Reliability Analysis of Data Storage Systems

THÈSE N^{O} 5531 (2012)

PRÉSENTÉE LE 28 SEPTEMBRE 2012

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE DE THÉORIE DES COMMUNICATIONS

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Vinodh VENKATESAN

acceptée sur proposition du jury:

Prof. B. Falsafi, président du jury

Prof. R. Urbanke, Prof. C. Fragouli, directeurs de thèse

Prof. A. G. Dimakis, rapporteur

Dr E. Eleftheriou, rapporteur

Prof. P. Thiran, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse

2012

Abstract

Modern data storage systems are extremely large and consist of several tens or hundreds of nodes. In such systems, node failures are daily events, and safeguarding data from them poses a serious design challenge. The focus of this thesis is on the data reliability analysis of storage systems and, in particular, on the effect of different design choices and parameters on the system reliability.

Data redundancy, in the form of replication or advanced erasure codes, is used to protect data from node failures. By storing redundant data across several nodes, the surviving redundant data on surviving nodes can be used to rebuild the data lost by the failed nodes if node failures occur. As these rebuild processes take a finite amount of time to complete, there exists a non-zero probability of additional node failures during rebuild, which eventually may lead to a situation in which some of the data have lost so much redundancy that they become irrecoverably lost from the system. The average time taken by the system to suffer an irrecoverable data loss, also known as the mean time to data loss (MTTDL), is a measure of data reliability that is commonly used to compare different redundancy schemes and to study the effect of various design parameters. The theoretical analysis of MTTDL, however, is a challenging problem for non-exponential real-world failure and rebuild time distributions and for general data placement schemes. To address this issue, a methodology for reliability analysis is developed in this thesis that is based on the probability of direct path to data loss during rebuild. The reliability analysis is detailed in the sense that it accounts for the rebuild times involved, the amounts of partially rebuilt data when additional nodes fail during rebuild, and the fact that modern systems use an intelligent rebuild process that will first rebuild the data having the least amount of redundancy left. Through rigorous arguments and simulations it is established that the methodology developed is well-suited for the reliability analysis of real-world data storage systems. Applying this methodology to data storage systems with different types of redundancy, various data placement schemes, and rebuild constraints, the effect of the design parameters on the system reliability is studied.

When sufficient network bandwidth is available for rebuild processes, it is shown that spreading the redundant data corresponding to the data on each node across a higher number of other nodes and using a distributed and intel-

ligent rebuild process will improve the system MTTDL. In particular, declustered placement, which corresponds to spreading the redundant data corresponding to each node equally across all other nodes of the system, is found to potentially have significantly higher MTTDL values than other placement schemes, especially for large storage systems. This implies that more reliable data storage systems can be designed merely by changing the data placement without compromising on the storage efficiency or performance. The effect of a limited network rebuild bandwidth on the system reliability is also analyzed, and it is shown that, for certain redundancy schemes, spreading redundant data across more number of nodes can actually have a detrimental effect on reliability.

It is also shown that the MTTDL values are invariant in a large class of node failure time distributions with the same mean. This class includes the exponential distribution as well as the real-world distributions, such as Weibull or gamma. This result implies that the system MTTDL will not be affected if the failure distribution is changed to a corresponding exponential one with the same mean. This observation is also of great importance because it suggests that the MTTDL results obtained in the literature by assuming exponential node failure distributions may still be valid for real-world storage systems despite the fact that real-world failure distributions are non-exponential. In contrast, it is shown that the MTTDL is sensitive to the node rebuild time distribution.

A storage system reliability simulator is built to verify the theoretical results mentioned above. The simulator is sufficiently complex to perform all required failure events and rebuild tasks in a storage system, to use real-world failure and rebuild time distributions for scheduling failures and rebuilds, to take into account partial rebuilds when additional node failures occur, and to simulate different data placement schemes and compare their reliability. The simulation results are found to match the theoretical predictions with high confidence for a wide range of system parameters, thereby validating the methodology of reliability analysis developed.

Keywords: data storage, reliability, MTTDL, data placement, erasure codes, replication, simulation.

Résumé

Les systèmes modernes de stockage de données sont de très grande taille et se composent de plusieurs dizaines ou centaines d'éléments ou nœuds. Les pannes de ces nœuds sont des événements quotidiens pour de tels systèmes, d'où le défi qui consiste à protéger les données stockées contre ces pannes. Cette thèse se concentre sur l'analyse de la fiabilité du stockage de données et plus particulièrement de quelle façon différents choix de conception de ces systèmes peuvent influencer cette fiabilité.

La redondance sert à protéger les données contre des pannes des noeuds à l'aide de méthodes basées sur la replication ou des codes correcteurs. En stockant ces données redondantes sur plusieurs nœuds, les nœuds survivants servent à reconstruire les données perdues par ceux tombés en panne. Comme le processus de reconstruction de ces données n'est pas instantané, il existe un risque non nul que d'autres nœuds tombent en panne dans l'intervalle. Ceci peut conduire à une perte des données si une part trop grande des données redondantes est perdue. Le temps moyen jusqu'à ce qu'un tel système de stockage subisse une perte irrémédiable des données (MTTDL en anglais) est une mesure de la fiabilité du système qui est communément utilisée pour comparer différentes méthodes de redondance et étudier les effets de différents paramètres. Cependant, dans la pratique, une évaluation analytique de ce temps moyen (MTTDL) reste un défi car les distributions des pannes et des temps de reconstruction ne sont pas exponentielles ; en outre, ce MTTDL est aussi influencé par les diverses méthodes de placement des données redondantes.

Cette thèse adresse ce problème en développant une méthodologie pour l'analyse de la fiabilité de ces systèmes qui se base sur la probabilité de la succession d'événements menant le plus directement à la perte de données durant la reconstruction. L'analyse de la fiabilité tient compte des temps de reconstruction, de la quantité de données partiellement reconstruites lorsque de nouvelles pannes surviennent, et du fait que les systèmes actuels procèdent à une reconstruction intelligente en commençant par les données qui bénéficient de la redondance la plus faible. Une argumentation rigoureuse ainsi que des simulations montrent que cette méthodologie est adaptée à l'analyse de la fiabilité des systèmes de stockage pratiques. Cette méthodologie est ensuite appliquée à des systèmes de stockage utilisant différents types de redondance,

méthodes de placement, et contraintes de reconstruction, afin d'étudier l'effet de ces paramètres sur la fiabilité du système.

Dès lors que le débit du réseau d'interconnexion des nœuds permet des reconstructions intelligentes en parallèle, il ressort que la fiabilité du système augmente avec le nombre de nœuds utilisés pour répartir la redondance des données d'un nœud donné. En particulier, le placement dégroupé où la redondance est répartie également sur tous les autres nœuds du système bénéfice d'un MTTDL sensiblement supérieur à d'autres méthodes de placement, surtout pour des systèmes de grande taille. Ceci implique que la fiabilité des systèmes de stockage peut être améliorée simplement en changeant de méthode de placement sans altérer l'efficacité ou la performance du stockage. L'effet sur la fiabilité d'un débit limité du réseau est également analysé : pour certaines méthodes de redondance, une répartition de la redondance sur davantage de nœuds peut s'avérer alors préjudiciable.

Il est également démontré que le MTTDL est invariant pour une grande classe de distributions des pannes avec la même moyenne. Cette classe inclus non seulement la distribution exponentielle mais également des distributions pratiques telles que Weibull et gamma. Ceci implique que le MTTDL du système est inchangé si la distribution des pannes est remplacée par une distribution exponentielle de même moyenne. Cette observation est très importante vu les résultats de MTTDL obtenus précédemment dans la litérature qui considèrent des distributions exponentielles des pannes, car ces résultats deviennent valables pour des systèmes pratiques bien que les distributions pratiques ne soient pas exponentielles. En revanche, il est démontré que le MTTDL dépend du type de distribution utilisée pour le temps de reconstruction.

Afin de vérifier ces résultats théoriques, un simulateur a été développé pour mesurer la fiabilité des systèmes de stockage. Ce simulateur est suffisamment sophistiqué pour simuler les pannes et les reconstructions et utilise des distributions pratiques pour les pannes et les temps de reconstruction, en tenant compte des reconstructions partielles quand de nouvelles pannes surviennent, et en permettant de choisir différentes méthodes de placement. Les résultats des simulations correspondent aux résultats analytiques avec un haut degré de confiance pour une large gamme de paramètres du système, et valident ainsi la méthodologie choisie pour l'analyse de la fiabilité.

Mots-clés: stockage de données, fiabilité, MTTDL, placement des données, codes correcteurs, replication, simulation.

Acknowledgments

Firstly, I thank my advisers Prof. Rüdiger Urbanke and Prof. Christina Fragouli for their guidance and support throughout my doctoral studies. Their insightful feedback on my research has greatly helped me in asking the right questions and finding meaningful answers.

I would also like to thank my adviser at IBM Research, Dr. Ilias Iliadis, for the many hours of lengthy technical discussions on my work. His comments have significantly helped in improving the quality of the articles I wrote during my doctoral studies. My thanks also go to Dr. Robert Haas at IBM Research for his encouragement and for setting the direction of research for my doctoral thesis. Special thanks go to Dr. Evangelos Eleftheriou for providing me an opportunity to collaborate with IBM.

My thanks also go out to my colleagues at IBM who have made my stay there during the last four years a motivating, enriching, and wonderful experience. I would also like to thank all members of the Information Processing Group (IPG) and the group for Algorithmic Research in Network Information (ARNI) for making me feel at home during my visits to EPFL.

To amma and appa.

Contents

1 Introduction

In today's information age, more and more information is created, captured, stored, and replicated digitally – e-mails, photos, music, videos, medical records, employee records, government records, weather data, scientific measurements, online businesses, etc. – and the amount of new data created and stored is increasing at an exponential rate [1]. According to a study done in 2008 by the International Data Corporation (IDC), a market research and analysis firm, the total amount of data stored in 2007 was 281 billion gigabytes (or 281 exabytes) and it is expected to increase tenfold in five years [1].

Designing data storage systems that can handle this huge amount of information presents many challenges, chief among which are guaranteeing reliability, that is, being able to store data without any loss even while several components in the storage system may fail, providing high performance, that is, achieving low latency and high speed in delivering a requested piece of stored data to the user or writing a piece of user data to the system, and achieving good storage and energy efficiency, that is, reducing the actual amount of storage space used to store a certain amount of user data and the amount of energy needed to write, read, and maintain a certain amount of data reliably over the lifetime of the system.

In this thesis, we will focus on the reliability of data storage systems and, in particular, how different design choices and parameters affect the system reliability.

1.1 Motivation

Reliability of a data storage system corresponds to the *data reliability*, that is, how reliable the system is in storing data. A perfectly reliable storage system is one that never loses any data stored in it unless the data is explicitly deleted

by the user. However very few devices can store data reliably for extremely long periods of time. Compared to cave paintings and ancient scrolls that have survived for thousands of years, today's digital storage devices can hardly survive for more than a few years [2, 3], let alone decades or centuries. The inevitability of a data deluge and the unreliability of data storage devices pose a serious design challenge in building reliable data storage systems.

One of the widely-used methods to improve data reliability is replication of data on several storage devices [4]. As an example, consider a storage system that consists of just two storage devices. These devices could be a couple of hard disk drives, or a couple of tape drives, or even a couple of computers (or storage nodes). All the data that is stored on one of the devices is also replicated on the second device. Eventually, at some point in time, one of the devices will fail. At this point, the failed device is replaced by a new device and the surviving replica of the data from the other device is copied to the new device. In this way, the system attempts to maintain two replicas of the data at all times. Unfortunately, the replica restoration process takes a finite amount of time and this gives rise to a non-zero probability of failure of the second device during this time, leading to irrecoverable data loss. As the time taken for a device to fail is random, the time taken for the system to end up in irrecoverable data loss is also random. Understanding the characteristics of the time to data loss and how it is affected by different system designs and parameters can help in building more reliable storage systems. However, the reliability analysis, that is, the characterization of the time to data loss, for even such a simple storage system with two devices can be quite cumbersome for general distributions of times to device failures and times to rebuild devices. The reliability analysis becomes extremely non-trivial especially for large systems with certain replica placement schemes or when other forms of redundancy, such as erasure codes, are used.

In this thesis, we develop analytical techniques to overcome these challenges in reliability analysis of data storage systems and propose methodologies to analyze and compare different system designs in terms of reliability.

1.2 Related Work

This section provides an overview of some of the earlier work related to the reliability analysis of data storage systems and how this thesis differs from or improves upon the existing body of literature on this topic.

1.2.1 Replication-based Systems

The problem of using replication to reliably maintain state in a distributed system for time spans that far exceed the lifetimes of individual replicas is addressed in [4]. This scenario is relevant for any system consisting of a potentially large and selectable number of replicated components, each of which

may be unreliable, where the goal is to have enough replicas to keep the system "alive" (meaning at least one replica is working or available) for a certain expected period of time, i.e., the system's lifetime. In particular, this applies to recent efforts to build highly available storage systems based on the peer-to-peer paradigm. The paper studies the impact of practical considerations, such as storage and bandwidth limits on the system, and presents methods to optimally choose system parameters so as to maximize lifetime. The analysis presented reveals that, for most practical scenarios, it is better to invest the available repair bandwidth in aggressively maintaining a small number of replicas than in spreading it across a large number of replicas. Only one particular replica placement scheme is considered, where the number of nodes is equal to the number of replicas. The analysis also assumes the failure and rebuild times to be independent and exponentially distributed. A drawback of the continuous-time Markov chain based analysis presented in this paper is that it cannot be easily extended to other placement schemes that employ distributed rebuild strategies. This is because these schemes can enter into a combinatorially large number of different states leading to an extremely large Markov model. Also, the use of such Markov models is not possible when the failure and rebuild time distributions are non-exponential. In this thesis, we address both these problems and develop an analytical framework that enables the reliability estimation of more general placement schemes under non-exponential failure and rebuild time distributions.

1.2.2 Redundancy Placement

Replication-based decentralized storage systems, such as CFS [5], OceanStore [6], Ivy [7], and Glacier [8], employ a variety of different strategies for replica placement and maintenance. In architectures that employ distributed hash tables (DHT), the choice of algorithm for data replication and maintenance can have a significant impact on both performance and reliability [9]. The paper presents a comparative analysis of replication algorithms that are based upon a specific DHT design. It also presents a novel maintenance algorithm for dynamic replica placement and considers the reliability of the resulting designs at the system level. That work proposes five different data placement schemes and the most reliable scheme is shown to be the block placement scheme. In this scheme, the system is divided into disjoint sets of r nodes and each set of r nodes store r copies (replicas) of the same data, where r is the replication factor. Similar results are also presented in [10]. However, the reliability analysis in these works is done for the case when there are no rebuild operations performed. This is a serious drawback of the analysis as the rebuilds are a crucial factor in determining the reliability of a system. As we will show in this thesis, accounting for rebuild times leads to a more accurate reliability analysis of practical storage systems.

Placement of redundant data with emphasis on erasure coding has been considered in [11]. In that work, the comparison of the means time to data loss

of different placement schemes have been done through simulations. However, there is no analytical characterization of the mean times to data loss in terms of the system parameters. In this thesis, we provide an analytical expression for the mean time to data loss of different redundancy placement schemes in an erasure-coded storage system that gives an insight into the effect of different parameters on the system reliability.

1.2.3 Recovery Mechanism

Both the recovery mechanism and the replica placement scheme affect the reliability of a system. Fast recovery mechanisms, such as rebuilding onto reserved spare space on surviving storage nodes instead of on a new spare node, reduce the window of vulnerability and improve the system reliability [12, 13, 14]. The replica placement scheme also plays an important role in determining the duration of rebuilds. In particular, distributing replicas over many storage nodes in the system reduces the rebuild times, but also increases the exposure of data to failure. For a replication factor of two, these two effects cancel out, and therefore, all placement schemes have similar reliability [15]. The distributed fast recovery scheme proposed in that work not only improves performance during data recovery, but also improves reliability. However, their analysis is based on an idealistic assumption that replica-sets (referred to as redundancy sets and groups in [12, 13], and as objects in [14]) fail independently. In contrast, in our analysis we assume that nodes fail independently, and take into account the correlations among different replica-sets that this induces. As we show in this thesis, this leads to different results.

1.2.4 Failure and Rebuild Time Distributions

Several of the earliest works on the analysis of reliability of storage systems [16] have assumed independent and exponentially distributed times to failure. A vast majority of the publications have also assumed exponentially distributed times to rebuild as this allows the use of continuous-time Markov chain models to estimate the reliability of the system [4, 12, 14]. A few works have used other probabilistic methods [9, 10], however, the probability of data loss in these works is obtained for the case when there are no rebuild operations performed. Publications based on real world failure data have shown that the distribution of failures is neither exponential nor independent [3]. Failure distributions other than exponential have been studied extensively through simulations [11, 13, 17]. In particular, it has been shown that the expected number of double disk failures in RAID-5 systems within a given time period can vary depending on the failure time distribution [17]. In contrast, we consider the expected time to the first data loss event (which is equivalent to a double disk failure in case of a RAID-5 system) and we show that it tends to be insensitive to the failure time distribution. More generally, in this thesis, we show that the mean time to data loss of a system tends to be insensitive to a large class of failure time

distributions including, most importantly, the non-exponential distributions that are observed in real-world storage nodes.

1.2.5 Network Rebuild Bandwidth Constraints

Dependence of system reliability on the data placement scheme without any network rebuild bandwidth limitations has been studied extensively in the literature [10, 12, 13, 14, 15, 18]. Effect of network rebuild bandwidth constraints on system reliability has been studied in [14]. Our results closely match the mean time to data loss results of [14]. In addition, we derive closed form expressions for the mean time to data loss and provide further insight into the effect of placement schemes on the reliability behavior of systems under network rebuild bandwidth constraints.

Other works in literature [19, 20] have proposed novel coding schemes that address the problem of reducing the bandwidth usage during reconstruction. The reliability analysis of these and other related works, however, have not accounted for rebuild times.

1.3 Summary of Results

The following gives a brief summary of the results of this thesis:

- Developed a methodology for reliability analysis of data storage systems.
  - such analysis was challenging due to several factors including
    - the complexity of failure and rebuild processes of storage nodes in a large-scale system
    - the difficulty in using continuous-time Markov chain models for data placement schemes that can enter into an extremely large number of possible states due to failures and rebuilds
    - the fact that real-world failure and rebuild time distributions are not exponential
  - due to the above mentioned challenges, time-consuming simulations were required to study the reliability of large-scale storage systems
  - the methodology developed enables one to assess system reliability analytically and obtain greater insight into the effects of different system designs and parameters on the reliability through closed-form relations
- Showed that, when unlimited network rebuild bandwidth is available, spreading redundant data (replicas or codewords) corresponding to each storage node across more number of other nodes improves the mean time to data loss (a measure of data reliability).

- Established the applicability of the developed methodology to a wide variety of failure and rebuild time distributions including, most importantly, real-world distributions.
- Studied the effect of limited network rebuild bandwidth on the system reliability and showed that, for certain redundancy schemes, spreading redundant data across more number of nodes can actually have a detrimental impact on reliability.
- Built a storage system simulator to verify the theoretical results described above. The simulator was sufficiently complex
  - to perform all required failure events and rebuild tasks in a storage system
  - to use real-world failure and rebuild time distributions for scheduling failures and rebuilds
  - to take into account partial rebuilds when additional node failures occur
  - to simulate different data placement schemes and compare their reliability

1.4 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 describes the modeling of data storage systems. The methodology of estimation of system reliability is detailed in Chapter 3. Chapter 4 applies the reliability analysis methodology developed in the previous chapter to replication-based systems and shows how replica placement affects system reliability. Impact of rebuild constraints on system reliability is analyzed in Chapter 5. The reliability analysis methodology is then applied to systems based on erasure-codes in Chapter 6. Chapter 7 describes the simulator built to verify the theoretical results. Finally, Chapter 8 concludes this thesis and points to possible directions for future work.

2 System Model

Modern data storage systems are complex in nature consisting of several components of hardware and software. To perform a reliability analysis, we require a model that abstracts the reliability behavior of this complex system and lends itself to theoretical analysis, but at the same time, preserves the core features that affect the system failures and rebuilds. In this chapter, we develop and describe a relatively simple yet powerful model that captures the essential reliability behavior of a storage system.

2.1 Storage System

The storage system is modeled as a collection of n identical *storage nodes* each of which stores c amount of data. The storage nodes are identical in the sense that their reliability behaviors are the same, that is, they have the same time to failure distribution, the same read/write bandwidth available for rebuild, and the same time to rebuild distributions. Most modern storage systems today store data on the same type of storage devices and distribute data equally across all devices for performance reasons. Therefore, this is a reasonable modeling assumption for today's storage systems. Future workload optimized storage systems may store data on different types of devices and the amount of data stored on a device may depend on its type [21]. If the data stored on different types of devices are independent, that is, if different types of devices do not share replicas or redundancies, such systems may be modeled as consisting of a number of smaller independent storage subsystems each of which are made of identical nodes storing equal amounts of data.

The total amount of data stored in the system is nc and this includes the redundant data that is created to improve data reliability. For example, in a two-way replicated system, the total data in the system, nc, consists of two

Table 2.1: Parameters of a storage system

c	amount of data stored on each storage node (bytes)
n	number of storage nodes
$1/\lambda$	mean time to failure of a storage node (s)
$1/\mu$	mean time to read/write c amount of data from/to a node (s)

copies (replicas) of $nc/2$ amount of *user data*. By user data, we refer to the data that the user of the storage system stores in the system. This excludes the redundancies that the system creates within itself to improve the data reliability. For example, in a three-way replicated system, the total amount of user data is $nc/3$. In addition to the c amount of data that is stored, each node is assumed to have sufficient spare space that may be used for a distributed rebuild process when other nodes fails. Although the exact amount of spare space required to handle all distributed rebuilds throughout the lifetime of the system may depend on a number of factors, including node failure and rebuild times, it can be shown to be of the order of c/n. The main parameters used in the storage system model are listed in Table 2.1.

As the primary aim is to analyze the *data* reliability and all the data is stored on storage nodes, only the failures and rebuilds of these storge nodes are modeled. The outage of network resources used to access the storage nodes is not explicitly modeled. Such an outage may cause one or more nodes to become unreachable temporarily. In such cases, the affected nodes are said to be *unavailable*. Nodes may also become unavailable for other reasons, such as, software upgrades, node reboots, etc. [22].

To extend the lifetime of data far beyond the lifetimes of individual storage nodes, the user data is stored along with redundant data. This redundancy may be simple replicas of the user data or may be some form of erasure codes. When a node failure occurs, the redundant data corresponding to the data on the failed node may still be available on other surviving storage nodes. This redundant data is then used to restore the lost data on a new replacement node. This restoration (or rebuild) process can be of two types, namely, direct rebuild, or distributed rebuild. In direct rebuild, the redundant data is read from surviving nodes, the lost data is reconstructed in a streaming manner (which may just involve copying, in the case of replication-based systems, or some form of decoding, in the case of erasure codes based systems), and the reconstructed data is directly stored on a new replacement node. In direct rebuild, the read-write bandwidth of the new replacement node may typically be the bottleneck for the rebuild process. In distributed rebuild, the redundant data is read from surviving nodes, the lost data is reconstructed in a streaming manner, and the reconstructed data is stored in the spare space of surviving nodes (making sure that redundancies corresponding to the same piece of data are not stored on the same node). Once all lost data has been restored, the

newly restored data is transferred from the surviving nodes to a new node. Distributed rebuild makes use of the read-write bandwidth available at all relevant surviving nodes and is therefore typically faster than direct rebuild.

The rebuild process following each node failure takes a finite amount of time and this leads to the possibility of additional node failures within this time, which can lead to an increased loss of redundancy. For example, in a three-way replicated system, a second node failure during the rebuild of the first may (depending on the replica placement and on which nodes failed) lead some data to lose its second copy. Most of the time, the rebuild processes complete, and all lost redundancies are restored. However, eventually, a series of catastrophic node failures occur that wipe out some data along with all its redundancy. In this thesis, we refer to this situation as *irrecoverable data loss*, or simply *data loss*. The time taken for a system to suffer data loss depends on system parameters and is a random variable. In our reliability analysis, we wish to estimate characteristics of this time to data loss and understand how it is affected by various system parameters and designs.

2.2 Storage Node

Each storage node, in and of itself, is a fairly complex entity and comprises of disks, memory, processor, network interface, and power supply. Any of these components can fail and cause the node to either become temporarily unavailable or permanently fail. As an abstraction, the details of which component failure led to a node failure is left out in our model. It is assumed that there is some mechanism, such as regular pinging of each node, in place to detect node failures as they occur. In large-scale storage systems, diagnosing the exact cause of a detected node failure and fixing the problem immediately may not be a viable option. Therefore, a detection of a node failure automatically triggers a rebuild process to restore the data.

2.2.1 Node Unavailability vs. Node Failure

The difference between node failure and temporary unavailability is crucial to the reliability model. Temporary node unavailability may result in temporary data unavailability, that is, data may become temporarily unavailable but not completely lost from the system. On the other hand, node failures may result in irrecoverable data loss, which is a more serious issue. The primary focus of this thesis will be on the study of irrecoverable data loss, although some of methodologies developed may also be applicable to the study of data unavailability.

As noted in [22], more than 90% of the node unavailabilities are transient and do not last for more than 15 minutes. Furthermore, it is also observed that the majority of these node unavailabilities are due to planned reboots, and that unplanned reboots and other unknown events are only a small proportion of all

events that cause nodes to become unavailable. As most of the unavailabilities are transient, a node rebuild process is initiated only if a node stays unavailable for more than 15 minutes [22]. In other word, node unavailabilities lasting longer than a certain amount of time are treated as node failures.

2.2.2 Independence of Node Failures

Node unavailabilities have been observed to exhibit strong correlation that may be due to short power outages in the datacenter, or part of rolling reboot or upgrade activity at the datacenter management layer [22]. However, only less than 10% of the node unavailabilities last longer than 15 minutes and are treated as node failures which trigger a rebuild process. There is no indication that correlations exist among such node failures. It has been argued that disk (as opposed to node) replacement rates in large storage systems show correlations [3]. However, as disks have been observed to be more reliable than other components of a node [23], the failure of a node is mainly determined by the failure of these other components. As there is no evidence that correlations exist among node failures (or permanent unavailabilities), we assume node failures to be independent in our model.

2.3 Redundancy

A data storage system is made of storage nodes that have relatively short lifetimes. To extend the lifetime of user data far beyond the lifetimes of the individual nodes, the system creates redundant data corresponding to the user data and stores it across different nodes so that when node failures occur this redundant data can be used to restore the data lost by the failed nodes.

One of the simplest forms of redundancy is replication. Given a replication factor r, the system stores r copies of the user data in the system such that no two copies are stored on the same node. Therefore, for a system storing a total amount of data nc, the corresponding user data is only nc/r, resulting in a *storage efficiency* of $1/r$. Storage efficiency is defined as the ratio of the amount of user data to the corresponding amount of data stored in the system.

Another form of redundancy are the so-called erasure codes, where the user data is divided into blocks of a fixed size (or symbols) and each set of l blocks is encoded into a set of $m > l$ blocks, called a codeword, before storing them on m distinct nodes. The encoding is done in such a way that *some* subset of m symbols of a codeword can be used to decode the l symbols of user data corresponding to that codeword. The storage efficiency of an erasure code is clearly l/m. Optimal erasure codes or *maximum distance separable* codes (MDS codes) have the property that any l out of m symbols can be used to decode a codeword. This type of redundancy is called a (l, m)-erasure code.

Typically, the advantage of an erasure coded system over a replication-based system is that it can offer much better reliability for the same storage

efficiency, or much higher storage efficiency for the same reliability. The advantage of a replication-based system over an erasure coded system is in performance. Erasure coded systems typically offer only one copy of the user data whereas replication-based systems offer r copies. Furthermore, an update of any one block of user data in an erasure code system will require reading of the existing codeword corresponding to that block, updating that codeword and writing the codeword back to the system. In contrast, an update of any piece of user data just requires overwriting the existing replicas of that piece in the system and does not require any additional reads or processing.

2.4 Data Placement

In a large storage system, the number of nodes, n, is typically much larger than the replication factor, r, or the codeword length, m. Therefore, there exist many ways in which r replicas or a codeword of m blocks can be stored across n nodes.

2.4.1 Clustered Placement

If n is divisible by r (or m), one simple way would be to divide the n nodes into disjoint sets of r (or m) nodes each and store the replicas (or codeword blocks) across the nodes in each set. We refer to this type of data placement as *clustered* placement, and each of these disjoint sets of nodes as *clusters*. It can be seen that, in such a placement scheme, no clusters share replicas or redundancies corresponding to data on another cluster. The storage system can essentially be modeled as consisting of n/r (or n/m) independent clusters. Reliability behavior of a cluster under exponential failure and rebuild time distributions is well-known [4, 16]. The reliability analysis of cluster placement is relatively straightforward compared to other more general placement schemes because each cluster is independent of the others and can be modeled by a continuous-time Markov chain when the failure and rebuild times are assumed to be exponentially distributed. More general placement schemes, however, can enter into a combinatorially large number of states leading to an extremely large Markov model. In addition, real-world failure and rebuild time distributions are known to be non-exponential [3], and this prevents one from using Markov models for the analysis.

2.4.2 Declustered Placement

A placement scheme that can potentially offer far higher reliability than the clustered placement scheme, especially as the number of nodes in the system grows, is the *declustered* placement scheme. There exists $\binom{n}{r}$ (or $\binom{n}{m}$) different ways of placing r replicas of a user data block (or m symbols of a codeword) in n nodes. In this scheme, all these $\binom{n}{r}$ (or $\binom{n}{m}$) possible ways are equally used to

store data. It can be seen that, in such a placement scheme, when a node fails, the redundancy corresponding to the data on the failed node is equally spread across the remaining surviving nodes. This allows one to use the rebuild read-write bandwidth available at all surviving nodes to do a *distributed* rebuild in parallel, which can be extremely fast when the number of nodes is large. In contrast, in clustered placement scheme, when a node fails, the redundancy corresponding to the data on the failed node is only spread across the remaining nodes of a cluster. Therefore, a fast parallel rebuild process that scales with the number of nodes is not possible for clustered placement. As it turns out, this is one of the main reasons why declustered placement can offer significantly higher reliability than clustered placement for large systems.

2.4.3 Spread Factor

A broader set of placement schemes can be defined using the concept of *spread factor*. For each node in the system, its *redundancy spread factor* is defined as the number of nodes over which the data on that node and its corresponding redundant data are spread. In a replication-based (or erasure coded) system, when a node fails, its spread factor determines the number of nodes which have replicas of (or the codeword symbols corresponding to) the lost data, and this in turn determines the degree of parallelism that can be used in rebuilding the data lost by that node. In this thesis, we will consider symmetric placement schemes in which the spread factor of each node is the same, denoted by k. In a symmetric placement scheme, the $r-1$ replicas of (or the $m-1$ codeword symbols corresponding to) the data on each node are *equally* spread across $k-1$ other nodes, the $r-2$ replicas of (or the $m-2$ codeword symbols corresponding to) the data shared by any two nodes are equally spread across $k-2$ other nodes, and so on. One example of such a symmetric placement scheme is the clustered placement scheme for which the spread factor, k, is equal to the replication factor, r (or the codeword length, m). Another example of a symmetric placement scheme is the declustered placement scheme for which the spread factor, k, is equal to the number of nodes, n. A number of different placement schemes can be generated by varying the spread factor k. The spread factor of a placement scheme is important in two ways: (a) it determines the number of nodes over which data of a failed node is spread and therefore, the degree of parallelism that can be used in the rebuild process of that node, and (b) it determines the amount of data that becomes critical, that is, the amount of data with the most number of replicas or codeword symbols lost, which needs to be rebuilt first when additional node failures occur. It can be seen that, any two nodes sharing replicas of some data, share replicas of exactly $\frac{r-1}{k-1}c$ amount of user data. Likewise, any two nodes storing codeword symbols of some data, store the codeword symbols of exactly $\frac{m-1}{k-1}c$ amount of user data. In general, any set of $\tilde{n}$ nodes ($\tilde{n} \leq r$) sharing replicas of some data, share replicas of exactly $c\prod_{i=1}^{\tilde{n}-1}\left(\frac{r-i}{k-i}\right)$ amount of user data. Similarly,

any set of $\tilde{n}$ nodes ($\tilde{n} \leq m$) storing codeword symbols of some data, store the codeword symbols corresponding to exactly $c\prod_{i=1}^{\tilde{n}-1}\left(\frac{m-i}{k-i}\right)$ amount of user data.

2.5 Node Failure

Based on the discussion in Section 2.2, the times to node failures are modeled as independent and identically distributed random variables. Let T_F denote the time to failure of a node. Its cumulative distribution function is denoted by $F_\lambda(\cdot)$, that is,

$$F_\lambda(t) := \Pr\{T_F \leq t\}, \quad t \geq 0, \tag{2.1}$$

its probability density function is denoted by $f_\lambda(\cdot)$, that is,

$$f_\lambda(t) := \frac{dF_\lambda(t)}{dt}, \quad t \geq 0, \tag{2.2}$$

and its mean is denoted by $1/\lambda$, that is,

$$\frac{1}{\lambda} := \int_0^\infty (1 - F_\lambda(t))dt. \tag{2.3}$$

Typically, F_λ is assumed to be exponential as this allows one to use Markov models for analysis. However, it has been observed that real-world storage devices do not have exponentially distributed failure times [3, 2]. This can also be seen to be intuitively true because an exponentially distributed failure time implies that the device has a constant failure rate independent of its age. However, it is common experience that devices age over time and that the failure rates of very old devices are much higher. It has been seen that Weibull and gamma distributions are a much better fit to the empirical data on storage device lifetimes [3]. It has also been noted that infant mortality, that is, extremely short lifetimes, is not evident in real-world storage systems. This may be due to pre-stressing devices before installing them in a system, thereby making sure that devices with short lifetimes are discarded during the pre-stressing process. An interesting result of this thesis is that the mean time to data loss of a storage system tends to be invariant within a large class of failure time distributions, that includes the exponential distribution and, most importantly, real-world distributions like Weibull and gamma (see Chapter 3).

2.6 Node Rebuild

We will describe the node rebuild process in a replication-based system here. The rebuild process for an erasure-coded system is the same except for the fact that we have m codeword blocks instead of r replicas.

In a replication-based storage system, when nodes fail, data blocks lose one or more of their r replicas. The purpose of the rebuild process is to recover all

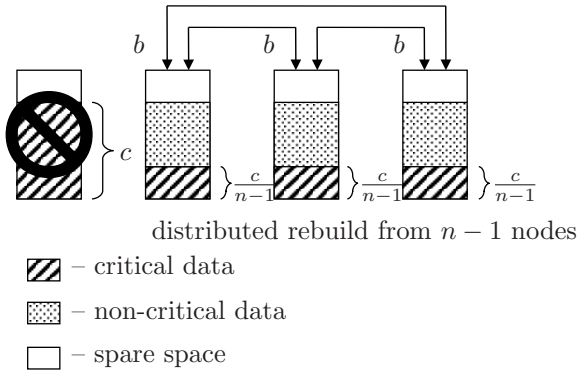


Figure 2.1: Example of the distributed rebuild model for a two-way replicated system. When one node fails, the critical data blocks are equally spread across the $n-1$ surviving nodes. The distributed rebuild process creates replicas of these critical blocks by copying them from one surviving node to another in parallel.

replicas lost, so that all data have r replicas. A good rebuild process needs to be both *intelligent* and *distributed*.

In an intelligent rebuild process, the system attempts to first recover the replicas of the blocks that have the least number of replicas left. As an example, consider a system that has $D_0, D_1, \cdots, D_{e-1}$, and D_e distinct number of data blocks which have lost $0, 1, \cdots, e-1$, and e replicas, respectively, and no blocks that have lost more than e replicas, for some e between 1 and $r-1$. An intelligent rebuild process attempts to first create an additional copy of the D_e blocks that have lost e replicas, because these are the blocks that are the most vulnerable to data loss if additional nodes fail. If it is successful and if no other failure occurs in between, then the system will have $D_0, D_1, \cdots, D_{e-2}$, and $D_{e-1} + D_e$ distinct data blocks which have lost $0, 1, \cdots, e-2$, and $e-1$ replicas, respectively. Then the rebuild process creates an additional copy of the $D_{e-1}+D_e$ data blocks and so forth until all replicas lost have been restored. In contrast to the intelligent rebuild, one may consider a *blind* rebuild, where lost replicas are being recovered in an order that is not specifically aimed at recovering the data blocks with the least number of replicas first. Clearly, an unintelligent rebuild is more vulnerable to data loss. In the remainder of the paper we consider only intelligent rebuild.

In placement schemes such as the declustered scheme, the surviving replicas that the system needs to read to recover the lost replicas may be spread across several, or even all, surviving nodes. Broadly speaking, two approaches can be taken when recovering the lost replicas: the data blocks to be rebuilt can be read from all the nodes in which they are present, and either (i) copied directly to a new node, or (ii) copied to (reserved) spare space in all surviving nodes first and then to a new node. The latter method is referred to as distributed

rebuild and has a clear advantage in terms of time to rebuild because it exploits parallelism when writing to many (surviving) nodes versus writing to only one (new) node. The time required to copy the restored data to the new node is irrelevant from a data reliability point of view, as additional node failures that occur after the lost replicas are restored do not cause the system to become critical. As an example, consider a two-way replicated system with declustered placement as shown in Fig. 2.1. When the first node fails, the surviving replicas of the c amount of data on the failed node are spread equally across all the remaining $n-1$ nodes. A distributed rebuild process copies the replicas of the critical data from one surviving node to another such that no two replicas of the same data end up on the same node. Once the distributed rebuild is complete, the system has two replicas of all data spread across $n-1$ nodes. If another node fails before the restored data is transferred to a new replacement node, then some of the data lose only one replica; no data loses two replicas. So it essentially functions as a system with $n-1$ nodes, each with $cn/(n-1)$ amount of data, until the restored data of the first failed node is fully copied to a new replacement node.

During the rebuild process, an average read-write bandwidth of $c\mu$ bytes/s is assumed to be reserved at each node exclusively for the rebuild. This implies that the average time required to read (or write) c amount of data from (or to) a node is equal to $1/\mu$. The average rebuild bandwidth is usually only a fraction of the total bandwidth available at each node; the remainder is being used to serve user requests. Let T_R denote the time required to read (or write) c amount of data from (or to) a node. Denote its cumulative distribution function by $G_\mu(\cdot)$, that is,

$$G_\mu(t) := \Pr\{T_R \le t\}, \quad t \ge 0, \tag{2.4}$$

and its probability density function by $g_\mu(\cdot)$, that is,

$$g_\mu(t) := \frac{dG_\mu(t)}{dt}, \quad t \ge 0. \tag{2.5}$$

In clustered placement, it is assumed that there are spare nodes, and when a node fails, data is read from any *one* of the surviving nodes[1] of the cluster to which the failed node belonged and written to a spare node at an average rate of $c\mu$. Let $T_{R_\alpha}^{\text{clus.}}$ be the time taken to rebuild a fraction α of a node in clustered placement, that is, read αc amount of data from one of the surviving nodes of the cluster and write to a new node at an average rate of $c\mu$. Denote its mean by $1/\mu_\alpha^{\text{clus.}}$. It is clear that

$$\frac{1}{\mu_\alpha^{\text{clus.}}} = \frac{\alpha}{\mu}. \tag{2.6}$$

Denote the cumulative distribution function of $T_{R_\alpha}^{\text{clus.}}$ by $G_{\mu_\alpha^{\text{clus.}}}(\cdot)$, that is,

$$G_{\mu_\alpha^{\text{clus.}}}(t) := \Pr\{T_{R_\alpha}^{\text{clus.}} \le t\}, \quad t \ge 0. \tag{2.7}$$

[1]In the case of an (l, m)-MDS code, the data is read from any l of the surviving nodes.

In declustered placement, it is assumed that sufficient spare space is reserved in each node for rebuild. During rebuild, the data to be rebuilt is read from *all* surviving nodes and copied to the spare space reserved in these nodes in such a way that no data block is copied to the spare space of a node in which a copy of it is already present. Since data is being read from and written to each surviving node, the total average read-write rebuild bandwidth $c\mu$ of each node is equally split between the reads and the writes. So if there are $\tilde{n}$ surviving nodes, the total average speed of rebuild in the system is $(\tilde{n}c\mu)/2$. Let $T_{R_\alpha}^{\text{declus.}}$ denote the time taken to rebuild a fraction α of a node in declustered placement. Denote its mean by $1/\mu_\alpha^{\text{declus.}}$. It is clear that

$$\frac{1}{\mu_\alpha^{\text{declus.}}} = \frac{\alpha}{(\tilde{n}\mu/2)} \tag{2.8}$$

It can be seen by comparing (2.8) to (2.6) that the rebuild time for declustered placement scheme can be much lesser than that for clustered placement scheme for the same amount of data, especially for systems with a large number of storage nodes. Denote the cumulative distribution function of $T_{R_\alpha}^{\text{declus.}}$ by $G_{\mu_\alpha^{\text{declus.}}}(\cdot)$, that is,

$$G_{\mu_\alpha^{\text{declus.}}}(t) := \Pr\{T_{R_\alpha}^{\text{declus.}} \leq t\}, \quad t \geq 0. \tag{2.9}$$

2.7 Failure and Rebuild Time Distributions

It is known that real world storage nodes are *generally reliable*, that is, the mean time to repair a node (which is typically of the order of tens of hours) is much smaller than the mean time to failure of a node (which is typically at least of the order of thousands of hours). As $1/\lambda$ denotes the mean time to failure of node and $1/\mu$ denotes the mean time to read (or write) c amount of data from (or to) a storage node, it follows that generally reliable nodes satisfy the following condition:

$$\frac{1}{\mu} \ll \frac{1}{\lambda}, \quad \text{or} \quad \frac{\lambda}{\mu} \ll 1. \tag{2.10}$$

In the subsequent analysis, this condition implies that terms involving powers of λ/μ greater than one are negligible compared to λ/μ and can be ignored.

Let the cumulative distribution functions F_λ and G_μ satisfy the following condition:

$$\mu \int_0^\infty F_\lambda(t)(1 - G_\mu(t))dt \ll 1, \quad \text{with } \frac{\lambda}{\mu} \ll 1. \tag{2.11}$$

The results of this thesis are derived for the class of failure and rebuild distributions that satisfy the above condition. In particular, the mean time to data loss of a system is shown to be insensitive to the failure distributions within

this class. This result is of great importance because it turns out that this condition holds for a wide variety of failure and rebuild distributions including, most importantly, distributions that are seen in real-world storage systems.

As an illustration, let us consider the class of failure distributions that satisfy the above conditions, when the rebuild times are deterministic, that is,

$$G_\mu(t) = \begin{cases} 0, & \text{when } t < 1/\mu, \\ 1, & \text{when } t \geq 1/\mu. \end{cases} \tag{2.12}$$

Recognizing that F_λ is a monotonically non-decreasing function such that $F_\lambda(t) \leq F_\lambda(1/\mu)$ for $t \leq 1/\mu$, the left hand side of (2.11) reduces to

$$\mu \int_0^{1/\mu} F_\lambda(t)dt \leq F_\lambda(1/\mu). \tag{2.13}$$

When $\lambda/\mu \ll 1$, it can be seen that $F_\lambda(1/\mu) \ll 1$ for a wide variety of distributions including exponential, Weibull (with shape parameter greater than 1), and gamma (with shape parameter greater than 1). For instance, consider a Weibull distribution with shape parameter k and scale parameter β having the cumulative distribution function

$$F_\lambda^{\text{Weibull}}(t) = 1 - e^{-(t/\beta)^k}. \tag{2.14}$$

The mean of the Weibull distribution, $1/\lambda$, is equal to $\beta\Gamma(1+1/k)$, where $\Gamma(\cdot)$ denotes the gamma function. Therefore, the scale parameter β can be written in terms of the mean $1/\lambda$ as

$$\beta = 1/(\lambda\Gamma(1 + 1/k)). \tag{2.15}$$

Substituting (2.15) in (2.14) for $t = 1/\mu$ we get

$$F_\lambda^{\text{Weibull}}(1/\mu) = 1 - e^{-(\lambda\Gamma(1+1/k)/\mu)^k} \ll 1, \tag{2.16}$$

when $\lambda/\mu \ll 1$ and $k \geq 1$. However, if $k < 1$, the above inequality may not hold. Note that nodes that have Weibull lifetime distributions with $k < 1$ have high infant mortality rate, whereas those with Weibull lifetime distributions with $k > 1$ gracefully age over time.

In general, it can be observed that failure distributions with high infant mortality rates do not satisfy condition (2.11). However, it has been observed that infant mortality is not present in real world storage nodes [3]. Furthermore, the effects of infant mortality can be eliminated from the system by stressing new nodes before adding them to the system. It can also be observed that (2.11) is satisfied by a wide variety of distributions for rebuild times, in particular, distributions with bounded support. Therefore, condition (2.11) is realistic as it is satisfied by practical storage systems.

Condition (2.11) can also be stated in the following alternate way: if F_λ and G_μ belong to a *family* of distributions characterized by λ and μ, respectively, then (2.11) is equivalent to

$$\lim_{\lambda/\mu \to 0} \mu \int_0^\infty F_\lambda(t)(1 - G_\mu(t))dt = 0. \tag{2.17}$$

For a fixed μ, this implies that

$$\lim_{1/\lambda \to \infty} \mu \int_0^\infty F_\lambda(t)(1 - G_\mu(t))dt = 0. \tag{2.18}$$

As $F_\lambda(t)(1 - G_\mu(t)) \leq 1 - G_\mu(t)$ and $1 - G_\mu(t)$ is integrable, by the dominated convergence theorem, the order of limit and integral can be exchanged. Therefore,

$$\lim_{1/\lambda \to \infty} \mu \int_0^\infty F_\lambda(t)(1 - G_\mu(t))dt = \mu \int_0^\infty \lim_{1/\lambda \to \infty} F_\lambda(t)(1 - G_\mu(t))dt. \tag{2.19}$$

Therefore, for a fixed μ, (2.18) holds only when

$$\lim_{1/\lambda \to \infty} F_\lambda(t) = 0, \forall\, t \text{ where } G_\mu(t) < 1, \tag{2.20}$$

and the convergence of F_λ is pointwise. Similarly, it can be shown that, for a fixed λ, (2.17) holds only when

$$\lim_{1/\mu \to 0} \mu(1 - G_\mu(t)) = 0, \forall\, t \text{ where } F_\lambda(t) > 0, \tag{2.21}$$

and the convergence is pointwise. Note that (2.20) and (2.21) can be equivalently written as

$$F_\lambda(t) \ll 1 \text{ when } G_\mu(t) < 1 \text{ and } \lambda \ll \mu, \tag{2.22}$$

$$\mu(1 - G_\mu(t)) \ll 1 \text{ when } F_\lambda(t) > 0 \text{ and } \mu \gg \lambda. \tag{2.23}$$

3 Reliability Estimation

Reliability analysis of data storage systems is a non-trival problem for general failure and rebuild time distributions and for general data placement schemes. Traditional continuous-time Markov models based analysis is not applicable to real-world failure and rebuild time distributions, which are observed to be non-exponential [3], and for certain placement schemes such as the declustered placement, even a continuous-time Markov model (under the assumption of exponential distribution of failure and rebuild times) becomes extremely complex.

To overcome these challenges, we develop a methodology for reliability analysis in this chapter through a series of approximations each of which are justified for generally reliable nodes and for failure and rebuild time distributions belonging to a certain broad class (see Section 2.7). This class of distributions includes real-world distributions that are typically non-exponentially and are modelled by Weibull or gamma distributions. The methodology developed is powerful enough to be applied to a wide variety of data placement and redundancy schemes, and can also be used to study the impact of certain constraints on the system during rebuild. The theoretical estimates of mean times to data loss predicted using this methodology have also been shown to match with simulations, which avoid all the approximations made in the methodology, over a wide range of system parameters.

3.1 Measures of Reliability

In a replication-based system, a data loss is said to have occurred in the system if all replicas of at least one data block have been lost and cannot be restored by the system. Similarly, in an erasure coded system, a data loss is said to have occurred when sufficient number of blocks of at least one codeword have been

lost rendering the codeword(s) undecodeable. The time taken for a system to end up in data loss is a random variable and the purpose of a reliability analysis is to characterize this random variable and study how it is affected by the different system parameters.

