralized- and distributed-recovery implementations.

6.1 Parameter values

Our mathematical models have been solved numerically using a set of parameters values. The protocol parameters have been set as follows.

Original number of fragments s. Block sizes in P2P systems are usually set to either 256KB, 512KB or 1MB and the fragment size to 64KB. This yields an original number of fragments s in the set $\{4, 8, 16\}$. We will consider $s = 8$ through most of our computations. The impact of this parameter on the performance of the system is assessed through the computation of the complementary cumulative distribution function (CCDF) of the block lifetime in two scenarios.

Number of redundant fragments r and recovery threshold k. The amount of redundancy r will be varied from 1 to 30 and for each value of r, we vary the threshold k from 1 to r. Our aim is to provide guidelines on how to select these parameters so as to guarantee a desired level of data

Table 2: Data sets characteristics and corresponding peers parameters values

Data set	*LMG*	*CSIL*	*Condor*	*All-pairs-ping*
Context	Internet	LAN	Internet	PlanetLab
Covered period	3 months	8 weeks	6 weeks	21 months
Number of peers	1170	83	210	200–550
On-times distribution	H_3 [19] (best fit)	H_3 [19] (best fit)	H_2 [19] (best fit)	Exp. [15] (reasonable)
On-times parameters				
p_1	0.282	0.464	0.592	1
p_2	0.271	0.197	0.408	–
p_3	0.447	0.339	–	–
$1/\mu_1$ (hours)	910.7	250.3	0.094	181
$1/\mu_2$ (hours)	0.224	1.425	3.704	–
$1/\mu_3$ (hours)	199.8	33.39	–	–
Mean on-time (hours)	352.2	127.7	1.567	181 [15]
Mean off-time (hours)	48.43 [14]	48	1.567 or 0.522	61 [15]
Percentage of on-times	0.879	0.727	0.5 or 0.75	0.750
Persistence probability p	0.4	0.4	0.8	0.4

lifetime and or availability. Observe that the optimal amount of redundancy r comes as a trade-off between high data availability and high storage efficiency and is connected with the recovery threshold k. Smaller threshold values allow for smaller amounts of redundant data at the expense of higher bandwidth utilization. The trade-off here is between efficient storage use (small r) and efficient bandwidth use (large k).

Peers parameters λ, $\{p_i, \mu_i\}_{i=1,\ldots,n}$ and p. Concerning the peers parameters, we rely on the findings of [19, 15]. The three data sets analyzed in [19] 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 [15]. The basic characteristics of the four data sets considered here and the corresponding values of the peers parameters are reported in Table 2.

The *LMG* set has been collected by Long, Muir and Golding [14]. It is based on Poisson probes sent to 1170 nodes in the Internet every 10 minutes on average. Each poll returned either the time since the host was last initialized, or a failure. The sets *CSIL* and *Condor* have been collected by Nurmi, Brevik and Wolski [19]. The *CSIL* set reports uptime of machines in the Computer Science Instructional Laboratory (CSIL) at the University of California, Santa Barbara. As for the *Condor* set, it reports CPU idle times of peers in a Condor pool [20] at the University of Wisconsin, in other words, it reports the availability of peers to perform an external job (the Condor pool offers processing time to the whole Internet). This can be seen as the time during which a peer may participate in a storage system. The *All-pairs-ping* set has been obtained by Stribling [24] after the processing of ping requests between each pair of PlanetLab [23] nodes. Each node pings every other

node roughly 4 times an hour. A 10-probes ping is considered successful only if at least one probe response was received.

Out of the four scenarios considered, *Condor* experiences the highest dynamics. This behavior has been reported elsewhere concerning peers on the Internet. For instance, it has been observed in [8] that on average peers join/leave the Internet more than six times per day and that sessions last for one hour on average. The probability of finding a peer connected or equivalently the percentage of on-times in a peer life cycle is reported in the sixteenth row of Table 2 as a complement of information. We will refer to this metric as the *peers availability*. The smallest peers availability is considered in the *Condor* scenario for $1/\lambda = 1.567$.

We have arbitrarily set $p = 0.4$ when peers availability is very high, like in the *LMG*, *CSIL* and *All-pairs-ping* scenarios, and $p = 0.8$ in the *Condor* scenario.

Recovery rates $\beta_c(k)$ and β_d. The recovery time is composed of various durations: the download time of s fragments, the reconstruction of lost fragments, and the time to store the reconstructed fragments on their hosts. Observe that storage time depends on whether the storage is made on a local disk, like in the distributed implementation, or over the network on a remote disk, like in centralized implementation, in which case the network latency is to be accounted for. As a result, we should always have $\beta_c(k) < \beta_d$. From now on, the recovery rate in the centralized scheme is made constant.

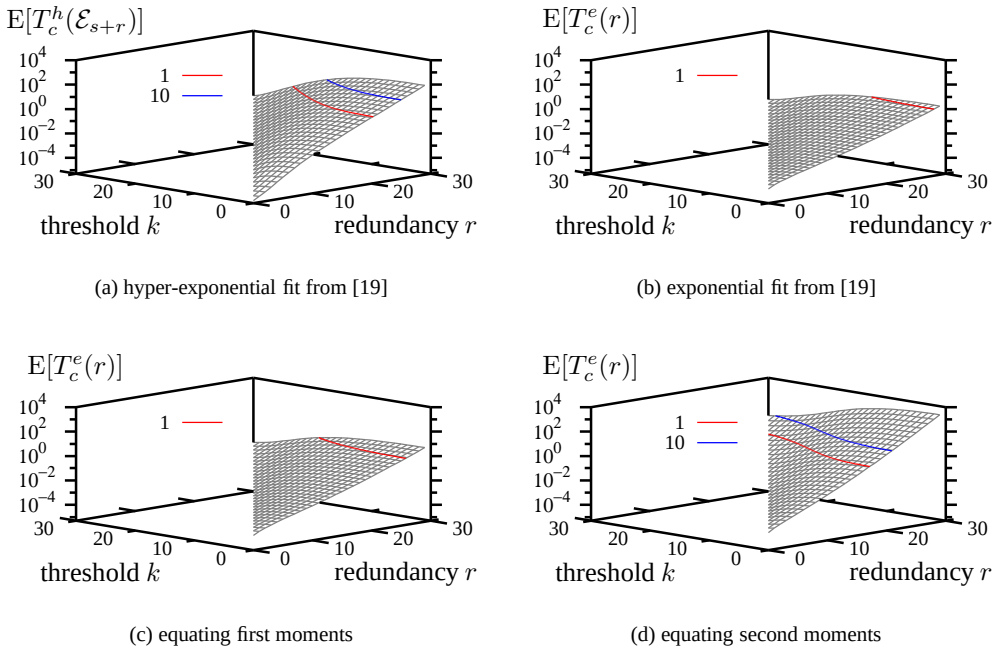
In scenarios where peers are highly available, as reported in the *LMG* and *CSIL* data sets, we consider $1/\beta_c = 22$ minutes and $1/\beta_d = 20$ minutes. In scenarios where peers are either very dynamic or disconnected for long periods, like with the *Condor* or the *All-pairs-ping* data sets, we expect the recovery process to last longer. Thus we set $1/\beta_c = 34$ minutes and $1/\beta_d = 30$ minutes.

In scenarios having the characteristics of either *LMG*, *CSIL* or *All-pairs-ping* data sets, the dynamics of storage systems therein deployed will have two timescales. This is not observed in contexts with the same characteristics as the *Condor* data set, where the recovery process evolves in the same timescale as the peers arrival and failure processes[1].

6.2 Validity of the basic model

We have analyzed two models for the evaluation of distributed storage systems. These models are identical except for the assumption on the distribution of peers on-times. However, solving the improved model is much more time consuming than solving the basic one. In this section, we want to assess whether the improved model is really a necessity or not. To this end, we will evaluate the lifetime of a block of data D using both models and compare the results. We deliberately select a scenario in which peers have been identified to have a non-exponential on-times distribution, namely the *Condor* scenario. In [19], a 2-stage hyper-exponential distribution is found to best fit the *Condor* data set, but the authors identify as well the parameter of the exponential distribution that best fits the same data.

[1] Concurrent failures may occur when the recovery rate and the failure rate are of the same order of magnitude (like in the *Condor* scenario). This important aspect needs to be addressed in future models.



(a) hyper-exponential fit from [19]

(b) exponential fit from [19]