3.1.1 Reliability Function

If the time to data loss is denoted by T_{DL}, the most general way to characterize this random variable is by describing its cumulative distribution function, or equivalently, its complementary cumulative distribution function (also known as the *reliability function*) denoted by $R(\cdot)$:

$$R(t) := \Pr\{T_{DL} > t\}, \quad t \geq 0. \tag{3.1}$$

The reliability function is a quantity of practical interest as it allows for designing systems that can provide reliability guarantees over given periods of interest. As an example, for a given time horizon T, for which the user intends to store data reliably, the reliability function $R(T)$ corresponding to a certain system design provides the probability that the system survives at least until T without data loss. Although a very useful reliability measure, closed form expressions of the reliability functions are extremely non-trivial to obtain, except for a handful of simple storage system models.

3.1.2 Mean Time to Data Loss (MTTDL)

Another measure of reliability is the mean time to data loss (MTTDL). It is an aggregate measure of reliability and is related to the reliability function by the following equation:

$$\text{MTTDL} = \int_0^\infty R(t)dt. \tag{3.2}$$

Although not as directly applicable as the reliability function the MTTDL is useful for assessing trade-offs, for comparing schemes, and for estimating the effect of the various parameters on the system reliability [24, 25]. From a theoretical perspective, as an aggregate measure, MTTDL is more amenable to analysis compared to the reliability function. Therefore, in this thesis, we use MTTDL as a measure of system reliability.

3.2 MTTDL Estimation

A few mathematical preliminaries that are needed for MTTDL estimation are discussed below.

3.2.1 Preliminaries

Node Availability

A node i operates for a certain period of time with distribution F_λ before failing. Following the failure of a node, the node and all of its data is restored after a period of time with distribution G_μ. Therefore, the timeline of the node consists of successive periods of operation and repair. For $t \geq 0$, let us define

$$\nu_t^{(i)} := \begin{cases} 1, & \text{if node is operational at time } t, \\ 0, & \text{if node is under rebuild at time } t. \end{cases} \tag{3.3}$$

Then the node availability at time t is given by the probability

$$a_t^{(i)} := \Pr\{\nu_t^{(i)} = 1\}. \tag{3.4}$$

The following result is well known in renewal theory [26, Chap. 2, pp. 109–114]:

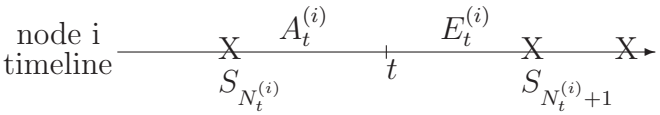
Lemma 3.1. *The steady-state node availability a is given by*

$$a := \lim_{t\to\infty} a_t^{(i)} = \frac{1/\lambda}{1/\lambda + 1/\mu}. \tag{3.5}$$

Note that the above result indicates that the steady-state node availability only depends on the means of the distributions F_λ and G_μ.

Age and Excess

Consider the timeline constructed by concatenating only the periods of operation of the node. In this timeline, let $N_t^{(i)}$ be the number of replacements of the node up to time t, and S_k be the time of the kth failure, for $k = 1, 2, \cdots$.



Define the *age* $A_t^{(i)}$ and the *excess* $E_t^{(i)}$ of the node as

$$A_t^{(i)} := t - S_{N_t^{(i)}}, \tag{3.6}$$

$$E_t^{(i)} := S_{N_t^{(i)}+1} - t. \tag{3.7}$$

As can be seen in the above picture, at a given time t, the age $A_t^{(i)}$ is equal to the time that has passed since the last replacement of the node, and the excess $E_t^{(i)}$ is equal to the time until the next failure of the node. A well known result in renewal theory is the following [26, Chapter 2, pp. 109–114]:

Lemma 3.2.

$$\lim_{t\to\infty} \Pr\{A_t^{(i)} \leq \tau\} = \lim_{t\to\infty} \Pr\{E_t^{(i)} \leq \tau\} = \widetilde{F}_\lambda(\tau), \tag{3.8}$$

where

$$\widetilde{F}_\lambda(\tau) \;:=\; \lambda \int_0^\tau (1 - F_\lambda(x))dx. \tag{3.9}$$

In other words, the cumulative distribution functions of $A_t^{(i)}$ and $E_t^{(i)}$ tend to $\widetilde{F}_\lambda$ as t tends to infinity. In fact, it can be shown that, if the probability density function, f_λ, corresponding to F_λ approaches zero exponentially fast, then the distributions of $A_t^{(i)}$ and $E_t^{(i)}$ also approach $\widetilde{F}_\lambda$ exponentially fast [26].

3.2.2 Probability of Data Loss during Rebuild

At any point of time, the system can be thought to be in one of two modes: fully-operational mode and rebuild mode. During the fully-operational mode, all data in the system has the original amount of redundancy and there is no active rebuild process. During the rebuild mode, some data in the system has less than the original amount of redundancy and there is an active rebuild process that is trying to restore the lost redundancy. A transition from fully-operational mode to rebuild mode occurs when a node fails; we refer to this node failure that causes a transition from the fully-operational mode to the rebuild mode as a *first-node* failure. Following a first-node failure, a complex sequence of rebuilds and subsequent node failures may occur, which eventually lead the system either to irrecoverable data loss, with probability P_{DL}, or back to the original fully-operational mode by restoring all replicas, with probability $1 - P_{DL}$. In other words, the probability of data loss in the rebuild mode, P_{DL}, is defined as follows:

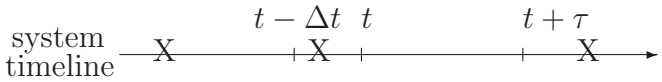
$$P_{DL} := \Pr\left\{ \begin{array}{c} \text{data loss occurs before} \\ \text{returning to the fully} \\ \text{-operational mode} \end{array} \middle| \begin{array}{c} \text{system enters} \\ \text{rebuild mode} \end{array} \right\}. \tag{3.10}$$

As the rebuild times are much shorter than the times to failure, the time taken for these complex sequence of events is negligible compared to the time between successive first-node failures, and therefore can be ignored. In other words, when computing the total time until data loss, the time spent by the system in rebuild mode is negligible compared to the time spent by the system in fully-operational mode and can therefore be ignored. This is due to the fact that nodes are generally reliable, that is, their mean times to failure are much larger compared to their mean times to rebuild (see Section 2.7). If we ignore the rebuild times, the system timeline consists of one first-node failure after another, each of which can end up in data loss with a probability P_{DL}. Therefore, if we can estimate the mean time between two successive first-node failures, that is, the mean fully-operational period of the system, we can easily compute the MTTDL using P_{DL}.

3.2.3 Mean Fully-Operational Period of the System

As mentioned in the previous section, the system timeline essentially consists of a series of first-node failures each of which can result in data loss with probability P_{DL}. Here, we compute the mean time between two successive first-node failures denoted by T.

Let $E(t, \Delta t)$ represent the event that the system was renewed to its original state in the interval $(t - \Delta t, t)$. Also, let $E(t, \Delta t, \tau)$ represent the event that the system was renewed to its fully-operational state in the interval $(t - \Delta t, t)$ and continues to operate without any failures in the interval $(t, t + \tau)$.



We are interested in the survival function of the system, that is, the probability $p_t(\tau)$ defined as:

$$\begin{aligned} p_t(\tau) &:= \lim_{\Delta t \to 0} \Pr\{E(t, \Delta t, \tau) | E(t, \Delta t)\} \\ &= \lim_{\Delta t \to 0} \frac{\Pr\{E(t, \Delta t, \tau)\}}{\Pr\{E(t, \Delta t)\}}. \end{aligned} \tag{3.11}$$

In other words, we are interested in the probability that a system survives without any node failures for a time period τ, given that the system was restored to its fully-operational state at some time t. Using this probability, the mean fully-operational period of the system at time t, T_t, can be computed as

$$T_t = \int_0^\infty p_t(\tau) d\tau. \tag{3.12}$$

In other words, T_t is the mean time period between the first-node failure at time t and the subsequent first-node failure. As the system becomes stationary, $p_t(\tau)$ converges to $p(\tau)$ and T_t converges to T. By computing $p(\tau)$, it can be shown that (see Appendix A)

$$T = \lim_{t \to \infty} T_t = \frac{1}{n\lambda}. \tag{3.13}$$

Note that the derivation of the mean fully-operational period holds for any distribution of the failure times. In addition, it can be shown that the system approaches stationarity exponentially fast if the density function, f_λ, corresponding to F_λ approaches zero exponentially fast [26].

3.2.4 MTTDL Estimate

As each first-node failure could result in data loss with probability P_{DL}, the expected number of first-node failures until data loss occurs is $1/P_{DL}$. By

neglecting the effect of the relatively short transient period of the system, the MTTDL is essentially the product of the expected time between two first-node-failure events, T, and the expected number of first-node-failure events, $1/P_{DL}$, that is,

$$\text{MTTDL} \approx \frac{T}{P_{DL}} = \frac{1}{n\lambda P_{DL}}. \tag{3.14}$$

Although (3.14) lets one estimate the MTTDL by computing the probability of data loss during rebuild, P_{DL}, a theoretical analysis is difficult because the paths, following a first-node failure, to either data loss or back to the fully-operational state are complex as they pass through a combinatorially large number of states. This makes the estimation of P_{DL} a non-trival problem. This issue is addressed in Chapters 4 and 6, where a methodology to estimate P_{DL} by approximating it by the probability of the *shortest path* to data loss is developed for replication-based systems and erasure coded systems, respectively.

Remark 3.1. *In obtaining the above expression, two approximations have been made. Firstly, the time spent by the system in the rebuild mode is ignored. This is motivated by the fact that nodes are generally reliable and that their mean times to failure are much larger than mean times to rebuild. Therefore, this approximation holds for systems with generally reliable nodes for which (2.10) holds. As a second approximation, the effect of the transient period of the system, where the expression (3.13) for the mean fully-operational period of the system (or equivalently, the mean time between two successive first-node failures) may not hold, is ignored. However, this is justified by the fact that the system approaches stationarity exponentially fast if the failure time density function, f_λ, approaches zero exponentially fast [26]. Real-world storage device failures are modeled well by Weibull distributions with shape parameters greater than one [3]. Such distributions have exponentially decaying densities and therefore, the second approximation is reasonable for real-world storage systems. It is also observed using simulations in this thesis that these approximations are reasonable for systems with generally reliable nodes.*

Remark 3.2. *If the failure distribution F_λ is assumed to be exponential, the node failures are memoryless and therefore, the expression (3.13) for the mean fully-operational period, T, holds for all time $t \geq 0$ and not just during the stationary period as t tends to infinity. This implies that the only approximation made in obtaining the expression (3.14) for MTTDL is ignoring the time spent by the system in rebuild mode, which is negligible compared to the time spent in fully-operational mode for systems with generally reliable nodes satisfying (2.10).*

3.2.5 Node vs. System Timelines

Note that there is a significant difference between the events on the node timeline and events on the system timeline. Events on the timeline of node i represent failures of node i (ignoring the rebuild time for that node). After each event on the node timeline, the failed node i is replaced by a new node whose failure time is independent of the previously failed node. Therefore, the events on a node timeline form a renewal process with independent interarrival times.

On the other hand, the events on the system timeline represent first-node failures in the system, that is, the failures that cause the system to go from the fully-operational mode to the rebuild mode (ignoring the time for the subsequent, potentially complex, sequence of node rebuilds and failures that eventually leads the system back to the fully-operational mode with probability $1 - P_{DL}$). After each event on the system timeline, only some of the nodes of the system may have been replaced. Consequently, the time until the next event also depends on the residual lifetimes of the remaining nodes. This implies that the interarrival times in the system timeline are not independent and so the events on this timeline do not form a renewal process. However, as shown by (A.20) in Appendix A, the distributions $p_t(\tau)$ (and hence the means) of the interarrival times become identical as the system becomes stationary. In other words, the interarrival times become identically distributed but are not independent as the system becomes stationary.

Replication-based Systems

4

Replication is a widely-used form of data redundancy that is employed in many of today's data storage systems. Besides improving data reliability, replication-based systems also improve the performance of the system because of the availability of several replicas of the user data.

In this chapter, we analyze the data reliability of replication-based systems in terms of its mean time to data loss (MTTDL) and show how different replica placement schemes and system parameters affect the system MTTDL. To do this, we make use of the relation (3.14) between MTTDL and the probability of data loss during rebuild, P_{DL}, which is a good approximation for real-world systems with generally reliable storage nodes (see Section 3.2). The estimation of P_{DL} is a non-trivial problem as the system can go through a complex sequence of node failures and rebuilds during the rebuild mode. Therefore, we approximate P_{DL} by the probability of the *shortest path* to data loss in rebuild mode and show that this approximation holds good for generally reliable nodes whose mean times to failure are much larger than their mean times to rebuild.

4.1 Estimation of the Probability of Data Loss during Rebuild

This section shows how the complex sequence of failure and rebuild events following a first-node failure, that is, a node failure that causes a transition of the system from fully-operational mode to the rebuild mode, is handled to be able to estimate the probability of data loss before all lost replicas are restored, namely, P_{DL}.

The general idea behind the estimation of P_{DL} is as follows. We model the reliability behavior of the system using *exposure levels* that range from zero to

r. Exposure level zero corresponds to a system where all data have all replicas intact, whereas exposure level r corresponds to a system where some data have lost all their replicas. In other words, the system starts at exposure level zero and eventually ends up in exposure level r, which corresponds to irrecoverable data loss. Rebuild processes cause the system to go to lower exposure levels, whereas node failures may, depending on the replica placement, cause the system to go to higher exposure levels. The probability P_{DL} is then equivalent to the probability that, once the system enters exposure level one, the system ends up in exposure level r before returning to exposure level zero. It is extremely non-trivial to evaluate this probability as there are infinitely many complex paths through which the system can traverse these exposure levels. The problem is not so much complicated by the infinite number of paths as it is by the fact that the probability of the next exposure level transition depends on how the system got to the current exposure level. So we approximate this probability, P_{DL}, of all possible paths to data loss by the probability of the direct path to data loss, namely, the path from exposure level one to two, two to three, and so on until r. We show that such an approximation holds for systems with generally reliable nodes (that is, nodes whose mean times to failure are much larger than their mean times to rebuild) in the sense that the relative error in the approximation tends to zero as the ratio of the mean time to rebuild to the mean time to failure tends to zero. However, even the computation of the probability of this direct path is quite involved. This is because, the probability of transition from one exposure level to the next not only depends on the current exposure level, but also on how the system arrived there. So we consider all possible *sample* direct paths from exposure level zero to r, compute their probabilities, and sum them up. This gives the probability of direct path to data loss which is then used as a good approximation for P_{DL}.

4.1.1 Exposure Levels

Consider a replication-based storage system with replication factor r. To keep the problem analytically tractable, we model the system as evolving from one *exposure level* to another as nodes fail and rebuilds complete. At time $t \geq 0$, let $D_l(t)$ be the amount of user data that have lost l replicas, with $0 \leq l \leq r$. The system is said to be in exposure level e at time t, $0 \leq e \leq r$, if

$$e = \max_{D_l(t)>0} l. \tag{4.1}$$

In other words, the system is in exposure level e if there exists some data with $r - e$ copies and no data with fewer than $r - e$ copies in the system, that is, $D_e(t) > 0$, and $D_l(t) = 0$ for all $l > e$. At $t = 0$, $D_l(0) = 0$ for all $l > 0$ and $D_0(0)$ is the total amount of user data stored in the system, which according to the parameters in Table 2.1, is equal to nc/r. Node failures and rebuild processes cause the values of $D_1(t), \cdots, D_r(t)$ and the exposure level

of the system to change over time. Data loss occurs when some data have lost all r replicas, that is, when $D_r(t) > 0$ for some time t. The smallest t for which $D_r(t) > 0$ is the first time the system ends up in data loss and is simply referred to as the *time to data loss*, T_{DL}:

$$T_{DL} = \min_{D_r(t)>0} t. \tag{4.2}$$

The time to data loss is a random variable and our goal is to estimate its mean, MTTDL.

4.1.2 Direct Path Approximation

A path to data loss following a first-node-failure event is a sequence of exposure level transitions that begins in exposure level 1 and ends in exposure level r (data loss) without going back to exposure level 0, that is, for some $m \geq r$, a sequence of $m-1$ exposure level transitions $e_1 \to e_2 \to \cdots \to e_m$ such that $e_1 = 1$, $e_m = r$, $e_2, \cdots, e_{m-1} \in \{1, \cdots, r-1\}$, and $|e_i - e_{i-1}| = 1$, $\forall\, i = 2, \cdots, m$. Note that this collection of paths excludes visits to exposure level 0, and therefore, only consists of all paths to data loss before all lost replicas are restored. To estimate P_{DL}, we need to estimate the probability of the union of *all* such paths to data loss following a first-node failure. As the set of events that can occur between exposure level 1 and exposure level r is complex, estimating P_{DL} is a non-trivial problem.

To circumvent this problem, we approximate P_{DL} by the probability of the direct path to data loss, that is, the probability of the path $1 \to 2 \to \cdots \to r$. It is shown in Appendix B that the probability of the direct path approximates well the probability of *all* paths, namely, P_{DL}, for a system with generally reliable nodes for which (2.10) holds. Thus, if we denote the probability of direct path to data loss by $P_{DL,\text{direct}}$, then

$$P_{DL} \approx P_{DL,\text{direct}}. \tag{4.3}$$

The proof of the above approximation relies only on the fact that the probabilities of transitions to higher exposure levels are extremely small, which is the case for systems with generally reliable nodes. The proof also does not make any assumptions on the failure and rebuild time distributions. Additionally, it is seen from the analysis in Appendix B that the approximation is quite good in the sense that the relative error of approximation tends to zero as the ratio, λ/μ, of mean time to rebuild to the mean time to failure tends to zero. In real-world storage systems, this ratio is observed to be generally small and therefore this is a reasonable approximation. The approximation is also seen to be quite good over a wide range of parameters using simulations which do not make this approximation.

4.1.3 Probability of the Direct Path to Data Loss

Consider the direct path to data loss, that is, the path $1 \to 2 \to \cdots \to r$ through the exposure levels. At each exposure level, the *intelligent* rebuild process attempts to rebuild the most-exposed data, that is, the data with the least number of replicas left (see Section 2.6). Let the rebuild times of the most-exposed data at each exposure level in this path be denoted by R_e, $e = 1, \cdots, r-1$. If no additional node failures occur during a rebuild in exposure level e that causes the system to go to exposure level $e+1$, then after a time period of R_e, the system will return to exposure level $e-1$. Therefore, determining the rebuild times at each exposure level is a key step in estimating the probability of the direct path to data loss.

The rebuild times at each exposure level are random variables that depend on the amount of most-exposed data to be rebuilt at that exposure level and the data placement scheme. The amount of most-exposed data to be rebuilt at a given exposure level, e, depends on when a node failure occurred during the rebuild in the previous exposure level, $e-1$, that caused the system to come to exposure level e. Let us illustrate this with a simple example: consider a system with six storage nodes divided into two clusters of three nodes each. Each node in a cluster stores a copy of the same data and there is no data with replicas on nodes in two different clusters. In our model, this is a replication-based system with replication factor three and clustered replica placement. The system is at exposure level zero until one of the nodes fails, at which point, the system enters exposure level one. The amount of data to be rebuilt is c and it takes an average of $1/\mu$ amount of time to make a copy of this data from one of the surviving nodes of the cluster to a new replacement node. In other words, the rebuild time R_1 has mean $1/\mu$. As the nodes are generally reliable, typically no additional node failures occur during this rebuild period and the system returns to exposure level zero. However, with a small probability, an additional node failure occurs. This node could either belong to the cluster being rebuilt, in which case the system enters exposure level two as some of the data lose a second replica, or the other cluster, in which case the system stays in the same exposure level as no data have lost more than one copy. To compute the probability of the direct path to data loss, we are interested in the probability of a node failure that causes the system to enter exposure level two. Suppose that this second node failure occurs when a fraction α of the data corresponding to the node that failed first is not yet rebuilt. Since the two failed nodes shared replicas of all their data, the amount of data that loses a second replica when the second failure occurs is αc. This data is now the most-exposed and it would now take an average of α/μ amount of time to rebuild this most-exposed data. In other words, the rebuild time R_2 has a conditional mean α/μ. We will now explicitly describe how one can estimate the rebuild times at each exposure level.

Let t_e, $e = 2, \cdots, r$, be the times of transitions from exposure level $e-1$ to e following a first-node failure, that is, a node failure that causes the system to

enter rebuild mode from the fully-operational mode. Let $\tilde{n}_e$ be the number of nodes in exposure level e whose failure before the rebuild of most-exposed data causes an exposure level transition to level $e+1$. For example, in clustered placement scheme, the failure of any of the surviving nodes of the cluster being rebuilt causes some data to lose an additional replica thereby leading the system to the next exposure level. Therefore, for clustered placement, $\tilde{n}_e = r - e$, as there are exactly $r - e$ surviving nodes in a cluster being rebuilt when the system is in exposure level e. Now, let

$$F_e := \min_{i \in \{1, \cdots, \tilde{n}_{e-1}\}} E_{t_{e-1}}^{(i)}, \quad e = 2, \cdots, r, \tag{4.4}$$

denote the time taken for a node failure to occur that can cause the system to enter exposure level e. Note that $E_{t_{e-1}}^{(i)}$, as defined in Section 3.2.1, denotes the time period from t_{e-1} until the next failure of node i. Therefore, F_e denotes the time until the first failure among the $\tilde{n}_{e-1}$ nodes that causes the system to enter exposure level e.

At exposure level e, let α_e be the fraction of the rebuild time, R_e, left when a node failure occurs causing an exposure level transition, that is, let

$$\alpha_e := \frac{R_e - F_{e+1}}{R_e}, \quad e = 1, \cdots, r-2. \tag{4.5}$$

In Appendix C, it is shown that α_e is uniformly distributed between zero and one, that is,

$$\alpha_e \sim U(0,1), \quad e = 1, \cdots, r-2. \tag{4.6}$$

Now, consider a direct path to data loss with $R_e = \tau_e$, $e = 1, \cdots, r-1$, and $\alpha_e = a_e$, $e = 1, \cdots, r-2$.[1] Denote the vector $(\tau_1, \cdots, \tau_{r-1})$ by $\vec{\tau}$ and $(a_1, \cdots, a_{r-2})$ by $\vec{a}$ for notational convenience. Then, the probability of this direct path, denoted by $P_{DL,\text{direct}}(\vec{\tau}, \vec{a})$, is given by

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\tau}, \vec{a}) \;=\; & \Pr\{R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1, R_2 = \tau_2, F_3 < R_2, \\ & \cdots, \alpha_{r-2} = a_{r-2}, R_{r-1} = \tau_{r-1}, F_r < R_{r-1}\}. \end{aligned} \tag{4.7}$$

In the above expression, the events $F_e < R_{e-1}$ represent the exposure level transitions from $e-1$ to e. Thus, the above expression gives the probability that the system will take this particular direct path to data loss with $R_e = \tau_e$ and $\alpha_e = a_e$. Expanding (4.7) by conditioning, we get

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\tau}, \vec{a}) \;=\; & \Pr\{R_1 = \tau_1\} \times \Pr\{F_2 < R_1 | R_1 = \tau_1\} \\ & \times \Pr\{\alpha_1 = a_1 | R_1 = \tau_1, F_2 < R_1\} \\ & \times \Pr\{R_2 = \tau_2 | R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1\} \\ & \times \Pr\{F_3 < R_2 | R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1, R_2 = \tau_2\} \\ & \cdots \times \Pr\{F_r < R_{r-1} | R_1 = \tau_1, \cdots, R_{r-1} = \tau_{r-1}\}. \end{aligned} \tag{4.8}$$

[1] More strictly, we consider a direct path to data loss with $\tau_e < R_e \leq \tau_e + \delta\tau_e$, $e = 1, \cdots, r-1$, and $a_e < \alpha_e \leq \delta a_e$, $e = 1, \cdots, r-2$, where $\delta\tau_e$ and δa_e are positive infinitesimal quantities, but we leave this out for notational convenience.

The first term in the above expansion is the probability $\Pr\{R_1 = \tau_1\}$. Denote the mean of R_1 by $1/\mu_1$, that is,

$$\frac{1}{\mu_1} := E[R_1]. \tag{4.9}$$

The actual value of the mean will depend on the underlying data placement and will be discussed further in the later sections. Based on the rebuild model described in Section 2.6, it follows that R_1 is distributed according to some distribution G_{μ_1} that satisfies (2.11):

$$R_1 \sim G_{\mu_1}. \tag{4.10}$$

Therefore, the first term reduces to

$$\Pr\{R_1 = \tau_1\} = g_{\mu_1}(\tau_1)\delta\tau_1, \tag{4.11}$$

where $\delta\tau_1$ denotes an infinitesimal increment in τ_1. The remaining terms in the expression for $P_{DL,\text{direct}}(\vec{\tau}, \vec{a})$ in (4.8) fall into three types:

$$\text{Type A: } \Pr\{F_e < R_{e-1} | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}\}, \tag{4.12}$$

$$\text{Type B: } \Pr\{\alpha_e = a_e | R_1 = \tau_1, \cdots, R_e = \tau_e, F_{e+1} < R_e\}, \tag{4.13}$$

$$\text{Type C: } \Pr\{R_e = \tau_e | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}, F_e < R_{e-1}, \alpha_{e-1} = a_{e-1}\}. \tag{4.14}$$

Expressions of type A denote the conditional probability of transition from exposure level $e-1$ to e, given that the system has traversed through exposure levels 1 to $e-1$ with corresponding rebuild times being equal to $\tau_1, \cdots, \tau_{e-1}$, and corresponding fractions of rebuild times still remaining when exposure levels occurred being equal to $a_1, \cdots, a_{e-1}$. Expressions of type B denote the conditional probability that the fraction α_e of the rebuild time, R_e, still left when an exposure level transition from e to $e+1$ occurred is equal to a_e, given that the system has traversed through exposure levels 1 to $e-1$ with corresponding rebuild times being equal to $\tau_1, \cdots, \tau_{e-1}$, and corresponding fractions of rebuild times still remaining when exposure levels occurred being equal to $a_1, \cdots, a_{e-1}$. Expressions of type C denote the conditional probability that the rebuild time in exposure level e is equal to τ_e, given that the system has traversed through exposure levels 1 to $e-1$ with corresponding rebuild times being equal to $\tau_1, \cdots, \tau_{e-1}$, and corresponding fractions of rebuild times still remaining when exposure levels occurred being equal to $a_1, \cdots, a_{e-1}$. Each of these types of expressions can be further simplified as follows.

Expressions of Type A

Terms of type A are the conditional probabilities of transitions to higher exposure levels. Given that the rebuild time $R_{e-1} = \tau_{e-1}$, the next exposure

transition event, $F_e < R_{e-1}$, is independent of the other conditioning terms in (4.12). Therefore, terms of the type (4.12) can be rewritten as

$$\begin{aligned}\Pr\{F_e < R_{e-1} | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}\} &= \Pr\{F_e < R_{e-1} | R_{e-1} = \tau_{e-1}\} \\ &= \Pr\{F_e < \tau_{e-1}\}. \end{aligned} \tag{4.15}$$

Here (4.15) follows from the fact that the time to next node failure, F_e, and the time to rebuild the most-exposed data, R_{e-1}, are independent. Substituting for F_e from (4.4), we have

$$\Pr\{F_e < \tau_{e-1}\} = \Pr\left\{\min_{i\in\{1,\cdots,\tilde{n}_{e-1}\}} E^{(i)}_{t_{e-1}} < \tau_{e-1}\right\} \tag{4.16}$$

$$= 1 - \Pr\left\{\min_{i\in\{1,\cdots,\tilde{n}_{e-1}\}} E^{(i)}_{t_{e-1}} \geq \tau_{e-1}\right\} \tag{4.17}$$

$$= 1 - \Pr\left\{E^{(i)}_{t_{e-1}} \geq \tau_{e-1}\, \forall\, i \in \{1, \cdots, \tilde{n}_{e-1}\}\right\} \tag{4.18}$$

$$= 1 - \left(\Pr\left\{E^{(1)}_{t_{e-1}} \geq \tau_{e-1}\right\}\right)^{\tilde{n}_{e-1}} \tag{4.19}$$

$$= 1 - \left(1 - \Pr\left\{E^{(1)}_{t_{e-1}} < \tau_{e-1}\right\}\right)^{\tilde{n}_{e-1}}. \tag{4.20}$$

Here, (4.19) follows from the fact that $E^{(i)}_{t_{e-1}}$ are independent and identically distributed random variables. From the results in Appendix D, it follows that

$$\Pr\left\{E^{(1)}_{t_{e-1}} < \tau_{e-1}\right\} = \lambda\tau_{e-1} + o\left(\lambda\tau_{e-1}\right). \tag{4.21}$$

Substituting (4.21) in (4.20), we get

$$\Pr\{F_e < \tau_{e-1}\} = 1 - \left(1 - \lambda\tau_{e-1} + o\left(\lambda\tau_{e-1}\right)\right)^{\tilde{n}_{e-1}} \tag{4.22}$$

$$= \tilde{n}_{e-1}\lambda\tau_{e-1} + o\left(\lambda\tau_{e-1}\right) \tag{4.23}$$

$$\approx \tilde{n}_{e-1}\lambda\tau_{e-1}, \tag{4.24}$$

where the approximation (4.24) holds good for systems with generally reliable nodes satisfying (2.10) and (2.11). From (4.15) and (4.24), we observe that the type A expressions of the form (4.12) can be reduced to

$$\Pr\{F_e < R_{e-1} | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}\} \approx \tilde{n}_{e-1}\lambda\tau_{e-1}, \tag{4.25}$$

for $e = 2, \cdots, r$, where the approximation holds good for systems with generally reliable node satisfying (2.10) and (2.11).

Remark 4.1. *Note that the expression (4.25) represents the conditional probability of transition from exposure level $e-1$ to e. The node rebuild times τ_{e-1} (which are typically of the order of a few hours) are much smaller compared to the mean time to node failure $1/\lambda$ (which are typically of the order of a few years) for generally reliable nodes. Therefore, the conditional probabilities of transition to higher exposure levels are extremely small for systems with generally reliable nodes.*

Expressions of Type B

Expressions of type B relate to the distribution of the fractions of most-exposed data that have not yet been rebuilt when a transition to a higher exposure level occurred. Given that $R_e = \tau_e$ and $F_{e+1} < R_e$, α_e is independent of the other conditioning terms in (4.13). Therefore, terms of the type (4.13) can be rewritten as

$$\begin{aligned}\Pr\{\alpha_e = a_e | R_1 = \tau_1, \cdots, R_e = \tau_e, F_{e+1} < R_e\} \\ = \Pr\{\alpha_e = a_e | R_e = \tau_e, F_{e+1} < R_e\}. \quad (4.26)\end{aligned}$$

Substituting for α_e from (4.5) into (4.26), we get

$$\begin{aligned}&\Pr\{\alpha_e = a_e | R_e = \tau_e, F_{e+1} < R_e\} \\ &= \Pr\left\{ \frac{R_e - F_{e+1}}{R_e} = a_e \,\middle|\, R_e = \tau_e, F_{e+1} < R_e \right\} && (4.27)\\ &= \frac{\Pr\left\{ \frac{R_e - F_{e+1}}{R_e} = a_e, R_e = \tau_e, F_{e+1} < R_e \right\}}{\Pr\{R_e = \tau_e, F_{e+1} < R_e\}} && (4.28)\\ &= \frac{\Pr\{F_{e+1} = \tau_e(1-a_e), R_e = \tau_e, F_{e+1} < \tau_e\}}{\Pr\{R_e = \tau_e, F_{e+1} < \tau_e\}} && (4.29)\\ &= \frac{\Pr\{F_{e+1} = \tau_e(1-a_e), F_{e+1} < \tau_e\} \Pr\{R_e = \tau_e\}}{\Pr\{F_{e+1} < \tau_e\} \Pr\{R_e = \tau_e\}} && (4.30)\\ &= \frac{\Pr\{F_{e+1} = \tau_e(1-a_e)\}}{\Pr\{F_{e+1} < \tau_e\}}. && (4.31)\end{aligned}$$

Here, (4.30) follows from the fact that the time to next node failure, F_e, and the time to rebuild the most-exposed data, R_{e-1}, are independent. From (4.24), we have

$$\Pr\{F_{e+1} < \tau_e\} = \tilde{n}(e)\lambda\tau_{e-1}, \quad (4.32)$$

and

$$\begin{aligned}\Pr\{F_{e+1} = \tau_e(1-a_e)\} &= \Pr\{\tau_e(1-(a_e+\delta a_e) < F_{e+1} \le \tau_e(1-a_e))\} \\ &= \Pr\{F_{e+1} \le \tau_e(1-a_e)\} \\ &\quad - \Pr\{F_{e+1} \le \tau_e(1-(a_e+\delta a_e))\} && (4.33)\\ &\approx \tilde{n}(e)\lambda\tau_{e-1}(1-a_e) \\ &\quad - \tilde{n}(e)\lambda\tau_{e-1}(1-(a_e+\delta a_e)) && (4.34)\\ &= \tilde{n}(e)\lambda\tau_{e-1}\delta a_e, && (4.35)\end{aligned}$$

where δa_e denotes an infinitesimal increment of a_e. From (4.26), (4.31), (4.32), and (4.35), we observe that type B terms of the form (4.13) can be reduced to

$$\Pr\{\alpha_e = a_e | R_1 = \tau_1, \cdots, R_e = \tau_e, F_{e+1} < R_e\} \approx \delta a_e, \quad (4.36)$$

for $e = 1, \cdots, r-2$, where the approximation holds good for systems with generally reliable node satisfying (2.10) and (2.11).

Expressions of Type C

Type C expressions of the form (4.14) give the conditional probabilities of the rebuild times in each exposure level. Given that $R_{e-1} = \tau_{e-1}$ and $\alpha_{e-1} = a_{e-1}$, the rebuild time in exposure level e, R_e, is independent of remaining conditioning terms in (4.14). Therefore, (4.14) can be rewritten as

$$\begin{aligned}\Pr\{R_e = \tau_e | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}, F_e < R_{e-1}, \alpha_{e-1} = a_{e-1}\} \\ = \Pr\{R_e = \tau_e | R_{e-1} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}\}. \quad (4.37)\end{aligned}$$

Denote the conditional means of R_e by $1/\mu_e$, that is,

$$\frac{1}{\mu_e} := E[R_e | R_{e-1}, \alpha_{e-1}], \quad e = 2, \cdots, r-1. \quad (4.38)$$

The actual values of $1/\mu_e$ depends on the data placement and this will be further discussed in later sections of this chapter. Now, the distribution of R_e given R_{e-1} and α_{e-1} could be modeled in several ways. We propose two models, namely,

$$\textbf{Model A:} \qquad R_e | R_{e-1}, \alpha_{e-1} \sim G_{\mu_e}, \quad (4.39)$$

$$\textbf{Model B:} \qquad R_e | R_{e-1}, \alpha_{e-1} = \frac{1}{\mu_e}. \quad (4.40)$$

In model A, we assume that, following a node failure, the system has to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where the node from which data was being rebuilt failed and hence the system has to rebuild from another node in the cluster. In model B, we assume that, following a node failure, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the case, for instance, in a clustered placement scheme where the newly failed node is different from the node from which data is being rebuilt. This is also the case in a declustered placement scheme, where the rebuild was being done from all nodes, and therefore, in a large system, the failure of one node does not significantly affect the rebuild process.

Therefore, type C expressions of the form (4.14) reduce, under models A and B, to

$$\begin{aligned}\Pr\{R_e = \tau_e | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}, F_e < R_{e-1}, \alpha_{e-1} = a_{e-1}\} \\ = \begin{cases} g_{\mu_e}(\tau_e)\delta\tau_e & \text{under model A,} \\ \delta(\tau_e - 1/\mu_e)\delta\tau_e & \text{under model B,} \end{cases} \quad (4.41)\end{aligned}$$

for $e = 2, \cdots, r-1$. Here, $g_{\mu_e}(\cdot)$ denotes the probability density function corresponding to the distribution G_{μ_e}, $\delta(\cdot - 1/\mu_e)$ denotes the Dirac delta function with a spike at $1/\mu_e$, and $\delta\tau_e$ denotes an infinitesimal increment of τ_e.

Probability of a Sample Direct Path

Substituting (4.11), (4.25), (4.36), and (4.41) in (4.8), the probability of a sample direct path with $R_e = \tau_e$, $e = 1, \cdots, r-1$, and $\alpha_e = a_e$,$e = 1, \cdots, r-2$, reduces to

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\tau}, \vec{a}) &\approx g_{\mu_1}(\tau_1)\delta\tau_1 \times \left(\prod_{e=2}^{r} \tilde{n}_{e-1}\lambda\tau_{e-1}\right) \times \left(\prod_{e'=1}^{r-1} \delta a_{e'}\right) \\ &\quad \times \begin{cases} \prod_{e''=2}^{r-2} g_{\mu_{e''}}(\tau_{e''})\delta\tau_{e''} & \text{(model A)} \\ \prod_{e''=2}^{r-2} \delta(\tau_{e''} - 1/\mu_{e''})\delta\tau_{e''} & \text{(model B)} \end{cases} \\ &= \lambda^{r-1} \times \tilde{n}_1 \cdots \tilde{n}_{r-1} \times \tau_1 \cdots \tau_{r-1} \times g_{\mu_1}(\tau_1) \\ &\quad \times \delta a_1 \cdots \delta a_{r-2} \times \delta\tau_1 \cdots \delta\tau_{r-1} \\ &\quad \times \begin{cases} g_{\mu_2}(\tau_2) \cdots g_{\mu_{r-1}}(\tau_{r-1}) & \text{(model A)} \\ \delta\left(\tau_2 - \frac{1}{\mu_2}\right) \cdots \delta\left(\tau_{r-1} - \frac{1}{\mu_{r-1}}\right) & \text{(model B)} \end{cases} \end{aligned} \tag{4.42}$$

Probability of Data Loss during Rebuild

As mentioned in Section 4.1.2, the probability of direct path to data loss, denoted by $P_{DL,\text{direct}}$, is a good approximation for the probability of data loss during rebuild, P_{DL}:

$$P_{DL} \approx P_{DL,\text{direct}}. \tag{4.43}$$

Also, the probability of the direct path to data loss, $P_{DL,\text{direct}}$, is the summation of the probabilities, $P_{DL,\text{direct}}(\vec{\tau}, \vec{a})$, of all possible sample direct paths. As the infinitesimal increments in (4.42) tend to zero, the summation becomes an integral. Therefore, the probability of all possible direct paths to data loss, $P_{DL,\text{direct}}$, and hence, P_{Dl}, becomes

$$\begin{aligned} P_{DL} &\approx \lambda^{r-1} \times \tilde{n}_1 \cdots \tilde{n}_{r-1} \\ &\quad \times \int_{\tau_1} \cdots \int_{\tau_{r-1}} \int_{a_1} \cdots \int_{a_{r-2}} \tau_1 \cdots \tau_{r-1} g_{\mu_1}(\tau_1) \cdots g_{\mu_{r-1}}(\tau_{r-1}) d\vec{a} d\vec{\tau} \end{aligned}$$

(model A) (4.44)

$$\begin{aligned} P_{DL} &\approx \lambda^{r-1} \times \tilde{n}_1 \cdots \tilde{n}_{r-1} \\ &\quad \times \int_{\tau_1} \cdots \int_{\tau_{r-1}} \int_{a_1} \cdots \int_{a_{r-2}} \Bigg(\tau_1 \cdots \tau_{r-1} g_{\mu_1}(\tau_1) \\ &\qquad \times \delta\left(\tau_2 - \frac{1}{\mu_2}\right) \cdots \delta\left(\tau_{r-1} - \frac{1}{\mu_{r-1}}\right) d\vec{a} d\vec{\tau} \Bigg) \end{aligned}$$

(model B). (4.45)

Here, the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$.

Remark 4.2. *The approximations in the expressions (4.44) and (4.45) for P_{DL} hold good for systems with generally reliable nodes that satisfy (2.10) and (2.11). The approximation is good in the sense that the relative error tends to zero as the ratio λ/μ of the mean time to node failure to the mean time to node rebuild tends to zero. The validity of the approximation has also been established for a wide rage of parameters using simulation.*

Remark 4.3. *The derivation of the expressions (4.44) and (4.45) for P_{DL} is quite general in the sense that it is applicable to all symmetric data placement schemes and to both replication-based systems and erasure-coded systems. It is also applicable to all node failure and rebuild distributions that satisfy (2.10) and (2.11). As can be observed from the expressions (4.44) and (4.45), the only unknowns in evaluating P_{DL} are the means μ_e, $e = 1, \cdots, r-1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e$, $e = 1, \cdots, r-1$. These quantities depend on the particular type of data placement or redundancy scheme used.*

Remark 4.4. *Erasure coded systems with an (l, m)-MDS code can survive the loss of upto $m - l$ blocks of a codeword. The loss of $m - l + 1$ blocks of any codeword results in irrecoverable data loss. Therefore, an erasure coded system can be modeled by exposure levels just like replication-based systems by replcing r by $m - l + 1$. The expressions (4.44) and (4.45) for P_{DL} continue to hold for erasure coded systems when r is replaced by $m - l + 1$.*

Remark 4.5. *It is clear from (4.44) and (4.45) that P_{DL} is invariant to the class of failure distributions satisfying (2.10) and (2.11) and only depends on the mean time to failure, $1/\lambda$. Furthermore, by the relation (3.14), the MTTDL is also invariant to this class of failure distributions. As this class of distributions includes real-world empirical distributions, such as the Weibull distribution, as well as the theoretically amenable exponential distribution, the benefit is two-fold. The fact that real-world failure distributions belongs to this class implies that these results are directly relevant to practical storage systems. On the other hand, the presence of the exponential distribution in this class means that MTTDL results obtained in the literature assuming unrealistic exponential distributions may be applicable to real-world storage systems as well.*

4.2 Effect of Replica Placement on Reliability

In this section, we consider different replica placement schemes as discussed in Section 2.4. We would like to estimate their reliability in terms of their MTTDL using the relations (3.14), (4.44), and (4.45), and understand how replica placement affects data reliability. To use the expressions (4.44) and (4.45) for P_{DL}, we need to compute the conditional means of rebuild times in each exposure level, μ_e, $e = 1, \cdots, r-1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e$, $e = 1, \cdots, r-1$.

The values of these quantities depend on the underlying replica placement and the nature of the rebuild process used.

Notation: Here, we introduce a few new notations and repeat the notations used in the previous section for the sake of clarity. Suppose that a first-node failure occurs at time t_1 causing the system to go from the fully-operational mode to exposure level 1 in the rebuild mode. Let t_e, $e = 2, \cdots, r$, denote the times of transitions frome exposure level $e-1$ to e. Let the rebuild times of the most-exposed data at each exposure level in this path be R_e, $e = 1, \cdots, r-1$, with conditional means $1/\mu_e$, $e = 1, \cdots, r-1$. Let $\tilde{n}_e$, $e = 1, \cdots, r-1$, be the number of nodes whose failure during rebuild can cause a transition to the next higher exposure level. Let $D_e(t)$, $e = 0, \cdots, r$, denote the amount of user data that have lost e replicas at time t. Around the times of exposure level transitions, t_e, let $D_e(t_e^-)$ and $D_e(t_e)$ denote the amounts of user data that have lost e replicas just before and just after time t_e, respectively. In addition, let S_e, $e = 1, \cdots, r-1$, denote the average speed (or rate) of rebuild in exposure level e. Also, let the kth raw moment of the rebuild distribution G_μ with mean $1/\mu$ be denoted by $M_k(G_\mu)$, that is,

$$M_k(G_\mu) := \int_t t^k dG_\mu(t), \qquad \text{for } k = 1, 2, \cdots. \tag{4.46}$$

By definition,

$$M_1(G_\mu) = \frac{1}{\mu}. \tag{4.47}$$

Note that, by Jensen's inequality,

$$M_1^k(G_\mu) \leq M_k(G_\mu), \qquad \text{for } k = 1, 2, \cdots, \tag{4.48}$$

that is, the kth power of the mean of G_μ is lesser than or equal to the kth raw moment of G_μ. Lastly, the superscripts 'clus.' and 'declus.' will be used to refer to quantities specific to clustered and declustered placement schemes, respectively.

4.2.1 Clustered Replica Placement

The goal of this section is to estimate the reliability of a clustered replica placement scheme in terms of its MTTDL. To achieve this goal, we first compute the conditional means of rebuild times in each exposure level, $\mu_e^{\text{clus.}}$, $e = 1, \cdots, r-1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e^{\text{clus.}}$, $e = 1, \cdots, r-1$. Using these quantities and expressions (4.44) and (4.45), we can compute the probability of data loss during rebuild, $P_{DL}^{\text{clus.}}$. The mean time to data loss, $\text{MTTDL}^{\text{clus.}}$, can then be obtained by using the relation (3.14).

Clustered Replica Placement: Exposure Level 1

Following a first-node failure at t_1, the system enters exposure level 1 and the rebuild process begins. The amount of data to be rebuilt at this exposure level is equal to the capacity of the failed node, c, that is,

$$D_1^{\text{clus.}}(t_1) = c. \tag{4.49}$$

As described in Section 2.6, the rebuild process in a clustered placement scheme involves copying the data corresponding to the failed node from one of the other surviving nodes of the cluster to a new spare node. This is done at an average bandwidth of $c\mu$, and therefore, the average rate (or speed) of rebuild in exposure level 1 is

$$S_1^{\text{clus.}} = c\mu. \tag{4.50}$$

The average time required for this rebuild, $1/\mu_1^{\text{clus.}}$, is obtained by dividing the amount of data to be rebuilt, given by (4.49), by the average speed of rebuild, given by (4.50). Thus,

$$\frac{1}{\mu_1^{\text{clus.}}} = E[R_1^{\text{clus.}}] = \frac{D_1^{\text{clus.}}(t_1)}{S_1^{\text{clus.}}} = \frac{1}{\mu}. \tag{4.51}$$

According to our model, the rebuild time, $R_1^{\text{clus.}}$, is distributed according to some distribution $G_{\mu_1^{\text{clus.}}}$ with mean $1/\mu_1^{\text{clus.}}$ that satisfies (2.11), that is,

$$R_1^{\text{clus.}} \sim G_{\mu_1^{\text{clus.}}} = G_\mu. \tag{4.52}$$

There are $r-1$ remaining nodes in the cluster of the failed node. The failure of any of these nodes during the rebuild period $R_1^{\text{clus.}}$ will cause the system to enter exposure level 2, whereas the failure of nodes belonging to any other cluster does not cause the system to enter exposure level 2. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_1^{\text{clus.}} = r - 1. \tag{4.53}$$

When one of the $\tilde{n}_1^{\text{clus.}}$ nodes fail before rebuild, the system enters exposure level 2.

Clustered Replica Placement: Exposure Level 2

The system enters exposure level 2 from exposure level 1 because one of the $\tilde{n}_1^{\text{clus.}}$ nodes fails during the rebuild period $R_1^{\text{clus.}}$. Consider an instance of the rebuild period,

$$R_1^{\text{clus.}} = \tau_1, \tag{4.54}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from 1 to 2 occurred,

$$\alpha_1 = a_1. \tag{4.55}$$

The remaining time to complete rebuild at exposure level 1 when the system entered exposure level 2 is the product of $R_1^{\text{clus.}}$ and α_1, namely, $a_1\tau_1$. As the average speed of rebuild in exposure level 1 is $S_1^{\text{clus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_1^{\text{clus.}}(t_2^-)$, is given by

$$D_1^{\text{clus.}}(t_2^-) = \alpha_1 R_1^{\text{clus.}} S_1^{\text{clus.}} = a_1\tau_1 c\mu, \tag{4.56}$$

which is essentially the product of (4.50), (4.54), and (4.55). At the time of transition from exposure level 1 to 2, t_2, all this $D_1^{\text{clus.}}(t_2^-)$ amount of data loses a second copy and is thus the most-exposed data in exposure level 2. This is due to the nature of the clustered replica placement scheme in which all nodes of a cluster share copies of the same data. Therefore, the amount of most-exposed data in exposure level 2, $D_2^{\text{clus.}}(t_2)$, is given by

$$D_2^{\text{clus.}}(t_2) = D_1^{\text{clus.}}(t_2^-) = a_1\tau_1 c\mu. \tag{4.57}$$

The average speed of rebuild remains unaffected as the system just copies from one of the surviving nodes of the cluster to a new replacement node at a rate of $c\mu$. Therefore,

$$S_2^{\text{clus.}} = c\mu. \tag{4.58}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_2^{\text{clus.}}$, of the rebuild time in the second exposure level, $R_2^{\text{clus.}}$, is obtained by dividing (4.57) by (4.58), that is,

$$\frac{1}{\mu_2^{\text{clus.}}} = E[R_2^{\text{clus.}} | R_1^{\text{clus.}} = \tau_1, \alpha_1 = a_1] = \frac{D_2^{\text{clus.}}(t_2)}{S_2^{\text{clus.}}} = a_1\tau_1. \tag{4.59}$$

There are now $r-2$ remaining nodes in the cluster of the failed node. The failure of any of these nodes during the rebuild time $R_2^{\text{clus.}}$ will cause the system to enter exposure level 3. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_2^{\text{clus.}} = r - 2. \tag{4.60}$$

Clustered Replica Placement: Exposure Level e

The computation of the conditional mean $1/\mu_e^{\text{clus.}}$ and the number of nodes $\tilde{n}_e^{\text{clus.}}$ for a general exposure level $e = 2, \cdots, r-1$, is similar to the computation of these quantities for exposure level 2 as described above. Firstly, we note that

the average speed of rebuild is unchanged in each exposure level for clustered placement, that is,

$$S_e^{\text{clus.}} = c\mu, \quad e = 1, \cdots, r-1. \tag{4.61}$$

This is due to the fact that the rebuild process always involves copying of data from one of the surviving nodes of the cluster under rebuild to a new replacement node.