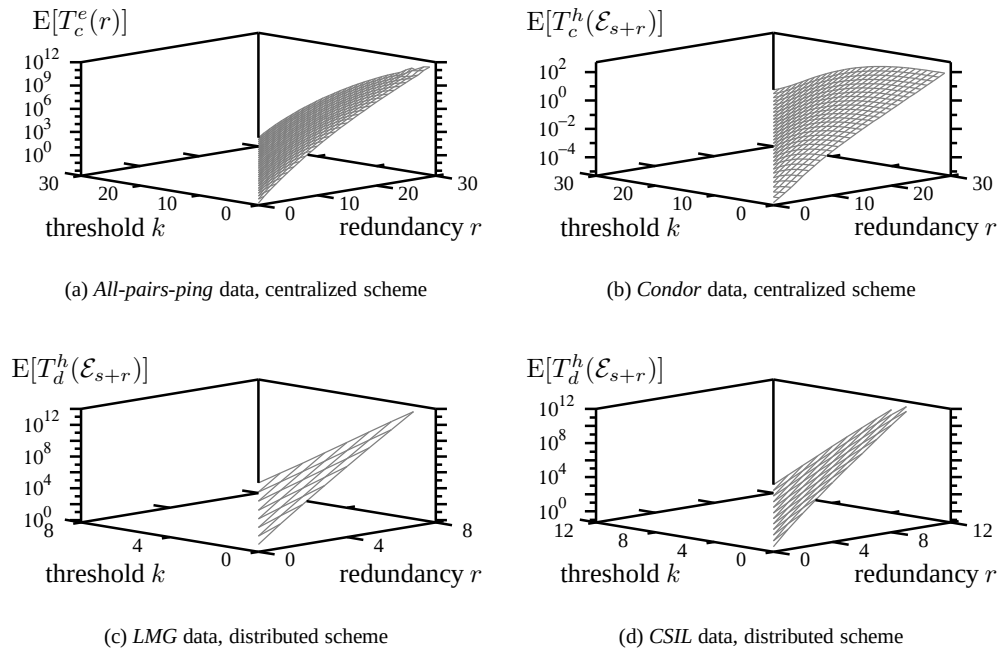
(c) equating first moments

(d) equating second moments

Figure 3: Expected data lifetime (expressed in years) in a *Condor* scenario using a centralized-recovery peer-to-peer storage system. Comparison between $\mathrm{E}[T_c^h(\mathcal{E}_{s+r})]$ (improved model) and $\mathrm{E}[T_c^e(r)]$ (basic model).

Figure 3 displays four plots of the expected data lifetime obtained with the centralized-recovery implementation versus the amount of redundancy r and the recovery threshold k. The improved model is used to derive the results depicted in Fig. 3(a) with $1/\mu_1 = 0.094$ hours, $1/\mu_2 = 3.704$ hours, $p_1 = 0.592$ and $p_2 = 1 - p_1$. The basic model has been solved (i) using the fit found in [19], namely $1/\mu = 1.543$ hours, (ii) using $1/\mu = 1.567$ which is the first moment of the H_2 distribution, and (iii) using $1/\mu = 2.367$ which corresponds to equating the second moments of on-times in both basic and improved model. The results are depicted in Figs. 3(b), 3(c) and 3(d) respectively. In all four cases, we have $s = 8$, $1/\lambda = 1.567$ hour, $1/\beta_c = 34$ minutes and $p = 0.8$.

Figure 3 reveals that the basic model returns substantially different results than the ones outcome of the improved model. Since the distribution of peers on-times is hyperexponential in the *Condor* scenario, the results obtained through the improved model are the correct ones. We conclude that the basic model does not capture the essence of the system performance when peers on-times are not exponentially distributed. *Henceforth, we will use the basic model in scenarios with the* All-pairs-



(a) *All-pairs-ping* data, centralized scheme

(b) *Condor* data, centralized scheme

(c) *LMG* data, distributed scheme

(d) *CSIL* data, distributed scheme

Figure 4: Expected lifetime (expressed in years) versus r and k.

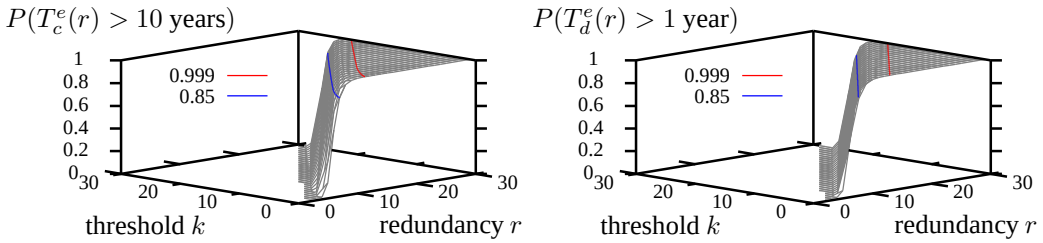
ping *characteristics, and the improved model in scenarios with the characteristics of either* LMG, CSIL, *or* Condor.

6.3 The conditional block lifetime

We 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 $T_c^e(r)$, $T_d^e(r)$, $T_c^h(\mathcal{E}_{s+r})$ and $T_d^h(\mathcal{E}_{s+r})$. The expectation is given in (3) and (15) and the CCDF results from (2) and (14). Four scenarios have been considered as detailed in Sect. 6.1. The results are graphically reported in Figs. 4–7.

The data lifetime in a PlanetLab-like environment (data set *All-pairs-ping*) and a *Condor*-like environment (with $1/\lambda = 0.522$ hour) is depicted in Figs. 4(a) and 4(b) respectively, assuming a centralized-repair scheme. The same metric using a distributed-repair scheme is displayed against the redundancy r and the threshold k in Figs. 4(c) and 4(d) for the scenarios /lmg and *CSIL* respectively.

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. When peers are



(a) centralized scheme

(b) distributed scheme

Figure 5: CCDF of data lifetime versus r and k using *All-pairs-ping* data.

highly available, like in the *LMG* and *CSIL* scenarios, the expected lifetime decreases exponentially with an increasing k. In the other cases (*All-pairs-ping* and *Condor*), the decrease is sub-exponential. Observe how relatively "small" is the expected lifetime in the *Condor* scenario when compared to that in the other three scenarios. Recall that the system dynamics in the *Condor* scenario has only one timescale, unlike the other three scenarios.

For the same values of r and k, we observe a higher data lifetime when the peers availability is higher. Using the same recovery mechanism, the lifetime in *LMG* (respectively *All-pairs-ping*) is several orders of magnitude larger than that in *CSIL* (respectively *Condor*).

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 results are shown against r and k in Fig. 5 (*All-pairs-ping*), Fig. 6 (*Condor*), and Fig. 7 (*LMG* and *CSIL*). 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 dynamics has two timescale. 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.

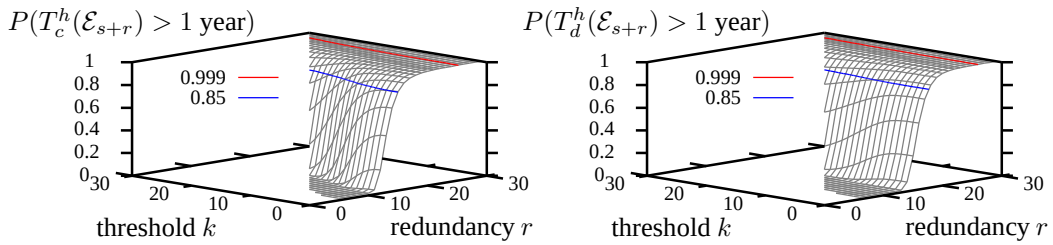
6.4 The availability metrics

We will start this section by the validation of the following approximation

$$\mathrm{E}\left[\frac{1}{T(i)}\int_0^{T(i)} \mathbb{1}\{X(t)=j\}dt\right] \approx \frac{\mathrm{E}[T(i,j)]}{\mathrm{E}[T(i)]}, \quad i,j \in \{0,\ldots,r\}, \tag{21}$$

which has been made for computing the two availability metrics in all models presented in this paper.

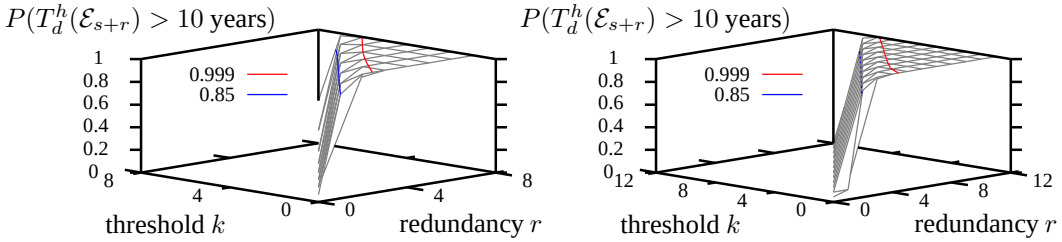
In order to do that, we have simulated the Markov chain depicted in Fig. 2 starting at time 0 in state r (thus $i = r$ in (21)), with $s = 8$, $1/\mu = 5$ hours, $1/\lambda = 3$ hours, $p = 0.8$, and $1/\beta_d = 30$ minutes. The amount of redundancy r is varied from 1 to 23 and for each value of r, we vary the



(a) centralized scheme

(b) distributed scheme

Figure 6: CCDF of data lifetime versus r and k using *Condor* data and $1/\lambda = 0.522$.



(a) *LMG* data

(b) *CSIL* data

Figure 7: CCDF of data lifetime versus r and k using (a) *LMG* and (b) *CSIL* data (distributed recovery).

threshold k from 1 to r. Hence we performed a total of $\sum_{r=1}^{23} r = 276$ different simulations, each being repeated 100 times.

The left-hand side (LHS) of (21), for $i = r$ and $j = 0, \ldots, r$, has been measured from the simulation results and the right-hand side (RHS) of it, also for $i = r$ and $j = 0, \ldots, r$, has been computed numerically using (4)-(3). Observe that for each value of the triple (r, k, j), there is one value of the RHS of (21) but there are 100 sample values of the LHS of (21), that have been averaged.

We then derived the relative error between the averaged LHS of (21) (the correct value) and the RHS of (21) (the approximate value). Given all considered values of the triple (r, k, j), we ended up with a set of $\sum_{r=1}^{23} r(r+1) = 4600$ values of the relative error.