Now, the system enters exposure level e from exposure level $e-1$ because one of the $\tilde{n}_{e-1}^{\text{clus.}}$ nodes fails during the rebuild period $R_{e-1}^{\text{clus.}}$. Consider an instance of the rebuild period,

$$R_{e-1}^{\text{clus.}} = \tau_{e-1}, \tag{4.62}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from $e-1$ to e occurred,

$$\alpha_{e-1} = a_{e-1}. \tag{4.63}$$

The remaining time to complete rebuild at exposure level $e-1$ when the system entered exposure level e is the product of $R_{e-1}^{\text{clus.}}$ and α_{e-1}, namely, $a_{e-1}\tau_{e-1}$. As the average speed of rebuild in exposure level $e-1$ is $S_{e-1}^{\text{clus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_{e-1}^{\text{clus.}}(t_e^-)$, is given by

$$D_{e-1}^{\text{clus.}}(t_e^-) = \alpha_{e-1} R_{e-1}^{\text{clus.}} S_{e-1}^{\text{clus.}} = a_{e-1}\tau_{e-1}c\mu, \tag{4.64}$$

which is essentially the product of (4.61), (4.62), and (4.63). At the time of transition from exposure level $e-1$ to e, t_e, all this $D_{e-1}^{\text{clus.}}(t_e^-)$ amount of data loses its eth copy and is thus the most-exposed data in exposure level e. This is due to the nature of the clustered replica placement scheme. Therefore, the amount of most-exposed data in exposure level e, $D_e^{\text{clus.}}(t_e)$, is given by

$$D_e^{\text{clus.}}(t_e) = D_e^{\text{clus.}}(t_e^-) = a_{e-1}\tau_{e-1}c\mu. \tag{4.65}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_e^{\text{clus.}}$, of the rebuild time in the eth exposure level, $R_e^{\text{clus.}}$, is obtained by dividing (4.65) by (4.61), that is,

$$\frac{1}{\mu_e^{\text{clus.}}} = E[R_e^{\text{clus.}} | R_{e-1}^{\text{clus.}} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}] = \frac{D_e^{\text{clus.}}(t_e)}{S_e^{\text{clus.}}} = a_{e-1}\tau_{e-1}. \tag{4.66}$$

There are now $r-e$ remaining nodes in the cluster under rebuild. The failure of any of these nodes during the rebuild time $R_e^{\text{clus.}}$ will cause the system to enter exposure level $e+1$. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_e^{\text{clus.}} = r - e. \tag{4.67}$$

Clustered Replica Placement: MTTDL under Model A

Recall that, under model A, following each exposure level transition, the system is assumed to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where the node from which data was being rebuilt failed and hence the system has to rebuild from another node in the cluster. This implies that the rebuild time in the new exposure level is a random variable, and only its mean depends on the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level. Given this mean, the rebuild time in the new exposure level is independent of the rebuild time in the previous exposure level. Having computed the key quantities $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ for $e = 1, \cdots, r-1$, we are now ready to compute $P_{DL}^{\text{clus.}}$ using the expression (4.44) for model A, and then $\text{MTTDL}^{\text{clus.}}$ using (3.14).

By substituting the values of $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ from (4.51), (4.66), and (4.67) into (4.44), we obtain

$$\begin{aligned} P_{DL}^{\text{clus.}} &\approx \lambda^{r-1} \times (r-1)! \\ &\quad \times \int_{\tau_1} \cdots \int_{\tau_{r-1}} \int_{a_1} \cdots \int_{a_{r-2}} \tau_1 \cdots \tau_{r-1} g_\mu(\tau_1) \cdots g_{\frac{1}{a_{r-2}\tau_{r-2}}}(\tau_{r-1}) d\vec{a} d\vec{\tau} \\ &\qquad \text{(model A)} \end{aligned} \tag{4.68}$$

As in (4.44) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$.

Replication factors $r \leq 3$: The expression (4.68) for $P_{DL}^{\text{clus.}}$ under model A cannot, in general, be further simplified without considering a particular family of rebuild distributions G_μ. However, it is worth noting that, for $r \leq 3$, a closed form expression for $P_{DL}^{\text{clus.}}$, and hence $\text{MTTDL}^{\text{clus.}}$, can be obtained under model A. This is illustrated by deriving the closed form expression for $r = 3$ by substituting $r = 3$ in (4.68) and simplifying as follows.

$$\begin{aligned} P_{DL}^{\text{clus.}} &\approx \lambda^2 \times 2! \times \int_{\tau_1=0}^{\infty} \int_{\tau_2=0}^{\infty} \int_{a_1=0}^{1} \tau_1 \tau_2 g_\mu(\tau_1) g_{\frac{1}{a_1\tau_1}}(\tau_2) da_1 d\tau_2 d\tau_1 \\ &= 2\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1 g_\mu(\tau_1) \int_{a_1=0}^{1} \int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{1}{a_1\tau_1}}(\tau_2) d\tau_2 da_1 d\tau_1 \\ &\qquad \text{for } r = 3 \text{ (model A)}. \end{aligned} \tag{4.69}$$

Noting that

$$\int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{1}{a_1\tau_1}}(\tau_2) d\tau_2 = a_1\tau_1, \tag{4.70}$$

we get

$$P_{DL}^{\text{clus.}} \approx 2\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_\mu(\tau_1) \int_{a_1=0}^{1} a_1 da_1 d\tau_1 \quad (4.71)$$

$$= \lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_\mu(\tau_1) d\tau_1 \quad (4.72)$$

$$= \lambda^2 M_2(G_\mu) \qquad \text{for } r = 3 \text{ (model A)}, \quad (4.73)$$

where $M_2(G_\mu)$, as defined in (4.46), denotes the second raw moment of the rebuild distribution G_μ. The expression for $\text{MTTDL}^{\text{clus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{clus.}}} \approx \frac{1}{n\lambda^3 M_2(G_\mu)} = \frac{\mu^2}{n\lambda^3} \frac{M_1^2(G_\mu)}{M_2(G_\mu)}$$
$$\text{for } r = 3 \text{ (model A)}. \quad (4.74)$$

Here, the last step is obtained by multipliying and dividing by square of the mean of the rebuild time distribution G_μ, $M_1^2(G_\mu)$, which is also equal to $1/\mu^2$. This is done to show the effect of the rebuild distribution on the MTTDL. For deterministic rebuild times, the second raw moment, $M_2(G_\mu)$, is equal to the square of the first raw moment, $M_1^2(G_\mu)$, and therefore, the term $M_1^2(G_\mu)/M_2(G_\mu)$ evaluates to one. However, if the rebuild times are random, the second raw moment is always greater than the square of the first raw moment by Jensen's inequality, and therefore, the term $M_1^2(G_\mu)/M_2(G_\mu)$ is smaller than one. The closed form expression for $r = 2$ can be derived similarly and is given by

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu}{n\lambda^2} \qquad \text{for } r = 2 \text{ (model A)}. \quad (4.75)$$

Replication factors $r > 3$: For replication factors $r > 3$, the evaluation of $P_{Dl}^{\text{clus.}}$ under model A involves computing the expectations of functions involving higher raw moments of G_μ, which cannot be done without considering a particular family of rebuild distributions. As an example, if G_μ is exponential, the expression for MTTDL under model A can be shown to be the following:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{1}{(r-1)!} \prod_{e=1}^{r-3} \frac{1}{(r-e-1)^e}, \text{ when } G_\mu \text{ is exponential}$$
$$\text{(model A)}. \quad (4.76)$$

Clustered Replica Placement: MTTDL under Model B

In contrast to model A, we assume in model B that, following an exposure level transition, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the

case, for instance, in a clustered placement scheme where the newly failed node is different from the node from which data is being rebuilt. This is also the case in a declustered placement scheme, where the rebuild was being done from all nodes, and therefore, the failure of one node does not signficantly affect the rebuild process. This impliles that the rebuild time in the new exposure level is completely determined by the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level.

By substituting the values of $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ from (4.51), (4.66), and (4.67) into (4.45), we obtain

$$\begin{aligned} P_{DL}^{\text{clus.}} \approx\ & \lambda^{r-1} \times (r-1)! \\ & \times \int_{\tau_1} \cdots \int_{\tau_{r-1}} \int_{a_1} \cdots \int_{a_{r-2}} \Big(\tau_1 \cdots \tau_{r-1} g_\mu(\tau_1) \\ & \times \delta(\tau_2 - a_1\tau_1) \cdots \delta(\tau_{r-1} - a_{r-2}\tau_{r-2})\, d\vec{a} d\vec{\tau} \Big) \\ & \text{(model B).} \end{aligned} \tag{4.77}$$

As in (4.45) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$. In contrast to model A, closed form expressions in terms of the raw moments of the rebuild distribution G_μ can be obtained for model B as follows. By changing the order of integrals in (4.77), we obtain

$$\begin{aligned} P_{DL}^{\text{clus.}} &\approx \lambda^{r-1} \times (r-1)! \\ &\quad \times \int_{a_1} \cdots \int_{a_{r-2}} \int_{\tau_1} \cdots \int_{\tau_{r-1}} \Big(\tau_1 \cdots \tau_{r-1} g_\mu(\tau_1) \\ &\quad \times \delta(\tau_2 - a_1\tau_1) \cdots \delta(\tau_{r-1} - a_{r-2}\tau_{r-2})\, d\tau_{r-1} \cdots d\tau_1 d\vec{a} \Big) \\ &= \lambda^{r-1} \times (r-1)! \\ &\quad \times \int_{a_1} \cdots \int_{a_{r-2}} \int_{\tau_1} \cdots \int_{\tau_{r-2}} \Big(\tau_1 \cdots \tau_{r-2}^2 a_{r-2} g_\mu(\tau_1) \\ &\quad \times \delta(\tau_2 - a_1\tau_1) \cdots \delta(\tau_{r-2} - a_{r-3}\tau_{r-3})\, d\tau_{r-2} \cdots d\tau_1 d\vec{a} \Big) \end{aligned} \tag{4.78}$$

$$= \lambda^{r-1} \times (r-1)! \times \int_{a_1} \cdots \int_{a_{r-2}} \int_{\tau_1} \cdots \int_{\tau_{r-3}} \Big(\tau_1 \cdots \tau_{r-3}^3 a_{r-3}^2 a_{r-2} g_\mu(\tau_1) \times \delta(\tau_2 - a_1\tau_1) \cdots \delta(\tau_{r-3} - a_{r-4}\tau_{r-4})\, d\tau_{r-3} \cdots d\tau_1 d\vec{a} \Big) \quad (4.79)$$

$$\vdots$$

$$= \lambda^{r-1} \times (r-1)! \times \int_{a_1} \cdots \int_{a_{r-2}} \int_{\tau_1} \tau_1^{r-1} a_1^{r-2} \cdots a_{r-3}^2 a_{r-2} g_\mu(\tau_1) d\tau_1 d\vec{a} \quad \text{(model B)}. \quad (4.80)$$

Here, steps (4.78)–(4.80) follow by successively integrating over $\tau_{r-1}, \cdots, \tau_2$, by using the Dirac delta function's property. Changing the order of the integrals and integrating out $a_1, \cdots, a_{r-2}$, we get

$$P_{DL}^{\text{clus.}} \approx \lambda^{r-1} \times (r-1)! \times \int_{\tau_1} \tau_1^{r-1} g_\mu(\tau_1) \frac{1}{(r-1)!} d\tau_1 \quad (4.81)$$

$$= \lambda^{r-1} M_{r-1}(G_\mu) \quad \text{(model B)} \quad (4.82)$$

where $M_{r-1}(G_\mu)$, as defined in (4.46), denotes the $(r-1)$th raw moment of the rebuild distribution G_μ. The expression for $\text{MTTDL}^{\text{clus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{clus.}}} \approx \frac{1}{n\lambda^r M_{r-1}(G_\mu)} = \frac{\mu^{r-1}}{n\lambda^r} \frac{M_1^{r-1}(G_\mu)}{M_{r-1}(G_\mu)} \quad \text{(model B)}. \quad (4.83)$$

Here, the last step is obtained by multipliying and dividing by $(r-1)$th power of the mean of the rebuild time distribution G_μ, $M_1^{r-1}(G_\mu)$, which is also equal to $1/\mu^{r-1}$. This is done to show the effect of the rebuild distribution on the MTTDL. For deterministic rebuild times, the $(r-1)$th raw moment, $M_{r-1}(G_\mu)$, is equal to the $(r-1)$th power of the first raw moment, $M_1^{r-1}(G_\mu)$, and therefore, the term $M_1^{r-1}(G_\mu)/M_{r-1}(G_\mu)$ evaluates to one. For random rebuild times, by the Jensen's inequality, the $(r-1)$th raw moment, $M_{r-1}(G_\mu)$, is always greater than the $(r-1)$th power of the first raw moment, $M_1^{r-1}(G_\mu)$, and therefore, the term $M_1^{r-1}(G_\mu)/M_{r-1}(G_\mu)$ evaluates to less than one.

As an example, if G_μ is exponential, the expression for $\text{MTTDL}^{\text{clus.}}$ reduces to the following:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{1}{(r-1)!} \quad \text{when } G_\mu \text{ is exponential (model B)}. \quad (4.84)$$

Remark 4.6. *The expressions (4.74), (4.76), (4.83), and (4.84) for the mean time to data loss of a storage system with clustered replica placement scheme under both models A and B are seen to be invariant within the class of node*

failure distributions that satisfy (2.10) and (2.11). In particular, the MTTDL only depends on the mean of the times to node failure, $1/\lambda$. As the conditions (2.10) and (2.11) hold true for real-world storage nodes as well, these MTTDL results are of practical significance.

Remark 4.7. *The expressions (4.74), (4.76), (4.83), and (4.84) for the mean time to data loss of a storage system with clustered replica placement scheme also reveal that the MTTDL is sensitive to the rebuild distribution G_μ and to the choice of the model A or B. It is observed that deterministic rebuild times have better MTTDL values compared to random rebuild times because the terms of the form $M_1^{r-1}(G_\mu)/M_{r-1}(G_\mu)$ are upper-bounded by one due to the Jensen's inequality, and the bound is achieved for deterministic rebuild times. The explanation for this fact is that, when rebuild times are random, given that a failure occurred during rebuild, it is more probable that the rebuild time was larger. This effect is known as the waiting time paradox. The waiting time paradox is also the reason for the MTTDL values to be higher under model B than under model A (compare (4.76) and (4.84)) because model A introduces additional randomness to the rebuild times at each exposure level whereas model B does not.*

Remark 4.8. *For a given rebuild distribution G_μ and for replication factors $r \leq 3$, we note that the expressions for MTTDL under model A, given by (4.74) and (4.75), and the expressions for MTTDL under model B, given by (4.83), are not different. However, for replication factors $r > 3$, as illustrated by the choice of an exponential rebuild distribution in (4.76) and (4.84), the MTTDL under model A may differ from the MTTDL under model B. Furthermore, if the rebuild times are deterministic, we note that the models A and B do not differ by definition (see (4.39) and (4.40)). Therefore, for deterministic rebuild times, the MTTDL values under both models are the same.*

Remark 4.9. *The MTTDL of a replication-based system with clustered placement scheme is observed to scale down inversely proportional to the number of nodes n. It is directly proportional to the rth power of the mean time to node failure $1/\lambda$, and inversely proportional to the $(r-1)$th power of the mean time to node rebuild $1/\mu$. As will be seen later, this is a general trend in the MTTDL behavior of data storage systems. This trend also holds for erasure coded systems with an (l, m)-MDS code, with r replaced by $m - l + 1$. Besides changing the parameters λ, μ, and r, another way to influence the MTTDL of a storage system is by changing the replica placement. By changing the replica placement scheme, one can influence the scaling of MTTDL with respect to the number of nodes n, resulting in a tremendous improvement in reliability for large storage systems.*

4.2.2 Declustered Replica Placement

In most storage systems, the mean times to node failure and mean times to node rebuilds are given constants because they depend on the particular type of nodes used. For a replication-based system with a given type of node, one way to improve reliability is to increase the replication factor r. However, this comes at the cost of storage efficiency. For a given replication factor, as mentioned in Remark 4.9, it may be possible to simply change the underlying replica placement and the way in which rebuild is done to gain significant improvements in reliability for large storage systems. Declustered replica placement is one of those ways in which the system reliability can be improved over clustered placement for large storage systems. The goal of this section is to estimate the reliability of the declustered replica placement scheme in terms of the mean time to data loss, and understand how this replica placement scheme can achieve high reliability in large systems. To achieve this goal, we first compute the conditional means of rebuild times in each exposure level, $\mu_e^{\text{declus.}}$, $e = 1, \cdots, r-1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e^{\text{declus.}}$, $e = 1, \cdots, r-1$. Using these quantities and expressions (4.44) and (4.45), we can compute the probability of data loss during rebuild, $P_{DL}^{\text{declus.}}$. The mean time to data loss, $\text{MTTDL}^{\text{declus.}}$, can then be obtained by using the relation (3.14).

Declustered Replica Placement: Exposure Level 1

Following a first-node failure at t_1, the system enters exposure level 1 and the rebuild process begins. The amount of data to be rebuilt at this exposure level is equal to the capacity of the failed node, c, that is,

$$D_1^{\text{declus.}}(t_1) = c. \tag{4.85}$$

By the nature of the declustered placement, the $r-1$ remaining replicas of the data corresponding to the failed node are spread equally across all the surviving $n-1$ nodes of the system. As described in Section 2.6, the distributed rebuild process in a declustered placement scheme involves reading the replicas of the data to be rebuilt from all the surviving nodes of the system and copying to the spare space of these nodes in such a way that no data is copied to a node in which its replica is already present. As each of the $n-1$ node has an average read-write rebuild bandwidth of $c\mu$, and as equal amounts of data is read and written to each node during the distributed rebuild process owing to its symmetry, the average rate of rebuild in exposure level 1 is

$$S_1^{\text{declus.}} = (n-1)\frac{c\mu}{2}. \tag{4.86}$$

The average time required for this rebuild, $1/\mu_1^{\text{declus.}}$, is obtained by dividing the amount of data to be rebuilt, given by (4.85), by the average speed of

rebuild, given by (4.86). Thus,

$$\frac{1}{\mu_1^{\text{declus.}}} = E[R_1^{\text{declus.}}] = \frac{D_1^{\text{declus.}}(t_1)}{S_1^{\text{declus.}}} = \frac{1}{(n-1)\mu/2}. \tag{4.87}$$

According to our model, the rebuild time, R_1, is distributed according to G_{μ_1}, that is,

$$R_1^{\text{declus.}} \sim G_{\mu_1^{\text{declus.}}} = G_{(n-1)\mu/2}. \tag{4.88}$$

There are $n-1$ surviving nodes in the system, each containting equal amounts of the replicas corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period $R_1^{\text{declus.}}$ will cause the system to enter exposure level 2. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_1^{\text{declus.}} = n - 1. \tag{4.89}$$

When one of the $\tilde{n}_1^{\text{declus.}}$ nodes fail before rebuild, the system enters exposure level 2.

Declustered Replica Placement: Exposure Level 2

The system enters exposure level 2 from exposure level 1 because one of the $\tilde{n}_1^{\text{declus.}}$ nodes fails during the rebuild period $R_1^{\text{declus.}}$. Consider an instance of the rebuild period,

$$R_1^{\text{declus.}} = \tau_1, \tag{4.90}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from 1 to 2 occurred,

$$\alpha_1 = a_1. \tag{4.91}$$

The remaining time to complete rebuild at exposure level 1 when the system entered exposure level 2 is the product of $R_1^{\text{declus.}}$ and α_1, namely, $a_1\tau_1$. As the average speed of rebuild in exposure is $S_1^{\text{declus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_1(t_2^-)$, is given by

$$D_1^{\text{declus.}}(t_2^-) = \alpha_1 R_1^{\text{declus.}} S_1^{\text{declus.}} = a_1\tau_1(n-1)\frac{c\mu}{2}, \tag{4.92}$$

which is essentially the product of (4.86), (4.90), and (4.91). In contrast to clustered placement, at the time of transition from exposure level 1 to 2, t_2, *not* all of this $D_1^{\text{declus.}}(t_2^-)$ amount of data loses a second copy. As described in Section 2.4, due to the nature of the declustered replica placement scheme, the two failed nodes share copies of only a fraction $\frac{r-1}{n-1}$ of this data. So during the exposure level transition, only $\frac{r-1}{n-1}D_1^{\text{declus.}}(t_2^-)$ amount of data loses

a second copy. Therefore, the amount of most-exposed data in exposure level 2, $D_2^{\text{declus.}}(t_2)$, is given by

$$D_2^{\text{declus.}}(t_2) = \frac{r-1}{n-1} D_1^{\text{declus.}}(t_2^-) = (r-1)a_1\tau_1\frac{c\mu}{2}. \tag{4.93}$$

By the nature of the declustered placement, the $r-2$ remaining replicas of the most-exposed data are spread equally across all the surviving $n-2$ nodes of the system. The distributed rebuild process involves reading these replicas from all the surviving nodes of the system and copying to the spare space of these nodes in such a way that no data is copied to a node in which its replica is already present. As each of the $n-2$ node has an average read-write rebuild bandwidth of $c\mu$, and as equal amounts of data is read and written to each node during the distributed rebuild process owing to its symmetry, the average rate of rebuild in exposure level 2 is given by

$$S_2^{\text{declus.}} = (n-2)\frac{c\mu}{2}. \tag{4.94}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_2^{\text{declus.}}$, of the rebuild time in the second exposure level, $R_2^{\text{declus.}}$, is obtained by dividing (4.93) by (4.94), that is,

$$\frac{1}{\mu_2^{\text{declus.}}} = E[R_2^{\text{declus.}} | R_1^{\text{declus.}} = \tau_1, \alpha_1 = a_1] = \frac{D_2^{\text{declus.}}(t_2)}{S_2^{\text{declus.}}} = \frac{r-1}{n-2}a_1\tau_1. \tag{4.95}$$

There are now $n-2$ surviving nodes in the system, each containing equal amounts of the replicas corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period $R_2^{\text{declus.}}$ will cause the system to enter exposure level 3. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_2^{\text{declus.}} = n-2. \tag{4.96}$$

Declustered Replica Placement: Exposure Level e

The computation of the conditional mean $1/\mu_e^{\text{declus.}}$ and the number of nodes $\tilde{n}_e^{\text{declus.}}$ for a general exposure level $e = 2, \cdots, r-1$ is similar to the computation of these quantities for exposure level 2 as described above. Firstly, we note that the distributed rebuild process in each exposure level e always involves reading replicas of the data to be rebuilt from all $n-e$ surviving nodes of the system and copying it to the spare spaces of these nodes in such a way that no data is copied to a node in which its replica is already present. Due to the nature of the declustered placement, this involves reading and writing equal amounts of data in each node. As the average read-write rebuild bandwidth at each node is $c\mu$, which is equally split between the reads and the writes, the average speed of rebuild in each exposure level for declustered placement is

$$S_e^{\text{declus.}} = (n-e)\frac{c\mu}{2}, \quad e = 1, \cdots, r-1. \tag{4.97}$$

Now, the system enters exposure level e from exposure level $e-1$ because one of the $\tilde{n}_{e-1}^{\text{declus.}}$ nodes fails during the rebuild period $R_{e-1}^{\text{declus.}}$. Consider an instance of the rebuild period,

$$R_{e-1}^{\text{declus.}} = \tau_{e-1}, \tag{4.98}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from $e-1$ to e occurred,

$$\alpha_{e-1} = a_{e-1}. \tag{4.99}$$

The remaining time to complete rebuild at exposure level $e-1$ when the system entered exposure level e is the product of $R_{e-1}^{\text{declus.}}$ and α_{e-1}, namely, $a_{e-1}\tau_{e-1}$. As the average speed of rebuild in exposure is $S_{e-1}^{\text{declus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_{e-1}^{\text{declus.}}(t_e^-)$, is given by

$$D_{e-1}^{\text{declus.}}(t_e^-) = \alpha_{e-1} R_{e-1}^{\text{declus.}} S_{e-1}^{\text{declus.}} = a_{e-1}\tau_{e-1}(n-e+1)\frac{c\mu}{2}, \tag{4.100}$$

which is essentially the product of (4.97), (4.98), and (4.99). At the time of transition from exposure level $e-1$ to e, t_{e-1}, *not* all of this $D_1^{\text{declus.}}(t_2^-)$ amount of data loses its eth copy. Due to the nature of the declustered replica placement scheme, the newly failed nodes shares copies of only a fraction $\frac{r-e+1}{n-e+1}$ of this data. So during the exposure level transition, only $\frac{r-e+1}{n-e+1} D_1^{\text{declus.}}(t_2^-)$ amount of data loses its eth copy. Therefore, the amount of most-exposed data in exposure level e, $D_e^{\text{declus.}}(t_e)$, is given by

$$D_e^{\text{declus.}}(t_e) = \frac{r-e+1}{n-e+1} D_e^{\text{declus.}}(t_e^-) = (r-e+1)a_{e-1}\tau_{e-1}\frac{c\mu}{2}. \tag{4.101}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_e^{\text{declus.}}$, of the rebuild time in the eth exposure level, $R_e^{\text{declus.}}$, is obtained by dividing (4.101) by (4.97), that is,

$$\begin{aligned}
\frac{1}{\mu_e^{\text{declus.}}} &= E[R_e^{\text{declus.}} | R_{e-1}^{\text{declus.}} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}] && (4.102)\\
&= \frac{D_e^{\text{declus.}}(t_e)}{S_e^{\text{declus.}}} && (4.103)\\
&= \frac{r-e+1}{n-e} a_{e-1}\tau_{e-1}. && (4.104)
\end{aligned}$$

There are now $n-e$ surviving nodes in the system, each containting equal amounts of the replicas corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period $R_e^{\text{declus.}}$ will cause the system to enter exposure level $e+1$. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_e^{\text{declus.}} = n-e. \tag{4.105}$$

Declustered Replica Placement: MTTDL under Model A

Recall that, under model A, following each exposure level transition, the system is assumed to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where the node from which data was being rebuilt failed and hence the system has to rebuild from another node in the cluster. However, in a declustered placement scheme, where the distributed rebuild was being done from all nodes, the failure of one node may not signficantly affect the rebuild process. Therefore, model B may be better suited for the declustered placement scheme than model A. Nonetheless, we will derive the expressions for declustered placement scheme under model A for the sake of completeness.

Having computed the key quantities $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ for $e = 1, \cdots, r-1$, we are now ready to compute $P_{DL}^{\text{declus.}}$ using the expression (4.44) for model A, and then $\text{MTTDL}^{\text{declus.}}$ using (3.14). By substituting the values of $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ from (4.87), (4.104), and (4.105) into (4.44), we obtain

$$P_{DL}^{\text{declus.}} \approx \lambda^{r-1} \times (n-1)\cdots(n-r+1) \times \int_{\tau_1}\cdots\int_{\tau_{r-1}} \int_{a_1}\cdots\int_{a_{r-2}} \Bigg(\tau_1 \cdots \tau_{r-1} \times g_{\frac{(n-1)\mu}{2}}(\tau_1) \cdots g_{\frac{n-r+1}{2a_{r-2}\tau_{r-2}}}(\tau_{r-1}) d\vec{a} d\vec{\tau} \Bigg) \quad \text{(model A)} \quad (4.106)$$

As in (4.44) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$.

Replication factors $r \leq 3$: Similar to the case of clustered placement, the expression (4.106) for $P_{DL}^{\text{declus.}}$ under model A cannot, in general, be further simplified without considering a particular family of rebuild distributions G_μ. However, for $r \leq 3$, a closed form expression for $P_{DL}^{\text{declus.}}$, and hence $\text{MTTDL}^{\text{declus.}}$, can be obtained under model A. This is illustrated by deriving the closed form expression for $r = 3$ by substituting $r = 3$ in (4.106) and simplifying as follows.

$$P_{DL}^{\text{declus.}} \approx \lambda^2 \times (n-1)(n-2) \times \int_{\tau_1=0}^{\infty}\int_{\tau_2=0}^{\infty}\int_{a_1=0}^{1} \Bigg(\tau_1\tau_2 \times g_{\frac{(n-1)\mu}{2}}(\tau_1) g_{\frac{n-2}{2a_1\tau_1}}(\tau_2) da_1 d\tau_2 d\tau_1 \Bigg) \quad (4.107)$$

$$= (n-1)(n-2)\lambda^2 \Bigg(\int_{\tau_1=0}^{\infty} \tau_1 g_{\frac{(n-1)\mu}{2}}(\tau_1) \int_{a_1=0}^{1} \int_{\tau_2=0}^{\infty} \Big(\tau_2 \times g_{\frac{n-2}{2a_1\tau_1}}(\tau_2) \Big) d\tau_2 da_1 d\tau_1 \Bigg) \quad \text{for } r=3 \text{ (model A)}. \tag{4.108}$$

Noting that

$$\int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{n-2}{2a_1\tau_1}}(\tau_2) d\tau_2 = \frac{2a_1\tau_1}{n-2}, \tag{4.109}$$

we get

$$P_{DL}^{\text{declus.}} \approx 2(n-1)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_{\frac{(n-1)\mu}{2}}(\tau_1) \int_{a_1=0}^{1} a_1 da_1 d\tau_1 \tag{4.110}$$

$$= (n-1)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_{\frac{(n-1)\mu}{2}}(\tau_1) d\tau_1 \tag{4.111}$$

$$= (n-1)\lambda^2 M_2(G_{(n-1)\mu/2}) \qquad \text{for } r=3 \text{ (model A)}, \tag{4.112}$$

where $M_2(G_{(n-1)\mu/2})$, as defined in (4.46), denotes the second raw moment of the rebuild distribution $G_{(n-1)\mu/2}$. The expression for $\text{MTTDL}^{\text{declus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{declus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{declus.}}} \tag{4.113}$$

$$\approx \frac{1}{n(n-1)\lambda^3 M_2(G_{(n-1)\mu/2})} \quad \text{for } r=3 \text{ (model A)}. \tag{4.114}$$

Multipliying and dividing (4.114) by square of the mean of the rebuild time distribution $G_{(n-1)\mu/2}$,

$$M_1^2(G_{(n-1)\mu/2}) = \frac{1}{((n-1)\mu/2)^2} \tag{4.115}$$

we get

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{4n\lambda^3} \frac{M_1^2(G_{(n-1)\mu/2})}{M_2(G_{(n-1)\mu/2})} \quad \text{for } r=3 \text{ (model A)}. \tag{4.116}$$

For deterministic rebuild times, the second raw moment, $M_2(G_{(n-1)\mu/2})$, is equal to the square of the first raw moment, $M_1^2(G_{(n-1)\mu/2})$, and therefore, the term $M_1^2(G_{(n-1)\mu/2})/M_2(G_{(n-1)\mu/2})$ evaluates to one. However, if the rebuild times are random, the second raw moment is always greater than the square of the first raw moment by Jensen's inequality, and therefore, the term $M_1^2(G_{(n-1)\mu/2})/M_2(G_{(n-1)\mu/2})$ is smaller than one. The closed form expression for $r=2$ can be derived similarly and is given by

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu}{2n\lambda^2} \qquad \text{for } r=2 \text{ (model A)}. \tag{4.117}$$

Replication factors $r > 3$: For replication factors $r > 3$, the evaluation of $P_{Dl}^{\text{declus.}}$ under model A involves computing the expectations of functions involving higher raw moments of G_μ, which cannot be done without considering a particular family of rebuild distributions. However, given a particular family of rebuild distributions, the derivation of MTTDL involves successively evaluating the integrals in (4.106) to compute $P_{DL}^{\text{declus.}}$, and then using (3.14) to obtain $\text{MTTDL}^{\text{declus.}}$.

Declustered Replica Placement: MTTDL under Model B

In contrast to model A, we assume in model B that, following an exposure level transition, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the case in a declustered placement scheme, where the rebuild was being done from all nodes, and therefore, in a large system, the failure of one node does not signficantly affect the rebuild process. This impliles that the rebuild time in the new exposure level is completely determined by the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level.

By substituting the values of $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ from (4.87), (4.104), and (4.105) into (4.45), we obtain

$$\begin{aligned} P_{DL}^{\text{declus.}} \approx{}& \lambda^{r-1} \times (n-1)\cdots(n-r+1) \\ &\times \int_{\tau_1}\cdots\int_{\tau_{r-1}} \int_{a_1}\cdots\int_{a_{r-2}} \Bigg(\tau_1\cdots\tau_{r-1} g_{\frac{(n-1)\mu}{2}}(\tau_1) \\ &\times \delta\left(\tau_2 - \frac{r-1}{n-2}a_1\tau_1\right)\cdots\delta\left(\tau_{r-1} - \frac{2}{n-r+1}a_{r-2}\tau_{r-2}\right) d\vec{a}\,d\vec{\tau}\Bigg) \\ &\text{(model B).} \end{aligned} \tag{4.118}$$

As in (4.45) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$. In contrast to model A, closed form expressions in terms of the raw moments of the rebuild distribution can be obtained for model B as follows. By changing the order of integrals in (4.118), we obtain

$$\begin{aligned} P_{DL}^{\text{declus.}} \approx{}& \lambda^{r-1} \times (n-1)\cdots(n-r+1) \\ &\times \int_{a_1}\cdots\int_{a_{r-2}} \int_{\tau_1}\cdots\int_{\tau_{r-1}} \Bigg(\tau_1\cdots\tau_{r-1} g_{\frac{(n-1)\mu}{2}}(\tau_1) \\ &\times \delta\left(\tau_2 - \frac{r-1}{n-2}a_1\tau_1\right)\cdots\delta\left(\tau_{r-1} - \frac{2}{n-r+1}a_{r-2}\tau_{r-2}\right) \\ &\times d\tau_{r-1}\cdots d\tau_1 d\vec{a}\Bigg) \end{aligned} \tag{4.119}$$

$$= \lambda^{r-1} \times (n-1)\cdots(n-r+1) \times \frac{2}{n-r+1} \times \int_{a_1}\cdots\int_{a_{r-2}}\int_{\tau_1}\cdots\int_{\tau_{r-2}} \left(\tau_1 \cdots \tau_{r-2}^2 a_{r-2} g_{\frac{(n-1)\mu}{2}}(\tau_1) \times \delta\left(\tau_2 - \frac{r-1}{n-2} a_1 \tau_1\right) \cdots \delta\left(\tau_{r-2} - \frac{3}{n-r+2} a_{r-3}\tau_{r-3}\right) \times d\tau_{r-2}\cdots d\tau_1 d\vec{a} \right) \quad (4.120)$$

$$= \lambda^{r-1} \times (n-1)\cdots(n-r+1) \times \frac{2}{(n-r+1)} \times \frac{3^2}{(n-r+2)^2} \times \int_{a_1}\cdots\int_{a_{r-2}}\int_{\tau_1}\cdots\int_{\tau_{r-3}} \left(\tau_1 \cdots \tau_{r-3}^3 a_{r-3}^2 a_{r-2} g_{\frac{(n-1)\mu}{2}}(\tau_1) \times \delta\left(\tau_2 - \frac{r-1}{n-2} a_1 \tau_1\right) \cdots \delta\left(\tau_{r-3} - \frac{4}{n-r+3} a_{r-4}\tau_{r-4}\right) \times d\tau_{r-3}\cdots d\tau_1 d\vec{a} \right) \quad (4.121)$$

$$\vdots$$

$$= \lambda^{r-1} \times (n-1)^{r-1} \times \prod_{e=1}^{r-2} \left(\frac{r-e}{n-e}\right)^{r-e-1} \times \int_{a_1}\cdots\int_{a_{r-2}}\int_{\tau_1} \tau_1^{r-1} a_1^{r-2} \cdots a_{r-3}^2 a_{r-2} g_{\frac{(n-1)\mu}{2}}(\tau_1) d\tau_1 d\vec{a} \quad \text{(model B)}. \quad (4.122)$$

Here, steps (4.120)–(4.122) follow by successively integrating over $\tau_{r-1}, \cdots, \tau_2$, using the Dirac delta function's property, cancelling out terms of the form $(n-r+e)$, $e = 1, \cdots, r-2$, and rewriting the terms outside the integral after multiplying and dividing by $(n-1)^{r-1}$. Changing the order of the integrals and integrating out $a_1, \cdots, a_{r-2}$, we get

$$P_{DL}^{\text{declus.}} \approx \lambda^{r-1}(n-1)^{r-1} \prod_{e=1}^{r-2} \left(\frac{r-e}{n-e}\right)^{r-e-1} \int_{\tau_1} \tau_1^{r-1} g_{\frac{(n-1)\mu}{2}}(\tau_1) \frac{1}{(r-1)!} d\tau_1$$
$$= \lambda^{r-1} M_{r-1}(G_{(n-1)\mu/2}) \frac{(n-1)^{r-1}}{(r-1)!} \prod_{e=1}^{r-2} \left(\frac{r-e}{n-e}\right)^{r-e-1} \quad \text{(model B)}, \quad (4.123)$$

where $M_{r-1}(G_{(n-1)\mu/2})$, as defined in (4.46), denotes the $(r-1)$th raw moment of the rebuild distribution $G_{(n-1)\mu/2}$. The expression for $\text{MTTDL}^{\text{declus.}}$ then

follows from (3.14):

$$\text{MTTDL}^{\text{declus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{declus.}}} \tag{4.124}$$

$$\approx \frac{1}{n\lambda^r M_{r-1}(G_{(n-1)\mu/2})} \frac{(r-1)!}{(n-1)^{r-1}} \prod_{e=1}^{r-2} \left(\frac{n-e}{r-e}\right)^{r-e-1}$$

$$\text{(model B).} \tag{4.125}$$

Multipliying and dividing (4.125) by the $(r-1)$th power of the mean of the rebuild time distribution $G_{(n-1)\mu/2}$,

$$M_1^{r-1}(G_{(n-1)\mu/2}) = \frac{1}{((n-1)\mu/2)^{r-1}} \tag{4.126}$$

we get

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{M_1^{r-1}(G_{(n-1)\mu/2})}{M_{r-1}(G_{(n-1)\mu/2})} \frac{(r-1)!}{2^{r-1}} \prod_{e=1}^{r-2} \left(\frac{n-e}{r-e}\right)^{r-e-1}$$

$$\text{(model B).} \tag{4.127}$$

For deterministic rebuild times, the $(r-1)$th raw moment, $M_{r-1}(G_{(n-1)\mu/2})$, is equal to the $(r-1)$th power of the first raw moment, $M_1^{r-1}(G_{(n-1)\mu/2})$, and therefore, the term $M_1^{r-1}(G_{(n-1)\mu/2})/M_{r-1}(G_{(n-1)\mu/2})$ evaluates to one. For random rebuild times, by the Jensen's inequality, the $(r-1)$th raw moment, $M_{r-1}(G_{(n-1)\mu/2})$, is always greater than the $(r-1)$th power of the first raw moment, $M_1^{r-1}(G_{(n-1)\mu/2})$, and therefore, their ratio evaluates to less than one.

As an example, if $G_{(n-1)\mu/2}$ is exponential, the expression for $\text{MTTDL}^{\text{declus.}}$ reduces to the following:

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{1}{2^{r-1}} \prod_{e=1}^{r-2} \left(\frac{n-e}{r-e}\right)^{r-e-1}$$

$$\text{when } G_{(n-1)\mu/2} \text{ is exponential (model B).} \tag{4.128}$$

Remark 4.10. *The expressions (4.116), (4.117), (4.127), and (4.128) for the mean time to data loss of a storage system with declustered replica placement scheme under both models A and B are seen to be invariant within the class of node failure distributions that satisfy (2.10) and (2.11). In particular, the MTTDL only depends on the mean of the times to node failure, $1/\lambda$. As the conditions (2.10) and (2.11) hold true for real-world storage nodes as well, these MTTDL results are of practical significance.*

Remark 4.11. *The expressions (4.116), (4.127), and (4.128) for the mean time to data loss of a storage system with declustered replica placement scheme also reveal that the MTTDL is sensitive to the rebuild distribution $G_{(n-1)\mu/2}$. It is observed that deterministic rebuild times have higher MTTDL values*

compared to random rebuild times. This is because the terms of the form $M_1^{r-1}(G_{(n-1)\mu/2})/M_{r-1}(G_{(n-1)\mu/2})$ *are upper-bounded by* 1 *due to the Jensen's inequality, and this bound is achieved for deterministic rebuild times. The explanation for this fact is that, when rebuild times are random, given that a failure occurred during rebuild, it is more probable that the rebuild time was larger. This effect is known as the waiting time paradox. Larger rebuild times imply that a larger amount of most-exposed data remains unrebuilt when the system enters a higher exposure level, thereby reducing the reliability.*

Remark 4.12. *Comparing the MTTDL values of clustered placement in (4.83) with those of declustered placement in (4.127), we observe that they are both directly proportional to the* r*th power of the mean time to node failure* $1/\lambda$*, and inversely proportional to the* $(r-1)$ *power of the mean time to node rebuild* $1/\mu$*. This is a general trend in the MTTDL behavior of data storage systems. This trend also holds for erasure coded systems with an* (l, m)*-MDS code, with* r *replaced by* $m-l+1$*. However, in contrast to clustered placement, the MTTDL of a replication-based system with declustered placement scheme is observed to scale differently with the number of nodes,* n*, for different replication factors,* r*. It can be seen from (4.127) that the MTTDL of declustered placement scales roughly as the* $(r(r-3)/2)$*th power of* n*. For* $r = 2$*, the MTTDL of declustered placement scales inversely proportional to* n*, just like in clustered placement. For* $r = 3$*, the MTTDL of declustered placement stays roughtly constant with* n*. For* $r > 3$*, the MTTDL of declustered placement increases with* n*. This shows that, by changing the replica placement scheme, one can influence the scaling of MTTDL with respect to the number of nodes* n*, resulting in a tremendous improvement in reliability for large storage systems.*

4.3 Clustered vs. Declustered Replica Placement

In this section, we compare and contrast the MTTDL of systems under clustered and declustered replica placement schemes using the help of figures. Note that models A and B, as described in Section 4.1.3, do not differ in the values of MTTDL for $r \leq 3$. Furthermore, the difference between models A and B for $r > 3$ is typically only a constant factor that depends on the rebuild distribution. Also, if the the rebuild times are deterministic, there is no difference between models A and B and therefore they agree on the MTTDL values for all replication factors. So, without loss of generality, we will only consider the MTTDL values under model B for further discussions in this section.

4.3.1 Replication Factor 2

Plugging $r = 2$ in (4.83) and (4.127), we obtain the MTTDL of two-way replicated systems for clustered and declustered placement schemes, respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu}{n\lambda^2} \quad \text{for } r = 2. \tag{4.129}$$

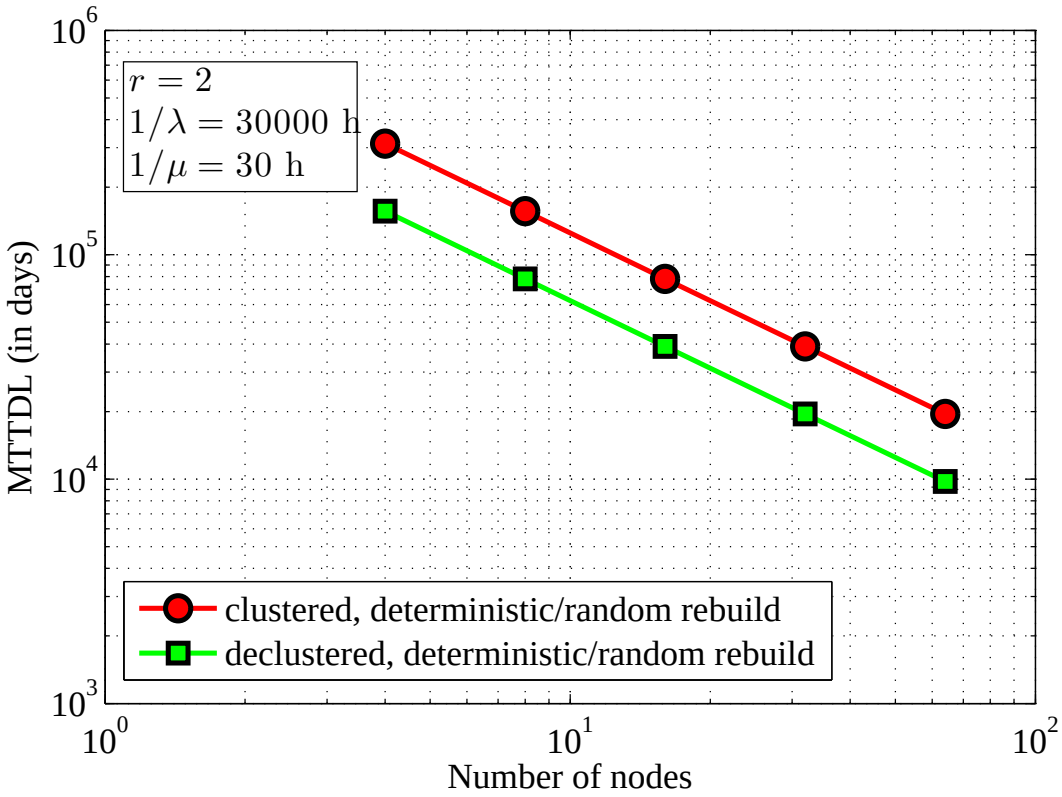


Figure 4.1: MTTDL as a function of the number of nodes for replication factor two.

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu}{2n\lambda^2} \qquad \text{for } r = 2. \tag{4.130}$$

From (4.129) and (4.130), it is observed that the MTTDL of two-way replicated systems under both placement schemes are directly proportional to the square of the mean time to node failure, $1/\lambda$, and inversely proportional to the mean time to read all contents of a node during rebuild, $1/\mu$. In addition, the MTTDL values are seen to be independent of the underlying rebuild distribution. Figure 4.1 illustrates the MTTDL behavior of two-way replicated systems with respect to the number of nodes in the system.