The resulting complementary cumulative distribution function of the relative error is displayed in Fig. 8. We observe that only 10% of the values are larger than 0.75×10^{-3} and, most importantly,

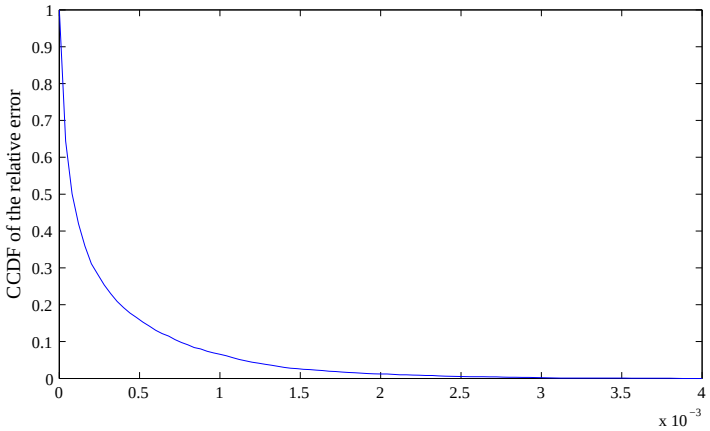


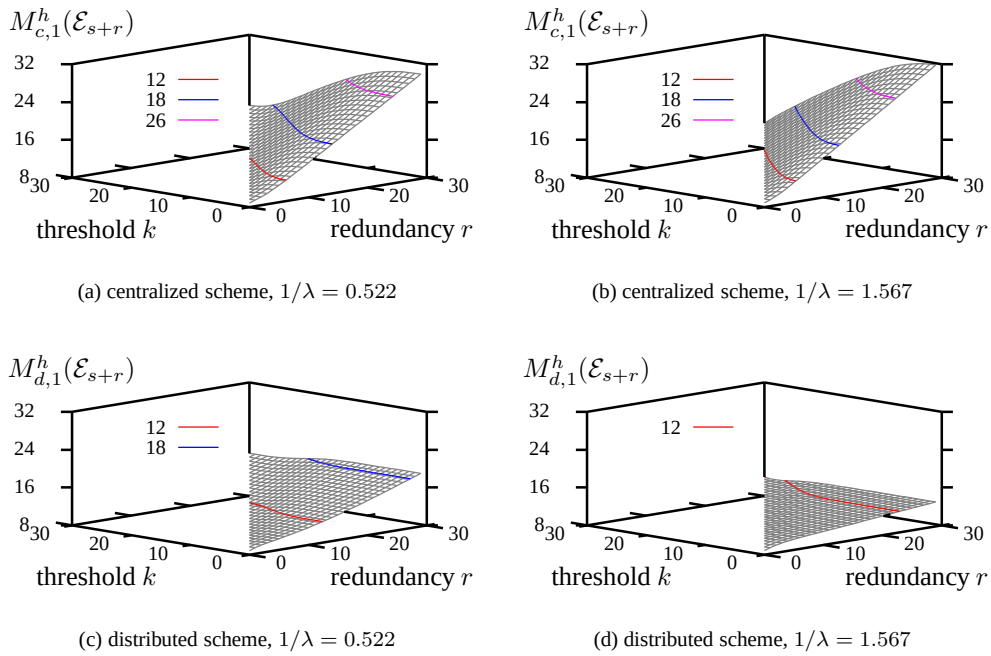
Figure 8: The complementary cumulative distribution function of the relative error induced by the approximation (21).

the maximum value of the relative error is 0.004. *We conclude that the approximation* (21) *is very good and will definitely not imperil the correctness of any result based on it.*

Having validated (21), the availability of data can safely be measured using the metrics defined in Sects. 4-5. We have computed the first availability metric in a *Condor*-like scenario using (17) and (18). The results are reported in Fig. 9: the centralized implementation of the recovery mechanism has been assumed when producing the top graphs whereas the bottom graphs assume a distributed-recovery mechanism; left-hand-side graphs correspond to $1/\lambda = 0.522$ whereas right-hand-side graphs correspond to $1/\lambda = 1.567$.

We see from Fig. 9 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: observe how the line cuts at $r = 30$ are different in Figs. 9(a)–9(b). Strange enough, the data availability at large r and small k is higher for smaller λ in the centralized-recovery implementation. A smaller λ implies a larger expected peer off-time, one could then expect the data to be less available as observed for large values of k in the centralized scheme and for all values of k and r in the distributed scheme; see Figs. 9(c)–9(d). We do not have an explanation for this counter-intuitive observation at the moment. Observe also that the parameter k seems to have a minor if not negligible effect on the availability metric $M^h_{d,1}(\mathcal{E}_{s+r})$ in the distributed implementation. As one could expect, the centralized scheme achieves higher availability than the distributed scheme.