4.3.2 Replication Factor 3

Plugging $r = 3$ in (4.83) and (4.127), we obtain the MTTDL values for three-way replicated systems for clustered and declustered placement schemes, respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^2}{n\lambda^3} \frac{M_1^2(G_\mu)}{M_2(G_\mu)} \qquad \text{for } r = 3. \tag{4.131}$$

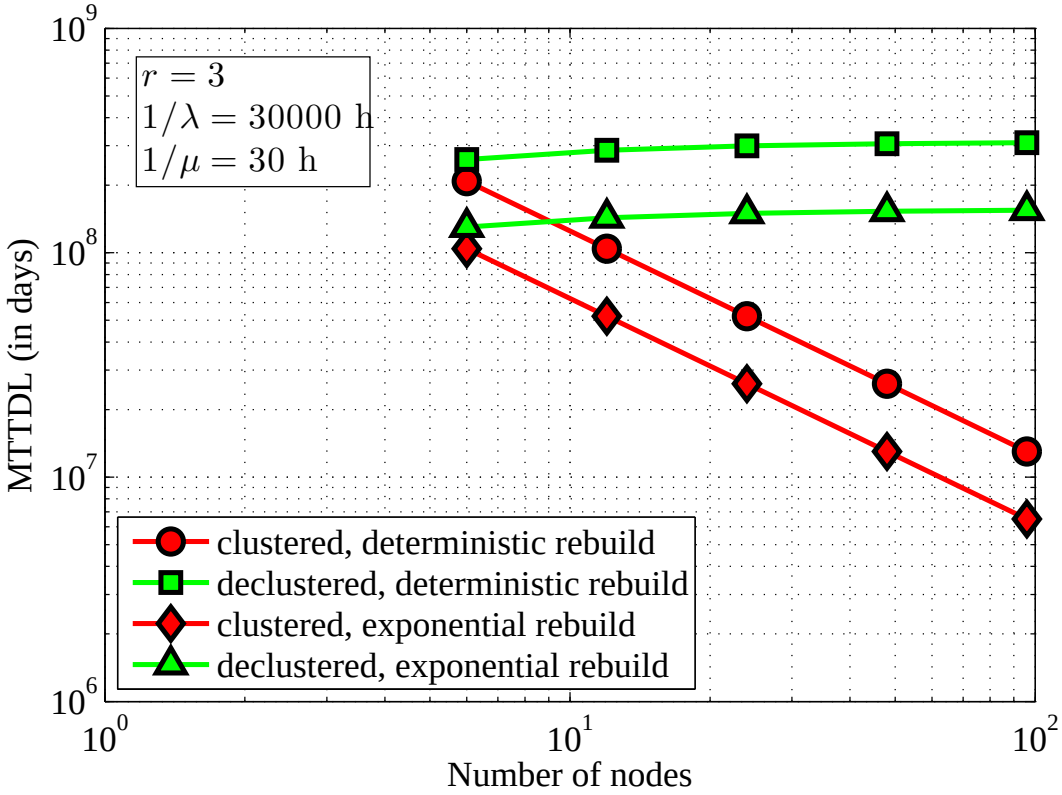


Figure 4.2: MTTDL as a function of the number of nodes for replication factor three.

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{4n\lambda^3} \frac{M_1^2\left(G_{\frac{n-1}{2}\mu}\right)}{M_2\left(G_{\frac{n-1}{2}\mu}\right)} \qquad \text{for } r = 3. \tag{4.132}$$

From (4.131) and (4.132), it is observed that the MTTDL of three-way replicated systems under both placement schemes are directly proportional to the cube of the mean time to node failure, $1/\lambda$, and inversely proportional to the square of the mean time to read all contents of a node during rebuild, $1/\mu$. In contrast to two-way replicated systems, it is seen that the MTTDL depends on the rebuild distribution. For deterministic rebuild times, the ratios $M_1^2(G_\mu)/M_2(G_\mu)$ and $M_1^2\left(G_{\frac{n-1}{2}\mu}\right)/M_2\left(G_{\frac{n-1}{2}\mu}\right)$ become one. However, for random rebuild times, these ratios are upper-bounded by one by Jensen's inequality. As an example, if the rebuild time distribution was exponential, these ratios are equal to $1/2$ and therefore,

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^2}{2n\lambda^3} \qquad \text{for } r = 3 \text{ (exponential rebuilds).} \tag{4.133}$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{8n\lambda^3} \qquad \text{for } r = 3 \text{ (exponential rebuilds).} \tag{4.134}$$

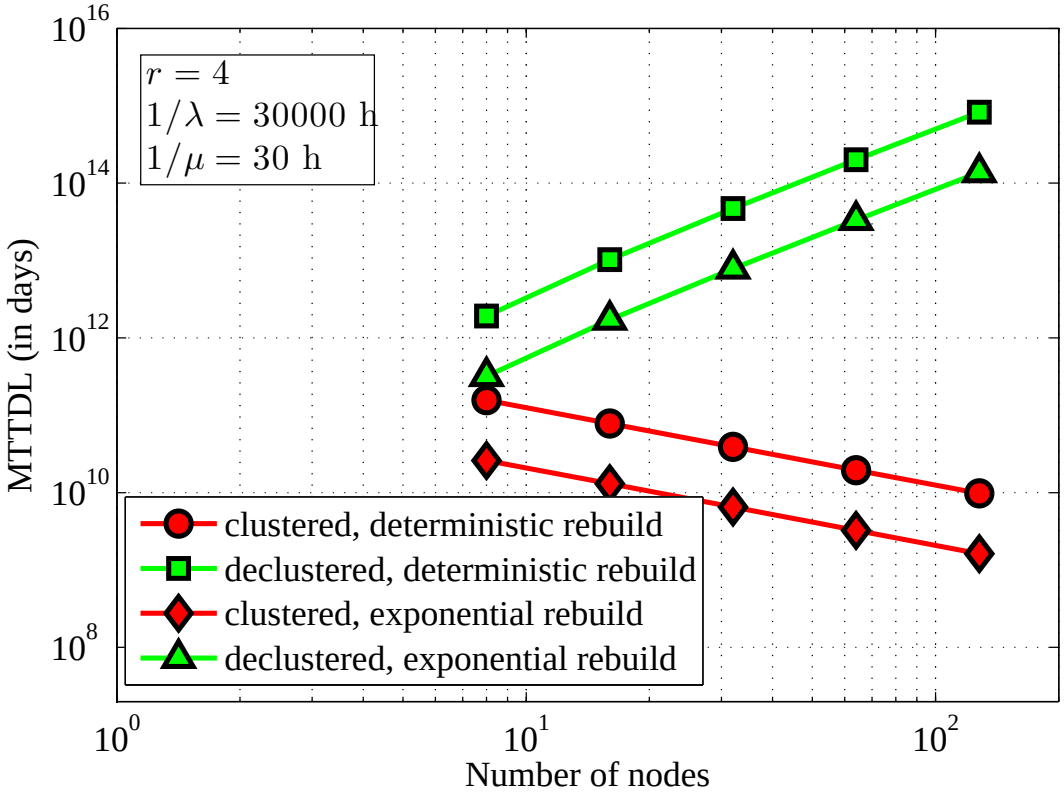


Figure 4.3: MTTDL as a function of the number of nodes for replication factor four.

The MTTDL of a system using a three-way replicated system is plotted against the number of nodes in the system for clustered and declustered placements, as well as for deterministic and exponential rebuild times, in Figure 4.2. It is observed that the rebuild time distribution scales down the MTTDL, but leaves the behavior with respect to the number of nodes, n, unaffected. This has also been verified by means of simulation in Chapter 7.

In contrast to two-way replicated systems, the difference in MTTDL between the two schemes for a three-way replicated system can be significant, depending on the number of nodes, n. This is because, as seen from (4.131) and (4.132), the MTTDL of clustered placement is inversely proportional to n, whereas the MTTDL of declustered placement is roughly invariant with respect to n. This is illustrated in Figure 4.2 in which MTTDL of double parity codes is plotted against the number of nodes, n, in a log-log scale. The lines corresponding to clustered placement have a slope of -1 indicating that the MTTDL is inversely proportional to n, whereas the lines corresponding to declustered placement have a slope of roughly 0 indicating that the MTTDL is invariant with respect to n.

4.3.3 Replication Factor 4

Plugging $r = 4$ in (4.83) and (4.127), we obtain the MTTDL values for four-way replicated systems for clustered and declustered placement schemes, respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^3}{n\lambda^4} \frac{M_1^3(G_\mu)}{M_3(G_\mu)} \quad \text{for } r = 4. \tag{4.135}$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)^2(n-2)\mu^3}{24n\lambda^4} \frac{M_1^3\left(G_{\frac{n-1}{2}\mu}\right)}{M_3\left(G_{\frac{n-1}{2}\mu}\right)} \quad \text{for } r = 4. \tag{4.136}$$

From (4.135) and (4.136), it is observed that the MTTDL of four-way replicated systems under both placement schemes are directly proportional to the fourth power of the mean time to node failure, $1/\lambda$, and inversely proportional to the cube of the mean time to read all contents of a node during rebuild, $1/\mu$. As was the case in double parity codes, the MTTDL depends on the rebuild distribution. For deterministic rebuild times, the ratios $M_1^3(G_\mu)/M_3(G_\mu)$ and $M_1^3\left(G_{\frac{n-1}{2}\mu}\right)/M_2\left(G_{\frac{n-1}{2}\mu}\right)$ become one. However, for random rebuild times, these ratios are upper-bounded by one by Jensen's inequality. As an example, if the rebuild time distribution was exponential, these ratios are equal to 1/6 and therefore,

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^3}{6n\lambda^4} \qquad \text{for } r = 4 \text{ (exponential rebuilds).} \tag{4.137}$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)^2(n-2)\mu^3}{144n\lambda^4} \qquad \text{for } r = 4 \text{ (exponential rebuilds).} \tag{4.138}$$

Comparing (4.138) with (4.136), it is observed that the rebuild time distribution scales down the MTTDL, but leaves the behavior with respect to the number of nodes, n, unaffected. This can be seen in the plots of MTTDL of a system using four-way replication against the number of nodes in the system for clustered and declustered placements, as well as for deterministic and exponential rebuild times, in Figure 4.3. Also, as in the case of double parity codes, the difference in MTTDL between the two schemes can be significant, depending on the number of nodes, n, in the system. This is because, as seen from (4.135) and (4.136), the MTTDL of clustered placement is inversely proportional to n, whereas the MTTDL of declustered placement is roughly proportional to the square of n. This is illustrated in Figure 4.3 in which MTTDL is plotted against the number of nodes, n, in a log-log scale. The lines corresponding to clustered placement have a slope of -1 indicating that the MTTDL is inversely proportional to n, whereas the lines corresponding to declustered placement have a slope of roughly 2 indicating that the MTTDL is proportional to the square of n.

Remark 4.13. *The relative error in the approximations (3.14), (4.44), and (4.45) tends to zero as the ratio λ/μ tends to zero. So, the expressions for MTTDL obtained in this thesis are better approximations for smaller values of λ/μ. This implies that, if simulation-based MTTDL values match the theoretically predicted MTTDL values for a certain value of λ/μ, it will also match for all smaller values of λ/μ. This fact is used in Chapter 7, where simulations are shown to match theory for values of λ/μ that are much larger than those observed in real-world storage systems, thereby establishing the applicability of the theoretical results to real-world storage systems.*

Impact of Limited Network Rebuild Bandwidth

5

As seen in the previous chapter, the manner in which redundant data are placed across the nodes in the system, that is, data placement, affects both how fast and how effective the rebuild process can be. There are two main ways in which the data placement, and hence the rebuild process, affects the reliability of the system. Firstly, if the redundant data are placed across several nodes in the system, the rebuild process can benefit by parallelizing the data restoration process. The restoration time can be minimal provided there is sufficient network bandwidth available. Minimal restoration time implies that there is a shorter window of time during which additional node failures can hinder the rebuild. Secondly, spreading the replicas of data across several nodes also exposes these replicas to the failure of any of these nodes, thereby increasing the probability of failure of the rebuild process. Interestingly, for two-way replicated systems, these two effects cancel each other out resulting in similar reliability for all data placement schemes [15]. For higher replication factors, however, the first effect is more dominant as the second effect tends to expose less amount of data to the danger of irrecoverable loss because of the spreading of replicas across several nodes [18]. When the network bandwidth is limited though, the rebuild times may be longer and therefore the former factor may be affected. This imbalance leads to interesting results in terms of the mean time to data loss (MTTDL) of the system. In this chapter, we explore this effect and show how the network bandwidth constraint affects the system MTTDL and how we can design schemes that achieve high reliability under these conditions. The results of this chapter can also be used to adapt the data placement schemes when the available network rebuild bandwidth or the number of nodes in the system changes so that system reliability is maintained at a high level at all times.

Table 5.1: Parameters of a storage system with limited network rebuild bandwidth

c	amount of data stored on each node (bytes)
n	number of storage nodes
$1/\lambda$	mean time to failure of a storage node (s)
b	average rebuild bandwidth at each storage node (bytes/s)
r	replication factor
k	spread factor of the data placement scheme
$B_{\max}$	maximum network rebuild bandwidth (bytes/s)
$1/\mu$	average time to read/write c amount of data from/to a node during rebuild ($1/\mu = c/b$)
N	effective maximum number of nodes from which distributed rebuild can occur at full speed in parallel ($N = B_{\max}/b$)
$B_{\text{eff}}(\tilde{k})$	effective distributed rebuild bandwidth involving $\tilde{k}$ nodes ($B_{\text{eff}}(\tilde{k}) = \min(\tilde{k}b, B)$)
$S_{\text{eff}}(\tilde{k})$	effective speed of distributed rebuild involving $\tilde{k}$ nodes ($S_{\text{eff}}(\tilde{k}) = B_{\text{eff}}(\tilde{k})/2$)

5.1 Limited Network Rebuild Bandwidth

In this section, we describe the system model under limited network rebuild bandwidth. Table 5.1 lists the parameters used. The upper and lower parts of the table list the set of independent and dependent parameters, respectively.

The rebuild process used to restore the data lost by failed nodes is assumed to be both *intelligent* and *distributed*. By an intelligent rebuild process, we mean that the system always attempts to first recover the copies (replicas) of the most critical data, that is, data that has the least number of replicas left in the system. In a distributed rebuild process, the data lost by a failed node is restored by reading surviving replicas and creating a new replica of the data in reserved spare space on surviving nodes as illustrated in Fig. 2.1. More specifically, if the surviving replicas of the most critical data are stored across $\tilde{k}$ nodes, these replicas are used to rebuild the lost data in the spare space on those $\tilde{k}$ nodes such that no two copies of the same data are stored on the same node. This is done so that the rebuild process can make use of the node rebuild bandwidth available at all $\tilde{k}$ nodes in parallel. Once all lost data is recovered, this newly recovered data is transferred to new replacement nodes. For clustered placement, the surviving replicas of the most critical data of a cluster are present on all the surviving nodes of that cluster. Therefore, the replicas of this data are read from one of the surviving nodes and written to a new spare node, as it is not possible to do a distributed rebuild as described earlier without creating two replicas of the same data on the same node.

During the rebuild process, an average read-write bandwidth of b bytes/s is assumed to be reserved at each node exclusively for the rebuild. This is usually

only a fraction of the total bandwidth available at each node; the remainder is being used to serve user requests. If $1/\mu$ is the time to rebuild a storage node in clustered placement, that is, the time required to read all contents of a node and write to a new spare node, then

$$\frac{1}{\mu} = \frac{c}{b}. \tag{5.1}$$

In a distributed rebuild process, if the surviving replicas of the most critical data are stored across $\tilde{k}$ nodes, then the total network bandwidth required to perform rebuild at full speed is $\tilde{k}b$. Let the maximum available network bandwidth for rebuilds be denoted by $B_{\max}$. We will assume that $B_{\max} \geq b$ as $B_{\max} < b$ is a degenerate case. So, if the available network rebuild bandwidth is $B_{\max}$, the total bandwidth that can be used by rebuilds cannot exceed $B_{\max}$. Therefore, the effective distributed rebuild bandwidth, $B_{\text{eff}}(\tilde{k})$, is given by

$$B_{\text{eff}}(\tilde{k}) := \min(\tilde{k}b, B_{\max}) = \min(\tilde{k}, N)b, \tag{5.2}$$

where N specifies the effective maximum number of nodes from which rebuild can occur in parallel at full speed and is given by

$$N := \frac{B_{\max}}{b}. \tag{5.3}$$

Note that N may not be an integer; it only represents the *effective* maximum number of nodes from which distributed rebuild can occur at full speed. Substituting for b from (5.1) into (5.2), we get

$$B_{\text{eff}} = \min(\tilde{k}, N)c\mu. \tag{5.4}$$

The distributed rebuild process in involves reading the replicas of the data to be rebuilt from $\tilde{k}$ nodes and copying to the spare space of these nodes in such a way that no data is copied to a node in which its replica is already present. As equal amounts of data are read from and written to each node during the distributed rebuild process owing to its symmetry, the average rate of rebuild, $S_{\text{eff}}(\tilde{k})$, is equal to half of the effective distributed rebuild bandwidth, $B_{\text{eff}}(\tilde{k})$, that is,

$$S_{\text{eff}}(\tilde{k}) = \frac{1}{2}B_{\text{eff}}(\tilde{k}) = \frac{1}{2}\min(\tilde{k}, N)c\mu. \tag{5.5}$$

For the sake of clarity and consistency with earlier chapters, we will only use expressions involving μ and N rather than b and $B_{\max}$ in the remainder of the paper. The implicit relationship between μ, N, b, and $B_{\max}$ is given in Table 2.1.

Clustered placement is an exception as it does not use distributed rebuild. The effective speed of rebuild for clustered placement is $c\mu$ because data is read from any *one* of the surviving nodes of the cluster to which the failed node belonged, and then written to a spare node.

5.2 Effect of Limited Network Rebuild Bandwidth on Reliability

In this section, we consider different replica placement schemes as discussed in Section 2.4. Under limited network rebuild bandwidth, we would like to estimate their reliability in terms of their MTTDL using the relations (3.14), (4.44), and (4.45), and understand how the limited network bandwidth and replica placement affect data reliability. To use the expressions (4.44) and (4.45) for P_{DL}, we need to compute the conditional means of rebuild times in each exposure level, μ_e, $e = 1, \cdots, r-1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e$, $e = 1, \cdots, r-1$. The values of these quantities depend on the underlying replica placement and the nature of the rebuild process used. The notations used in this chapter are the same those described in Section 4.2.

5.2.1 Clustered Replica Placement

Since $B_{\max} > b = c\mu$, clustered replica placement in not affected by the value of $B_{\max}$. This is because the rebuild process in clustered placement always involves reading data from any one of the surviving nodes at an average speed of $c\mu$ and writing it to a spare node at the same speed. Therefore, the MTTDL values for clustered placement are the same as in the previous chapter, and are given by (4.83).

5.2.2 Other Symmetric Replica Placement Schemes

For symmetric placement schemes other than clustered placement, that is, for spread factors $k > r$, the rebuilds may be affected by the value of $B_{\max}$ as indicated by (5.3), (5.4), and (5.5). This may also impact the MTTDL values of these replica placement schemes. Here, we derive the expressions for MTTDL of these replica placement schemes under limited network rebuild bandwidth.

Exposure Level 1

Following a first-node failure at t_1, the system enters exposure level 1 and the rebuild process begins. The amount of data to be rebuilt at this exposure level is equal to the capacity of the failed node, c, that is,

$$D_1(t_1) = c. \tag{5.6}$$

By the nature of the symmetric placement scheme with spread factor k, the $r-1$ remaining replicas of the data corresponding to the failed node are spread equally across $k-1$ other surviving nodes of the system. As described in Section 5.1, the distributed rebuild process involves reading the replicas of the

data to be rebuilt from the $k-1$ and copying to the spare space of these nodes in such a way that no data is copied to a node in which its replica is already present. As this distributed rebuild process involves $k-1$ nodes, the average rate of rebuild in exposure level 1 is given by (5.5) for $\tilde{k} = k-1$, that is,

$$S_1 = S_{\text{eff}}(k-1) = \frac{1}{2}\min(k-1, N)c\mu. \tag{5.7}$$

The average time required for this rebuild, $1/\mu_1$, is obtained by dividing the amount of data to be rebuilt, given by (5.6), by the average speed of rebuild, given by (5.7). Thus,

$$\frac{1}{\mu_1} = E[R_1] = \frac{D_1(t_1)}{S_1} = \frac{1}{\min(k-1, N)\mu/2}. \tag{5.8}$$

According to our model, the rebuild time, R_1, is distributed according to G_{μ_1}, that is,

$$R_1 \sim G_{\mu_1} = G_{\min(k-1,N)\mu/2}. \tag{5.9}$$

There are $k-1$ nodes in the system that contain equal amounts of the replicas corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period R_1 will cause the system to enter exposure level 2. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_1 = k-1. \tag{5.10}$$

When one of the $\tilde{n}_1$ nodes fail before rebuild, the system enters exposure level 2.

Exposure Level 2

The system enters exposure level 2 from exposure level 1 because one of the $\tilde{n}_1$ nodes fails during the rebuild period R_1. Consider an instance of the rebuild period,

$$R_1 = \tau_1, \tag{5.11}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from 1 to 2 occurred,

$$\alpha_1 = a_1. \tag{5.12}$$

The remaining time to complete rebuild at exposure level 1 when the system entered exposure level 2 is the product of R_1 and α_1, namely, $a_1\tau_1$. As the average speed of rebuild in exposure is S_1, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_1(t_2^-)$, is given by

$$D_1(t_2^-) = \alpha_1 R_1 S_1 = \frac{1}{2}\min(k-1, N)c\mu a_1\tau_1, \tag{5.13}$$

which is essentially the product of (5.7), (5.11), and (5.12). At the time of transition from exposure level 1 to 2, t_2, *not* all of this $D_1(t_2^-)$ amount of data loses a second copy. As described in Section 2.4, due to the nature of the symmetric placement scheme, the two failed nodes share copies of only a fraction $\frac{r-1}{k-1}$ of this data. So during the exposure level transition, only $\frac{r-1}{k-1}D_1(t_2^-)$ amount of data loses a second copy. Therefore, the amount of most-exposed data in exposure level 2, $D_2(t_2)$, is given by

$$D_2(t_2) = \frac{r-1}{k-1}D_1(t_2^-) = \frac{1}{2}\frac{r-1}{k-1}\min(k-1,N)c\mu a_1\tau_1. \tag{5.14}$$

By the nature of the symmetric placement scheme with spread factor k, the $r-2$ remaining replicas of the most-exposed data are spread equally across $k-2$ surviving nodes of the system. The distributed rebuild process involves reading these replicas from these nodes and copying to the spare space of these nodes in such a way that no data is copied to a node in which its replica is already present. As this distributed rebuild process involves $k-2$ nodes, the average rate of rebuild in exposure level 2 is given by (5.5) for $\tilde{k} = k-2$, that is,

$$S_2 = S_{\text{eff}}(k-2) = \frac{1}{2}\min(k-2,N)c\mu. \tag{5.15}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_2$, of the rebuild time in the second exposure level, R_2, is obtained by dividing (5.14) by (5.15), that is,

$$\frac{1}{\mu_2} = E[R_2|R_1 = \tau_1, \alpha_1 = a_1] = \frac{D_2(t_2)}{S_2} = \frac{r-1}{k-1}\frac{\min(k-1,N)}{\min(k-2,N)}a_1\tau_1. \tag{5.16}$$

There are now $k-3$ nodes in the system that contain equal amounts of the replicas corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period R_2 will cause the system to enter exposure level 3. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_2 = k-3. \tag{5.17}$$

Exposure Level e

The computation of the conditional mean $1/\mu_e$ and the number of nodes $\tilde{n}_e$ for a general exposure level $e = 2, \cdots, r-1$ is similar to the computation of these quantities for exposure level 2 as described above. Firstly, we note that the distributed rebuild process in each exposure level e always involves reading replicas of the data to be rebuilt from a set of $k-e$ surviving nodes and copying it to the spare spaces of these nodes in such a way that no data is copied to a node in which its replica is already present. As this distributed

rebuild process involves $k-e$ nodes, the average rate of rebuild in exposure level e is given by (5.5) for $\tilde{k} = k - e$, that is,

$$S_e = S_{\text{eff}}(k-e) = \frac{1}{2}\min(k-e, N)c\mu, \quad e = 1, \cdots, r-1. \tag{5.18}$$

Now, the system enters exposure level e from exposure level $e-1$ because one of the $\tilde{n}_{e-1}$ nodes fails during the rebuild period R_{e-1}. Consider an instance of the rebuild period,

$$R_{e-1} = \tau_{e-1}, \tag{5.19}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from $e-1$ to e occurred,

$$\alpha_{e-1} = a_{e-1}. \tag{5.20}$$

The remaining time to complete rebuild at exposure level $e-1$ when the system entered exposure level e is the product of R_{e-1} and α_{e-1}, namely, $a_{e-1}\tau_{e-1}$. As the average speed of rebuild in exposure is S_{e-1}, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_{e-1}(t_e^-)$, is given by

$$D_{e-1}(t_e^-) = \alpha_{e-1}R_{e-1}S_{e-1} = \frac{1}{2}\min(k-e+1, N)c\mu a_{e-1}\tau_{e-1}, \tag{5.21}$$

which is essentially the product of (5.18), (5.19), and (5.20). At the time of transition from exposure level $e-1$ to e, t_{e-1}, *not* all of this $D_1(t_2^-)$ amount of data loses its eth copy. Due to the nature of the symmetric placement scheme with spread factor k, the newly failed nodes shares copies of only a fraction $\frac{r-e+1}{k-e+1}$ of this data. So during the exposure level transition, only $\frac{r-e+1}{k-e+1}D_1(t_2^-)$ amount of data loses its eth copy. Therefore, the amount of most-exposed data in exposure level e, $D_e(t_e)$, is given by

$$D_e(t_e) = \frac{r-e+1}{k-e+1}D_e(t_e^-) = \frac{1}{2}\frac{r-e+1}{k-e+1}\min(k-e+1, N)c\mu a_{e-1}\tau_{e-1}. \tag{5.22}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_e$, of the rebuild time in the eth exposure level, R_e, is obtained by dividing (5.22) by (5.18), that is,

$$\begin{aligned}
\frac{1}{\mu_e} &= E[R_e | R_{e-1} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}] && (5.23)\\
&= \frac{D_e(t_e)}{S_e} && (5.24)\\
&= \frac{r-e+1}{k-e+1}\frac{\min(k-e+1, N)}{\min(k-e, N)}a_{e-1}\tau_{e-1}. && (5.25)
\end{aligned}$$

There are now $k-e$ nodes in the system that contain equal amounts of the replicas corresponding to the most-exposed data. So, the failure of any of these

nodes during the rebuild period R_e will cause the system to enter exposure level $e+1$. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_e = k - e. \tag{5.26}$$

MTTDL under Model A

Recall that, under model A, following each exposure level transition, the system is assumed to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where the node from which data was being rebuilt failed and hence the system has to rebuild from another node in the cluster. This model may also be applicable for symmetric placement schemes with a small spread factor k.

Having computed the key quantities $1/\mu_e$ and $\tilde{n}_e$ for $e = 1, \cdots, r-1$, we are now ready to compute P_{DL} using the expression (4.44) for model A, and then MTTDL using (3.14). By substituting the values of $1/\mu_e$ and $\tilde{n}_e$ from (5.8), (5.25), and (5.26) into (4.44), we obtain

$$\begin{aligned} P_{DL}(k) \;\approx\; \lambda^{r-1} \times (k-1)\cdots(k-r+1) \times \int_{\tau_1}\cdots\int_{\tau_{r-1}} \int_{a_1}\cdots\int_{a_{r-2}} \Bigg(\tau_1 \cdots \tau_{r-1} \\ \times g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \cdots g_{\frac{\min(k-r+1,N)}{\min(k-r+2,N)} \frac{k-r+2}{2a_{r-2}\tau_{r-2}}}(\tau_{r-1}) d\vec{a} d\vec{\tau} \Bigg) \\ \text{for } k = r+1, \cdots, n \text{ (model A).} \end{aligned} \tag{5.27}$$

As in (4.44) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$.

Replication factors $r \leq 3$: The expression (5.27) for P_{DL} under model A cannot, in general, be further simplified without considering a particular family of rebuild distributions G_μ. However, for $r \leq 3$, a closed form expression for P_{DL}, and hence MTTDL, can be obtained under model A. This is illustrated by deriving the closed form expression for $r = 3$ by substituting $r = 3$ in (5.27) and simplifying as follows.

$$\begin{aligned} P_{DL}(k) \approx \lambda^2 \times (k-1)(k-2) \times \int_{\tau_1=0}^{\infty} \int_{\tau_2=0}^{\infty} \int_{a_1=0}^{1} \Bigg(\tau_1 \tau_2 \\ \times g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) g_{\frac{\min(k-2,N)}{\min(k-1,N)} \frac{k-1}{2a_1\tau_1}}(\tau_2) da_1 d\tau_2 d\tau_1 \Bigg) \end{aligned} \tag{5.28}$$

$$= (k-1)(k-2)\lambda^2 \Bigg(\int_{\tau_1=0}^{\infty} \tau_1 g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \int_{a_1=0}^{1} \int_{\tau_2=0}^{\infty} \Big(\tau_2 \times g_{\frac{\min(k-2,N)}{\min(k-1,N)} \frac{k-1}{2a_1\tau_1}}(\tau_2)\Big) d\tau_2 da_1 d\tau_1 \Bigg)$$
$$\text{for } r = 3 \text{ and } k = 4, \cdots, n \text{ (model A)}. \tag{5.29}$$

Noting that

$$\int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{\min(k-2,N)}{\min(k-1,N)} \frac{k-1}{2a_1\tau_1}}(\tau_2) d\tau_2 = \frac{\min(k-1,N)}{\min(k-2,N)} \frac{2a_1\tau_1}{k-1}, \tag{5.30}$$

we get

$$P_{DL}(k) \approx 2(k-2) \frac{\min(k-1,N)}{\min(k-2,N)} \lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \int_{a_1=0}^{1} a_1 da_1 d\tau_1 \tag{5.31}$$

$$= (k-2) \frac{\min(k-1,N)}{\min(k-2,N)} \lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) d\tau_1 \tag{5.32}$$

$$= (k-2) \frac{\min(k-1,N)}{\min(k-2,N)} \lambda^2 M_2(G_{\min(k-1,N)\mu/2})$$
$$\text{for } r = 3 \text{ and } k = 4, \cdots, n \text{ (model A)}, \tag{5.33}$$

where $M_2(G_{\min(k-1,N)\mu/2})$, as defined in (4.46), denotes the second raw moment of the rebuild distribution $G_{\min(k-1,N)\mu/2}$. The expression for MTTDL then follows from (3.14):

$$\text{MTTDL}(k) \approx \frac{1}{n\lambda P_{DL}} \tag{5.34}$$

$$\approx \frac{\min(k-2,N)}{n(k-2)\min(k-1,N)\lambda^3 M_2(G_{\min(k-1,N)\mu/2})}$$
$$\text{for } r = 3 \text{ and } k = 4, \cdots, n \text{ (model A)}. \tag{5.35}$$

Multipliying and dividing (5.35) by square of the mean of the rebuild time distribution $G_{\min(k-1,N)\mu/2}$,

$$M_1^2(G_{\min(k-1,N)\mu/2}) = \frac{1}{(\min(k-1,N)\mu/2)^2} \tag{5.36}$$

we get

$$\text{MTTDL}(k) \approx \frac{\mu^2}{4n\lambda^3} \frac{\min(k-2,N)\min(k-1,N)}{k-2} \frac{M_1^2(G_{\min(k-1,N)\mu/2})}{M_2(G_{\min(k-1,N)\mu/2})}$$
$$\text{for } r = 3 \text{ and } k = 4, \cdots, n \text{ (model A)}. \tag{5.37}$$

For deterministic rebuild times, the second raw moment, $M_2(G_{\min(k-1,N)\mu/2})$, is equal to the square of the first raw moment, $M_1^2(G_{\min(k-1,N)\mu/2})$, and therefore,

the term $M_1^2(G_{\min(k-1,N)\mu/2})/M_2(G_{\min(k-1,N)\mu/2})$ evaluates to one. However, if the rebuild times are random, the second raw moment is always greater than the square of the first raw moment by Jensen's inequality, and therefore, the term $M_1^2(G_{\min(k-1,N)\mu/2})/M_2(G_{\min(k-1,N)\mu/2})$ is smaller than one. The closed form expression for $r = 2$ can be derived similarly and is given by

$$\text{MTTDL}(k) \approx \frac{\mu}{2n\lambda^2}\frac{\min(k-1,N)}{k-1} \quad \text{for } r = 2 \text{ and } k = 3, \cdots, n \text{ (model A).} \quad (5.38)$$

Replication factors $r > 3$: For replication factors $r > 3$, the evaluation of P_{Dl} under model A involves computing the expectations of functions involving higher raw moments of G_μ, which cannot be done without considering a particular family of rebuild distributions. However, given a particular family of rebuild distributions, the derivation of MTTDL involves successively evaluating the integrals in (5.27) to compute $P_{DL}(k)$, and then using (3.14) to obtain MTTDL(k).

Declustered Replica Placement: MTTDL under Model B

In contrast to model A, we assume in model B that, following an exposure level transition, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the case in a symmetric placement scheme with a large spread factor k, where the rebuild was being done from several nodes, and therefore, the failure of one node does not significantly affect the rebuild process. This implies that the rebuild time in the new exposure level is completely determined by the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level.

By substituting the values of $1/\mu_e$ and $\tilde{n}_e$ from (5.8), (5.25), and (5.26) into (4.45), we obtain

$$\begin{aligned} P_{DL}(k) \approx{}& \lambda^{r-1} \times (k-1)\cdots(k-r+1) \\ &\times \int_{\tau_1}\cdots\int_{\tau_{r-1}}\int_{a_1}\cdots\int_{a_{r-2}} \Bigg(\tau_1\cdots\tau_{r-1} g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \\ &\quad \times \delta\left(\tau_2 - \frac{r-1}{k-1}\frac{\min(k-1,N)}{\min(k-2,N)}a_1\tau_1\right) \\ &\quad \cdots \delta\left(\tau_{r-1} - \frac{2}{k-r+2}\frac{\min(k-r+2,N)}{\min(k-r+1,N)}a_{r-2}\tau_{r-2}\right) d\vec{a}d\vec{\tau}\Bigg) \\ &\text{for } k = r+1, \cdots, n \text{ (model B).} \end{aligned} \quad (5.39)$$

As in (4.45) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, r-1$, and from 0 to 1 for a_e, $e = 1, \cdots, r-2$. In contrast to model A, closed form expressions

in terms of the raw moments of the rebuild distribution can be obtained for model B as follows. By changing the order of integrals in (5.39), we obtain

$$
\begin{aligned}
P_{DL}(k) \approx{}& \lambda^{r-1} \times (k-1)\cdots(k-r+1) \\
&\times \int_{a_1}\cdots\int_{a_{r-2}} \int_{\tau_1}\cdots\int_{\tau_{r-1}} \Bigg(\tau_1 \cdots \tau_{r-1} g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \\
&\times \delta\left(\tau_2 - \frac{r-1}{k-1}\frac{\min(k-1,N)}{\min(k-2,N)} a_1\tau_1\right) \\
&\cdots \delta\left(\tau_{r-1} - \frac{2}{k-r+2}\frac{\min(k-r+2,N)}{\min(k-r+1,N)} a_{r-2}\tau_{r-2}\right) \\
&\times d\tau_{r-1}\cdots d\tau_1 d\vec{a}\Bigg) \qquad (5.40) \\
={}& \lambda^{r-1} \times (k-1)\cdots(k-r+1) \times \frac{2}{k-r+2}\frac{\min(k-r+2,N)}{\min(k-r+1,N)} \\
&\times \int_{a_1}\cdots\int_{a_{r-2}} \int_{\tau_1}\cdots\int_{\tau_{r-2}} \Bigg(\tau_1 \cdots \tau_{r-2}^2 a_{r-2} g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \\
&\times \delta\left(\tau_2 - \frac{r-1}{k-1}\frac{\min(k-1,N)}{\min(k-2,N)} a_1\tau_1\right) \\
&\cdots \delta\left(\tau_{r-2} - \frac{3}{k-r+3}\frac{\min(k-r+3,N)}{\min(k-r+2,N)} a_{r-3}\tau_{r-3}\right) \\
&\times d\tau_{r-2}\cdots d\tau_1 d\vec{a}\Bigg) \qquad (5.41) \\
={}& \lambda^{r-1} \times (k-1)\cdots(k-r+1) \times \frac{2}{k-r+2}\frac{\min(k-r+2,N)}{\min(k-r+1,N)} \\
&\times \left(\frac{3}{k-r+3}\frac{\min(k-r+3,N)}{\min(k-r+2,N)}\right)^2 \\
&\times \int_{a_1}\cdots\int_{a_{r-2}} \int_{\tau_1}\cdots\int_{\tau_{r-3}} \Bigg(\tau_1 \cdots \tau_{r-3}^3 a_{r-3}^2 a_{r-2} g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) \\
&\times \delta\left(\tau_2 - \frac{r-1}{k-1}\frac{\min(k-1,N)}{\min(k-2,N)} a_1\tau_1\right) \\
&\cdots \delta\left(\tau_{r-3} - \frac{4}{k-r+4}\frac{\min(k-r+4,N)}{\min(k-r+3,N)} a_{r-4}\tau_{r-4}\right) \\
&\times d\tau_{r-3}\cdots d\tau_1 d\vec{a}\Bigg) \qquad (5.42) \\
&\vdots
\end{aligned}
$$

$$= \lambda^{r-1} \times (\min(k-1,N))^{r-1} \times \prod_{e=1}^{r-2} \left(\frac{r-e}{k-e}\right)^{r-e-1} \times \prod_{e'=1}^{r-1} \frac{k-e'}{\min(k-e',N)}$$
$$\times \int_{a_1} \cdots \int_{a_{r-2}} \int_{\tau_1} \tau_1^{r-1} a_1^{r-2} \cdots a_{r-3}^2 a_{r-2} g_{\frac{\min(k-1,N)\mu}{2}}(\tau_1) d\tau_1 d\vec{a}$$
$$\text{for } k = r+1, \cdots, n \text{ (model B).} \qquad (5.43)$$

Here, steps (5.41)–(5.43) follow by successively integrating over $\tau_{r-1}, \cdots, \tau_2$, using the Dirac delta function's property, canceling out terms of the form $\min(k-r+e, N)$, $e = 1, \cdots, r-2$, and rewriting the terms outside the integral. Changing the order of the integrals and integrating out $a_1, \cdots, a_{r-2}$, we get

$$P_{DL}(k) \approx \lambda^{r-1} \times (\min(k-1,N))^{r-1} \times \prod_{e=1}^{r-2} \left(\frac{r-e}{k-e}\right)^{r-e-1} \times \prod_{e'=1}^{r-1} \frac{k-e'}{\min(k-e',N)}$$
$$\times \int_{\tau_1} \tau_1^{r-1} g_{\frac{(n-1)\mu}{2}}(\tau_1) \frac{1}{(r-1)!} d\tau_1$$
$$= \lambda^{r-1} \times \frac{(\min(k-1,N))^{r-1}}{(r-1)!} \times \prod_{e=1}^{r-2} \left(\frac{r-e}{k-e}\right)^{r-e-1} \times \prod_{e'=1}^{r-1} \frac{k-e'}{\min(k-e',N)}$$
$$\times M_{r-1}(G_{\min(k-1,N)\mu/2})$$
$$\text{for } k = r+1, \cdots, n \text{ (model B),} \qquad (5.44)$$

where $M_{r-1}(G_{\min(k-1,N)\mu/2})$, as defined in (4.46), denotes the $(r-1)$th raw moment of the rebuild distribution $G_{\min(k-1,N)\mu/2}$. The expression for MTTDL then follows from (3.14):

$$\text{MTTDL}(k) \approx \frac{1}{n\lambda P_{DL}} \qquad (5.45)$$
$$\approx \frac{1}{n\lambda^r M_{r-1}(G_{\min(k-1,N)\mu/2})} \times \frac{(r-1)!}{(\min(k-1,N))^{r-1}}$$
$$\times \prod_{e=1}^{r-2} \left(\frac{k-e}{r-e}\right)^{r-e-1} \times \prod_{e'=1}^{r-1} \frac{\min(k-e',N)}{k-e'}$$
$$\text{for } k = r+1, \cdots, n \text{ (model B).} \qquad (5.46)$$

Multiplying and dividing (5.46) by the $(r-1)$th power of the mean of the rebuild time distribution $G_{\min(k-1,N)\mu/2}$,

$$M_1^{r-1}(G_{\min(k-1,N)\mu/2}) = \frac{1}{(\min(k-1,N)\mu/2)^{r-1}} \qquad (5.47)$$

we get

$$\text{MTTDL}(k) \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{M_1^{r-1}(G_{\min(k-1,N)\mu/2})}{M_{r-1}(G_{\min(k-1,N)\mu/2})} \frac{(r-1)!}{2^{r-1}} \prod_{e=1}^{r-2} \left(\frac{k-e}{r-e}\right)^{r-e-1} \times \prod_{e'=1}^{r-1} \frac{\min(k-e', N)}{k-e'}$$
$$\text{for } k = r+1, \cdots, n \text{ (model B).} \quad (5.48)$$

For deterministic rebuild times, the $(r-1)$th raw moment, $M_{r-1}(G_{\min(k-1,N)\mu/2})$, is equal to the $(r-1)$th power of the first raw moment, $M_1^{r-1}(G_{\min(k-1,N)\mu/2})$, and therefore, the term $M_1^{r-1}(G_{\min(k-1,N)\mu/2})/M_{r-1}(G_{\min(k-1,N)\mu/2})$ evaluates to one. For random rebuild times, by the Jensen's inequality, the $(r-1)$th raw moment, $M_{r-1}(G_{\min(k-1,N)\mu/2})$, is always greater than the $(r-1)$th power of the first raw moment, $M_1^{r-1}(G_{\min(k-1,N)\mu/2})$, and therefore, their ratio evaluates to less than one.

As an example, if $G_{\min(k-1,N)\mu/2}$ is exponential, the expression for MTTDL reduces to the following:

$$\text{MTTDL}(k) \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{1}{2^{r-1}} \prod_{e=1}^{r-2} \left(\frac{k-e}{r-e}\right)^{r-e-1} \times \prod_{e'=1}^{r-1} \frac{\min(k-e', N)}{k-e'}$$
$$\text{for } k = r+1, \cdots, n, \text{ when } G_{\min(k-1,N)\mu/2} \text{ is exponential (model B).} \quad (5.49)$$

5.3 MTTDL vs. Network Rebuild Bandwidth

In this section, we study the MTTDL of storage systems under limited network rebuild bandwidth. Note that the MTTDL values under models A and B, as given by (5.37), (5.38), and (5.48), do not differ for $r \leq 3$. Furthermore, the difference between models A and B for $r > 3$ is typically only a constant factor that depends on the rebuild distribution. If the rebuild times are deterministic, there is no difference between models A and B, and therefore, they agree on the MTTDL values for all replication factors. So, without loss of generality, we only consider the MTTDL values under model B for further discussion in this section. Similarly, the effect of rebuild distribution under model B is seen to be a constant factor that depends on the rebuild distribution. Therefore, without loss of generality, we compare different placement schemes only under deterministic rebuilds, for which the ratio of the raw moments evaluates to one.

For comparison, we consider four different symmetric placement schemes defined by their spread factors. Firstly, we consider the clustered placement scheme, whose spread factor is $k = r$. Next, we consider the declustered placement scheme, whose spread factor scales with the number of nodes as $k = n$. It can be seen from (5.48) that, for constant spread factors that do not scale with n, the behavior of MTTDL with respect to n is similar to clustered

placement. Therefore, we consider two other placement schemes whose spread factors depend on the system size n: one with spread factor $k = \lfloor\sqrt{n}\rfloor$, and another with spread factor $k = \min(n, N)$.

The expression for MTTDL in (5.48) can be broken down as follows to understand the effect of limited network rebuild bandwidth. When there is sufficient network rebuild bandwidth for the distributed rebuild process to take place at full speed, that is, when the spread factor $k \leq N + 1$, the terms of the form $\min(k - e', N)$ become equal to $k - e'$ for $e' = 1, \cdots, r - 1$, and the second product in expression (5.48) for MTTDL becomes equal to one. For $k = n$, that is, for declustered placement, and for $k \leq N + 1$, expression (5.48) is the same as expression (4.127) in Chapter 4. On the other hand, when the spread factor $k \geq N + r - 1$, the network rebuild bandwidth is insufficient for a distributed rebuild process at full speed and therefore the system reliability is affected negatively. This can be seen from the fact that the second product in expression (5.48) for MTTDL becomes smaller than one and roughly scales as $k^{-(r-1)}$.

Now, if we denote the MTTDL under a network rebuild bandwidth constraint N by MTTDL_N and the MTTDL under no network bandwidth constraint, that is, $N = \infty$, by MTTDL_∞, then it follows from (5.48) that

$$\text{MTTDL}_N(k) = \text{MTTDL}_\infty(k) \times \prod_{e'=1}^{r-1} \frac{\min(k - e', N)}{k - e'}, \tag{5.50}$$

where $\text{MTTDL}_\infty(k)$ is given by

$$\text{MTTDL}_\infty(k) \approx \frac{\mu^{r-1}}{n\lambda^r} \frac{(r-1)!}{2^{r-1}} \prod_{e=1}^{r-2} \left(\frac{k-e}{r-e}\right)^{r-e-1}. \tag{5.51}$$

5.3.1 Replication Factor 2

For declustered placement, that is, for $k = n$, the expression for MTTDL (5.48) under deterministic rebuilds reduces to

$$\text{MTTDL}^{\text{declus.}} \approx \begin{cases} \dfrac{\mu}{2n\lambda^2} & \text{when } n \leq N + 1 \\ \dfrac{\mu N}{2n(n-1)\lambda^2} & \text{when } n \geq N + 1. \end{cases} \tag{5.52}$$

The above expressions show that, when the network rebuild bandwidth is not sufficient to perform distributed rebuild process at full speed, the MTTDL becomes inversely proportional to the *square* of the number of nodes instead of being inversely proportional to the number of nodes. This drastic change in the MTTDL behavior as the system scales is shown in Figure 5.1. The figure shows the plots of MTTDL as a function of the number of nodes for

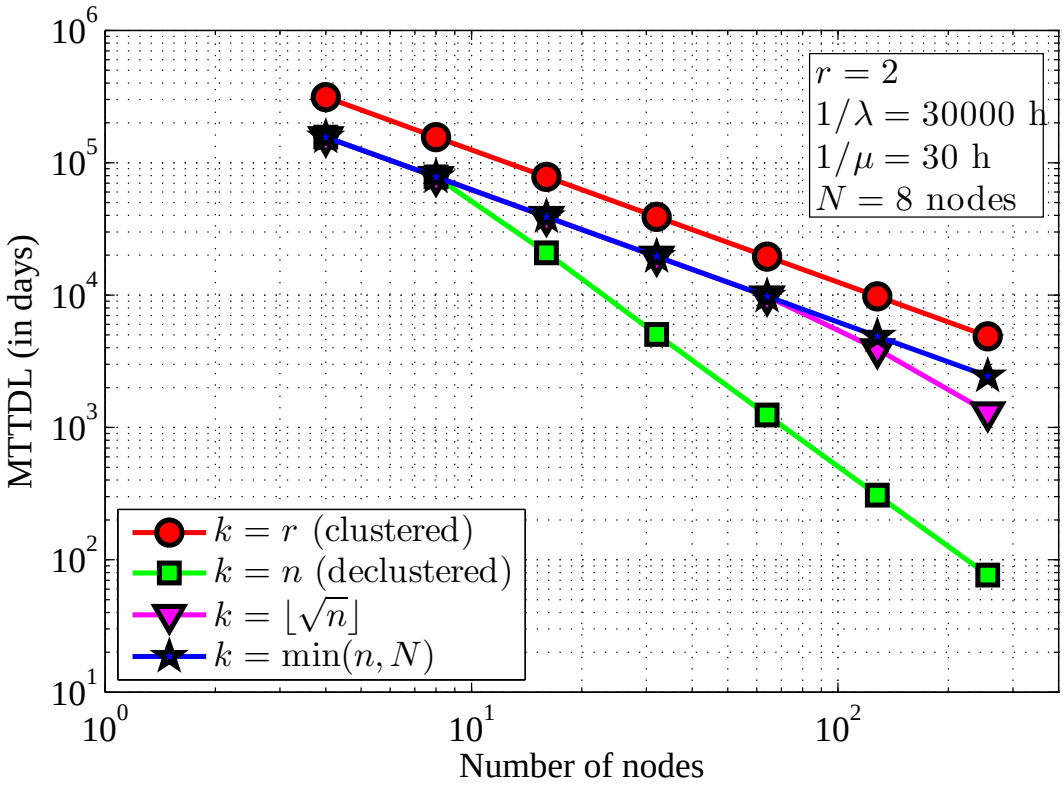


Figure 5.1: MTTDL as a function of the number of nodes for replication factor two when the network rebuild bandwidth can support only up to $N = 8$ nodes at full speed during distributed rebuild.

the four different placement schemes considered. When the network rebuild bandwidth can support only up to $N = 8$ nodes at full speed during distributed rebuild, it is seen that the MTTDL of declustered placement drops significantly compared to other placement schemes which are not affected (because their spread factors are less than N). For a scheme whose spread factor varies as $\lfloor\sqrt{n}\rfloor$, the change in MTTDL behavior is seen around $n = N^2 = 64$ nodes.