Regarding the second availability metric, we have computed it in all four scenarios considered assuming either implementation of the recovery mechanism. The results corresponding to a PlanetLab-like context (*All-pairs-ping*), computed using (6) and (18) with $m = r - k$, are depicted



(a) centralized scheme, $1/\lambda = 0.522$

(b) centralized scheme, $1/\lambda = 1.567$

(c) distributed scheme, $1/\lambda = 0.522$

(d) distributed scheme, $1/\lambda = 1.567$

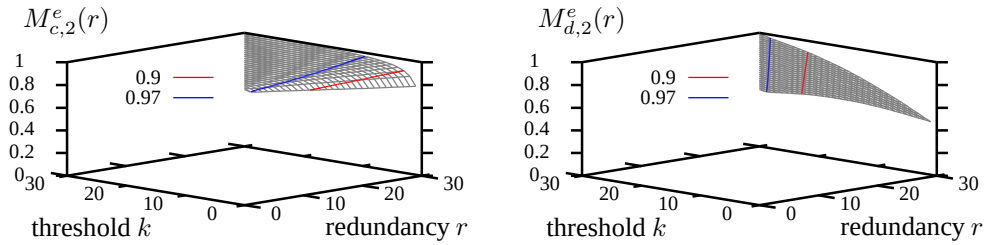
Figure 9: Availability metrics $M^h_{c,1}(\mathcal{E}_{s+r})$ and $M^h_{d,1}(\mathcal{E}_{s+r})$ versus r and k in *Condor* scenario.

in Fig. 10. When the scenario has the characteristics of the *Condor* data set with $1/\lambda = 0.522$, we use $m = s + r - k$ in (17); see results in Fig. 11.

In both considered scenarios, peers are connected to the storage system for the same percentage of time (75%) during their lifetime. However, a noticeable difference in the data availability is observed: according to the second availability metric, data in *Condor*-like systems would be much less available than if it were in a storage system with the characteristics of the *All-pairs-ping* data set, for the same protocol parameters s, r and k and the peers availability. This deterioration in the performance of *Condor*-like systems is mainly due to having a recovery process of the same order of magnitude than the peers arrival and failure processes.

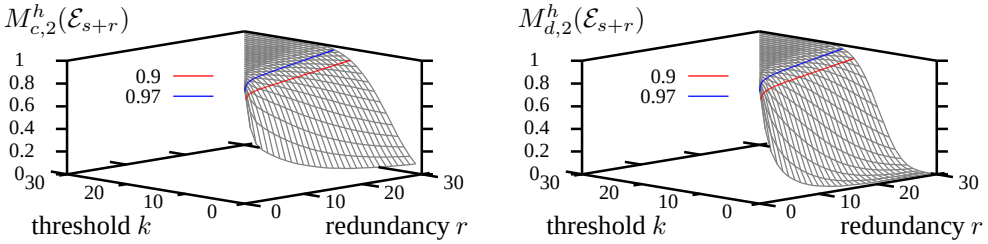
6.5 Engineering the system

Using our theoretical framework it is easy to tune the system parameters for fulfilling predefined requirements. As an illustration, we consider two scenarios implementing a centralized recovery scheme: the first has the *All-pairs-ping* characteristics and is analyzed with the basic model and the second has the *Condor* characteristics and is analyzed with the improved model.



(a) centralized scheme (b) distributed scheme

Figure 10: Availability metrics $M^e_{c,2}(r)$ and $M^e_{d,2}(r)$ for $m = r - k$ using the *All-pairs-ping* data.



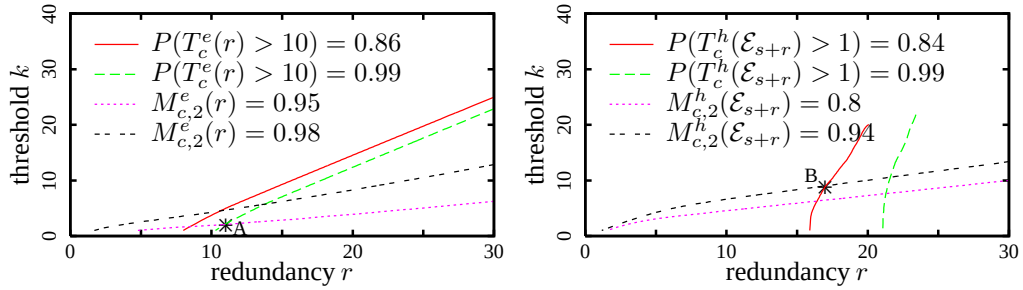
(a) centralized scheme (b) distributed scheme

Figure 11: Availability metrics $M^h_{c,2}(\mathcal{E}_{s+r})$ and $M^h_{d,2}(\mathcal{E}_{s+r})$ for $m = s + r - k$ using $1/\lambda = 0.522$ and the *Condor* data.

In the *All-pairs-ping* scenario, we select two contour lines of the function $P(T^e_c(r) > 10 \text{ year})$ depicted in Fig. 5(a) at values 0.86 and 0.99 and two contour lines of the availability metric $M^e_{c,2}(r)$ displayed in Fig. 10(a) at values 0.95 and 0.98. These four contour lines are reported in Fig. 12(a).

Consider point A which corresponds to $r = 11$ and $k = 2$ (recall $s = 8$). Selecting this point as the operating point of the storage system will ensure that $P(T^e_c(r) > 10) = 0.99$ and $M^e_{c,2}(r) = 0.95$. In other words, when $r = 11$ and $k = 2$, only 1% of the stored blocks would be lost after 10 years and for 95% of a block lifetime there will be 9 ($= r - k$) 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 1.375.

In the *Condor* scenario, the four contour lines selected in Fig. 12(b) are those of $P(T^h_c(\mathcal{E}_{s+r}) > 1)$ (illustrated in Fig. 6(a)) at values 0.84 and 0.99 and $M^h_{c,2}(\mathcal{E}_{s+r})$ (shown in Fig. 11(a)) at values 0.8 and 0.94. Similarly to the *All-pairs-ping* scenario, one can select the system operating point so as to satisfy a desired level of service delivered to users. For instance, selecting $r = 17$ and $k = 9$



(a) *All-pairs-ping* data

(b) *Condor* data, $1/\lambda = 0.522$

Figure 12: Selection of r and k according to predefined requirements assuming a centralized-recovery scheme.

(point B) achieves $P(T_c^h(\mathcal{E}_{s+r}) > 1) = 0.84$ and $M_{c,2}^h(\mathcal{E}_{s+r}) = 0.94$ for a storage overhead r/s equal to 2.125.

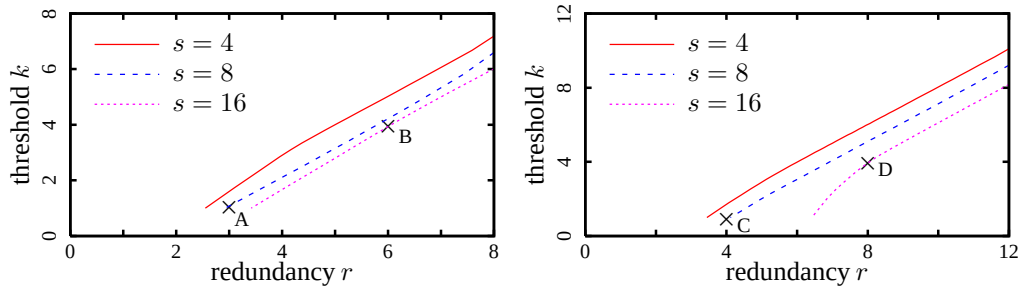
6.6 Impact of the original number of fragments

We now investigate the impact of the parameter s on the performance of the storage system. We have computed the function $P(T_d^h(\mathcal{E}_{s+r}) > 10 \text{ year})$ in scenarios like the ones reported in the *LMG* and *CSIL* data sets for $s = 4, 8, 16$, $r = 1, \ldots, 12$ and $k = 1, \ldots, r$. We assumed the recovery mechanism to be distributed.

In the *LMG* scenario (respectively the *CSIL* scenario), and for each value of s, the contour line at value 0.99 (respectively at line 0.993) has been selected. The resulting curves are plotted in Fig. 13. Any point selected from any line in Fig. 13(a) (respectively Fig. 13(b)) returns values of s, r and k that ensure only 1% (respectively 0.7%) of the blocks of data would be lost after 10 years. However, the storage overhead r/s and the bandwidth usage (related to k) will differ from one point to another.

For instance, consider points A ($s = 8$, $r = 3$ and $k = 1$) and B ($s = 16$, $r = 6$ and $k = 4$) in Fig. 13(a). The storage overhead is the same at both operating points and equals 0.375. The difference is that the threshold at A is smaller than the one at B. In other words, the operating point A ($s = 8$) incurs a higher bandwidth usage during recovery than the operating point B ($s = 16$) while providing the same lifetime guarantee and using the same storage overhead. A similar observation can be made based on points C ($s = 8$, $r = 4$ and $k = 1$) and D ($s = 16$, $r = 8$ and $k = 4$) in Fig. 13(b).

Another difference between points A and B is related to the block size which is equal to $s \times 64$KB. The discussion on the consequences of having a large or small block size is beyond the scope of this paper.