5.3.2 Replication Factor 3

For declustered placement, the expression for MTTDL (5.48) under deterministic rebuilds reduces to

$$\text{MTTDL}^{\text{declus.}} \approx \begin{cases} \dfrac{\mu^2(n-1)}{4n\lambda^3} & \text{when } n \leq N+1 \\ \dfrac{\mu^2 N^2}{4n(n-2)\lambda^3} & \text{when } n \geq N+2. \end{cases} \tag{5.53}$$

The change in the MTTDL behavior due to limited network rebuild bandwidth is greater than that observed for replication factor two; it goes from being

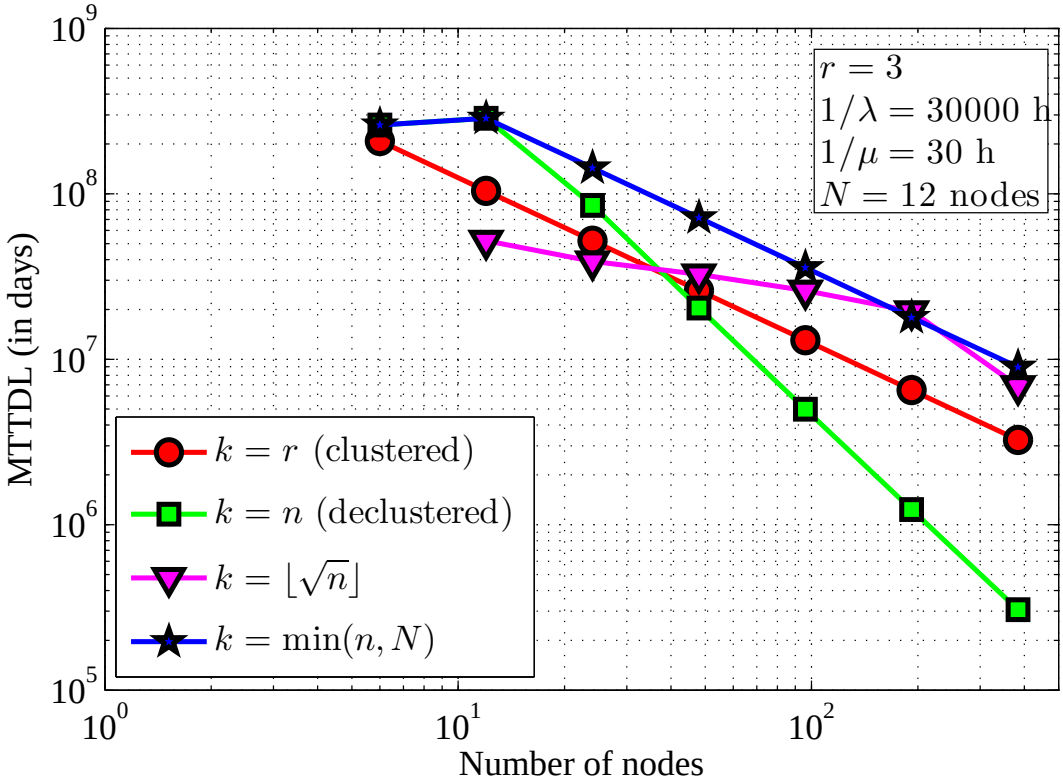


Figure 5.2: MTTDL as a function of the number of nodes for replication factor three when the network rebuild bandwidth can support only up to $N = 12$ nodes at full speed during distributed rebuild.

constant with respect to the number of nodes when network rebuild bandwidth is sufficient, to being inversely proportional to the square of the number of nodes when the network rebuild bandwidth is limited. This is also shown in Fig. 5.2. Interestingly, for $r = 3$, limiting the spread factor to N, that is, setting $k = \min(n, N)$, can achieve much higher MTTDL than the declustered placement scheme for $n \geq N + 2$.

5.3.3 Replication Factor 4

For declustered placement, the expression for MTTDL (5.48) reduces to

$$\text{MTTDL}^{\text{declus.}} \approx \begin{cases} \dfrac{\mu^3(n-1)^2(n-2)}{24n\lambda^4} & \text{when } n \leq N+1 \\ \dfrac{\mu^3 N^2(N+1)}{24(N+2)\lambda^4} & \text{when } n = N+2 \\ \dfrac{\mu^3(n-1)N^3}{24n(n-3)\lambda^4} & \text{when } n \geq N+3. \end{cases} \tag{5.54}$$

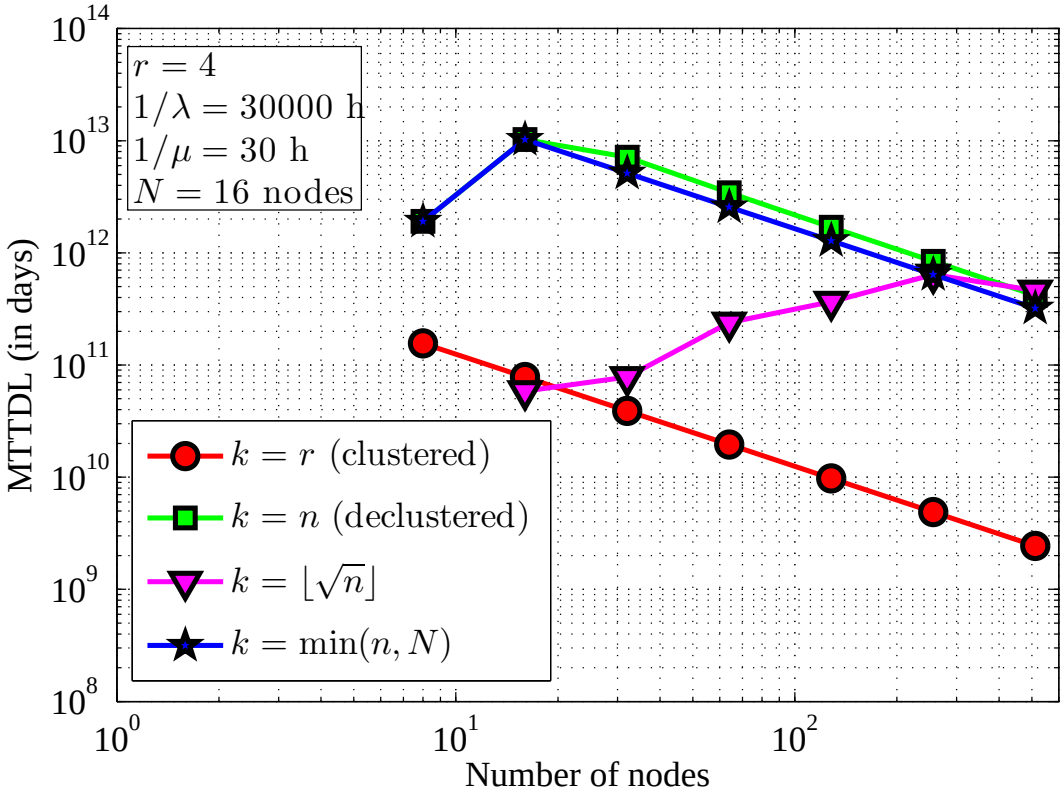


Figure 5.3: MTTDL as a function of the number of nodes for replication factor two when the network rebuild bandwidth can support only up to $N = 16$ nodes at full speed during distributed rebuild.

The above expressions are plotted in Fig. 5.3. For replication factor 4, the MTTDL values of a scheme that limits the spread factor to N, that is, $k = \min(n, N)$, is comparable to the MTTDL values of the declustered scheme for which $k = n$. This is because, although the limited network bandwidth slows down rebuilds in a declustered placement scheme, the amount of most-exposed data to be rebuilt as the system goes to higher exposure levels also decreases. It appears that, for declustered placement, the negative influence of limited network bandwidth is effectively countered by the positive influence of decreasing amounts of critical data as additional nodes fail.

5.3.4 Replication Factor 5

For declustered placement, the expression for MTTDL (5.48) under deterministic rebuilds reduces to

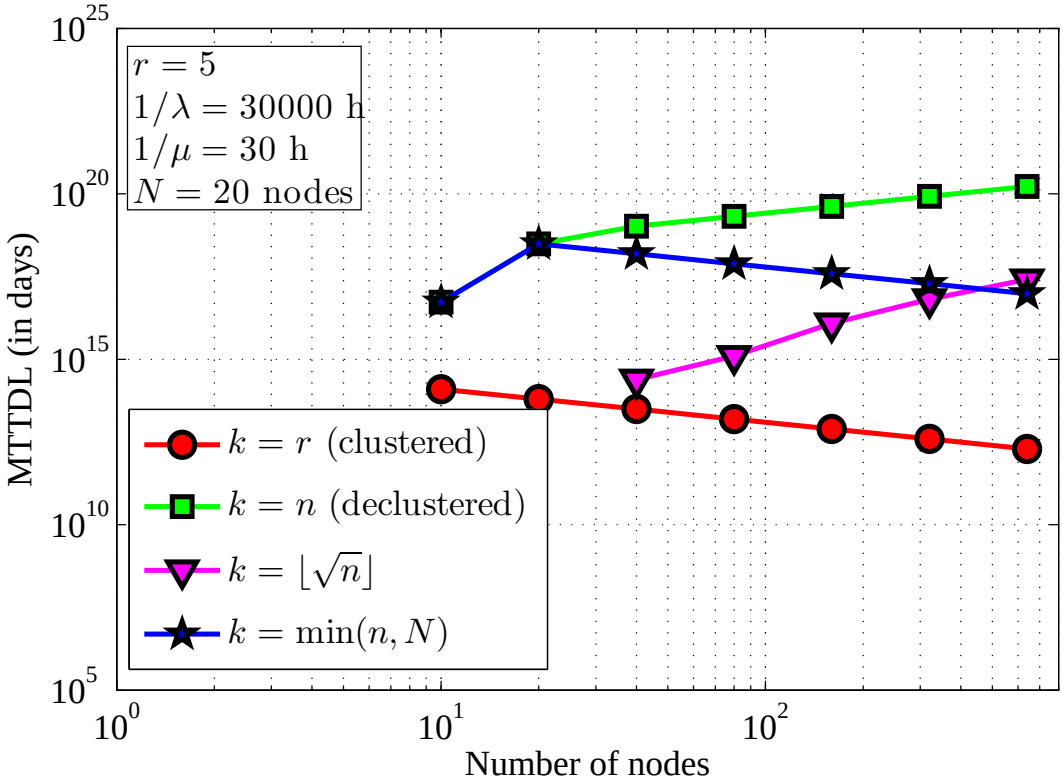


Figure 5.4: MTTDL as a function of the number of nodes for replication factor two when the network rebuild bandwidth can support only up to $N = 20$ nodes at full speed during distributed rebuild.

$$\text{MTTDL}^{\text{declus.}} \approx \begin{cases} \dfrac{\mu^4(n-1)^3(n-2)^2(n-3)}{768n\lambda^5} & \text{when } n \leq N+1 \\ \dfrac{\mu^4(N+1)^2N^3(N-1)}{768(N+2)\lambda^5} & \text{when } n = N+2 \\ \dfrac{\mu^4(N+2)^2(N+1)N^3}{768(N+3)\lambda^5} & \text{when } n = N+3 \\ \dfrac{\mu^4(n-1)^2(n-2)N^4}{768n(n-4)\lambda^5} & \text{when } n \geq N+4. \end{cases} \tag{5.55}$$

The above expressions are plotted in Fig. 5.4. The positive effect of decreasing amounts of critical data as additional nodes fail is stronger than that observed for replication factor four.

5.4 Optimal Data Placement for High Reliability

Using expression (5.48) for MTTDL, one can find the optimal value of the spread factor k for which the corresponding MTTDL is maximized. Clearly, the optimal spread factor depends on the number of nodes n and the maximum network rebuild bandwidth $B_{\max}$. In a dynamically changing storage system, the number of nodes and the available network rebuild bandwidth $B_{\max}$ may change over time. As a result, the optimal spread factor may change as well. In this case, one could consider redistributing the data in accordance to the new optimal spread factor. Such a scheme ensures that the system reliability constantly remains at a high level.

6 Erasure Coded Systems

As an alternative to replication, modern data storage systems employ advanced erasure codes to protect data from storage node failures because of their ability to provide high data reliability as well as high storage efficiency. The use of such erasure codes can be dated back to as early as the 1980s when they were applied for building systems with redundant arrays of inexpensive disks (RAID) [16]. When nodes fail, storage systems try to maintain the redundancy through node rebuild processes that use the data from the surviving nodes to reconstruct the lost data in new replacement nodes. As these rebuild processes take a finite amount of time, there exists a non-zero probability of further node failures during rebuild that can cause the system to lose enough redundant data to render some of the originally stored data irrecoverable. The average amount of time taken by the system to end up in irrecoverable data loss, also known as the mean time to data loss, or MTTDL, is a measure of reliability commonly used to compare different coding schemes and study the effect of various design parameters. As, for performance reasons, the length of codewords in an erasure coded system is typically much smaller than the number of storage nodes in the system, there exist a large number of possible ways in which codewords can be stored across the nodes of the system. However, most reliability analyses in the literature are performed under the assumption that the number of storage nodes is equal to the codeword length. In addition, some of the reliability analyses assume a constant probability of additional node failures during rebuild. For replication-based systems, it is well-known that the MTTDL is significantly affected by the choice of placement of replicas. In particular, it was shown in Chapter 4 that the declustered replica placement scheme can provide significantly higher reliability than clustered placement, especially for large storage systems. In this chapter, these results are extended to erasure coded systems and it is shown that a declus-

tered placement of codewords can significantly improve the system reliability. Just as in the case of replication-based systems, the reliability analysis of erasure coded systems in this chapter is detailed, in the sense that it accounts for the rebuild times involved, the amounts of partially rebuilt data when additional nodes fail during rebuild, and the fact that most modern systems utilize an intelligent rebuild process by rebuilding the most critical codewords first.

6.1 Erasure Codes in Data Storage

In data storage systems, an erasure code, in the most general sense, is a mapping from a set of l user data blocks (or symbols) to a set of $m > l$ blocks, called a codeword, in such as way that some subset of the m blocks of the codeword can be used to decode the l user data blocks. Optimal erasure codes or *maximum distance separable* codes (MDS codes) have the property that *any* l out of m symbols can be used to decode a codeword. Such an erasure code is referred to as an (l, m)-*MDS code*. Although the techniques developed here may be applicable to general non-MDS erasure codes as well, we will restrict our reliability analysis to MDS codes in this thesis.

Typically, the advantage of an erasure coded system over a replication-based system is that it can offer much better reliability for the same storage efficiency, or much higher storage efficiency for the same reliability. The advantage of a replication-based system over an erasure coded system is in performance. Erasure coded systems typically offer only one copy of the user data whereas replication-based systems offer r copies. Furthermore, an update of any block of user data in an erasure coded system will require reading of the existing codeword corresponding to that block, updating that codeword and writing the codeword back to the system. In contrast, in a replication-based system, an update of any piece of user data just requires overwriting the existing replicas of that data in the system and does not require any additional reads or processing.

6.2 Codeword Reconstruction

When storage nodes fail, codewords lose some of their symbols and this leads to a reduction in data redundancy. The system attempts to maintain the redundancy of the system by reconstructing the lost codeword symbols using the surviving symbols of the affected codewords. For a system using an (l, m)-MDS code for redundancy, a simple way to reconstruct a codeword that has lost up to $m - l$ symbols is to read any of its l symbols, decode the original l user data blocks, re-encode these l user data blocks using the (l, m)-MDS code, and recover the lost codeword symbols. As an alternative to this method of reconstruction, other methods, based on regenerating codes, have been proposed as a solution to minimize the amount of data *transferred* over the storage network

during reconstruction [19, 20]. Although regenerating codes help in reducing the amount of network reconstruction traffic, the proposed reconstruction process relies on being able to read considerably larger amounts of data from each node and processing that data before transferring it to other nodes over the network. When sufficient network rebuild bandwidth is available, the speed of this reconstruction process is determined completely by the amount of data to be read from each node. From a data reliability point of view, reading larger amounts of data and the resulting slower reconstruction process can significantly degrade the reliability of the system. Therefore, in this thesis, we only consider the simple way of reconstruction, that is, reading any l symbols of a codeword, decoding the original l user data blocks, re-encoding these l user data blocks using the (l, m)-MDS code, and recovering the lost codeword symbols.

The reconstruction process takes a finite amount of time, which depends on the amount of data to read, the time taken for decoding and re-encoding this data, and the amount of data to write. Typically the amount of time taken for decoding and re-encoding this data is much smaller than the time taken to read the required data and write the re-encoded data. It is assumed that the decoding and re-encoding of data is done in a streaming fashion, that is, as the data is being read, the decoding and re-encoding is assumed to be done on-the-fly which converts a stream of input data to a stream of output data. This implies that the time taken for the reconstruction is equal to the time taken to stream the input and output data. As this streaming time is a non-zero quantity, there exists a non-zero probability of further node failures within this reconstruction time period which may result in further loss of redundancy. Due to this probability, eventually, the system loses $m - l + 1$ symbols or more of some codewords, thereby rendering these codewords undecodeable and resulting in irrecoverable data loss. Similar to replication-based systems, we wish to analyze the data reliability of replication-based systems in terms of its mean time to data loss (MTTDL) and show how different codeword placement schemes and system parameters affect the system MTTDL. To do this, we make use of the relation (3.14) between MTTDL and the probability of data loss during rebuild, P_{DL}, namely,

$$\text{MTTDL} \approx \frac{1}{n\lambda P_{DL}},$$

which is a good approximation for real-world systems with generally reliable storage nodes (see Section 3.2). Note that this relation was derived without assuming any specific redundancy scheme or data placement. In addition, as noted in Remark 3.1, this is a reasonable approximation for real-world data storage systems.

Just like in the case of replication-based systems, the estimation of P_{DL} is a non-trivial problem as the system can go through a complex sequence of node failures and rebuilds during the rebuild mode. Therefore, we approximate P_{DL} by the probability of the *shortest path* to data loss in rebuild mode and show

that this approximation holds good for generally reliable nodes whose mean times to failure are much larger than their mean times to rebuild.

6.3 Estimation of the Probability of Data Loss during Rebuild

This section shows how the complex sequence of failure and rebuild events following a first-node failure, that is, a node failure that causes a transition of the system from fully-operational mode to the rebuild mode, is handled to be able to estimate the probability of data loss before all lost replicas are restored, namely, P_{DL}.

The general idea behind the estimation of P_{DL} is as follows. We model the reliability behavior of the system using *exposure levels* that range from zero to $m - l + 1$. Exposure level zero corresponds to a system where all codewords have all their symbols intact, whereas exposure level $m - l + 1$ corresponds to a system where some codewords have lost $m - l + 1$ symbols and are therefore undecodeable. In other words, the system starts at exposure level zero and eventually ends up in exposure level $m - l + 1$, which corresponds to irrecoverable data loss. Rebuild processes cause the system to go down exposure levels whereas node failures may, depending on the codeword placement, cause the system to go up exposure levels. The probability P_{DL} is then equivalent to the probability that, once the system enters exposure level one, the system ends up in exposure level $m - l + 1$ before returning to exposure level zero. It is extremely non-trivial to evaluate this probability as there are infinitely many complex paths through which the system can traverse these exposure levels. So we approximate this probability, P_{DL}, of all possible paths to data loss by the probability of the direct path to data loss, namely, the path from exposure level one to two, two to three, and so on until $m - l + 1$. We show that such an approximation holds good for systems with generally reliable nodes (that is, nodes whose mean times to failure are much larger than their mean times to rebuild) in the sense that the relative error in the approximation tends to zero as the ratio of the mean time to rebuild to the mean time to failure tends to zero. However, even the computation of the probability of this direct path is quite involved. This is because, the probability of transition from one exposure level to the next not only depends upon the current exposure level, but also on how the system arrived there in terms of how much critical data needs to be rebuilt. So we consider all possible sample direct paths from exposure level zero to $m - l + 1$, compute their probabilities, and sum them up. This gives the probability of direct path to data loss which is then used as a good approximation for P_{DL}.

6.3.1 Exposure Levels

Consider an erasure coded storage system with an (l, m)-MDS code. Let

$$\tilde{r} := m - l + 1. \tag{6.1}$$

We model the system as evolving from one exposure level to another as nodes fail and rebuilds complete. At time $t \geq 0$, let $D_j(t)$ be the amount of user data that have lost j symbols of their corresponding codewords, for $0 \leq j \leq \tilde{r}$. The system is said to be in exposure level e at time t, $0 \leq e \leq \tilde{r}$, if

$$e = \max_{D_j(t)>0} j. \tag{6.2}$$

In other words, the system is in exposure level e if there exists some data with only $\tilde{r} - e$ symbols of their corresponding codewords and no data with fewer than $\tilde{r} - e$ symbols of their corresponding codewords in the system, that is, $D_e(t) > 0$, and $D_j(t) = 0$ for all $j > e$. At $t = 0$, $D_j(0) = 0$ for all $j > 0$ and $D_0(0)$ is the total amount of user data stored in the system, which according to the parameters in Table 2.1, is equal to $nc/(m/l)$. Node failures and rebuild processes cause the values of $D_1(t), \cdots, D_{\tilde{r}}(t)$ and the exposure level of the system to change over time. Data loss occurs when some data have lost $\tilde{r}$ codeword symbols, that is, when $D_{\tilde{r}}(t) > 0$ for some time t. The smallest t for which $D_{\tilde{r}}(t) > 0$ is the first time the system ends up in data loss and is simply referred to as the time to data loss, T_{DL}:

$$T_{DL} = \min_{D_{\tilde{r}}(t)>0} t. \tag{6.3}$$

The time to data loss is a random variable and our goal is to estimate its mean, MTTDL.

6.3.2 Direct Path Approximation

A path to data loss following a first-node-failure event is a sequence of exposure level transitions that begins in exposure level 1 and ends in exposure level $\tilde{r}$ (data loss) without going back to exposure level 0, that is, for some $j \geq r$, a sequence of $j-1$ exposure level transitions $e_1 \to e_2 \to \cdots \to e_j$ such that $e_1 = 1$, $e_j = r$, $e_2, \cdots, e_{j-1} \in \{1, \cdots, \tilde{r}-1\}$, and $|e_i - e_{i-1}| = 1$, $\forall\, i = 2, \cdots, j$. Note that this collection of paths excludes visits to exposure level 0 and therefore only consists of all paths to data loss before all lost replicas are restored. To estimate P_{DL}, we need to estimate the probability of the union of *all* such paths to data loss following a first-node failure. As the set of events that can occur between exposure level 1 and exposure level $\tilde{r}$ is complex, estimating P_{DL} is a non-trivial problem.

To circumvent this problem, we approximate P_{DL} by the probability of the direct path to data loss, that is, the probability of the path $1 \to 2 \to \cdots \to \tilde{r}$. It is shown in Appendix B that the probability of the direct path approximates

well the probability of *all* paths, namely, P_{DL}, for a system with generally reliable nodes for which (2.10) holds. Thus, if we denote the probability of direct path to data loss by $P_{DL,\text{direct}}$, then

$$P_{DL} \approx P_{DL,\text{direct}}. \tag{6.4}$$

The proof of the above approximation relies only on the fact that the probabilities of transitions to higher exposure levels are extremely small, which is the case for systems with generally reliable nodes. The proof also does not make any assumptions on the failure and rebuild time distributions. Additionally, it is seen from the analysis in Appendix B that the approximation is quite good in the sense that the relative error of approximation tends to zero as the ratio, λ/μ, of mean time to rebuild to the mean time to failure tends to zero. In real-world storage systems, this ratio is observed to be generally small and therefore this is a reasonable approximation. The approximation is also seen to be quite good over a wide range of parameters using simulations which do not make this approximation.

6.3.3 Probability of the Direct Path to Data Loss

Consider the direct path to data loss, that is, the path $1 \to 2 \to \cdots \to \tilde{r}$ through the exposure levels. At each exposure level, the *intelligent* rebuild process attempts to rebuild the most-exposed data, that is, the data with the least number of codeword symbols left (see Section 2.6). Let the rebuild times of the most-exposed data at each exposure level in this path be denoted by R_e, $e = 1, \cdots, \tilde{r} - 1$. If no additional node failures occur during a rebuild in exposure level e that cause the system to go to exposure level $e+1$, then after a time period of R_e, the system will return to exposure level $e-1$. Therefore, determining the rebuild times at each exposure level is a key step in estimating the probability of the direct path to data loss.

The rebuild times at each exposure level are random variables that depend on the amount of most-exposed data to be rebuilt at that exposure level and the data placement scheme. The amount of most-exposed data to be rebuilt at a given exposure level, e, depends on when a node failure occurred during the rebuild in the previous exposure level, $e-1$, that caused the system to enter exposure level e. Let us illustrate this with a simple example: consider a system using an (l, m)-MDS code with $n = 2m$ storage nodes. The $2m$ storage nodes are divided into two clusters of m nodes each. The codewords of length m are stored (or *striped*) across the m nodes of either one cluster or the other. In other words, there is no codeword which is striped across nodes of two different clusters. In our model, this is an erasure coded system with an (l, m)-MDS code and clustered codeword placement. The system is at exposure level zero until one of the nodes fails, at which point, the system enters exposure level one. As the amount of data stored in the failed node was c, it follows that the codewords corresponding to this c amount of data have lost one symbol. As

described in Section 6.2, the codeword reconstruction process involves reading l of the surviving symbols of these affected codewords, decoding the user data corresponding to these codewords, re-encoding the lost symbols, and writing them to a new replacement node. As the processing during reconstruction, that is, the decoding and re-encoding, is assumed to be done on the fly, the reconstruction time depends on the time required to read the required data from the surviving nodes and write the reconstructed data to the new node. As described in Section 2.6, in clustered placement, this data is assumed to be read from any l of the $m-1$ surviving nodes of the cluster. The amount of data to be reconstructed is c and therefore, the total amount of data to be read is lc and the total amount of data to be written is c. However, due to the fact that the lc amount of data is read from l nodes in parallel with c amount from each node, and the fact that the average amount of time required to read (or write) c amount of data from (or to) a node is $1/\mu$, it takes an average of $1/\mu$ amount of time for this reconstruction process to complete. In other words, the rebuild time R_1 has mean $1/\mu$. As the nodes are generally reliable, most of the time, no additional nodes failures occur during this rebuild period and the system returns to exposure level zero. However, with a small probability, an additional node failure occurs. This node could either belong to the cluster being rebuilt, in which case the system enters exposure level two as some of the data lose a second codeword symbol, or the other cluster, in which case the system stays in the same exposure level as no data have lost more than one codeword symbol. To compute the probability of the direct path to data loss, we are interested in the probability of a node failure that causes the system to enter exposure level two. Suppose that this second node failure occurs when a fraction α of the data corresponding to the node that failed first is not yet rebuilt. As the two failed nodes shared codewords of all their data, the amount of data that loses a second codeword symbol when the second failure occurs is αc. This data is now the most-exposed and, and by similar arguments as above, it would now take an average of α/μ amount of time to rebuild this most-exposed data. In other words, the rebuild time R_2 has a conditional mean α/μ. We will now explicitly describe how one can estimate the rebuild times at each exposure level.

Let t_e, $e = 2, \cdots, \tilde{r}$, be the times of transitions from exposure level $e-1$ to e following a first-node failure, that is, a node failure that causes the system to enter rebuild mode from the fully-operational mode. Let $\tilde{n}_e$ be the number of nodes in exposure level e whose failure before the rebuild of most-exposed data causes an exposure level transition to level $e+1$. For example, in clustered placement scheme, the failure of any of the surviving nodes of the cluster being rebuilt causes some data to lose an additional codeword symbol thereby leading the system to the next exposure level. Therefore, for clustered placement, $\tilde{n}_e = \tilde{r} - e$ as there are exactly $\tilde{r} - e$ surviving nodes in a cluster being rebuilt when the system is in exposure level e. Now, let

$$F_e := \min_{i \in \{1, \cdots, \tilde{n}_{e-1}\}} E^{(i)}_{t_{e-1}}, \quad e = 2, \cdots, \tilde{r}, \tag{6.5}$$

denote the time taken for a node failure to occur that can cause the system to enter exposure level e. Note that $E^{(i)}_{t_{e-1}}$, as defined in Section 3.2.1, denotes the time period from t_{e-1} until the next failure of node i. Therefore, F_e denotes the time until the first failure among the $\tilde{n}_{e-1}$ nodes that causes the system to enter exposure level e.

At exposure level e, let α_e be the fraction of the rebuild time R_e still left when a node failure occurs causing an exposure level transition, that is, let

$$\alpha_e := \frac{R_e - F_{e+1}}{R_e}, \quad e = 1, \cdots, \tilde{r} - 2. \tag{6.6}$$

In Appendix C, it is shown that α_e is uniformly distributed between zero and one, that is,

$$\alpha_e \sim U(0,1), \quad e = 1, \cdots, \tilde{r} - 2. \tag{6.7}$$

Now, consider a direct path to data loss with $R_e = \tau_e$, $e = 1, \cdots, \tilde{r} - 1$, and $\alpha_e = a_e$, $e = 1, \cdots, \tilde{r} - 2$.[1] Denote the vector $(\tau_1, \cdots, \tau_{\tilde{r}-1})$ by $\vec{\tau}$ and $(a_1, \cdots, a_{\tilde{r}-2})$ by $\vec{a}$ for notational convenience. Then, the probability of this direct path, denoted by $P_{DL,\text{direct}}(\vec{\tau}, \vec{a})$, is given by

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\tau}, \vec{a}) &= \Pr\{R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1, R_2 = \tau_2, F_3 < R_2, \\ &\quad \cdots, \alpha_{\tilde{r}-2} = a_{\tilde{r}-2}, R_{\tilde{r}-1} = \tau_{\tilde{r}-1}, F_{\tilde{r}} < R_{\tilde{r}-1}\}. \end{aligned} \tag{6.8}$$

In the above expression, the events $F_e < R_{e-1}$ represent the exposure level transitions from $e-1$ to e. Thus, the above expression gives the probability that the system will take this particular direct path to data loss with $R_e = \tau_e$ and $\alpha_e = a_e$. Expanding (6.8) by conditioning, we get

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\tau}, \vec{a}) &= \Pr\{R_1 = \tau_1\} \times \Pr\{F_2 < R_1 | R_1 = \tau_1\} \\ &\quad \times \Pr\{\alpha_1 = a_1 | R_1 = \tau_1, F_2 < R_1\} \\ &\quad \times \Pr\{R_2 = \tau_2 | R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1\} \\ &\quad \times \Pr\{F_3 < R_2 | R_1 = \tau_1, F_2 < R_1, \alpha_1 = a_1, R_2 = \tau_2\} \\ &\quad \cdots \times \Pr\{F_{\tilde{r}} < R_{\tilde{r}-1} | R_1 = \tau_1, \cdots, R_{\tilde{r}-1} = \tau_{\tilde{r}-1}\}. \end{aligned} \tag{6.9}$$

The first term in the above expansion is the probability $\Pr\{R_1 = \tau_1\}$. Denote the mean of R_1 by $1/\mu_1$, that is,

$$\frac{1}{\mu_1} := E[R_1]. \tag{6.10}$$

The actual value of the mean will depend on the underlying data placement and will be discussed further in the later sections. Based on the rebuild model

[1] More strictly, we consider a direct path to data loss with $\tau_e < R_e \leq \tau_e + \delta\tau_e$, $e = 1, \cdots, \tilde{r}-1$, and $a_e < \alpha_e \leq \delta a_e$, $e = 1, \cdots, \tilde{r}-2$, where $\delta\tau_e$ and δa_e are positive infinitesimal quantities, but we leave this out for notational convenience.

described in Section 2.6, it follows that R_1 is distributed according to some distribution G_{μ_1} that satisfies (2.11):

$$R_1 \sim G_{\mu_1}. \tag{6.11}$$

Therefore, the first term in (6.9) reduces to

$$\Pr\{R_1 = \tau_1\} = g_{\mu_1}(\tau_1)\delta\tau_1, \tag{6.12}$$

where $\delta\tau_1$ denotes an infinitesimal increment in τ_1. The remaining terms in the expression for $P_{DL,\text{direct}}(\vec{r},\vec{a})$ in (6.9) fall into three types:

$$\text{Type A: } \Pr\{F_e < R_{e-1} | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}\}, \tag{6.13}$$

$$\text{Type B: } \Pr\{\alpha_e = a_e | R_1 = \tau_1, \cdots, R_e = \tau_e, F_{e+1} < R_e\}, \tag{6.14}$$

$$\text{Type C: } \Pr\{R_e = \tau_e | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}, F_e < R_{e-1}, \alpha_{e-1} = a_{e-1}\}. \tag{6.15}$$

As shown in Section 6.3.3, each of these types of expressions can be further simplified as follows.

Expressions of Type A

Expressions of the form (6.13), which denote the conditional probability of transition from exposure level $e-1$ to e, can be reduced to

$$\Pr\{F_e < R_{e-1} | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}\} \approx \tilde{n}_{e-1}\lambda\tau_{e-1}, \tag{6.16}$$

for $e = 2, \cdots, \tilde{r}$, where the approximation holds good for systems with generally reliable node satisfying (2.10) and (2.11).

Remark 6.1. *For real-world storage systems, which are made of generally reliable nodes, the node rebuild times τ_{e-1} (which are typically of the order of a few hours) are much smaller compared to the mean time to node failure $1/\lambda$ (which are typically of the order of a few years). Therefore, the conditional probabilities of transition to higher exposure levels, represented by (6.16), are extremely small for such systems.*

Expressions of Type B

Type B terms of the form (6.14), which denote the conditional probability that the fraction of rebuild time, R_e, still left when an exposure level transition from e to $e+1$ occurred is equal to a_e, can be reduced to

$$\Pr\{\alpha_e = a_e | R_1 = \tau_1, \cdots, R_e = \tau_e, F_{e+1} < R_e\} \approx \delta a_e, \tag{6.17}$$

for $e = 1, \cdots, \tilde{r} - 2$, where the approximation holds good for systems with generally reliable node satisfying (2.10) and (2.11). Here, δa_e denotes an infinitesimal increment of a_e.

Expressions of Type C

Type C expressions of the form (6.15), which denote the conditional probability that the rebuild time in exposure level e is equal to τ_e, reduce to

$$\begin{aligned}\Pr\{R_e = \tau_e | R_1 = \tau_1, \cdots, R_{e-1} = \tau_{e-1}, F_e < R_{e-1}, \alpha_{e-1} = a_{e-1}\} \\ = \begin{cases} g_{\mu_e}(\tau_e)\delta\tau_e & \text{under model A,} \\ \delta(\tau_e - 1/\mu_e)\delta\tau_e & \text{under model B,} \end{cases}\end{aligned} \tag{6.18}$$

for $e = 2, \cdots, \tilde{r} - 1$. Here, $g_{\mu_e}(\cdot)$ denotes the probability density function corresponding to the distribution G_{μ_e}, $\delta(\cdot - 1/\mu_e)$ denotes the Dirac delta function with a spike at $1/\mu_e$, $\delta\tau_e$ denotes an infinitesimal increment of τ_e, and models A and B describe the distribution of R_e given R_{e-1} and α_{e-1} as shown below:

$$\textbf{Model A:} \qquad R_e | R_{e-1}, \alpha_{e-1} \sim G_{\mu_e}, \tag{6.19}$$

$$\textbf{Model B:} \qquad R_e | R_{e-1}, \alpha_{e-1} = \frac{1}{\mu_e}. \tag{6.20}$$

In model A, we assume that, following a node failure, the system has to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where one of the l nodes from which data was being rebuilt failed and hence the system has to rebuild from another set of l nodes in the cluster. If l is small, this may involve a significant reconfiguration of the rebuild process. In model B, we assume that, following a node failure, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the case, for instance, in a clustered placement scheme where the newly failed node is not among the set of l nodes from which data is being rebuilt. This is also the case in a declustered placement scheme, where the rebuild was being done from all nodes, and therefore, in a large system, the failure of one node does not significantly affect the rebuild process.

Probability of a Sample Direct Path

Substituting (6.12), (6.16), (6.17), and (6.18) in (6.9), the probability of a sample direct path with $R_e = \tau_e$, $e = 1, \cdots, \tilde{r}-1$, and $\alpha_e = a_e$,$e = 1, \cdots, \tilde{r}-2$,

reduces to

$$\begin{aligned} P_{DL,\text{direct}}(\vec{\tau},\vec{a}) &\approx g_{\mu_1}(\tau_1)\delta\tau_1 \times \left(\prod_{e=2}^{\tilde{r}} \tilde{n}_{e-1}\lambda\tau_{e-1}\right) \times \left(\prod_{e'=1}^{\tilde{r}-1} \delta a_{e'}\right) \\ &\quad \times \begin{cases} \prod_{e''=2}^{\tilde{r}-2} g_{\mu_{e''}}(\tau_{e''})\delta\tau_{e''} & \text{(model A)} \\ \prod_{e''=2}^{\tilde{r}-2} \delta(\tau_{e''} - 1/\mu_{e''})\delta\tau_{e''} & \text{(model B)} \end{cases} \\ &= \lambda^{\tilde{r}-1} \times \tilde{n}_1 \cdots \tilde{n}_{\tilde{r}-1} \times \tau_1 \cdots \tau_{\tilde{r}-1} \times g_{\mu_1}(\tau_1) \\ &\quad \times \delta a_1 \cdots \delta a_{\tilde{r}-2} \times \delta\tau_1 \cdots \delta\tau_{\tilde{r}-1} \\ &\quad \times \begin{cases} g_{\mu_2}(\tau_2) \cdots g_{\mu_{\tilde{r}-1}}(\tau_{\tilde{r}-1}) & \text{(model A)} \\ \delta\left(\tau_2 - \frac{1}{\mu_2}\right) \cdots \delta\left(\tau_{\tilde{r}-1} - \frac{1}{\mu_{\tilde{r}-1}}\right) & \text{(model B)} \end{cases} \end{aligned} \tag{6.21}$$

Probability of Data Loss during Rebuild

As mentioned in Section 6.3.2, the probability of direct path to data loss, denoted by $P_{DL,\text{direct}}$, is a good approximation for the probability of data loss during rebuild, P_{DL}:

$$P_{DL} \approx P_{DL,\text{direct}}. \tag{6.22}$$

Also, the probability of the direct path to data loss, $P_{DL,\text{direct}}$, is the summation of the probabilities, $P_{DL,\text{direct}}(\vec{\tau},\vec{a})$, of all possible sample direct paths. As the infinitesimal increments in (6.21) tend to zero, the summation becomes an integral. Therefore, the probability of all possible direct paths to data loss, $P_{DL,\text{direct}}$, and hence, P_{Dl}, becomes

$$\begin{aligned} P_{DL} &\approx \lambda^{\tilde{r}-1} \times \tilde{n}_1 \cdots \tilde{n}_{\tilde{r}-1} \\ &\quad \times \int_{\tau_1} \cdots \int_{\tau_{\tilde{r}-1}} \int_{a_1} \cdots \int_{a_{\tilde{r}-2}} \tau_1 \cdots \tau_{\tilde{r}-1} g_{\mu_1}(\tau_1) \cdots g_{\mu_{\tilde{r}-1}}(\tau_{\tilde{r}-1}) d\vec{a} d\vec{\tau} \end{aligned} \quad \text{(model A)} \tag{6.23}$$

$$\begin{aligned} P_{DL} &\approx \lambda^{\tilde{r}-1} \times \tilde{n}_1 \cdots \tilde{n}_{\tilde{r}-1} \\ &\quad \times \int_{\tau_1} \cdots \int_{\tau_{\tilde{r}-1}} \int_{a_1} \cdots \int_{a_{\tilde{r}-2}} \left(\tau_1 \cdots \tau_{\tilde{r}-1} g_{\mu_1}(\tau_1) \right. \\ &\qquad \left. \times \delta\left(\tau_2 - \frac{1}{\mu_2}\right) \cdots \delta\left(\tau_{\tilde{r}-1} - \frac{1}{\mu_{\tilde{r}-1}}\right) d\vec{a} d\vec{\tau} \right) \end{aligned} \quad \text{(model B)}. \tag{6.24}$$

Here, the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, \tilde{r}-1$, and from 0 to 1 for a_e, $e = 1, \cdots, \tilde{r}-2$.

Remark 6.2. *The approximations in the expressions (6.23) and (6.24) for P_{DL} hold good for systems with generally reliable nodes that satisfy (2.10) and (2.11). The approximation is good in the sense that the relative error tends to zero as the ratio λ/μ of the mean time to node failure to the mean time to node rebuild tends to zero. The validity of the approximation has also been established for a wide rage of parameters using simulation.*

Remark 6.3. *The derivation of the expressions (6.23) and (6.24) for P_{DL} is quite general in the sense that it is applicable to all symmetric data placement schemes and to both replication-based systems and erasure-coded systems. It is also applicable to all node failure and rebuild distributions that satisfy (2.10) and (2.11). As can be observed from the expressions (6.23) and (6.24), the only unknowns in evaluating P_{DL} are the means μ_e, $e = 1, \cdots, \tilde{r} - 1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e$, $e = 1, \cdots, \tilde{r} - 1$. These quantities depend on the particular type of data placement or redundancy scheme used.*

Remark 6.4. *Replication-based storage systems with replication factor r are a special case of erasure coded systems with an (l, m)-MDS code, where the parameters l and m are equal to 1 and r, respectively. This can also be observed by comparing the expressions (4.44) and (4.45) for the P_{DL} of replication-based systems with the expressions (6.23) and (6.24) for the P_{DL} of systems using MDS codes.*

Remark 6.5. *It is clear from (6.23) and (6.24) that P_{DL} is invariant to the class of failure distributions satisfying (2.10) and (2.11) and only depends on the mean time to failure, $1/\lambda$. Furthermore, by the relation (3.14), the MTTDL is also invariant to this class of failure distributions. As this class of distributions includes real-world empirical distributions, such as the Weibull distribution, as well as the theoretically amenable exponential distribution, the benefit is two-fold. The fact that real-world failure distributions belongs to this class implies that these results are directly relevant to practical storage systems. On the other hand, the presence of the exponential distribution in this class means that MTTDL results obtained in the literature assuming unrealistic exponential distributions may be applicable to real-world storage systems as well.*

6.4 Effect of Codeword Placement on Reliability

In this section, we consider different codeword placement schemes as discussed in Section 2.4. We would like to estimate their reliability in terms of their MTTDL using the relations (3.14), (6.23), and (6.24), and understand how codeword placement affects data reliability. To use the expressions (6.23) and (6.24) for P_{DL}, we need to compute the conditional means of rebuild times in each exposure level, μ_e, $e = 1, \cdots, \tilde{r} - 1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e$, $e = 1, \cdots, \tilde{r} - 1$.

The values of these quantities depend on the underlying codeword placement and the nature of the rebuild process used.

Notation: Here, we introduce a few new notations and repeat the notations used in the previous section for the sake of clarity. Suppose that a first-node failure occurs at time t_1 causing the system to go from the fully-operational mode to exposure level 1 in the rebuild mode. Let t_e, $e = 2, \cdots, \tilde{r}$, denote the times of transitions from exposure level $e-1$ to e. Let the rebuild times of the most-exposed data at each exposure level in this path be R_e, $e = 1, \cdots, \tilde{r}-1$, with conditional means $1/\mu_e$, $e = 1, \cdots, \tilde{r}-1$. Let $\tilde{n}_e$, $e = 1, \cdots, \tilde{r}-1$, be the number of nodes whose failure during rebuild can cause a transition to the next higher exposure level. Let $D_e(t)$, $e = 0, \cdots, \tilde{r}$, denote the amount of user data that have lost e of their codeword symbols at time t. Around the times of exposure level transitions, t_e, let $D_e(t_e^-)$ and $D_e(t_e)$ denote the amounts of user data that have lost e of their codeword symbols just before and just after time t_e, respectively. In addition, let S_e, $e = 1, \cdots, \tilde{r}-1$, denote the average speed (or rate) of rebuild in exposure level e. Also, let the kth raw moment of the rebuild distribution G_μ with mean $1/\mu$ be denoted by $M_k(G_\mu)$, that is,

$$M_k(G_\mu) := \int_t t^k dG_\mu(t), \qquad \text{for } k = 1, 2, \cdots. \tag{6.25}$$

By definition,

$$M_1(G_\mu) = \frac{1}{\mu}. \tag{6.26}$$

Note that, by Jensen's inequality,

$$M_1^k(G_\mu) \leq M_k(G_\mu), \qquad \text{for } k = 1, 2, \cdots, \tag{6.27}$$

that is, the kth power of the mean of G_μ is lesser than or equal to the kth raw moment of G_μ. Lastly, the superscripts 'clus.' and 'declus.' will be used to refer to quantities specific to clustered and declustered placement schemes, respectively.

6.4.1 Clustered Codeword Placement

The goal of this section is to estimate the reliability of a clustered codeword placement scheme in terms of its MTTDL. To achieve this goal, we first compute the conditional means of rebuild times in each exposure level, $\mu_e^{\text{clus.}}$, $e = 1, \cdots, \tilde{r}-1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e^{\text{clus.}}$, $e = 1, \cdots, \tilde{r}-1$. Using these quantities and expressions (6.23) and (6.24), we can compute the probability of data loss during rebuild, $P_{DL}^{\text{clus.}}$. The mean time to data loss, $\text{MTTDL}^{\text{clus.}}$, can then be obtained by using the relation (3.14).

Clustered Codeword Placement: Exposure Level 1

Following a first-node failure at t_1, the codewords of some cluster lost one of their symbols. Therefore, the system enters exposure level 1 and the rebuild process begins. The amount of data to be rebuilt at this exposure level is equal to the capacity of the failed node, c, that is,

$$D_1^{\text{clus.}}(t_1) = c. \tag{6.28}$$

As described in Sections 2.6 and 6.2, the rebuild process in a system using an (l, m)-MDS code with clustered codeword placement involves reading l symbols of the codewords corresponding to the data on the failed node from l of the other $m-1$ surviving nodes of the affected cluster, computing the lost codeword symbols, and writing them to a new spare node. As the data is being read in parallel from l nodes at an average bandwidth of $c\mu$ from each node, and written to a new node at an average bandwidth of $c\mu$, the average rate (or speed) of rebuild in exposure level 1 is

$$S_1^{\text{clus.}} = c\mu. \tag{6.29}$$

The average time required for this rebuild, $1/\mu_1^{\text{clus.}}$, is obtained by dividing the amount of data to be rebuilt, given by (6.28), by the average speed of rebuild, given by (6.29). Thus,

$$\frac{1}{\mu_1^{\text{clus.}}} = E[R_1^{\text{clus.}}] = \frac{D_1^{\text{clus.}}(t_1)}{S_1^{\text{clus.}}} = \frac{1}{\mu}. \tag{6.30}$$

According to our model, the rebuild time, $R_1^{\text{clus.}}$, is distributed according to some distribution $G_{\mu_1^{\text{clus.}}}$ with mean $1/\mu_1^{\text{clus.}}$ that satisfies (2.11), that is,

$$R_1^{\text{clus.}} \sim G_{\mu_1^{\text{clus.}}} = G_\mu. \tag{6.31}$$

There are $m-1$ remaining nodes in the cluster of the failed node. Due to the nature of clustered codeword placement, the failure of any of these nodes during the rebuild period $R_1^{\text{clus.}}$ will cause the system to enter exposure level 2, whereas the failure of nodes belonging to any other cluster does not cause the system to enter exposure level 2. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_1^{\text{clus.}} = m - 1. \tag{6.32}$$

When one of the $\tilde{n}_1^{\text{clus.}}$ nodes fail before rebuild, the system enters exposure level 2.

Clustered Codeword Placement: Exposure Level 2

The system enters exposure level 2 from exposure level 1 because one of the $\tilde{n}_1^{\text{clus.}}$ nodes fails during the rebuild period $R_1^{\text{clus.}}$. Consider an instance of the rebuild period,

$$R_1^{\text{clus.}} = \tau_1, \tag{6.33}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from 1 to 2 occurred,

$$\alpha_1 = a_1. \tag{6.34}$$

The remaining time to complete rebuild at exposure level 1 when the system entered exposure level 2 is the product of $R_1^{\text{clus.}}$ and α_1, namely, $a_1\tau_1$. As the average speed of rebuild in exposure level 1 is $S_1^{\text{clus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_1^{\text{clus.}}(t_2^-)$, is given by

$$D_1^{\text{clus.}}(t_2^-) = \alpha_1 R_1^{\text{clus.}} S_1^{\text{clus.}} = a_1\tau_1 c\mu, \tag{6.35}$$

which is essentially the product of (6.29), (6.33), and (6.34). At the time of transition from exposure level 1 to 2, t_2, all this $D_1^{\text{clus.}}(t_2^-)$ amount of data loses its second codeword symbol and is thus the most-exposed data in exposure level 2. This is due to the nature of the clustered codeword placement scheme in which all nodes of a cluster store codewords of the same data. Therefore, the amount of most-exposed data in exposure level 2, $D_2^{\text{clus.}}(t_2)$, is given by

$$D_2^{\text{clus.}}(t_2) = D_1^{\text{clus.}}(t_2^-) = a_1\tau_1 c\mu. \tag{6.36}$$

The average speed of rebuild remains unchanged as the rebuild process still involves reading data from l of the remaining $m-2$ nodes of the affected cluster at an average bandwidth of $c\mu$ from each node, computing the lost codeword symbols on-the-fly, and writing them to a spare node at an average bandwidth of $c\mu$. Therefore,

$$S_2^{\text{clus.}} = c\mu. \tag{6.37}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_2^{\text{clus.}}$, of the rebuild time in the second exposure level, $R_2^{\text{clus.}}$, is obtained by dividing (6.36) by (6.37), that is,

$$\frac{1}{\mu_2^{\text{clus.}}} = E[R_2^{\text{clus.}} | R_1^{\text{clus.}} = \tau_1, \alpha_1 = a_1] = \frac{D_2^{\text{clus.}}(t_2)}{S_2^{\text{clus.}}} = a_1\tau_1. \tag{6.38}$$

There are now $m-2$ remaining nodes in the cluster of the failed node. The failure of any of these nodes during the rebuild time $R_2^{\text{clus.}}$ will cause the system to enter exposure level 3. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_2^{\text{clus.}} = m - 2. \tag{6.39}$$

Clustered Codeword Placement: Exposure Level e

The computation of the conditional mean $1/\mu_e^{\text{clus.}}$ and the number of nodes $\tilde{n}_e^{\text{clus.}}$ for a general exposure level $e = 2, \cdots, \tilde{r}-1$, is similar to the computation of these quantities for exposure level 2 as described above. Firstly, we note that the average speed of rebuild is unchanged in each exposure level for clustered placement, that is,

$$S_e^{\text{clus.}} = c\mu, \quad e = 1, \cdots, \tilde{r} - 1. \tag{6.40}$$

This is due to the fact that the rebuild process always involves reading data from l of the affected cluster at an average bandwidth of $c\mu$ from each node, computing the lost codeword symbols on-the-fly, and writing them to a spare node at an average bandwidth of $c\mu$.

Now, the system enters exposure level e from exposure level $e-1$ because one of the $\tilde{n}_{e-1}^{\text{clus.}}$ nodes fails during the rebuild period $R_{e-1}^{\text{clus.}}$. Consider an instance of the rebuild period,

$$R_{e-1}^{\text{clus.}} = \tau_{e-1}, \tag{6.41}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from $e-1$ to e occurred,

$$\alpha_{e-1} = a_{e-1}. \tag{6.42}$$

The remaining time to complete rebuild at exposure level $e-1$ when the system entered exposure level e is the product of $R_{e-1}^{\text{clus.}}$ and α_{e-1}, namely, $a_{e-1}\tau_{e-1}$. As the average speed of rebuild in exposure level $e-1$ is $S_{e-1}^{\text{clus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_{e-1}^{\text{clus.}}(t_e^-)$, is given by

$$D_{e-1}^{\text{clus.}}(t_e^-) = \alpha_{e-1} R_{e-1}^{\text{clus.}} S_{e-1}^{\text{clus.}} = a_{e-1}\tau_{e-1}c\mu, \tag{6.43}$$

which is essentially the product of (6.40), (6.41), and (6.42). At the time of transition from exposure level $e-1$ to e, t_e, all this $D_{e-1}^{\text{clus.}}(t_e^-)$ amount of data loses its eth codeword symbol and is thus the most-exposed data in exposure level e. This is due to the nature of the clustered codeword placement scheme. Therefore, the amount of most-exposed data in exposure level e, $D_e^{\text{clus.}}(t_e)$, is given by

$$D_e^{\text{clus.}}(t_e) = D_e^{\text{clus.}}(t_e^-) = a_{e-1}\tau_{e-1}c\mu. \tag{6.44}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_e^{\text{clus.}}$, of the rebuild time in the eth exposure level, $R_e^{\text{clus.}}$, is obtained by dividing (6.44) by (6.40), that is,

$$\frac{1}{\mu_e^{\text{clus.}}} = E[R_e^{\text{clus.}} | R_{e-1}^{\text{clus.}} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}] = \frac{D_e^{\text{clus.}}(t_e)}{S_e^{\text{clus.}}} = a_{e-1}\tau_{e-1}. \tag{6.45}$$

There are now $m-e$ remaining nodes in the cluster under rebuild. The failure of any of these nodes during the rebuild time $R_e^{\text{clus.}}$ will cause the system to enter exposure level $e+1$. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_e^{\text{clus.}} = m - e. \tag{6.46}$$

Clustered Codeword Placement: MTTDL under Model A

Recall that, under model A, following each exposure level transition, the system is assumed to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where one of the l nodes from which data was being rebuilt failed and hence the system has to rebuild from another node set of l nodes in the cluster. If l is small, this may involve a significant reconfiguration of the rebuild process. This implies that the rebuild time in the new exposure level is a random variable, and only its mean depends on the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level. Given this mean, the rebuild time in the new exposure level is independent of the rebuild time in the previous exposure level. Having computed the key quantities $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ for $e = 1, \cdots, \tilde{r}-1$, we are now ready to compute $P_{DL}^{\text{clus.}}$ using the expression (6.23) for model A, and then $\text{MTTDL}^{\text{clus.}}$ using (3.14).

By substituting the values of $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ from (6.30), (6.45), and (6.46) into (6.23), we obtain

$$\begin{aligned} P_{DL}^{\text{clus.}} \approx\ & \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}+1) \\ & \times \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-1}} \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \tau_1 \cdots \tau_{\tilde{r}-1} g_\mu(\tau_1)\cdots g_{\frac{1}{a_{\tilde{r}-2}\tau_{\tilde{r}-2}}}(\tau_{\tilde{r}-1}) d\vec{a} d\vec{\tau} \end{aligned}$$

(model A) (6.47)

As in (6.23) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, \tilde{r}-1$, and from 0 to 1 for a_e, $e = 1, \cdots, \tilde{r}-2$.

(l, m)-MDS codes with $m - l \leq 2$: The expression (6.47) for $P_{DL}^{\text{clus.}}$ under model A cannot, in general, be further simplified without considering a particular family of rebuild distributions G_μ. However, it is worth noting that, for $\tilde{r} \leq 3$, a closed form expression for $P_{DL}^{\text{clus.}}$, and hence $\text{MTTDL}^{\text{clus.}}$, can be obtained under model A. By definition (6.1), this corresponds to the case when $m - l \leq 2$. This is illustrated by deriving the closed form expression for $\tilde{r} = 3$, or equivalently, $m - l = 2$, by substituting $\tilde{r} = 3$ in (6.47) and simplifying as

follows.

$$\begin{aligned} P_{DL}^{\text{clus.}} &\approx (m-1)(m-2)\lambda^2 \int_{\tau_1=0}^{\infty}\int_{\tau_2=0}^{\infty}\int_{a_1=0}^{1} \tau_1\tau_2 g_\mu(\tau_1) g_{\frac{1}{a_1\tau_1}}(\tau_2) da_1 d\tau_2 d\tau_1 \\ &= (m-1)(m-2)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1 g_\mu(\tau_1) \int_{a_1=0}^{1}\int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{1}{a_1\tau_1}}(\tau_2) d\tau_2 da_1 d\tau_1 \\ &\qquad \text{for } m-l=2 \text{ (model A).} \end{aligned} \tag{6.48}$$

Noting that

$$\int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{1}{a_1\tau_1}}(\tau_2) d\tau_2 = a_1\tau_1, \tag{6.49}$$

we get

$$P_{DL}^{\text{clus.}} \approx (m-1)(m-2)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_\mu(\tau_1) \int_{a_1=0}^{1} a_1 da_1 d\tau_1 \tag{6.50}$$

$$= \frac{1}{2}(m-1)(m-2)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_\mu(\tau_1) d\tau_1 \tag{6.51}$$

$$= \frac{1}{2}(m-1)(m-2)\lambda^2 M_2(G_\mu) \qquad \text{for } m-l=2 \text{ (model A),} \tag{6.52}$$

where $M_2(G_\mu)$, as defined in (6.25), denotes the second raw moment of the rebuild distribution G_μ. The expression for $\text{MTTDL}^{\text{clus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{clus.}}} \tag{6.53}$$

$$\approx \frac{2}{n(m-1)(m-2)\lambda^3 M_2(G_\mu)} \tag{6.54}$$

$$= \frac{\mu^2}{n\lambda^3} \frac{2}{(m-1)(m-2)} \frac{M_1^2(G_\mu)}{M_2(G_\mu)} \qquad \text{for } m-l=2 \text{ (model A).} \tag{6.55}$$

Here, the last step is obtained by multiplying and dividing by square of the mean of the rebuild time distribution G_μ, $M_1^2(G_\mu)$, which is also equal to $1/\mu^2$. This is done to show the effect of the rebuild distribution on the MTTDL. For deterministic rebuild times, the second raw moment, $M_2(G_\mu)$, is equal to the square of the first raw moment, $M_1^2(G_\mu)$, and therefore, the term $M_1^2(G_\mu)/M_2(G_\mu)$ evaluates to one. However, if the rebuild times are random, the second raw moment is always greater than the square of the first raw moment by Jensen's inequality, and therefore, the term $M_1^2(G_\mu)/M_2(G_\mu)$ is smaller than one. The closed form expression for $m-l=1$, or equivalently, $\tilde{r}=2$, can be derived similarly and is given by

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu}{n\lambda^2} \frac{1}{(m-1)} \qquad \text{for } m-l=1 \text{ (model A).} \tag{6.56}$$

(l, m)-MDS codes with $m-l > 2$: For (l, m)-MDS codes with $m-l > 2$, or equivalently, $\tilde{r} > 3$, the evaluation of $P_{Dl}^{\text{clus.}}$ under model A involves computing the expectations of functions involving higher raw moments of G_μ, which cannot be done without considering a particular family of rebuild distributions. However, given a particular family of rebuild distributions, the derivation of MTTDL involves successively evaluating the integrals in (6.47) to compute $P_{DL}^{\text{clus.}}$, and then using (3.14) to obtain $\text{MTTDL}^{\text{clus.}}$.

Clustered Codeword Placement: MTTDL under Model B

In contrast to model A, we assume in model B that, following an exposure level transition, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the case, for instance, in a clustered placement scheme where the newly failed node is not in the set of l nodes from which data is being rebuilt. This is also the case in a declustered placement scheme, where the rebuild is being done from all nodes, and therefore, in a large system, the failure of one node does not significantly affect the rebuild process. This implies that the rebuild time in the new exposure level is completely determined by the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level.

By substituting the values of $1/\mu_e^{\text{clus.}}$ and $\tilde{n}_e^{\text{clus.}}$ from (6.30), (6.45), and (6.46) into (6.24), we obtain

$$\begin{aligned} P_{DL}^{\text{clus.}} \approx\ & \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}-1) \\ & \times \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-1}} \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \Big(\tau_1 \cdots \tau_{\tilde{r}-1} g_\mu(\tau_1) \\ & \qquad \times \delta(\tau_2 - a_1\tau_1)\cdots\delta(\tau_{\tilde{r}-1} - a_{\tilde{r}-2}\tau_{\tilde{r}-2})\, d\vec{a}\, d\vec{\tau} \Big) \\ & \qquad\qquad \text{(model B).} \end{aligned} \tag{6.57}$$

As in (6.24) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, \tilde{r}-1$, and from 0 to 1 for a_e, $e = 1, \cdots, \tilde{r}-2$. In contrast to model A, closed form expressions in terms of the raw moments of the rebuild distribution G_μ can be obtained for model B as follows. By changing the order of integrals in (6.57), we obtain

$$\begin{aligned} P_{DL}^{\text{clus.}} \approx\ & \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}-1) \\ & \times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-1}} \Big(\tau_1 \cdots \tau_{\tilde{r}-1} g_\mu(\tau_1) \\ & \qquad \times \delta(\tau_2 - a_1\tau_1)\cdots\delta(\tau_{\tilde{r}-1} - a_{\tilde{r}-2}\tau_{\tilde{r}-2})\, d\tau_{\tilde{r}-1}\cdots d\tau_1 d\vec{a} \Big) \end{aligned}$$

$$= \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}-1)$$
$$\times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-2}} \Bigg(\tau_1 \cdots \tau_{\tilde{r}-2}^2 a_{\tilde{r}-2} g_\mu(\tau_1)$$
$$\times \delta(\tau_2 - a_1\tau_1)\cdots\delta(\tau_{\tilde{r}-2} - a_{\tilde{r}-3}\tau_{\tilde{r}-3})\, d\tau_{\tilde{r}-2}\cdots d\tau_1 d\vec{a} \Bigg) \quad (6.58)$$

$$= \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}-1)$$
$$\times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-3}} \Bigg(\tau_1 \cdots \tau_{\tilde{r}-3}^3 a_{\tilde{r}-3}^2 a_{\tilde{r}-2} g_\mu(\tau_1)$$
$$\times \delta(\tau_2 - a_1\tau_1)\cdots\delta(\tau_{\tilde{r}-3} - a_{\tilde{r}-4}\tau_{\tilde{r}-4})\, d\tau_{\tilde{r}-3}\cdots d\tau_1 d\vec{a} \Bigg) \quad (6.59)$$

$$\vdots$$

$$= \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}-1)$$
$$\times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1} \tau_1^{\tilde{r}-1} a_1^{\tilde{r}-2}\cdots a_{\tilde{r}-3}^2 a_{\tilde{r}-2} g_\mu(\tau_1) d\tau_1 d\vec{a}$$
(model B). (6.60)

Here, steps (6.58)–(6.60) follow by successively integrating over $\tau_{\tilde{r}-1}, \cdots, \tau_2$, by using the Dirac delta function's property. Changing the order of the integrals and integrating out $a_1, \cdots, a_{\tilde{r}-2}$, we get

$$P_{DL}^{\text{clus.}} \approx \lambda^{\tilde{r}-1} \times (m-1)\cdots(m-\tilde{r}-1) \times \int_{\tau_1} \tau_1^{\tilde{r}-1} g_\mu(\tau_1) \frac{1}{(\tilde{r}-1)!} d\tau_1 \quad (6.61)$$
$$= \lambda^{\tilde{r}-1} \binom{m-1}{\tilde{r}-1} M_{\tilde{r}-1}(G_\mu) \quad (6.62)$$
$$= \lambda^{m-l} \binom{m-1}{m-l} M_{m-l}(G_\mu) \quad (6.63)$$
$$= \lambda^{m-l} \binom{m-1}{l-1} M_{m-l}(G_\mu) \quad \text{(model B)} \quad (6.64)$$

where $M_{m-l}(G_\mu)$, as defined in (6.25), denotes the $(m-l)$th raw moment of the rebuild distribution G_μ. Here, we have substituted for $\tilde{r}$ from its definition (6.1), that is, $\tilde{r} = m - l + 1$. The expression for $\text{MTTDL}^{\text{clus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{clus.}}} \quad (6.65)$$
$$\approx \frac{1}{n\lambda^{m-l+1}\binom{m-1}{l-1} M_{m-l}(G_\mu)} \quad (6.66)$$
$$= \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{1}{\binom{m-1}{l-1}} \frac{M_1^{m-l}(G_\mu)}{M_{m-l}(G_\mu)} \quad \text{(model B)}. \quad (6.67)$$

Here, the last step is obtained by multiplying and dividing by $(m-l)$th power of the mean of the rebuild time distribution G_μ, $M_1^{m-l}(G_\mu)$, which is also equal to $1/\mu^{m-l}$. This is done to show the effect of the rebuild distribution on the MTTDL. For deterministic rebuild times, the $(m-l)$th raw moment, $M_{m-l}(G_\mu)$, is equal to the $(m-l)$th power of the first raw moment, $M_1^{m-l}(G_\mu)$, and therefore, the term $M_1^{m-l}(G_\mu)/M_{m-l}(G_\mu)$ evaluates to one. For random rebuild times, by the Jensen's inequality, the $(m-l)$th raw moment, $M_{m-l}(G_\mu)$, is always greater than the $(m-l)$th power of the first raw moment, $M_1^{m-l}(G_\mu)$, and therefore, the term $M_1^{m-l}(G_\mu)/M_{m-l}(G_\mu)$ evaluates to less than one.

As an example, if G_μ is exponential, the expression for $\text{MTTDL}^{\text{clus.}}$ reduces to the following:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{1}{\binom{m-1}{l-1}} \frac{1}{(m-l)!} \tag{6.68}$$

$$= \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{(l-1)!}{(m-1)!} \quad \text{when } G_\mu \text{ is exponential (model B)} \tag{6.69}$$

Remark 6.6. *The expressions (6.55), (6.67), and (6.69) for the mean time to data loss of a storage system with clustered codeword placement scheme under both models A and B are seen to be invariant within the class of node failure distributions that satisfy (2.10) and (2.11). In particular, the MTTDL only depends on the mean of the times to node failure, $1/\lambda$. As the conditions (2.10) and (2.11) hold true for real-world storage nodes as well, these MTTDL results are of practical significance.*

Remark 6.7. *The expressions (6.55), (6.67), and (6.69) for the mean time to data loss of a storage system with clustered codeword placement scheme also reveal that the MTTDL is sensitive to the rebuild distribution G_μ and to the choice of the model A or B. It is observed that deterministic rebuild times have better MTTDL values compared to random rebuild times because the terms of the form $M_1^{m-l}(G_\mu)/M_{m-l}(G_\mu)$ are upper-bounded by one due to the Jensen's inequality, and the bound is achieved for deterministic rebuild times. The explanation for this fact is that, when rebuild times are random, given that a failure occurred during rebuild, it is more probable that the rebuild time was larger. This effect is known as the waiting time paradox. The waiting time paradox is also the reason for the MTTDL values to be higher under model B than under model A because model A introduces additional randomness to the rebuild times at each exposure level whereas model B does not.*

Remark 6.8. *For a given rebuild distribution G_μ and for (l,m)-MDS codes with $m-l \leq 2$, we note that the expressions for MTTDL under model A, given by (6.55) and (6.56), and the expressions for MTTDL under model B, given by (6.67), are not different. However, when $m-l > 2$, the MTTDL under model A may differ from the MTTDL under model B. Furthermore, if the rebuild times are deterministic, we note that the models A and B do not differ by*

definition (see (6.19) and (6.20)). Therefore, for deterministic rebuild times, the MTTDL values under both models are the same.

Remark 6.9. *The MTTDL of an erasure coded system with clustered placement scheme is observed to scale down inversely proportional to the number of nodes n. It is directly proportional to the $(m-l+1)$th power of the mean time to node failure $1/\lambda$, and inversely proportional to the $(m-l)$th power of the mean time to node rebuild $1/\mu$. This was also the case for replication-based systems and is seen to be a general trend in the MTTDL behavior of data storage systems.*

Remark 6.10. *Replication with replication factor r is equivalent to an (l,m)-MDS code with parameters $l=1$ and $m=r$. Therefore, as expected, it is seen that the MTTDL expressions (6.55), (6.67), and (6.69) for clustered placement in erasure coded systems with an (l,m)-MDS code respectively reduce to the MTTDL expressions (4.74), (4.83), and (4.84) for clustered placement in replication-based storage systems with replication factor r, when the parameters $l=1$ and $m=r$.*

Remark 6.11. *The use of erasure codes in data storage can be dated back to the 1980s when it was used in designing systems with redundant arrays of inexpensive disks, or RAID. The so-called RAID-5 system consists of clusters of m disks where the user data blocks are striped across $m-1$ disks of each cluster along with one parity block on the mth disk of that cluster [16]. In our model, this corresponds to a clustered codeword placement of an $(m-1,m)$-MDS code. The expression for MTTDL of such a system, (6.67), is seen to match with corresponding expression in [16] for a RAID-5 system. Similarly, the MTTDL expressions for the so-called RAID-6 system [27], which corresponds to a clustered codeword placement of an $(m-2,m)$-MDS code, is also seen to match with (6.67).*

6.4.2 Declustered Codeword Placement

In most storage systems, the mean times to node failure and mean times to node rebuilds are given constants because they depend on the particular type of nodes used. For an (l,m)-MDS code based system with a given type of nodes, one way to improve reliability is to increase $m-l$. However, this comes at the cost of performance because each update to a user data block requires the system to read its corresponding codeword, recompute the new codeword, and write the new codeword blocks. The other alternative to improving reliability may be to simply change the underlying codeword placement and the way in which rebuild is done to gain significant improvements in reliability for large storage systems. Declustered codeword placement is one of those ways in which the system reliability can be improved over clustered placement for large storage systems. The goal of this section is to estimate the reliability of the declustered codeword placement scheme in terms of the mean time to data

loss, and understand how this codeword placement scheme can achieve high reliability in large systems. To achieve this goal, we first compute the conditional means of rebuild times in each exposure level, $\mu_e^{\text{declus.}}$, $e = 1, \cdots, \tilde{r} - 1$, and the number of nodes whose failure can cause a transition to the next exposure level, $\tilde{n}_e^{\text{declus.}}$, $e = 1, \cdots, \tilde{r} - 1$. Using these quantities and expressions (6.23) and (6.24), we can compute the probability of data loss during rebuild, $P_{DL}^{\text{declus.}}$. The mean time to data loss, $\text{MTTDL}^{\text{declus.}}$, can then be obtained by using the relation (3.14).

Declustered Codeword Placement: Exposure Level 1

Following a first-node failure at t_1, the system enters exposure level 1 and the rebuild process begins. The amount of data to be rebuilt at this exposure level is equal to the capacity of the failed node, c, that is,

$$D_1^{\text{declus.}}(t_1) = c. \tag{6.70}$$

By the nature of the declustered placement, the $m-1$ remaining symbols of the codewords corresponding to the failed node are spread equally across all the surviving $n-1$ nodes of the system. As described in Sections 2.6 and 6.2, the distributed rebuild process in a declustered placement scheme involves reading the required codeword symbols of the data to be rebuilt from all the surviving nodes of the system, computing the lost codeword symbols, and writing them to the spare space of these nodes in such a way that no codeword symbol is written to a node in which another codeword symbol corresponding to the same codeword is already present. This process requires reading lc amount of data, as well as writing c amount of data, from and to all $n-1$ surviving nodes in parallel. As each of the $n-1$ nodes has an average read-write rebuild bandwidth of $c\mu$, and as l times more data is read from each node than what is written during the distributed rebuild process, the average rate of rebuild in exposure level 1 is

$$S_1^{\text{declus.}} = \frac{n-1}{l+1} c\mu. \tag{6.71}$$

It can be seen that (6.71) reduces to the corresponding expression (4.86) for replication based systems where the parameter l is one. The average time required for this rebuild, $1/\mu_1^{\text{declus.}}$, is obtained by dividing the amount of data to be rebuilt, given by (6.70), by the average speed of rebuild, given by (6.71). Thus,

$$\frac{1}{\mu_1^{\text{declus.}}} = E[R_1^{\text{declus.}}] = \frac{D_1^{\text{declus.}}(t_1)}{S_1^{\text{declus.}}} = \frac{l+1}{(n-1)\mu}. \tag{6.72}$$

According to our model, the rebuild time, R_1, is distributed according to G_{μ_1}, that is,

$$R_1^{\text{declus.}} \sim G_{\mu_1^{\text{declus.}}} = G_{\frac{n-1}{l+1}\mu}. \tag{6.73}$$

There are $n-1$ surviving nodes in the system, each containing equal amounts of codewords symbols corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period $R_1^{\text{declus.}}$ will cause the system to enter exposure level 2. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_1^{\text{declus.}} = n - 1. \tag{6.74}$$

When one of the $\tilde{n}_1^{\text{declus.}}$ nodes fail before rebuild, the system enters exposure level 2.

Declustered Codeword Placement: Exposure Level 2

The system enters exposure level 2 from exposure level 1 because one of the $\tilde{n}_1^{\text{declus.}}$ nodes fails during the rebuild period $R_1^{\text{declus.}}$. Consider an instance of the rebuild period,

$$R_1^{\text{declus.}} = \tau_1, \tag{6.75}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from 1 to 2 occurred,

$$\alpha_1 = a_1. \tag{6.76}$$

The remaining time to complete rebuild at exposure level 1 when the system entered exposure level 2 is the product of $R_1^{\text{declus.}}$ and α_1, namely, $a_1\tau_1$. As the average speed of rebuild in exposure is $S_1^{\text{declus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_1(t_2^-)$, is given by

$$D_1^{\text{declus.}}(t_2^-) = \alpha_1 R_1^{\text{declus.}} S_1^{\text{declus.}} = a_1\tau_1 \frac{n-1}{l+1} c\mu, \tag{6.77}$$

which is essentially the product of (6.71), (6.75), and (6.76). In contrast to clustered placement, at the time of transition from exposure level 1 to 2, t_2, *not* all of this $D_1^{\text{declus.}}(t_2^-)$ amount of data loses a second codeword symbol. As discussed in Section 2.4, due to the nature of the declustered codeword placement scheme, the two failed nodes store codewords of only a fraction $\frac{m-1}{n-1}$ of this data. So during the exposure level transition, only $\frac{m-1}{n-1} D_1^{\text{declus.}}(t_2^-)$ amount of user data loses a second codeword symbol. Therefore, the amount of most-exposed data in exposure level 2, $D_2^{\text{declus.}}(t_2)$, is given by

$$D_2^{\text{declus.}}(t_2) = \frac{m-1}{n-1} D_1^{\text{declus.}}(t_2^-) = \frac{m-1}{l+1} a_1\tau_1 c\mu. \tag{6.78}$$

By the nature of the declustered placement, the $m-2$ remaining codeword symbols of the most-exposed data are spread equally across all the surviving $n-2$ nodes of the system. As described in Sections 2.6 and 6.2, the distributed rebuild process in a declustered placement scheme involves reading the required

codeword symbols of the data to be rebuilt from all the surviving nodes of the system, computing the lost codeword symbols, and writing them to the spare space of these nodes in such a way that no codeword symbol is written to a node in which another codeword symbol corresponding to the same codeword is already present. This process requires reading lc amount of data, as well as writing c amount of data, from and to all $n-2$ surviving nodes in parallel. As each of the $n-2$ nodes has an average read-write rebuild bandwidth of $c\mu$, and as l times more data is read from each node than what is written during the distributed rebuild process, the average rate of rebuild in exposure level 2 is given by

$$S_2^{\text{declus.}} = \frac{n-2}{l+1} c\mu. \tag{6.79}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_2^{\text{declus.}}$, of the rebuild time in the second exposure level, $R_2^{\text{declus.}}$, is obtained by dividing (6.78) by (6.79), that is,

$$\frac{1}{\mu_2^{\text{declus.}}} = E[R_2^{\text{declus.}} | R_1^{\text{declus.}} = \tau_1, \alpha_1 = a_1] = \frac{D_2^{\text{declus.}}(t_2)}{S_2^{\text{declus.}}} = \frac{m-1}{n-2} a_1 \tau_1. \tag{6.80}$$

There are now $n-2$ surviving nodes in the system, each containing equal amounts of the codeword symbols corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period $R_2^{\text{declus.}}$ will cause the system to enter exposure level 3. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_2^{\text{declus.}} = n-2. \tag{6.81}$$

Declustered Codeword Placement: Exposure Level e

The computation of the conditional mean $1/\mu_e^{\text{declus.}}$ and the number of nodes $\tilde{n}_e^{\text{declus.}}$ for a general exposure level $e = 2, \cdots, \tilde{r}-1$ is similar to the computation of these quantities for exposure level 2 as described above. Firstly, we note that the distributed rebuild process in each exposure level e always involves reading the required codeword symbols of the data to be rebuilt from all the $n-e$ surviving nodes of the system, computing the lost codeword symbols, and writing them to the spare space of these nodes in such a way that no codeword symbol is written to a node in which another codeword symbol corresponding to the same codeword is already present. This process requires reading lc amount of data, as well as writing c amount of data, from and to all $n-e$ surviving nodes in parallel. Due to the nature of the declustered placement, this involves reading and writing equal amounts of data in each node. As each of the $n-e$ nodes has an average read-write rebuild bandwidth of $c\mu$, and as l times more data is read from each node than what is written during the

distributed rebuild process, the average rate of rebuild in exposure level e is given by

$$S_e^{\text{declus.}} = \frac{n-e}{l+1}c\mu, \quad e = 1, \cdots, \tilde{r}-1. \tag{6.82}$$

Now, the system enters exposure level e from exposure level $e-1$ because one of the $\tilde{n}_{e-1}^{\text{declus.}}$ nodes fails during the rebuild period $R_{e-1}^{\text{declus.}}$. Consider an instance of the rebuild period,

$$R_{e-1}^{\text{declus.}} = \tau_{e-1}, \tag{6.83}$$

and an instance of the fraction of the rebuild period still left when the exposure level transition from $e-1$ to e occurred,

$$\alpha_{e-1} = a_{e-1}. \tag{6.84}$$

The remaining time to complete rebuild at exposure level $e-1$ when the system entered exposure level e is the product of $R_{e-1}^{\text{declus.}}$ and α_{e-1}, namely, $a_{e-1}\tau_{e-1}$. As the average speed of rebuild in exposure is $S_{e-1}^{\text{declus.}}$, it follows that the amount of the most-exposed data not rebuilt when the exposure level transition occurred, $D_{e-1}^{\text{declus.}}(t_e^-)$, is given by

$$D_{e-1}^{\text{declus.}}(t_e^-) = \alpha_{e-1}R_{e-1}^{\text{declus.}}S_{e-1}^{\text{declus.}} = a_{e-1}\tau_{e-1}\frac{n-e+1}{l+1}c\mu, \tag{6.85}$$

which is essentially the product of (6.82), (6.83), and (6.84). At the time of transition from exposure level $e-1$ to e, t_{e-1}, *not* all of this $D_1^{\text{declus.}}(t_2^-)$ amount of user data loses its eth codeword symbol. Due to the nature of the declustered codeword placement scheme, the newly failed nodes store codewords of only a fraction $\frac{m-e+1}{n-e+1}$ of this user data. So during the exposure level transition, only $\frac{m-e+1}{n-e+1}D_1^{\text{declus.}}(t_2^-)$ amount of user data loses its eth codeword symbol. Therefore, the amount of most-exposed data in exposure level 2, $D_2^{\text{declus.}}(t_2)$, is given by

$$D_e^{\text{declus.}}(t_e) = \frac{m-e+1}{n-e+1}D_e^{\text{declus.}}(t_e^-) = \frac{m-e+1}{l+1}a_{e-1}\tau_{e-1}c\mu. \tag{6.86}$$

As the rebuild process is assumed to be *intelligent*, that is, the most-exposed data are rebuilt first, the conditional mean, $1/\mu_e^{\text{declus.}}$, of the rebuild time in the eth exposure level, $R_e^{\text{declus.}}$, is obtained by dividing (6.86) by (6.82), that is,

$$\frac{1}{\mu_e^{\text{declus.}}} = E[R_e^{\text{declus.}} | R_{e-1}^{\text{declus.}} = \tau_{e-1}, \alpha_{e-1} = a_{e-1}] \tag{6.87}$$

$$= \frac{D_e^{\text{declus.}}(t_e)}{S_e^{\text{declus.}}} \tag{6.88}$$

$$= \frac{m-e+1}{n-e}a_{e-1}\tau_{e-1}. \tag{6.89}$$

There are now $n-e$ surviving nodes in the system, each containing equal amounts of the codeword symbols corresponding to the most-exposed data. So, the failure of any of these nodes during the rebuild period $R_e^{\text{declus.}}$ will cause the system to enter exposure level $e+1$. Therefore, the number of nodes whose failure during rebuild can cause a transition to the next exposure level is given by

$$\tilde{n}_e^{\text{declus.}} = n - e. \tag{6.90}$$

Declustered Codeword Placement: MTTDL under Model A

Recall that, under model A, following each exposure level transition, the system is assumed to reconfigure its rebuild process entirely to rebuild the most-exposed data blocks in the new exposure level. This model may be applicable for a clustered placement scheme, where the l nodes from which data was being rebuilt failed and hence the system has to rebuild from another set of l node in the cluster. However, in a declustered placement scheme, where the distributed rebuild was being done from all n nodes, the failure of one node may not significantly affect the rebuild process in a large system. Therefore, model B may be better suited for the declustered placement scheme than model A. Nonetheless, we will derive the expressions for declustered placement scheme under model A for the sake of completeness.

Having computed the key quantities $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ for $e = 1, \cdots, \tilde{r}-1$, we are now ready to compute $P_{DL}^{\text{declus.}}$ using the expression (6.23) for model A, and then $\text{MTTDL}^{\text{declus.}}$ using (3.14). By substituting the values of $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ from (6.72), (6.89), and (6.90) into (6.23), we obtain

$$P_{DL}^{\text{declus.}} \approx \lambda^{\tilde{r}-1} \times (n-1)\cdots(n-\tilde{r}+1) \times \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-1}} \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \Bigg(\tau_1 \cdots \tau_{\tilde{r}-1} \times g_{\frac{n-1}{l+1}\mu}(\tau_1)\cdots g_{\frac{n-\tilde{r}+1}{(m-\tilde{r}+2)a_{\tilde{r}-2}\tau_{\tilde{r}-2}}}(\tau_{\tilde{r}-1}) d\vec{a} d\vec{\tau} \Bigg) \quad \text{(model A)} \tag{6.91}$$

As in (6.23) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, \tilde{r}-1$, and from 0 to 1 for a_e, $e = 1, \cdots, \tilde{r}-2$.

(l, m)-MDS codes with $m - l \leq 2$: Similar to the case of clustered placement, the expression (6.91) for $P_{DL}^{\text{declus.}}$ under model A cannot, in general, be further simplified without considering a particular family of rebuild distributions G_μ. However, for $m - l \leq 2$, or equivalently, for $r \leq 3$, a closed form expression for $P_{DL}^{\text{declus.}}$, and hence $\text{MTTDL}^{\text{declus.}}$, can be obtained under model A. This is illustrated by deriving the closed form expression for $m - l = 2$ by

substituting $\tilde{r} = 3$ in (6.91) and simplifying as follows.

$$P_{DL}^{\text{declus.}} \approx \lambda^2 \times (n-1)(n-2) \times \int_{\tau_1=0}^{\infty} \int_{\tau_2=0}^{\infty} \int_{a_1=0}^{1} \Bigg(\tau_1 \tau_2 \times g_{\frac{n-1}{l+1}\mu}(\tau_1) g_{\frac{n-2}{(m-1)a_1\tau_1}}(\tau_2) da_1 d\tau_2 d\tau_1 \Bigg) \tag{6.92}$$

$$= (n-1)(n-2)\lambda^2 \Bigg(\int_{\tau_1=0}^{\infty} \tau_1 g_{\frac{n-1}{l+1}\mu}(\tau_1) \int_{a_1=0}^{1} \int_{\tau_2=0}^{\infty} \Big(\tau_2 \times g_{\frac{n-2}{(m-1)a_1\tau_1}}(\tau_2) \Big) d\tau_2 da_1 d\tau_1 \Bigg)$$
$$\text{for } m-l=2 \text{ (model A).} \tag{6.93}$$

Noting that

$$\int_{\tau_2=0}^{\infty} \tau_2 g_{\frac{n-2}{(m-1)a_1\tau_1}}(\tau_2) d\tau_2 = \frac{(m-1)a_1\tau_1}{n-2}, \tag{6.94}$$

we get

$$P_{DL}^{\text{declus.}} \approx (n-1)(m-1)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_{\frac{n-1}{l+1}\mu}(\tau_1) \int_{a_1=0}^{1} a_1 da_1 d\tau_1 \tag{6.95}$$

$$= \frac{1}{2}(n-1)(m-1)\lambda^2 \int_{\tau_1=0}^{\infty} \tau_1^2 g_{\frac{n-1}{l+1}\mu}(\tau_1) d\tau_1 \tag{6.96}$$

$$= \frac{1}{2}(n-1)(m-1)\lambda^2 M_2\left(G_{\frac{n-1}{l+1}\mu}\right) \quad \text{for } m-l=2 \text{ (model A),} \tag{6.97}$$

where $M_2\left(G_{\frac{n-1}{l+1}\mu}\right)$, as defined in (6.25), denotes the second raw moment of the rebuild distribution $G_{\frac{n-1}{l+1}\mu}$. The expression for $\text{MTTDL}^{\text{declus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{declus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{declus.}}} \tag{6.98}$$

$$\approx \frac{2}{n(n-1)(m-1)\lambda^3 M_2\left(G_{\frac{n-1}{l+1}\mu}\right)} \quad \text{for } m-l=2 \text{ (model A).} \tag{6.99}$$

Multipliying and dividing (6.99) by square of the mean of the rebuild time distribution $G_{\frac{n-1}{l+1}\mu}$,

$$M_1^2\left(G_{\frac{n-1}{l+1}\mu}\right) = \left(\frac{l+1}{(n-1)\mu}\right)^2 \tag{6.100}$$

we get

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{n\lambda^3}\frac{2}{(m-1)(l+1)^2}\frac{M_1^2\left(G_{\frac{n-1}{l+1}\mu}\right)}{M_2\left(G_{\frac{n-1}{l+1}\mu}\right)} \tag{6.101}$$

$$= \frac{(n-1)\mu^2}{n\lambda^3}\frac{2}{(m-1)^3}\frac{M_1^2\left(G_{\frac{n-1}{l+1}\mu}\right)}{M_2\left(G_{\frac{n-1}{l+1}\mu}\right)} \quad \text{for } m-l=2 \text{ (model A).} \tag{6.102}$$

For deterministic rebuild times, the second raw moment, $M_2\left(G_{\frac{n-1}{l+1}\mu}\right)$, is equal to the square of the first raw moment, $M_1^2\left(G_{\frac{n-1}{l+1}\mu}\right)$, and therefore, the term $M_1^2\left(G_{\frac{n-1}{l+1}\mu}\right)/M_2\left(G_{\frac{n-1}{l+1}\mu}\right)$ evaluates to one. However, if the rebuild times are random, the second raw moment is always greater than the square of the first raw moment by Jensen's inequality, and therefore, the term $M_1^2\left(G_{\frac{n-1}{l+1}\mu}\right)/M_2\left(G_{\frac{n-1}{l+1}\mu}\right)$ is smaller than one. The closed form expression for $m-l=1$ can be derived similarly and is given by

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu}{n\lambda^2}\frac{1}{m} \qquad \text{for } m-l=1 \text{ (model A).} \tag{6.103}$$

(l,m)-MDS codes with $m-l>2$: For (l,m)-MDS codes with $m-l>2$, or equivalently, for $\tilde{r}>3$, the evaluation of $P_{Dl}^{\text{declus.}}$ under model A involves computing the expectations of functions involving higher raw moments of G_μ, which cannot be done without considering a particular family of rebuild distributions. However, given a particular family of rebuild distributions, the derivation of MTTDL involves successively evaluating the integrals in (6.91) to compute $P_{DL}^{\text{declus.}}$, and then using (3.14) to obtain $\text{MTTDL}^{\text{declus.}}$.

Declustered Codeword Placement: MTTDL under Model B

In contrast to model A, we assume in model B that, following an exposure level transition, the system has to do little or no reconfiguration of the rebuild process to rebuild the most-exposed data in the new exposure level. This is the case in a declustered placement scheme, where the rebuild was being done from all nodes, and therefore, in a large system, the failure of one node does not signficantly affect the rebuild process. This impliles that the rebuild time in the new exposure level is completely determined by the rebuild time and the fraction of most-exposed data not rebuilt in the previous exposure level.

By substituting the values of $1/\mu_e^{\text{declus.}}$ and $\tilde{n}_e^{\text{declus.}}$ from (6.72), (6.89), and (6.90) into (6.24), we obtain

$$P_{DL}^{\text{declus.}} \approx \lambda^{\tilde{r}-1} \times (n-1)\cdots(n-\tilde{r}+1) \times \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-1}} \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \left(\tau_1 \cdots \tau_{\tilde{r}-1} g_{\frac{n-1}{l+1}\mu}(\tau_1) \times \delta\left(\tau_2 - \frac{m-1}{n-2}a_1\tau_1\right)\cdots\delta\left(\tau_{\tilde{r}-1} - \frac{m-\tilde{r}+2}{n-\tilde{r}+1}a_{\tilde{r}-2}\tau_{\tilde{r}-2}\right) d\vec{a}d\vec{\tau} \right) \quad \text{(model B).} \quad (6.104)$$

As in (6.24) the integrals are from 0 to ∞ for τ_e, $e = 1, \cdots, \tilde{r}-1$, and from 0 to 1 for a_e, $e = 1, \cdots, \tilde{r}-2$. In contrast to model A, closed form expressions in terms of the raw moments of the rebuild distribution can be obtained for model B as follows. By changing the order of integrals in (6.104), we obtain

$$P_{DL}^{\text{declus.}} \approx \lambda^{\tilde{r}-1} \times (n-1)\cdots(n-\tilde{r}+1) \times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-1}} \left(\tau_1 \cdots \tau_{\tilde{r}-1} g_{\frac{n-1}{l+1}\mu}(\tau_1) \times \delta\left(\tau_2 - \frac{m-1}{n-2}a_1\tau_1\right)\cdots\delta\left(\tau_{\tilde{r}-1} - \frac{m-\tilde{r}+2}{n-\tilde{r}+1}a_{\tilde{r}-2}\tau_{\tilde{r}-2}\right) \times d\tau_{\tilde{r}-1}\cdots d\tau_1 d\vec{a} \right) \quad (6.105)$$

$$= \lambda^{\tilde{r}-1} \times (n-1)\cdots(n-\tilde{r}+1) \times \frac{m-\tilde{r}+2}{n-\tilde{r}+1} \times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-2}} \left(\tau_1 \cdots \tau_{\tilde{r}-2}^2 a_{\tilde{r}-2} g_{\frac{n-1}{l+1}\mu}(\tau_1) \times \delta\left(\tau_2 - \frac{m-1}{n-2}a_1\tau_1\right)\cdots\delta\left(\tau_{\tilde{r}-2} - \frac{m-\tilde{r}+3}{n-\tilde{r}+2}a_{\tilde{r}-3}\tau_{\tilde{r}-3}\right) \times d\tau_{\tilde{r}-2}\cdots d\tau_1 d\vec{a} \right) \quad (6.106)$$

$$= \lambda^{\tilde{r}-1} \times (n-1)\cdots(n-\tilde{r}+1) \times \frac{(m-\tilde{r}+2)}{(n-\tilde{r}+1)} \times \frac{(m-\tilde{r}+3)^2}{(n-\tilde{r}+2)^2} \times \int_{a_1}\cdots\int_{a_{\tilde{r}-2}} \int_{\tau_1}\cdots\int_{\tau_{\tilde{r}-3}} \left(\tau_1 \cdots \tau_{\tilde{r}-3}^3 a_{\tilde{r}-3}^2 a_{\tilde{r}-2} g_{\frac{n-1}{l+1}\mu}(\tau_1) \times \delta\left(\tau_2 - \frac{m-1}{n-2}a_1\tau_1\right)\cdots\delta\left(\tau_{\tilde{r}-3} - \frac{m-\tilde{r}+4}{n-\tilde{r}+3}a_{\tilde{r}-4}\tau_{\tilde{r}-4}\right) \times d\tau_{\tilde{r}-3}\cdots d\tau_1 d\vec{a} \right) \quad (6.107)$$

$$\vdots$$

$$= \lambda^{\tilde{r}-1} \times (n-1)^{\tilde{r}-1} \times \prod_{e=1}^{\tilde{r}-2} \left(\frac{m-e}{n-e}\right)^{\tilde{r}-e-1} \times \int_{a_1} \cdots \int_{a_{\tilde{r}-2}} \int_{\tau_1} \tau_1^{\tilde{r}-1} a_1^{\tilde{r}-2} \cdots a_{\tilde{r}-3}^2 a_{\tilde{r}-2} g_{\frac{n-1}{l+1}\mu}(\tau_1) d\tau_1 d\vec{a}$$

(model B). (6.108)

Here, steps (6.106)–(6.108) follow by successively integrating over $\tau_{\tilde{r}-1}, \cdots, \tau_2$, using the Dirac delta function's property, cancelling out terms of the form $(n-\tilde{r}+e)$, $e = 1, \cdots, \tilde{r}-2$, and rewriting the terms outside the integral after multiplying and dividing by $(n-1)^{\tilde{r}-1}$. Changing the order of the integrals and integrating out $a_1, \cdots, a_{\tilde{r}-2}$, we get

$$P_{DL}^{\text{declus.}} \approx \lambda^{\tilde{r}-1}(n-1)^{\tilde{r}-1} \prod_{e=1}^{\tilde{r}-2} \left(\frac{m-e}{n-e}\right)^{\tilde{r}-e-1} \int_{\tau_1} \tau_1^{\tilde{r}-1} g_{\frac{n-1}{l+1}\mu}(\tau_1) \frac{1}{(\tilde{r}-1)!} d\tau_1$$

$$= \lambda^{\tilde{r}-1} M_{\tilde{r}-1}\left(G_{\frac{n-1}{l+1}\mu}\right) \frac{(n-1)^{\tilde{r}-1}}{(\tilde{r}-1)!} \prod_{e=1}^{\tilde{r}-2} \left(\frac{m-e}{n-e}\right)^{\tilde{r}-e-1} \quad (6.109)$$

$$= \lambda^{m-l} M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right) \frac{(n-1)^{m-l}}{(m-l)!} \prod_{e=1}^{m-l-1} \left(\frac{m-e}{n-e}\right)^{m-l-e}$$

(model B), (6.110)

where $M_{m-l}(G_{\frac{n-1}{l+1}\mu})$, as defined in (6.25), denotes the $(m-l)$th raw moment of the rebuild distribution $G_{\frac{n-1}{l+1}\mu}$. Here, the last step was obtained by substituting for $\tilde{r}$ from its definition (6.1), that is, $\tilde{r} = m-l+1$. The expression for $\text{MTTDL}^{\text{declus.}}$ then follows from (3.14):

$$\text{MTTDL}^{\text{declus.}} \approx \frac{1}{n\lambda P_{DL}^{\text{declus.}}} \quad (6.111)$$

$$\approx \frac{1}{n\lambda^{m-l+1} M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)} \frac{(m-l)!}{(n-1)^{m-l}} \prod_{e=1}^{m-l-1} \left(\frac{n-e}{m-e}\right)^{m-l-e}$$

(model B). (6.112)

Multiplying and dividing (6.112) by the $(m-l)$th power of the mean of the rebuild time distribution $G_{\frac{n-1}{l+1}\mu}$,

$$M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right) = \left(\frac{l+1}{(n-1)\mu}\right)^{m-l} \quad (6.113)$$

we get

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)}{M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)} \frac{(m-l)!}{(l+1)^{m-l}} \prod_{e=1}^{m-l-1} \left(\frac{n-e}{m-e}\right)^{m-l-e} \quad \text{(model B).} \quad (6.114)$$

For deterministic rebuild times, the $(m-l)$th raw moment, $M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$, is equal to the $(m-l)$th power of the first raw moment, $M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$, and therefore, the term $M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)/M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$ evaluates to one. For random rebuild times, by the Jensen's inequality, the $(m-l)$th raw moment, $M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$, is always greater than the $(m-l)$th power of the first raw moment, $M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$, and therefore, their ratio evaluates to less than one.