(a) *LMG* data, $P(T_d^h(\mathcal{E}_{s+r}) > 10) = 0.99$

(b) *CSIL* data, $P(T_d^h(\mathcal{E}_{s+r}) > 10) = 0.993$

Figure 13: Protocol parameters yielding same $P(T_d^h(\mathcal{E}_{s+r}) > 10)$ using (a) *LMG* and (b) *CSIL* data.

7 Conclusion

We have proposed analytical models for evaluating the performance of two approaches for recovering lost data in distributed storage systems. One approach relies on a centralized server to recover the data; in the other approach, new peers perform this task in a distributed way. We have analyzed the lifetime and the availability of data achieved by both centralized- and distributed-repair systems through Markovian analysis and fluid approximations considering two different assumptions on the distribution of peers on-times. Extensice numerical computations have been undertaken to support the analysis and illustrate several issues of the performance. We conclude from the numerical results that (i) modeling peers on-times by an exponential distribution when this distribution is not found in practice leads to incorrect results and does not accurately reflect the behavior of the P2P storage systems (ii) when the system dynamics have only one time scale, the quality of service delivered to users is severly impaired. Using our theoretical framework it is easy to tune and optimize the system parameters for fulfilling predefined requirements.

References

[1] C. Grinstead, J. Laurie Snell, Introduction to Probability, American Mathematical Society, 1997.

[2] L. Kleinrock, Queueing Systems, Vol. 1, J. Wiley, New York, 1975.

[3] M. Neuts, Matrix Geometric Solutions in Stochastic Models. An Algorithmic Approach, John Hopkins University Press, Baltimore, 1981.

[4] R. Wolff, Stochastic Modeling and the Theory of Queues, Prentice-Hall, 1989.

[5] T. Kurtz, Solutions of ordinary differential equations as limits of pure jump markov processes, Journal of Applied Probability 7 (1) (1970) 49–58.

[6] I. Reed, G. Solomon, Polynomial codes over certain finite fields, Journal of SIAM 8 (2) (1960) 300–304.

[7] 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.

[8] R. Bhagwan, K. Tati, Y. Cheng, S. Savage, G. Voelker, Total Recall: System support for automated availability management, in: Proc. of ACM/USENIX NSDI '04, San Francisco, California, 2004, pp. 337–350.

[9] C. Blake, R. Rodrigues, High availability, scalable storage, dynamic peer networks: Pick two, in: Proc. of HotOS IX, Lihue, Hawaii, 2003.

[10] I. Clarke, O. Sandberg, B. Wiley, T. Hong, Freenet: A distributed anonymous information storage and retrieval system, in: Proc. of Workshop on Design Issues in Anonymity and Unobservability, Berkeley, California, Vol. 2009 of Lecture Notes in Computer Science, 2000, pp. 46–66.

[11] F. Dabek, F. Kaashoek, D. Karger, R. Morris, I. Stoica, Wide-area cooperative storage with CFS, in: Proc. of ACM SOSP '01, Banff, Canada, 2001, pp. 202–215.

[12] A. Goldberg, P. Yianilos, Towards an archival Intermemory, in: Proc. of ADL '98, Santa Barbara, California, 1998, pp. 147–156.

[13] W. Lin, D. Chiu, Y. Lee, Erasure code replication revisited, in: Proc. of IEEE P2P '04, Zurich, Switzerland, 2004, pp. 90–97.

[14] 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.

[15] S. Ramabhadran, J. Pasquale, Analysis of long-running replicated systems, in: Proc. of IEEE Infocom '06, Barcelona, Spain, 2006.

[16] A. Rowstron, P. Druschel, Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility, in: Proc. of ACM SOSP '01, Banff, Canada, 2001, pp. 188–201.

[17] G. Utard, A. Vernois, Data durability in peer to peer storage systems, in: Proc. of IEEE/ACM CCGRID 2004 (GP2PC 2004), Chicago, Illinois, 2004, pp. 90–94.

[18] H. Weatherspoon, J. Kubiatowicz, Erasure coding vs. replication: A quantitative comparison, in: Proc. of IPTPS '02, Cambridge, Massachusetts, Vol. 2429 of Lecture Notes in Computer Science, 2002, pp. 328–337.

[19] 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).

[20] Condor: High throughput computing, `http://www.cs.wisc.edu/condor/` (2007).

[21] Farsite: Federated, available, and reliable storage for an incompletely trusted environment, `http://research.microsoft.com/Farsite/` (2006).

[22] The OceanStore project: Providing global-scale persistent data, `http://oceanstore.cs.berkeley.edu/` (2005).

[23] PlanetLab, an open platform for developing, deploying, and accessing planetary-scale services, `http://www.planet-lab.org/` (2007).

[24] J. Stribling, PlanetLab - All Pairs Pings, `http://pdos.csail.mit.edu/~strib/pl_app` (2005).

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399