As an example, if $G_{\frac{n-1}{l+1}\mu}$ is exponential, the expression for $\text{MTTDL}^{\text{declus.}}$ reduces to the following:

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{1}{(l+1)^{m-l}} \prod_{e=1}^{m-l-1} \left(\frac{n-e}{m-e}\right)^{m-l-e} \quad \text{when } G_{\frac{n-1}{l+1}\mu} \text{ is exponential (model B).} \quad (6.115)$$

Remark 6.12. *The expressions (6.102), (6.103), (6.114), and (6.115) for the mean time to data loss of a storage system with declustered codeword placement scheme under both models A and B are seen to be invariant within the class of node failure distributions that satisfy (2.10) and (2.11). In particular, the MTTDL only depends on the mean of the times to node failure, $1/\lambda$. As the conditions (2.10) and (2.11) hold true for real-world storage nodes as well, these MTTDL results are of practical significance.*

Remark 6.13. *The expressions (6.102), (6.114), and (6.115) for the mean time to data loss of a storage system with declustered codeword placement scheme also reveal that the MTTDL is sensitive to the rebuild distribution $G_{\frac{n-1}{l+1}\mu}$. It is observed that deterministic rebuild times have higher MTTDL values compared to random rebuild times. This is because the terms of the form $M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)/M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)$ are upper-bounded by 1 due to the Jensen's inequality, and this bound is achieved for deterministic rebuild times. The explanation for this fact is that, when rebuild times are random, given that a failure occurred during rebuild, it is more probable that the rebuild time was larger. This effect is known as the waiting time paradox. Larger rebuild times imply that a larger amount of most-exposed data remains unrebuilt when the system enters a higher exposure level, thereby reducing the reliability.*

Remark 6.14. *Comparing the MTTDL values of clustered placement in (6.67) with those of declustered placement in (6.114), we observe that they are both directly proportional to the $(m-l+1)$th power of the mean time to node failure $1/\lambda$, and inversely proportional to the $(m-l)$ power of the mean time to node rebuild $1/\mu$. This is a general trend in the MTTDL behavior of data storage systems. However, in contrast to clustered placement, the MTTDL of declustered placement is observed to scale differently with the number of nodes, n, for different values of $\tilde{r} = m-l+1$. It can be seen from (6.114) that the MTTDL of declustered placement scales roughly as the $(\tilde{r}(\tilde{r}-3)/2)$th power of n, or equivalently as the $(m-l+1)(m-l-2)/2)$th power of n. For $m-l=1$, the MTTDL of declustered placement scales inversely proportional to n, just like in clustered placement. For $m-l=2$, the MTTDL of declustered placement stays roughly constant with n. For $m-l>2$, the MTTDL of declustered placement increases with n. This shows that, by changing the codeword placement scheme, one can influence the scaling of MTTDL with respect to the number of nodes n, resulting in a tremendous improvement in reliability for large storage systems.*

6.4.3 Other Symmetric Codeword Placement Schemes

As discussed in Section 2.4, a broader set of symmetric placement schemes can be defined using the concept of *spread factor*. For each node in the system, its redundancy spread factor is defined as the number of nodes over which the data on that node and its corresponding redundant data are spread. In an erasure coded system, when a node fails, its spread factor determines the number of nodes which have the codeword symbols corresponding to the lost data, and this in turn determines the degree of parallelism that can be used in rebuilding the data lost by that node. In this thesis, we will consider symmetric placement schemes for which the spread factor of each node is the same, denoted by k. In a symmetric placement scheme, the $m-1$ codeword symbols corresponding to the data on each node are equally spread across $k-1$ other nodes, the $m-2$ codeword symbols corresponding to the codewords shared by any two nodes are equally spread across $k-2$ other nodes, and so on. One example of such a symmetric placement scheme is the clustered placement scheme for which the spread factor, k, is equal to the codeword length, m. Another example of a symmetric placement scheme is the declustered placement scheme for which the spread factor, k, is equal to the number of nodes, n. A number of different placement schemes can be generated by varying the spread factor k between m and n. By similar arguments and computations as in the previous subsection, the MTTDL of a scheme with spread factor $k > m$, denoted by, MTTDL(k), is given by

$$\text{MTTDL}(k) \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{M_1^{m-l}\left(G_{\frac{k-1}{l+1}\mu}\right)}{M_{m-l}\left(G_{\frac{k-1}{l+1}\mu}\right)} \frac{(m-l)!}{(l+1)^{m-l}} \prod_{e=1}^{m-l-1} \left(\frac{k-e}{m-e}\right)^{m-l-e}$$
$$\text{for } k = m+1, \cdots, n \text{ (model B).} \quad (6.116)$$

Spread factor $k = m$ corresponds to clustered placement and therefore, its MTTDL is given by (6.67), that is,

$$\text{MTTDL}(m) \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{1}{\binom{m-1}{l-1}} \frac{M_1^{m-l}(G_\mu)}{M_{m-l}(G_\mu)} \qquad \text{(model B)}. \quad (6.117)$$

The difference in the derivation, and hence the final expressions, of MTTDL for $k > m$ and for $k = m$ stems from the assumptions made on the rebuild process. For a system with spread factor $k > m$, we assume that a distributed rebuild process is used that involves, at each exposure level e, reading the required codeword symbols from $k - e$ nodes, computing the most-exposed codeword symbols, and writing them to the spare space of these $k - e$ nodes in such a way that no codeword symbol is written to a node in which another codeword symbol corresponding to the same codeword is already present. However, when the spread factor $k = m$, such a rebuild process cannot be done because it is not possible to write the reconstructed codeword symbols to the spare space of the $m - e$ nodes in such a way that no codeword symbol is written to a node in which another codeword symbol corresponding to the same codeword is already present. Therefore, it assumed that the reconstructed codeword symbols are written to a new replacement node directly. This affects the rate of rebuild processes thereby affecting the derivation and the final expressions of MTTDL.

6.5 Placement, Storage Efficiency, and Reliability

In this section, we compare the MTTDLs of (l, m)-MDS code based systems for clustered and declustered placement schemes for various choice of parameters l and m with the help of figures. Typically, the MDS codes used for data storage are such that l out of m codeword symbols of each codeword are exactly the same as the l blocks of user data used to encode the codeword, and the $m - l$ remaining codeword symbols are referred to as *parities*. The number of parities in a codeword, namely, $m - l$, is an important factor that determines the performance as well as reliability of a storage system. Any update to a user data block may require the system to read the parities of the codeword corresponding to that data block, recompute these parities, and update these parities along with the user data block. Therefore, the number of parities directly affect the number of IO operations to be performed at each update. A larger number of parities will require a larger number of IO operations which in turn negatively affects the performance of the system. Secondly, the number of parities, $m - l$, affects the reliability of the system as it directly relates to the number of node failures that need to occur to cause data loss. In addition, the number of parities also affects the reliability *behavior* of various placement schemes as the number of nodes in the system increases. Therefore, we will

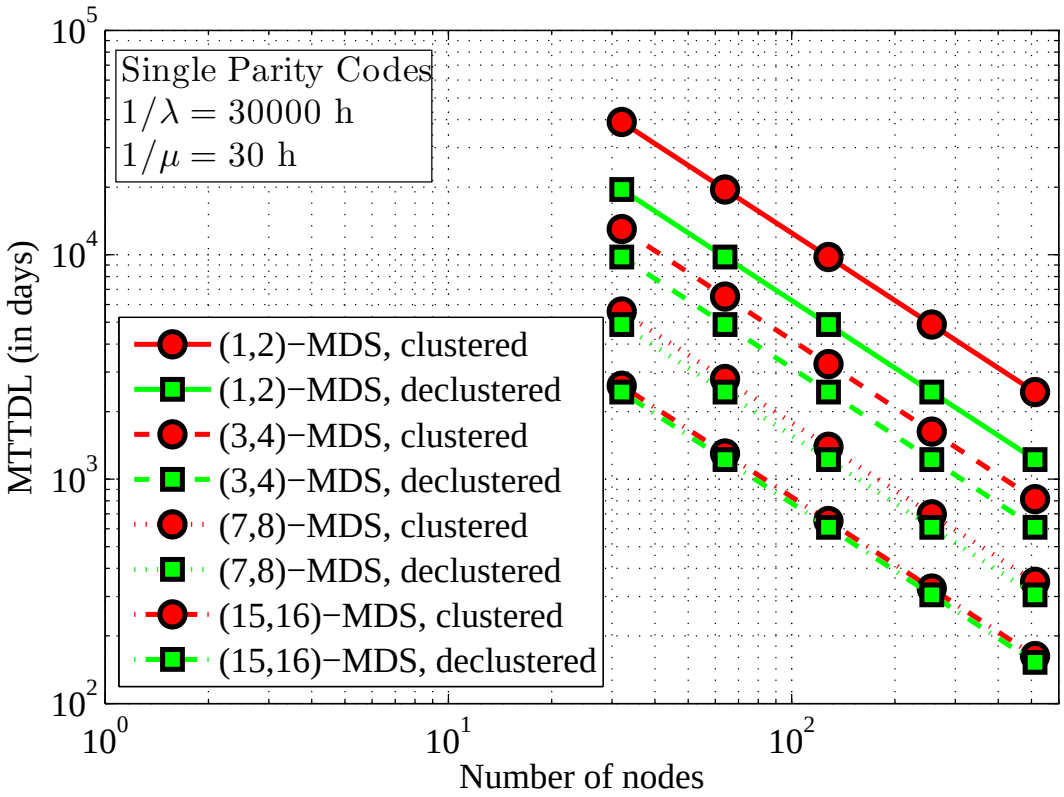


Figure 6.1: MTTDL of single parity codes vs. the number of nodes for mean time to node failure $1/\lambda = 30000$ h and mean time to read all contents of a node during rebuild $1/\mu = 30$ h.

compare the MTTDLs of various placement schemes for a given number of parities.

Note that models A and B, as described in Section 6.3.3, do not differ in the values of MTTDL for $m - l \leq 2$. Furthermore, the difference between models A and B for $m - l > 2$ is typically only a constant factor that depends on the rebuild distribution. Also, if the the rebuild times are deterministic, there is no difference between models A and B and therefore they agree on the MTTDL values for all $m - l \geq 1$. So, without loss of generality, we will only consider the MTTDL values under model B for further discussions in this section.

6.5.1 Single Parity Codes

Single parity (l, m)-MDS codes correspond to the case where $m - l = 1$. When $l = 1$, this corresponds to two-way replication. For higher values of l, this corresponds to RAID-5 [16]. Plugging $l = m - 1$ in (6.67) and (6.114), we obtain the MTTDL values for single parity codes for clustered and declustered

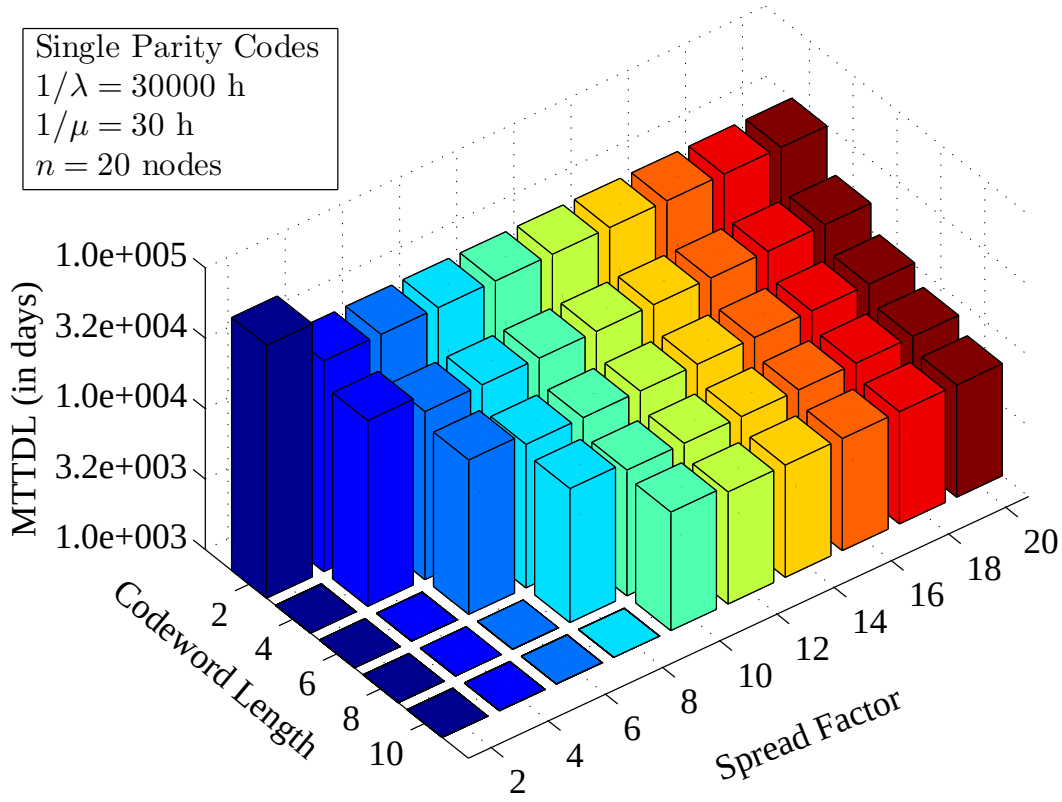


Figure 6.2: MTTDL of single parity codes as a function of codeword length m and spread factor k for a system with number of nodes $n = 20$.

placement schemes, respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu}{n\lambda^2}\frac{1}{m-1} \quad \text{for } m = 2, \cdots, n \text{ (single parity codes).} \quad (6.118)$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu}{n\lambda^2}\frac{1}{m} \quad \text{for } m = 2, \cdots, n \text{ (single parity codes).} \quad (6.119)$$

From (6.118) and (6.119), it is observed that the MTTDL of single parity codes under both placement schemes are directly proportional to the square of the mean time to node failure, $1/\lambda$, and inversely proportional to the mean time to read all contents of a node during rebuild, $1/\mu$. In addition, the MTTDL values are seen to be independent of the underlying rebuild distribution. The result (6.118) for clustered placement is well known since the 1980s when the reliability of RAID-5 systems were studied [16]. The difference in MTTDL between the two schemes is only a factor $m/(m-1)$ which is at most 2, when $m = 2$, that is, for two-way replicated systems. For higher values of m, the difference in MTTDL between clustered and declustered is smaller. as is illustrated by Figure 6.1. For a symmetric placement scheme with spread factor $k > m$, the MTTDL of single parity codes follows from (6.116) by

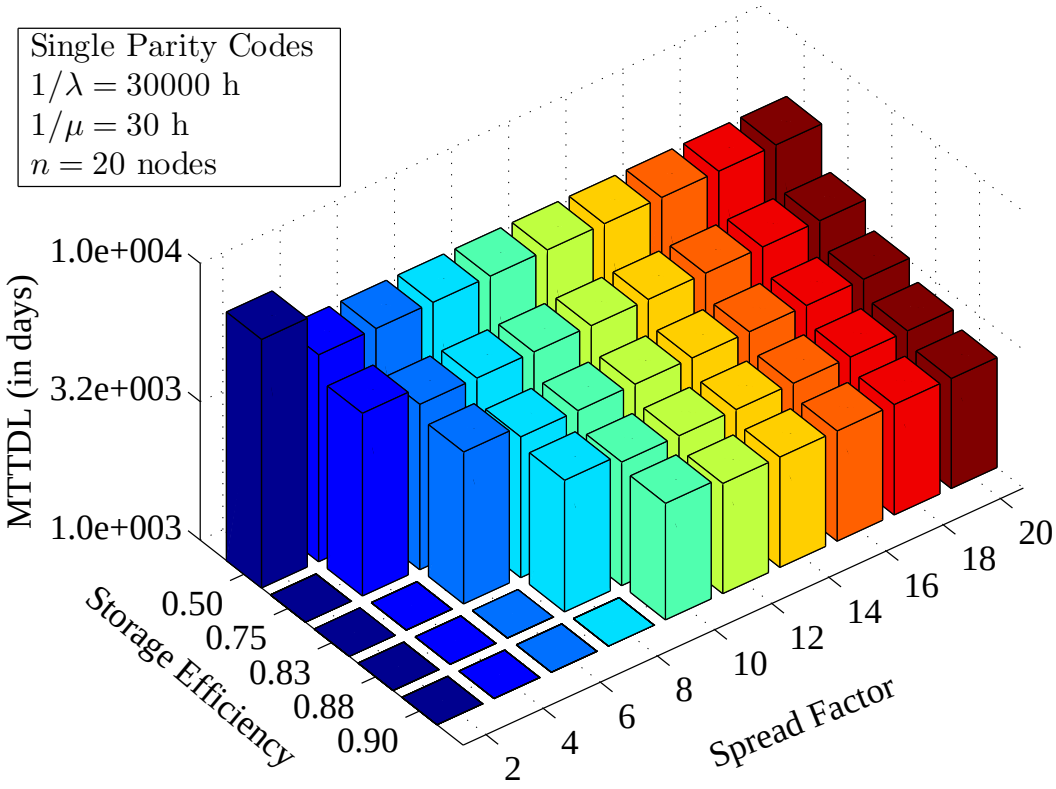


Figure 6.3: MTTDL of single parity codes as a function of storage efficiency $(m-1)/m$ and spread factor k for a system with number of nodes $n = 20$.

substituting $l = m - 1$:

$$\text{MTTDL}(k) \approx \frac{\mu}{n\lambda^2}\frac{1}{m} \qquad \text{for } k = m+1, \cdots, n, \text{ and } m = 2, \cdots, n \text{ (single parity codes).} \quad (6.120)$$

Spread factor $k = m$ corresponds to clustered placement scheme, and so its MTTDL is given by (6.118):

$$\text{MTTDL}(m) \approx \frac{\mu}{n\lambda^2}\frac{1}{m-1} \quad \text{for } m = 2, \cdots, n \text{ (single parity codes).} \quad (6.121)$$

It is observed from (6.120) that changing the spread factor k between $m+1$ and n does not have any effect on the MTTDL. This is because, although increasing factor speeds up the rebuild process as more number of nodes are involved in a parallel rebuild process, it also increases the number of nodes, whose failure during rebuild can cause some codewords to lose additional symbols, by a proportional amount. These effects cancel each other out resulting in an MTTDL that is unaffected by a change in spread factor. In fact, for replication factor 2, it can be shown that all possible placement schemes, not only symmetric placement schemes, have MTTDL values that differ by at most a factor two [15]. Figure 6.2 shows how the MTTDL varies as a function of both

the codeword length m and the spread factor k for single parity codes, for a given number of nodes n. In Figure 6.2, two-way replicated systems correspond to the case where the codeword length is 2, clustered placement corresponds to the cases where the spread factor is equal to the codeword length, and declustered placement corresponds to the case where the spread factor is equal to the number of nodes. It is observed that the clustered placement scheme has slightly higher MTTDL values than other placement schemes, and that increasing the codeword length decreases the MTTDL.

The storage efficiency of a redundancy scheme is equal to the ratio of the amount of user data to the actual amount of data stored in the system. A higher storage efficiency is desirable as it implies a lower cost of storage media. The storage efficiency of a system using a single parity code with codeword length m is equal to $(m-1)/m$, as each set of $m-1$ user data blocks requires storing m codeword blocks in the system. Therefore, Figure 6.2 is easily transformed into Figure 6.3 to show the MTTDL as a function of storage efficiency, $(m-1)/m$, and spread factor, k.

6.5.2 Double Parity Codes

Double parity (l, m)-MDS codes correspond to the case where $m - l = 2$. When $l = 1$, this corresponds to three-way replication. For higher values of l, this corresponds to RAID-6 [27]. Plugging $l = m-2$ in (6.67) and (6.114), we obtain the MTTDL values for double parity codes for clustered and declustered placement schemes, respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^2}{n\lambda^3} \frac{2}{(m-1)(m-2)} \frac{M_1^2(G_\mu)}{M_2(G_\mu)} \qquad \text{for } m = 3, \cdots, n \text{ (double parity codes).} \quad (6.122)$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{n\lambda^3} \frac{2}{(m-1)^3} \frac{M_1^2\left(G_{\frac{n-1}{m-1}\mu}\right)}{M_2\left(G_{\frac{n-1}{m-1}\mu}\right)} \qquad \text{for } m = 3, \cdots, n \text{ (double parity codes).} \quad (6.123)$$

From (6.122) and (6.123), it is observed that the MTTDL of double parity codes under both placement schemes are directly proportional to the cube of the mean time to node failure, $1/\lambda$, and inversely proportional to the square of the mean time to read all contents of a node during rebuild, $1/\mu$. In contrast to single parity codes, it is seen that the MTTDL depends on the rebuild distribution. For deterministic rebuild times, the ratios $M_1^2(G_\mu)/M_2(G_\mu)$ and $M_1^2\left(G_{\frac{n-1}{m-1}\mu}\right)/M_2\left(G_{\frac{n-1}{m-1}\mu}\right)$ become one. However, for random rebuild times, these ratios are upper-bounded by one by Jensen's inequality. As an example, if the rebuild time distribution was exponential, these ratios are equal to 1/2

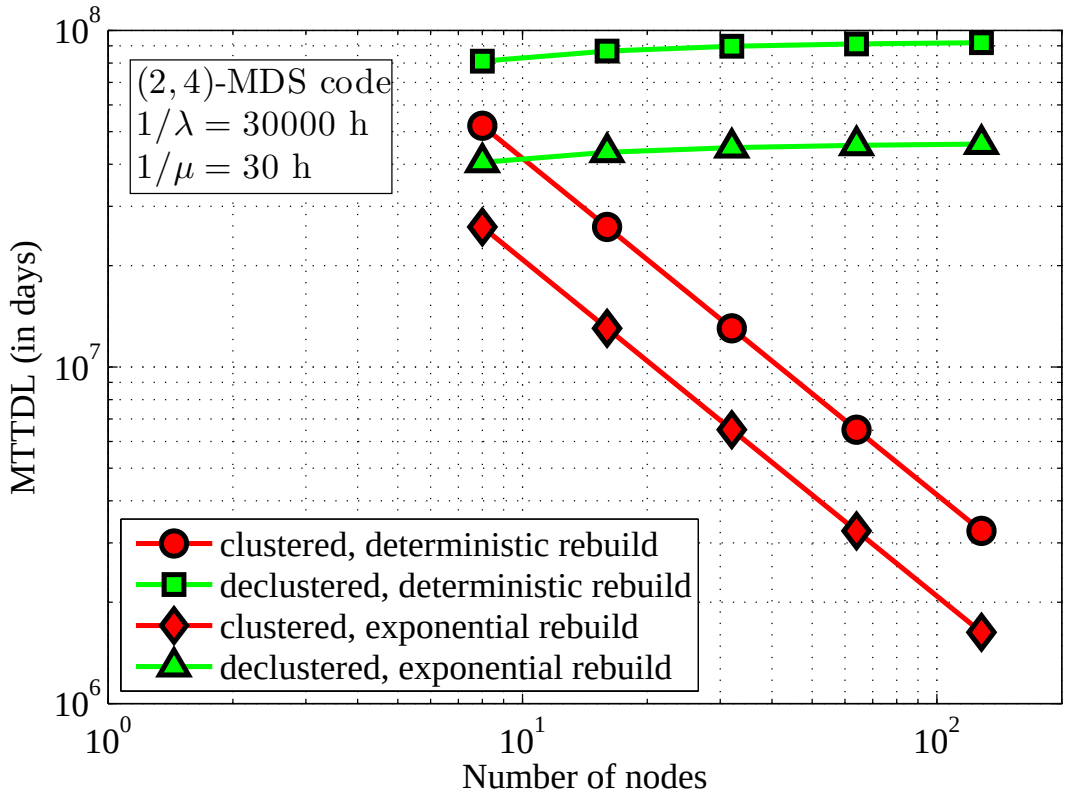


Figure 6.4: MTTDL of a $(2,4)$-MDS code vs. the number of nodes for mean time to node failure $1/\lambda = 30000$ h and mean time to read all contents of a node during rebuild $1/\mu = 30$ h.

and therefore

$$\mathrm{MTTDL}^{\mathrm{clus.}} \approx \frac{\mu^2}{n\lambda^3}\frac{1}{(m-1)(m-2)} \qquad \text{for } m = 3, \cdots, n$$
$$\text{(double parity codes, exponential rebuild times).} \quad (6.124)$$

$$\mathrm{MTTDL}^{\mathrm{declus.}} \approx \frac{(n-1)\mu^2}{n\lambda^3}\frac{1}{(m-1)^3} \qquad \text{for } m = 3, \cdots, n$$
$$\text{(double parity codes, exponential rebuild times).} \quad (6.125)$$

The result (6.124) for clustered placement is well known in the context of RAID-6 systems [27]. The MTTDL of a system using a $(2,4)$-MDS code is plotted against the number of nodes in the system for clustered and declustered placements, as well as for deterministic and exponential rebuild times, in Figure 6.4. It is observed that the rebuild time distribution scales down the MTTDL, but leaves the behavior with respect to the number of nodes, n, unaffected. This has also been verified by means of simulation in Chapter 7.

In contrast to single parity codes, the difference in MTTDL between the two schemes can be significant, depending on the number of nodes, n, in

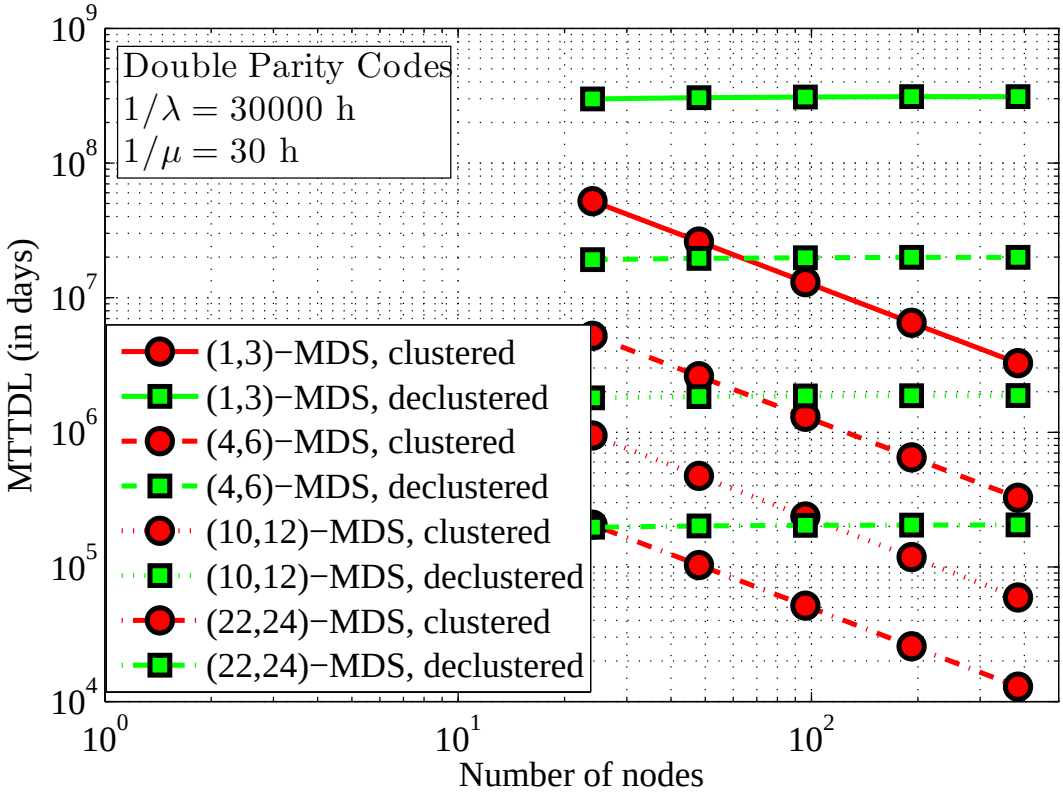


Figure 6.5: MTTDL of double parity codes vs. the number of nodes for mean time to node failure $1/\lambda = 30000$ h and mean time to read all contents of a node during rebuild $1/\mu = 30$ h.

the system. This is because, as seen from (6.122) and (6.123), the MTTDL of clustered placement is inversely proportional to n, whereas the MTTDL of declustered placement is roughly invariant with respect to n. This is illustrated in Figure 6.5 in which MTTDL of double parity codes is plotted against the number of nodes, n, in a log-log scale. The lines corresponding to clustered placement have a slope of -1 indicating that the MTTDL is inversely proportional to n, whereas the lines corresponding to declustered placement have a slope of roughly 0 indicating that the MTTDL is invariant with respect to n. It is also observed from Figure 6.5 that longer codes, which are more desirable as they have higher storage efficiency, can have better MTTDL with declustered placement than shorter codes with clustered placement for large systems. This is seen, for example, by observing the lines corresponding to $(4, 6)$-MDS code with declustered placement and $(1, 3)$-MDS code with clustered placement, for $n > 100$. Just like in the case of single parity codes, the difference in MTTDL between clustered and declustered is observed to be smaller for larger values of the codeword length, m.

For a symmetric placement scheme with spread factor $k > m$, the MTTDL

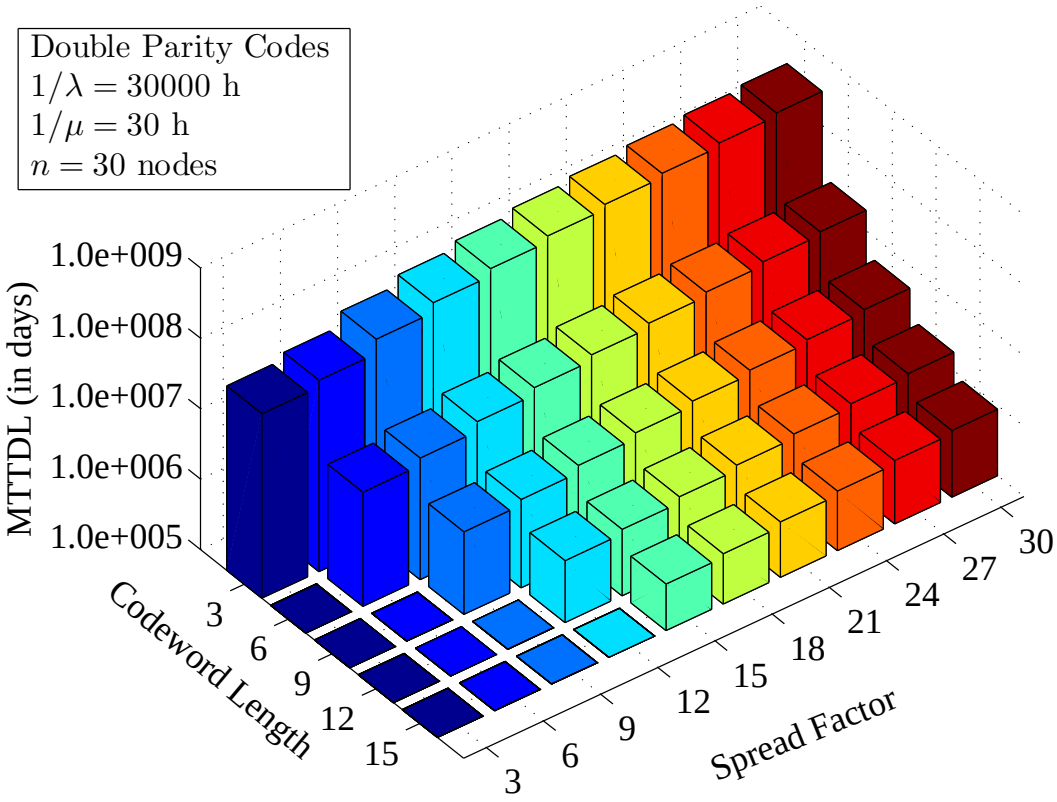


Figure 6.6: MTTDL of double parity codes as a function of codeword length m and spread factor k for a system with number of nodes $n = 20$.

of double parity codes follows from (6.116) by substituting $l = m - 2$:

$$\text{MTTDL}(k) \approx \frac{(k-1)\mu^2}{n\lambda^3} \frac{2}{(m-1)^3} \frac{M_1^2\left(G_{\frac{k-1}{m-1}\mu}\right)}{M_2\left(G_{\frac{k-1}{m-1}\mu}\right)} \quad \text{for } k = m+1, \cdots, n, \text{ and } m = 3, \cdots, n \text{ (double parity codes).} \quad (6.126)$$

Spread factor $k = m$ corresponds to clustered placement scheme, and so its MTTDL is given by (6.122):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^2}{n\lambda^3} \frac{2}{(m-1)(m-2)} \frac{M_1^2(G_\mu)}{M_2(G_\mu)} \quad \text{for } m = 3, \cdots, n \text{ (double parity codes).} \quad (6.127)$$

It is observed from (6.126) that, in contrast to single parity codes, increasing the spread factor k improves the MTTDL proportional to k. This is because, due to the spreading of codewords over more number of nodes, the amount of most-exposed data at each successive exposure level decreases rapidly, thereby reducing the chances of data loss. It is due to the fact that the amount of most-exposed data decreases at each exposure level, and not necessarily because

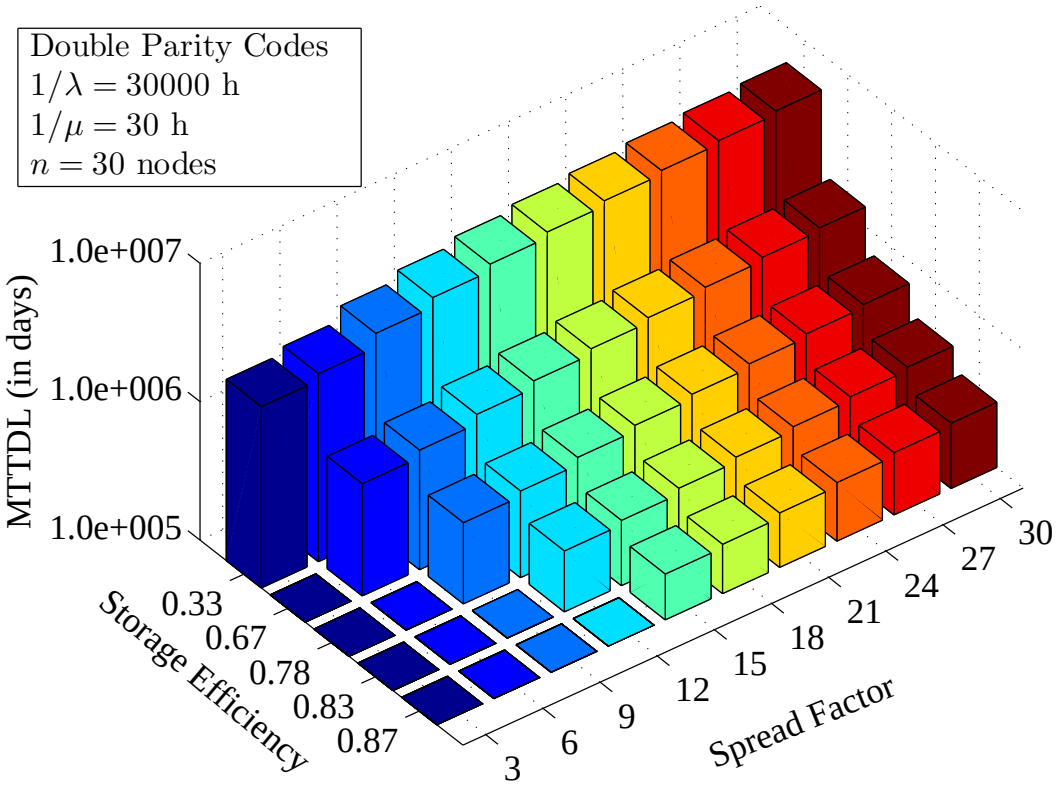


Figure 6.7: MTTDL of double parity codes as a function of storage efficiency $(m-2)/m$ and spread factor k for a system with number of nodes $n = 20$.

the rebuild times are much faster, that the MTTDL values increase with the spread factor k. Figure 6.6 shows how the MTTDL varies as a function of both the codeword length m and the spread factor k for double parity codes, for a given number of nodes, n. In Figure 6.6, three-way replicated systems correspond to the case where the codeword length is 3, clustered placement corresponds to the cases where the spread factor is equal to the codeword length, and declustered placement corresponds to the case where the spread factor is equal to the number of nodes. It is observed that increasing the spread factor increases the MTTDL, and that increasing the codeword length decreases the MTTDL.

The storage efficiency of a system using a double parity code with codeword length m is equal to $(m-2)/m$, as each set of $m-2$ user data blocks requires storing m codeword blocks in the system. Therefore, Figure 6.6 is easily transformed into Figure 6.7 to show the MTTDL as a function of storage efficiency, $(m-2)/m$, and spread factor, k.

6.5.3 Triple Parity Codes

Triple parity (l, m)-MDS codes correspond to the case where $m-l = 3$. When $l = 1$, this corresponds to four-way replication. Plugging $l = m-2$ in (6.67)

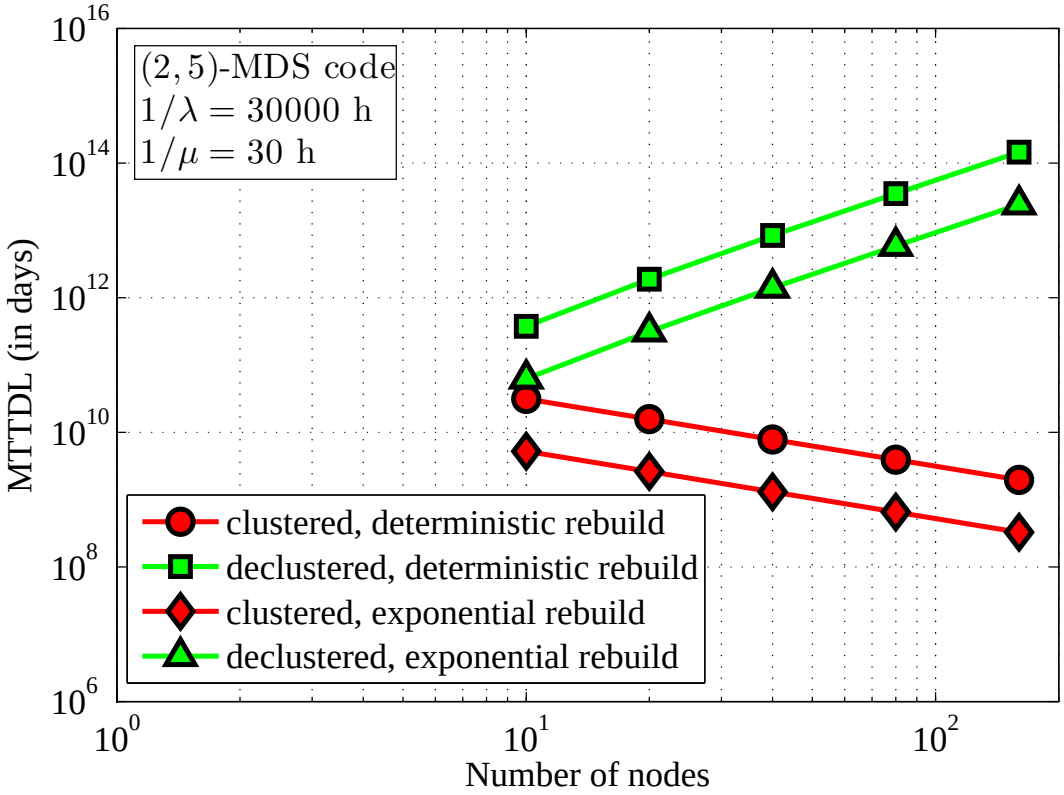


Figure 6.8: MTTDL of a $(2,5)$-MDS code vs. the number of nodes for mean time to node failure $1/\lambda = 30000$ h and mean time to read all contents of a node during rebuild $1/\mu = 30$ h.

and (6.114), we obtain the MTTDL values for triple parity codes for clustered and declustered placement schemes, respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^3}{n\lambda^4} \frac{6}{(m-1)(m-2)(m-3)} \frac{M_1^3(G_\mu)}{M_3(G_\mu)} \quad \text{for } m = 4, \cdots, n \text{ (triple parity codes).} \tag{6.128}$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)^2(n-2)\mu^3}{n\lambda^4} \frac{6}{(m-1)^2(m-2)^4} \frac{M_1^3\left(G_{\frac{n-1}{m-2}\mu}\right)}{M_3\left(G_{\frac{n-1}{m-2}\mu}\right)} \quad \text{for } m = 4, \cdots, n \text{ (triple parity codes).} \tag{6.129}$$

From (6.128) and (6.129), it is observed that the MTTDL of triple parity codes under both placement schemes are directly proportional to the fourth power of the mean time to node failure, $1/\lambda$, and inversely proportional to the cube of the mean time to read all contents of a node during rebuild, $1/\mu$. As was the case in double parity codes, the MTTDL depends on the rebuild distribution. For deterministic rebuild times, the ratios $M_1^3(G_\mu)/M_3(G_\mu)$ and

$M_1^3\left(G_{\frac{n-1}{m-2}\mu}\right)/M_2\left(G_{\frac{n-1}{m-2}\mu}\right)$ become one. However, for random rebuild times, these ratios are upper-bounded by one by Jensen's inequality. As an example, if the rebuild time distribution was exponential, these ratios are equal to 1/6 and therefore

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^3}{n\lambda^4}\frac{1}{(m-1)(m-2)(m-3)} \qquad \text{for } m=4,\cdots,n$$
$$\text{(triple parity codes, exponential rebuild times).} \quad (6.130)$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)^2(n-2)\mu^3}{n\lambda^4}\frac{1}{(m-1)^2(m-2)^4} \qquad \text{for } m=4,\cdots,n$$
$$\text{(triple parity codes, exponential rebuild times).} \quad (6.131)$$

Comparing (6.131) with (6.129), it is observed that the rebuild time distribution scales down the MTTDL, but leaves the behavior with respect to the number of nodes, n, unaffected. This can be seen in the plots of MTTDL of a system using a $(2,5)$-MDS code against the number of nodes in the system for clustered and declustered placements, as well as for deterministic and exponential rebuild times, in Figure 6.8. Also, as in the case of double parity codes, the difference in MTTDL between the two schemes can be significant, depending on the number of nodes, n, in the system. This is because, as seen from (6.128) and (6.129), the MTTDL of clustered placement is inversely proportional to n, whereas the MTTDL of declustered placement is roughly proportional to the square of n. This is illustrated in Figure 6.8 in which MTTDL is plotted against the number of nodes, n, in a log-log scale. The lines corresponding to clustered placement have a slope of -1 indicating that the MTTDL is inversely proportional to n, whereas the lines corresponding to declustered placement have a slope of roughly 2 indicating that the MTTDL is proportional to the square to n.

For a symmetric placement scheme with spread factor $k > m$, the MTTDL of triple parity codes follows from (6.116) by substituting $l = m-2$:

$$\text{MTTDL}(k) \approx \frac{(k-1)^2(k-2)\mu^3}{n\lambda^4}\frac{6}{(m-1)^2(m-2)^4}\frac{M_1^3\left(G_{\frac{k-1}{m-2}\mu}\right)}{M_3\left(G_{\frac{k-1}{m-2}\mu}\right)}$$
$$\text{for } k=m+1,\cdots,n \text{ and } m=4,\cdots,n \text{ (triple parity codes).} \quad (6.132)$$

Spread factor $k = m$ corresponds to clustered placement scheme, and so its MTTDL is given by (6.128):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^3}{n\lambda^4}\frac{6}{(m-1)(m-2)(m-3)}\frac{M_1^3(G_\mu)}{M_3(G_\mu)} \quad \text{for } m=4,\cdots,n$$
$$\text{(triple parity codes).} \quad (6.133)$$

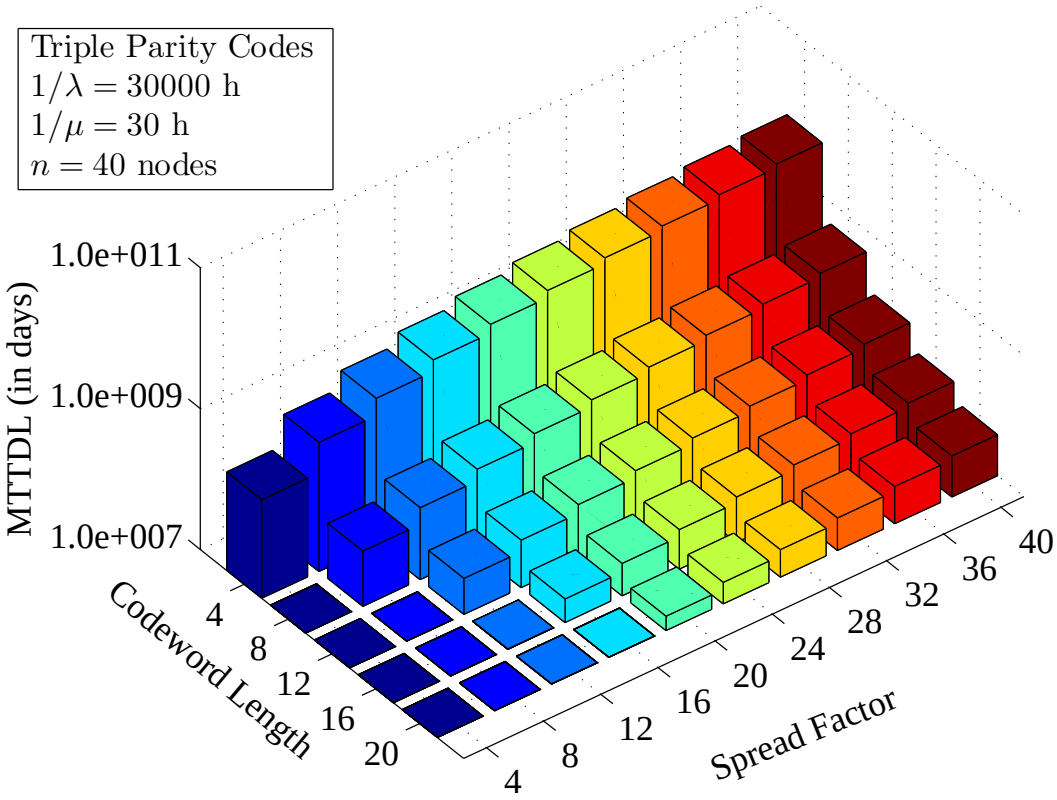


Figure 6.9: MTTDL of triple parity codes as a function of codeword length m and spread factor k for a system with number of nodes $n = 20$.

It is observed from (6.132) that, like in the case of double parity codes, increasing the spread factor k improves the MTTDL. The improvement, however, is much larger because the MTTDL is roughly proportional to the cube of k. The reason for this is the same as in the case of double parity codes, namely, due to the spreading of codewords over more number of nodes, the amount of most-exposed data at each successive exposure level decreases rapidly, thereby reducing the chances of data loss. Figure 6.9 shows how the MTTDL varies as a function of both the codeword length m and the spread factor k for triple parity codes, for a given number of nodes, n. In Figure 6.9, four-way replicated systems correspond to the case where the codeword length is 3, clustered placement corresponds to the cases where the spread factor is equal to the codeword length, and declustered placement corresponds to the case where the spread factor is equal to the number of nodes. It is observed that increasing the spread factor increases the MTTDL significantly, and that increasing the codeword length decreases the MTTDL significantly.

The storage efficiency of a system using a triple parity code with codeword length m is equal to $(m-3)/m$, as each set of $m-3$ user data blocks requires storing m codeword blocks in the system. Therefore, Figure 6.9 is easily transformed into Figure 6.10 to show the MTTDL as a function of storage efficiency,

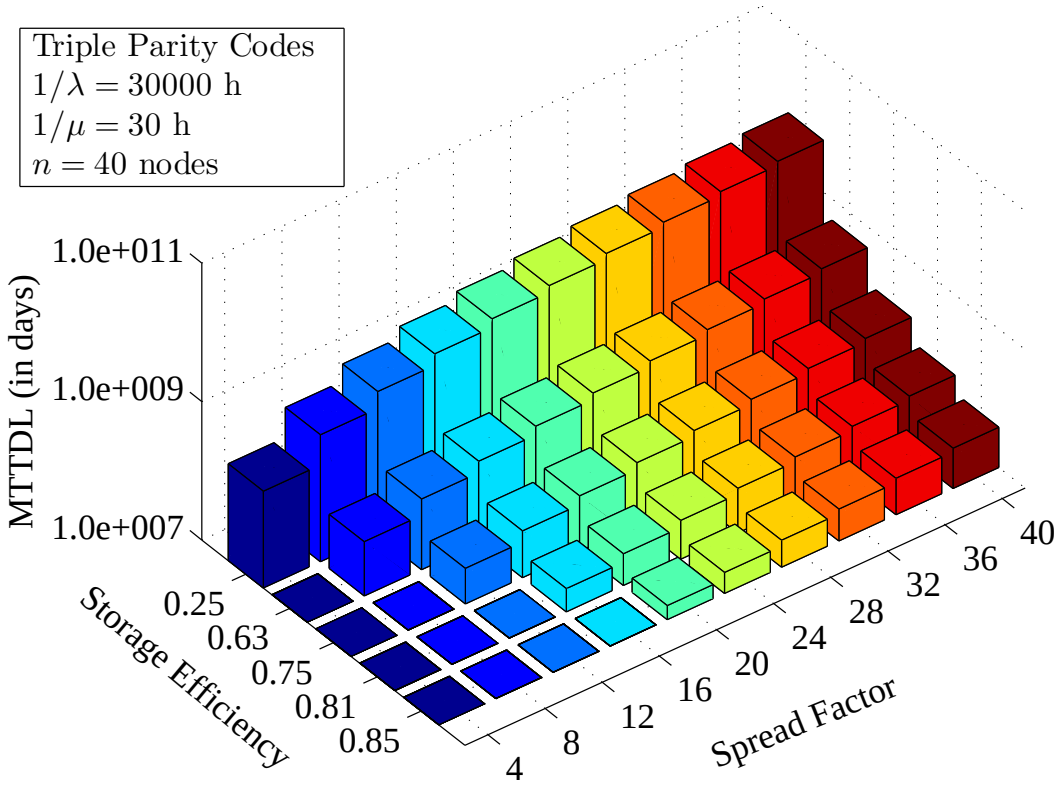


Figure 6.10: MTTDL of triple parity codes as a function of storage efficiency $(m-2)/m$ and spread factor k for a system with number of nodes $n = 20$.

$(m-3)/m$, and spread factor, k.

7 Reliability Simulations

Event-driven simulations are used to verify the theoretical estimates of MTTDL of replication-based systems for two placement schemes, namely, clustered and declustered, under various rebuild and failure time distributions, and under network rebuild bandwidth constraints. The simulations are more involved than the theoretical analysis as they do not make any of the approximations made in theory. Despite this fact, it is found that the simulations match theoretical estimates for a wide range of parameters, including the parameters generally observed in practice, thereby validating the applicability of the reliability analysis to real-world storage systems. A detailed description of the simulation method used and a comparison of simulation results and theory for a variety of storage system models is presented in this chapter.

7.1 Simulation Method

The storage system is simulated using an event-driven simulation with three types of events that drive the simulation time forward: *(a) failure events*, *(b) rebuild-complete events*, and *(c) node-restore events*. The state of the system is maintained by the following variables: `time`, the simulated time, `nActiveNodes`, the number of active (surviving) nodes in the system, `failTimes`, the times of next failure of each active node generated according to the distribution F_λ, `failedNodes`, the indices of all failed nodes, `exposureLevel`, the exposure level, and a vector of length $(r+1)$, `dataExposure` $= (D_0, \cdots, D_r)$, where D_e is the amount of user data that have lost e replicas, $e = 1, \cdots, r$. The values of these variables are updated at each event, and when $D_r > 0$, data loss is said to have occurred and the simulation ends.

For each set of parameters, the simulation is run 100 times, and the MTTDL and its 95% confidence intervals are computed. Whereas for declustered place-

ment, the simulation is run for n nodes, for clustered placement, the simulations are run only for one cluster, that is, r nodes, and the obtained MTTDL of the cluster is divided by n/r to obtain the MTTDL of the system. This is because clusters are independent of each other and the number of clusters is n/r.

7.1.1 Failure Event

Besides updating `time`, a failure event triggers the following: (i) decreasing `activeNodes` by one and increasing `exposureLevel` by one (recall that, for the declustered scheme, any node failure causes an exposure level transition, and that, for the clustered scheme, only one cluster is being simulated and therefore any node failure in that cluster causes an exposure level transition), (ii) scheduling the next failure event based on `failTimes`, (iii) updating `dataExposure` by taking partial rebuild of the most exposed data into account, and (iv) scheduling the rebuild-complete event based on the most exposed data in `dataExposure`, the placement scheme used (which determines the parallelism that can be exploited and therefore the speed of rebuild), network rebuild bandwidth limitations, and the rebuild distribution. By the nature of the rebuild process, data placement is preserved, that is, declustered remains declustered and clustered remains clustered. This is because, when the placement is declustered, critical blocks are read from and written to all nodes at the same time and the new replicas are placed such that declustering is preserved. When the placement is clustered, the replicas are created in a new node directly which preserves the placement. A main difference between declustered and clustered placement is how the data exposure vector changes at each exposure level transition. It was shown in the previous chapters that the main reason for declustered placement to have higher reliability is the fact that the amount of most-exposed data at each exposure level decreases significantly as the system enters higher exposure levels. Therefore, proper computation of data exposure vector at each exposure level transition for declustered placement is an important step in its reliability simulation. Whereas the computation of data exposure vector for clustered placement is fairly straightforward, the computation of data exposure vector for declustered placement is more involved.

Data Exposure Vector for Declustered Placement

For declustered placement at exposure level e, when a failure occurs, the data exposure vector, `dataExposure`, is updated from $(D_0, D_1, \cdots, D_e, 0, \cdots, 0)$ to $(D'_0, D'_1, \cdots, D'_e, D'_{e+1}, 0, \cdots, 0)$ as follows. Let $\tilde{n}$ denote the number of active nodes in the system at exposure level e. For $j = 0, \cdots, e-1$, the amount of user data that has $r-j$ surviving replicas in exposure level e is equal to D_j. These $r-j$ replicas are equally spread across the $\tilde{n}$ surviving nodes of the system due to the nature of declustered placement and distributed rebuild.

Therefore, when an additional node failure occurs, $\frac{r-j}{\tilde{n}}D_j$ loses its $(j+1)$th replica, for $j = 0, \cdots, e-1$. So, D'_j, $j = 0, \cdots, e-2$ is given by

$$D'_0 = D_0 - \frac{r}{\tilde{n}}D_0, \tag{7.1}$$

$$D'_j = D_j - \frac{r-j}{\tilde{n}}D_j + \frac{r-j+1}{\tilde{n}}D_{j-1}, \quad \text{for } j = 1, \cdots, e-2, \tag{7.2}$$

If α denotes the fraction of rebuild time at exposure level e still left when a transition to exposure level $e+1$ occurred, then D'_{e+1} follows from (4.101):

$$D'_{e+1} = \frac{r-e}{\tilde{n}}\alpha D_e. \tag{7.3}$$

This is because, $\frac{r-e}{\tilde{n}}\alpha D_e$ amount of user data loses its $(e+1)$th replica during the exposure level transition. However, an additional replica of $(1-\alpha)D_e$ amount of user data was created by the rebuild process in exposure level e. Therefore,

$$D'_{e-1} = D_{e-1} - \frac{r-e+1}{\tilde{n}}D_{e-1} + \frac{r-e+2}{\tilde{n}}D_{e-2} + (1-\alpha)D_e, \tag{7.4}$$

In addition, $\frac{r-e+1}{\tilde{n}}D_{e-1}$ amount of user data loses its eth replica during the exposure level transition. Therefore, it follows that D'_e is given by

$$D'_e = D_e - \frac{r-e}{\tilde{n}}\alpha D_e + \frac{r-e+1}{\tilde{n}}D_{e-1} - (1-\alpha)D_e. \tag{7.5}$$

Data loss occurs when D_r becomes positive.

7.1.2 Rebuild-Complete Event

A rebuild-complete event updates `time` and triggers the following: (i) decreasing `exposureLevel` by one, (ii) at exposure level e, $e = 1, \cdots, r-1$, updating `dataExposure` by adding D_e to D_{e-1} and setting D_e to zero (this means that the rebuild process always creates replicas of the most exposed data first, or in other words, an intelligent rebuild is done), (iii) scheduling the next rebuild-complete event based on the most exposed data, the placement scheme, network rebuild bandwidth limitations, and the rebuild distribution. Besides these, there are a few other updates that differ based on placement: for declustered placement, when all data have r copies, that is, when the exposure level becomes 0, a node-restore event is scheduled. A node-restore event occurs at the time when all the replicas that were newly created have been successfully transferred to new replacement nodes and the number of nodes is brought back to n. The number of nodes to restore is stored in `nodesToRestore`. For clustered placement, `activeNodes` is increased by one (because copies are being directly created in a new node and so a node-restore event is not required), and a failure time for the newly restored node is generated using the failure distribution F_λ.

7.1.3 Node-Restore Event

A node-restore event is scheduled only for declustered placement. Besides updating the simulated time, this event increases `activeNodes` by `nodesToRestore` and sets `nodesToRestore` to zero. Failure times for the newly restored nodes are scheduled using the failure distribution F_λ.

7.2 Theory vs. Simulation

Although some of the assumptions used in the theoretical analysis, such as independence of node failures, are also used in the simulation, the simulation results reflect a more realistic picture of the systems's reliability. This is because of the following key differences between the theoretical analysis and the simulations. The theoretical estimate of MTTDL in (3.14) takes into account only the time spent by the system in the fully-operational mode and ignores the time spent in rebuild mode, whereas the simulations do not ignore the rebuild times when calculating the times to data loss. Furthermore, in (4.3), P_{DL} is approximated by the probability of the direct path to data loss. In simulations however, all the complex trajectories of the system through the different exposure levels are simulated by simulating random node failure events and updating the data exposure vector by taking partial rebuilds into account. In the theoretical analysis, the time required to restore new nodes in a declustered placement scheme (following successful rebuild of lost replicas in the spare space of surviving nodes) is ignored, whereas in the simulations, the time to restore new nodes is simulated as well. In addition, other approximations made in the analysis, such as neglecting the effect of the transient period of the system, are implicitly avoided in the simulations. Therefore, the simulations reflect a more comprehensive picture of the system behavior than what is assumed in theory.

7.3 Simulation Results

Table 7.1 shows the range of parameters used for the simulations. Typical values for practical systems are used for all parameters, except for the mean times to failure of a node, which have been chosen artificially low (10000 h, 1000 h, and 400 h for replication factors 2, 3, and 4, respectively) to run the simulations fast. The running times of simulations with practical values of the mean times to node failure, which are of the order of 10000 h or higher, are prohibitively high; this is due to the fact that P_{DL} becomes extremely low thereby making the number of first-node-failure events that need to be simulated (along with the other complex set of events that restore all lost replicas following each first-node-failure event) extremely high for each run of the simulation. It is seen that, despite the unrealistically low values of mean times to node failure, the simulation-based values are a good match to

Table 7.1: Range of values of different simulation parameters

Parameter	Meaning	Range
c	amount of data stored on each node	12 TB
n	number of storage nodes	4 to 100
r	replication factor	$2, 3, 4$
b	average rebuild bandwidth at each storage node	96 MB/s
N	effective maximum number of nodes from which distributed rebuild can occur at full speed in parallel	no limit; 12 nodes
$1/\lambda$	mean time to failure of a node	400 h to 10000 h
$1/\mu$	average time to read/write c amount of data from/to a node during rebuild ($1/\mu = c/b$)	35 h
F_λ	node failure time distribution	exponential; Weibull with shape 0.7 to 5
G_μ	node rebuild time distribution	deterministic; exponential

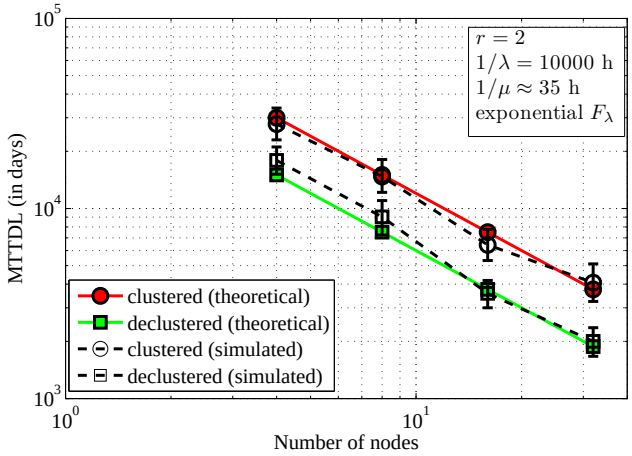
the theoretical estimates. This observation in conjunction with Remark 4.13 implies that the theoretical estimates will also be accurate for realistic values of mean times to node failure, $1/\lambda$, which are generally much higher.

7.3.1 Replication Factor 2

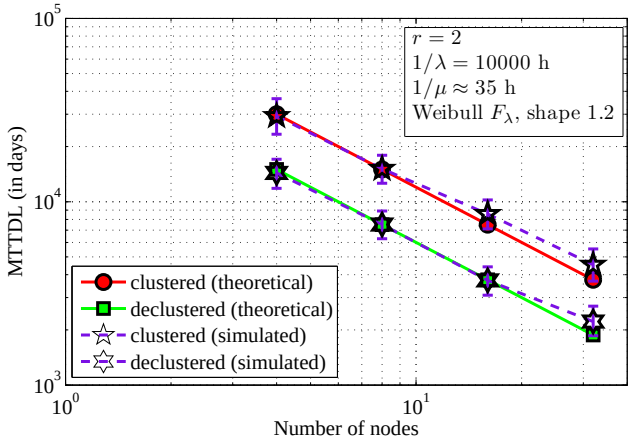
When sufficient network rebuild bandwidth is available, the MTTDL of two-way replicated systems for clustered and declustered placement schemes are given by (4.129)and (4.130), respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu}{n\lambda^2} \quad \text{for } r = 2.$$
$$\text{MTTDL}^{\text{declus.}} \approx \frac{\mu}{2n\lambda^2} \quad \text{for } r = 2.$$

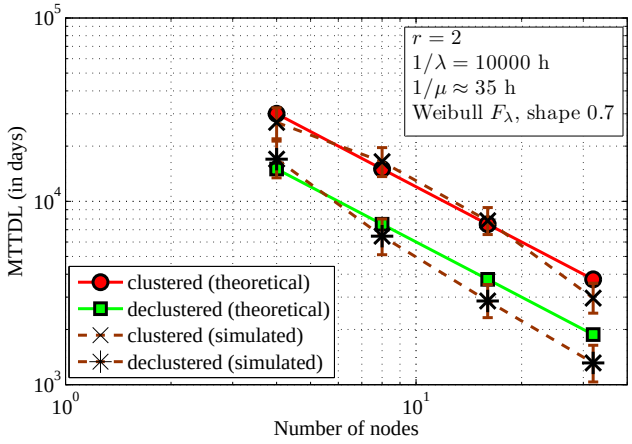
Figure 7.1 shows the comparison of theoretically predicted and simulation-based MTTDL values for a system with replication factor 2 as the number of nodes n in the system is varied. Figures 7.1a, 7.1b, and 7.1c, show the MTTDL when the node failure distribution, F_λ, is exponential, Weibull with shape parameter 1.2, and Weibull with shape parameter 0.7, respectively. It is observed that the theoretically predicted values, although approximate, are a good match to the simulation-based values as they typically lie within the 95% confidence intervals. In conjunction with Remark 4.13, this establishes that the approximations used in theoretical analysis for replication factor two are valid for values of $\lambda/\mu \leq 35/10000 = 0.0035$.



(a) MTTDL of two-way replicated systems for exponential node failure time distribution.



(b) MTTDL of two-way replicated systems for Weibull node failure time distribution with shape parameter 1.2.



(c) MTTDL of two-way replicated systems for Weibull node failure time distribution with shape parameter 0.7.

Figure 7.1: MTTDL of two-way replicated systems with mean time to node failure $1/\lambda = 10000$ h and mean time to read all contents of a node during rebuild $1/\mu \approx 35$ h.

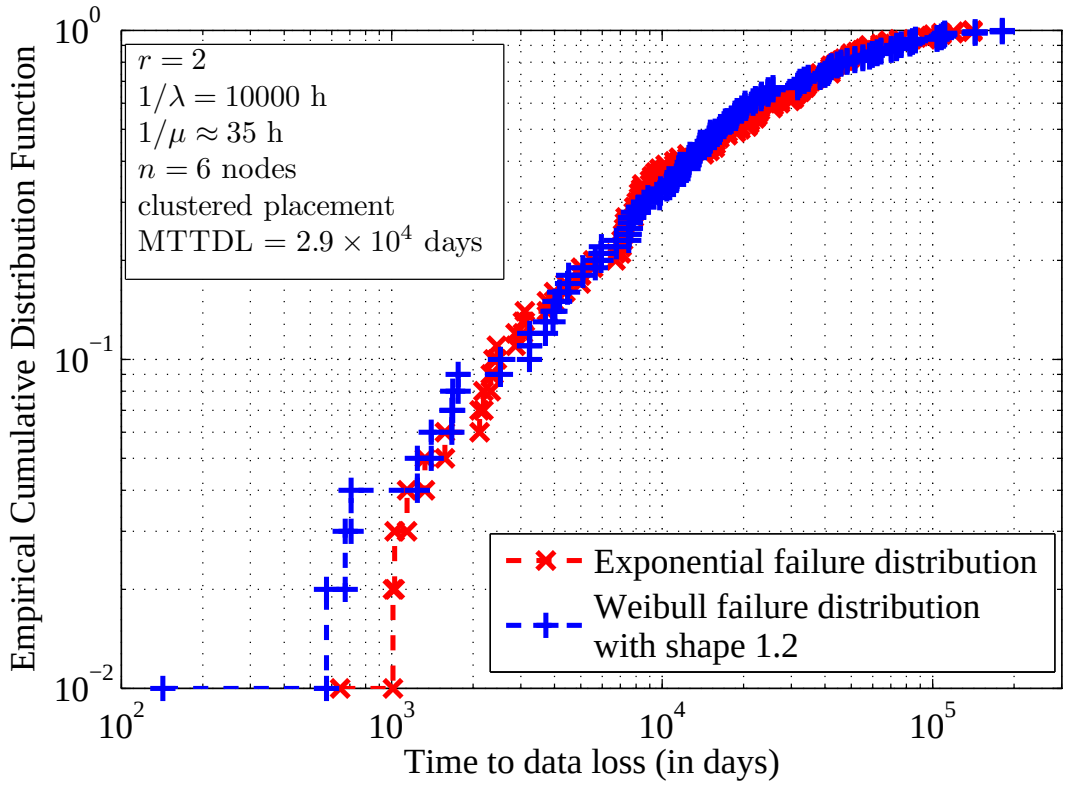


Figure 7.2: Empirical cumulative distribution function of a two-way replicated system.

From Figures 7.1a, 7.1b, and 7.1c, it is seen that the MTTDL is invariant with respect to the failure distribution. However, the cumulative distribution function does seem to depend on the underlying node failure distribution. This is illustrated by Figure 7.2, in which the empirical cumulative distribution function of a two-way replicated system with clustered placement is plotted for exponential failure distribution and for Weibull failure distribution with shape parameter 1.2. Although the MTTDL is the same under the two distributions, there are observable differences in the cumulative distribution functions of their times to data loss, especially for times to data loss less than 1000 days. The probability that data loss occurs within shorter durations (of the order of 1000 days) is much higher for Weibull distribution with shape parameter 1.2 than for exponential distribution.

When the network rebuild bandwidth is limited, and can only support up to N nodes at full speed during distributed rebuild, the MTTDL for declustered placement is given by (5.52):

$$\text{MTTDL}^{\text{declus.}} \approx \begin{cases} \dfrac{\mu}{2n\lambda^2} & \text{when } n \leq N+1 \\ \dfrac{\mu N}{2n(n-1)\lambda^2} & \text{when } n \geq N+1 \end{cases} \quad \text{for } r = 2.$$

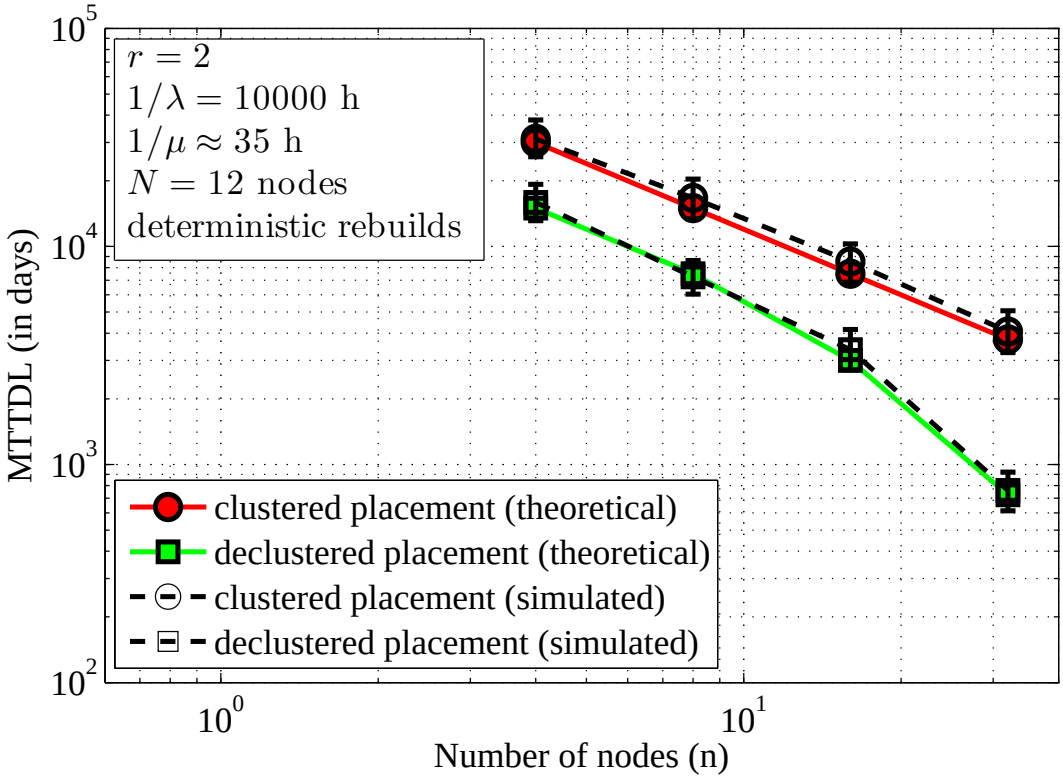


Figure 7.3: MTTDL of two-way replicated systems when the network rebuild bandwidth can support only up to $N = 12$ nodes at full speed during distributed rebuild.

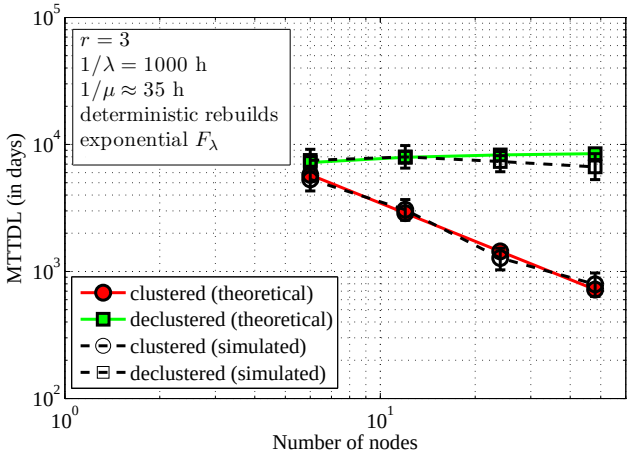
The above expressions show that, when the network rebuild bandwidth is not sufficient to perform distributed rebuild process at full speed, the MTTDL becomes inversely proportional to the *square* of the number of nodes instead of being inversely proportional to the number of nodes. This drastic change is also confirmed by simulations, which matches theoretically predicted MTTDL behavior as shown in Figure 7.3.

7.3.2 Replication Factor 3

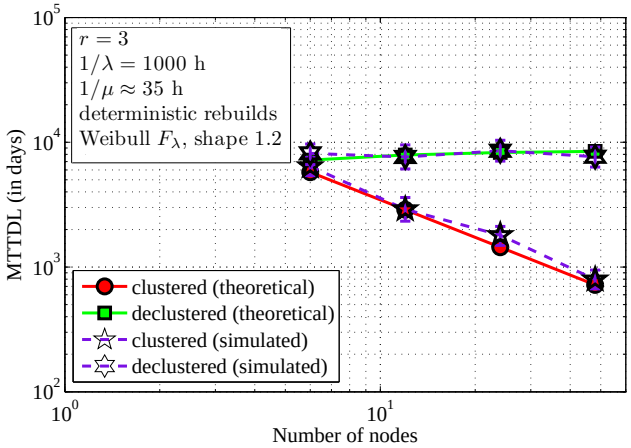
When sufficient network rebuild bandwidth is available, the MTTDL of three-way replicated systems for clustered and declustered placement schemes is given by (4.131) and (4.132), respectively:

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^2}{n\lambda^3} \frac{M_1^2(G_\mu)}{M_2(G_\mu)} \qquad \text{for } r = 3.$$

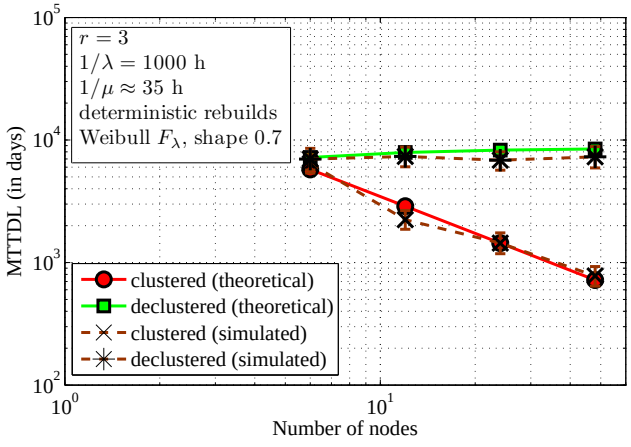
$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{4n\lambda^3} \frac{M_1^2\left(G_{\frac{n-1}{2}\mu}\right)}{M_2\left(G_{\frac{n-1}{2}\mu}\right)} \qquad \text{for } r = 3.$$



(a) MTTDL of three-way replicated systems for exponential node failure time distribution.



(b) MTTDL of three-way replicated systems for Weibull node failure time distribution with shape parameter 1.2.



(c) MTTDL of three-way replicated systems for Weibull node failure time distribution with shape parameter 0.7.

Figure 7.4: MTTDL of three-way replicated systems with mean time to node failure $1/\lambda = 1000$ h and mean time to read all contents of a node during rebuild $1/\mu \approx 35$ h.

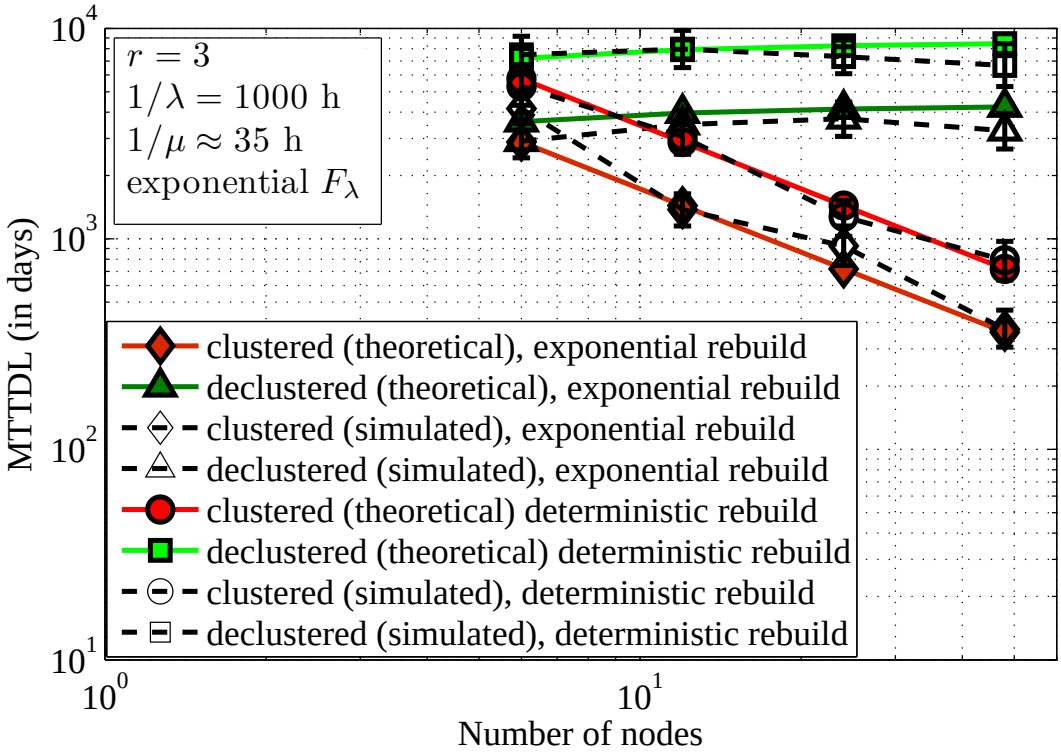


Figure 7.5: MTTDL of a three-way replicated system under deterministic and exponential rebuild time distributions.

Figure 7.4 shows the comparison of theoretically predicted and simulation-based MTTDL values for a system with replication factor 3 and deterministic rebuilds, as the number of nodes n in the system is varied. For deterministic rebuilds, the ratios of the moments in the above expressions evaluate to one. Figures 7.4a, 7.4b, and 7.4c, show the MTTDL when the node failure distribution, F_λ, is exponential, Weibull with shape parameter 1.2, and Weibull with shape parameter 0.7, respectively. It is observed that the theoretically predicted values, although approximate, are a good match to the simulation-based values as they typically lie within the 95% confidence intervals. In conjunction with Remark 4.13, this establishes that the approximations used in theoretical analysis for replication factor three are valid for values of $\lambda/\mu \leq 35/1000 = 0.035$. It is also observed from Figures 7.1a, 7.1b, and 7.1c, that the MTTDL is invariant with respect to the failure distribution. On the hand, the MTTDL is dependent on the rebuild distribution. As an example, if the rebuild distribution were exponential, the corresponding MTTDLs are given by (4.133) and (4.133):

$$\text{MTTDL}^{\text{clus.}} \approx \frac{\mu^2}{2n\lambda^3} \qquad \text{for } r = 3 \text{ (exponential rebuilds).}$$

$$\text{MTTDL}^{\text{declus.}} \approx \frac{(n-1)\mu^2}{8n\lambda^3} \qquad \text{for } r = 3 \text{ (exponential rebuilds).}$$

This is also confirmed by simulations, as shown in Figure 7.5.

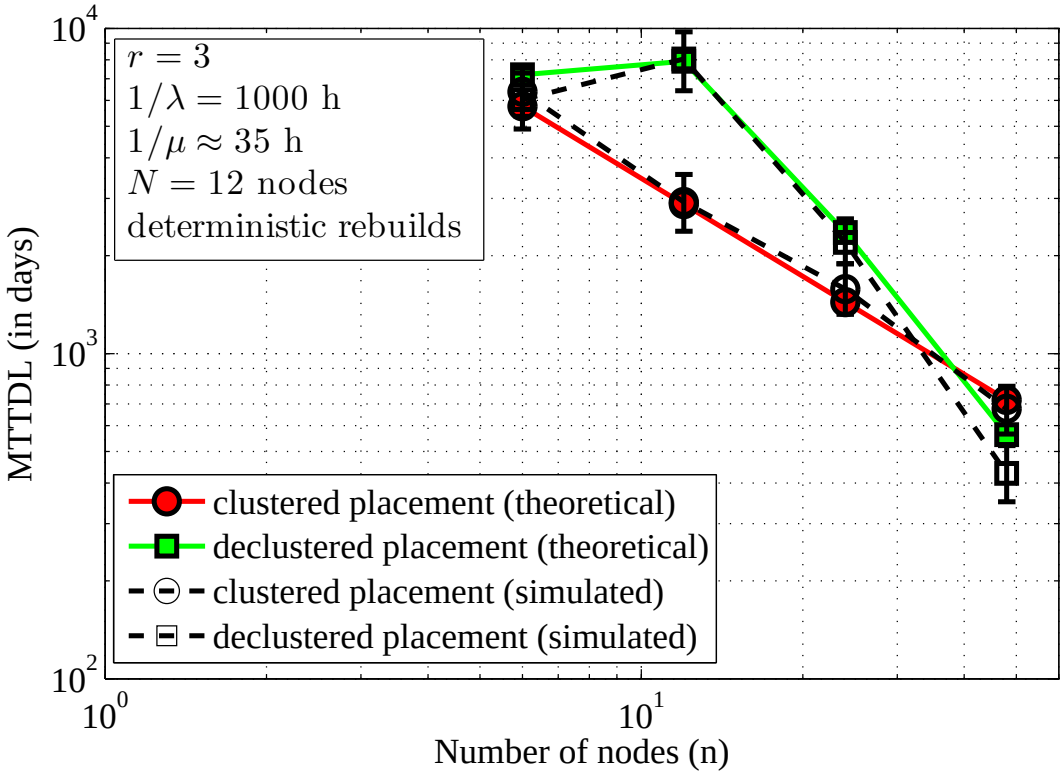


Figure 7.6: MTTDL of three-way replicated systems when the network rebuild bandwidth can support only up to $N = 12$ nodes at full speed during distributed rebuild.

When the network rebuild bandwidth is limited, and can only support up to N nodes at full speed during distributed rebuild, the MTTDL of declustered placement is given by (5.53) for deterministic rebuild times:

$$\text{MTTDL}^{\text{declus.}} \approx \begin{cases} \dfrac{\mu^2(n-1)}{4n\lambda^3} & \text{when } n \leq N+1 \\ \dfrac{\mu^2 N^2}{4n(n-2)\lambda^3} & \text{when } n \geq N+2 \end{cases} \quad \text{for } r = 3.$$

The change in the MTTDL behavior due to limited network rebuild bandwidth is greater than that observed for replication factor two; it goes from being constant with respect to the number of nodes when network rebuild bandwidth is sufficient, to being inversely proportional to the square of the number of nodes when the network rebuild bandwidth is limited. This is also confirmed by simulations, as shown in Figure 7.6.

7.3.3 Replication Factor 4

When sufficient network rebuild bandwidth is available, the MTTDL of four-way replicated systems for clustered and declustered placement schemes is

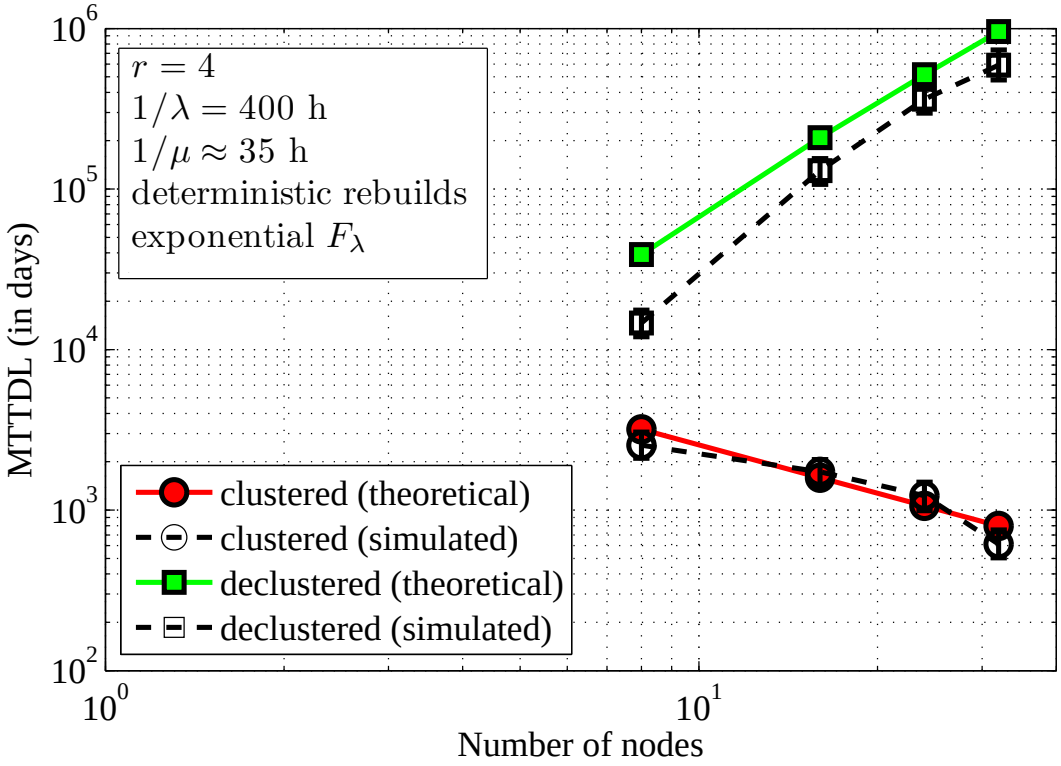


Figure 7.7: MTTDL of four-way replicated systems with mean time to node failure $1/\lambda = 400$ h and mean time to read all contents of a node during rebuild $1/\mu \approx 35$ h.

given by (4.135) and (4.136), respectively:

$$\mathrm{MTTDL}^{\mathrm{clus.}} \approx \frac{\mu^3}{n\lambda^4} \frac{M_1^3(G_\mu)}{M_3(G_\mu)} \quad \text{for } r = 4.$$

$$\mathrm{MTTDL}^{\mathrm{declus.}} \approx \frac{(n-1)^2(n-2)\mu^3}{24n\lambda^4} \frac{M_1^3\left(G_{\frac{n-1}{2}\mu}\right)}{M_3\left(G_{\frac{n-1}{2}\mu}\right)} \quad \text{for } r = 4.$$

For replication factors 4 and above, the simulation run times become prohibitively high even for large values of λ/μ. Furthermore, if λ/μ is made too big, then the theoretical approximations may no longer hold. Figure 7.7 shows the MTTDL of systems with $\lambda/\mu = 35/400 = 0.0875$, for which the simulations are close to the theoretically predicted values but do not match them as well as they did for replication factors two and three. However, it was observed through numerous simulations that the simulation curves were approaching the corresponding theoretical curves as the ratio λ/μ was decreased. It is also observed that, in spite of this difference in MTTDL values between theory and simulation, the behavior of MTTDL with respect to the number of nodes is well captured by the theoretical analysis. This is supported by the fact

that the simulation curves have roughly the same slopes as the theoretically predicted curves.

7.3.4 Erasure Coded Systems

Although the simulation based MTTDL values of erasure coded systems is not presented here, it is expected that they will also closely match the corresponding theoretically predicted values. This is because, the main differences between the simulation of replication based systems and erasure coded systems, are that the rebuilds in erasure coded systems take a longer time, and the number of nodes in an erasure coded system whose failure during rebuild can cause an exposure level transition is higher.

8 Conclusions and Future Work

The problem of reliably maintaining data in a data storage system for time periods far exceeding the lifetimes of individual storage nodes was considered in this thesis. A model of a modern storage system was developed, which was sufficiently comprehensive in including all the intricacies of the reliability behavior of the system, but at the same time was simple enough to abstract away from details that do not affect the system reliability. The validity of this model was established by thoroughly considering all relevant empirical studies of real-world storage systems.

A reliability analysis framework was developed around this model that enabled the analytical computation of mean time to data loss (MTTDL) of a system under a variety of different design and parameter choices. This framework was based on a series of approximations, each of which were shown to hold true for real-world storage systems through rigorous arguments as well as simulations. At the crux of this analytical framework lies the *direct path approximation*. Using this approximation, it is shown that the MTTDL is inversely proportional to the probability of the direct path to data loss during rebuild. Then, using the concept of exposure levels, the probability of direct path to data loss is written as a series of integrals which depend on the conditional means of rebuild times at each exposure level, and the number of nodes whose failure can cause an exposure level transition. The values of these required quantities depend on the data placement, redundancy scheme, and network bandwidth limitations. Therefore, by computing these quantities for a specific data placement, redundancy scheme, and network bandwidth limitation, and evaluating the integrals, the MTTDL of the system can be found.

Firstly, the analytical framework described above was applied to replication based systems using clustered and declustered data placement. It was found

that the MTTDL of clustered placement scheme was inversely proportional to the number of nodes in the system for all replication factors, whereas the MTTDL of declustered placement scheme scaled differently with the number of nodes for different replication factors. For a replication factor of two, the MTTDL of declustered placement was found to scale inversely proportional to the number of nodes, just like the MTTDL of clustered placement. For a replication factor of three, the MTTDL of declustered placement was found to be roughly invariant with respect to the number of nodes. This implies that, for a sufficiently large system, the declustered placement scheme will have a higher MTTDL than the clustered placement scheme. For a replication factor higher than three, the MTTDL of declustered placement was found to increase with the number of nodes in the system. There were two main reasons for this. Firstly, by placing replicas across several nodes in the system, a system with declustered placement was able to benefit from an extremely fast distributed rebuild process that rebuilt from all surviving nodes in parallel. A fast rebuild process reduces the probability that further node failures occur before the completion of rebuild that will eventually cause data loss. Secondly, by placing replicas across several nodes in the system, a system with declustered placement was able to benefit from an intelligent rebuild process that first rebuilds the data with the least number of surviving replicas. This benefit came from the fact that the amount of data with the least number of surviving replicas decreases significantly at each successive exposure level. However, the aforementioned MTTDL results for declustered placement hold true only as long as there is sufficient network rebuild bandwidth that is capable of supporting the distributed rebuild process at full speed using the read-write bandwidth available at all surviving nodes.

The computation of MTTDL under network rebuild bandwidth limitations yields completely different results for declustered placement, especially when the distributed rebuild process is severely hindered by limited network bandwidth. For a replication factor of two, the MTTDL of a declustered placement scheme, whose distributed rebuild process is unable to take place at full speed due to limited network bandwidth, scales inversely proportional to the square of the number of nodes, instead of inversely proportional to the number of nodes (which was the case when there was no network bandwidth limitation). Similarly, for a replication factor of three, the MTTDL of declustered placement scheme is seen to be inversely proportional to the square of the number of nodes under limited network rebuild bandwidth. For a replication factor of three, it was shown that much higher reliability can be achieved by choosing a symmetric data placement with a spread factor that is limited to the maximum number of nodes that the network can support at full speed during rebuild. More generally, the MTTDL expressions under network rebuild bandwidth limitations were obtained for all replication factors and for all possible spread factors. In a dynamically changing storage system, using these expressions, one can find the placement scheme with a spread factor that maximizes the reliability for a given number of nodes and for a given network rebuild band-

width, and continually adapt the data placement to changes in the system to maintain a high level reliability.

Next, the reliability of erasure coded systems, which are a superset of replication based systems, was studied using the analytical framework. It was found that the MTTDL behavior of erasure coded systems with respect to the number of nodes in the system is similar to the MTTDL behavior of replication-based systems. The behavior was found to be the same if "replication factor" was replaced by "number of parities". The MTTDL expressions for general maximum distance separable codes were computed and it was seen that there is a trade-off between storage efficiency and reliability.

Finally, detailed simulations were performed to test the validity of the analytical framework and its predictions. Despite the fact that the simulations avoided all the approximations made in the analytical framework, it was found that the simulation-based MTTDL values matched the theoretically predicted values for a wide range of system parameters, including real-world parameters. This provided a strong evidence for the applicability of the reliability analysis framework to real-world data storage systems.

For future work, the developed reliability analysis framework can be applied or extended to other system models. For instance, latent, or undetected, errors in the data are known to be a serious reliability concern in large storage systems. Such errors may render the rebuild processes ineffective and therefore increases the chances of data loss. One direction of research would be to include a model for latent errors within this framework and study its effects on the system reliability. Another example is the reliability of data storage systems that use different levels of redundancy to store different types of data. An interesting problem would be to find out what data placement is best for such systems in terms of reliability. Another direction of research is to characterize data unavailability due to temporary node unavailabilities. This is a challenging problem as node unavailabilities are known to be correlated and occur more frequently than node failures.

Mean Fully-Operational Period of the System

A

The following derivation of the mean fully-operational period of a system has been adapted from [26, Chap. 2, pp. 139–140]. Consider a storage system consisting of n nodes each of which have a mean lifetime of $1/\lambda$. The node failures are assumed to be independent and identically distributed with distribution function F_λ. Following each node failure, the failed node is replaced by a new node and lost data is restored after an average time of $1/\mu$. For $t \geq 0$,

$$\nu_t^{(i)} := \begin{cases} 1, & \text{if node is operational at time } t, \\ 0, & \text{if node is under rebuild at time } t. \end{cases} \tag{A.1}$$

Then the node availability at time t is given by the probability

$$a_t^{(i)} = \Pr\{\nu_t^{(i)} = 1\}. \tag{A.2}$$

By Lemma 1, it follows that the sequences $a_t^{(i)}$ converge to a:

$$\lim_{t\to\infty} a_t^{(i)} = \frac{1/\lambda}{1/\lambda + 1/\mu} = a, \text{ for } i = 1, \cdots, n. \tag{A.3}$$

For each node i, $i = 1, \cdots, n$, let $A_t^{(i)}$ and $E_t^{(i)}$ be the age of the node since its last replacement at time t, and the time until the next failure of the node at time time t, respectively. In addition, let

$$\widetilde{F}_{\lambda,A_t}^{(i)}(\tau) := \Pr\{A_t^{(i)} \leq \tau | \nu_t^{(i)} = 1\}, \tag{A.4}$$

$$\widetilde{F}_{\lambda,E_t}^{(i)}(\tau) := \Pr\{E_t^{(i)} \leq \tau | \nu_t^{(i)} = 1\}, \tag{A.5}$$

denote the distributions of the age and excess of node i at time t, respectively. According to Lemma 2, the above sequences, $\widetilde{F}_{\lambda,A_t}^{(i)}$ and $\widetilde{F}_{\lambda,E_t}^{(i)}$, converge pointwise to $\widetilde{F}_\lambda$, that is,

$$\lim_{t\to\infty} \widetilde{F}_{\lambda,A_t}^{(i)}(\tau) = \lim_{t\to\infty} \widetilde{F}_{\lambda,E_t}^{(i)}(\tau) = \widetilde{F}_\lambda(\tau), \tag{A.6}$$

for $i = 1, \cdots, n$, where $\widetilde{F}_\lambda$ is given by

$$\widetilde{F}_\lambda(\tau) = \lambda \int_0^\tau (1 - F_\lambda(x))dx. \tag{A.7}$$

If $E^{(i)}(t, \Delta t, \tau)$ denotes the event that the node i was renewed in the interval $(t - \Delta t, t)$, that it operates without failure in $(t, t + \tau)$, and that the remaining nodes operate without failure in $(t, t + \tau)$, then the event $E(t, \Delta t, \tau)$ can be written as the disjoint union

$$E(t, \Delta t, \tau) = E^{(1)}(t, \Delta t, \tau) \cup \cdots \cup E^{(n)}(t, \Delta t, \tau), \tag{A.8}$$

by ignoring events that have probabilities of higher order in Δt, such as more than one rebuild event within a Δt time period. Therefore,

$$\begin{aligned} \Pr\{E(t, \Delta t, \tau)\} &= \sum_{i=1}^{n} \Pr\{E^{(i)}(t, \Delta t, \tau)\} \\ &= \sum_{i=1}^{n} \Bigg[\Pr\{A_t^{(i)} \le \Delta t, E_t^{(i)} > \tau, \nu_t^{(i)} = 1\} \\ &\qquad \times \prod_{\substack{j=1 \\ j \neq i}}^{n} \Pr\{E_t^{(j)} > \tau, \nu_t^{(j)} = 1\} \Bigg]. \end{aligned} \tag{A.9}$$

The first term in the summation above can be expanded as

$$\begin{aligned} \Pr\{A_t^{(i)} \le \Delta t, E_t^{(i)} > \tau, \nu_t^{(i)} = 1\} &= \Pr\{\nu_t^{(i)} = 1\} \Pr\{A_t^{(i)} \le \Delta t | \nu_t^{(i)} = 1\} \\ &\qquad \cdot \Pr\{E_t^{(i)} > \tau | A_t^{(i)} \le \Delta t, \nu_t^{(i)} = 1\} \\ &= a_t^{(i)} \widetilde{F}_{\lambda, A_t}^{(i)}(\Delta t)(1 - F_{\lambda, \Delta t}(\tau)) \end{aligned} \tag{A.10}$$

where,

$$F_{\lambda, \Delta t}(\tau) := \Pr\{E_t^{(i)} \le \tau | A_t^{(i)} \le \Delta t, \nu_t^{(i)} = 1\}. \tag{A.11}$$

It can be seen that (A.10) follows from (A.2) and (A.4). The term $F_{\lambda, \Delta t}(\tau)$ can be seen to converge pointwise to $F_\lambda(\tau)$ as Δt tends to zero:

$$\begin{aligned} \lim_{\Delta t \to 0} F_{\lambda, \Delta t}(\tau) &= \lim_{\Delta t \to 0} \Pr\{E_t^{(i)} \le \tau | A_t^{(i)} \le \Delta t, \nu_t^{(i)} = 1\} \\ &= \Pr\{T_F^{(i)} \le \tau\} && \text{(A.12)} \\ &= F_\lambda(\tau). && \text{(A.13)} \end{aligned}$$

Here, (A.12) follows from the fact that, as Δt tends to zero, the excess time of node i, $E_t^{(i)}$, given that its age, $A_t^{(i)}$, is less than Δt, tends to the node's lifetime, $T_F^{(i)}$.

Furthermore, as $\widetilde{F}^{(i)}_{\lambda,A_t}(\Delta t)$ converges to $\widetilde{F}_{\lambda}(\Delta t)$ by Lemma 2, using (A.7), we can write $\widetilde{F}^{(i)}_{\lambda,A_t}(\Delta t)$ as

$$\widetilde{F}^{(i)}_{\lambda,A_t}(\Delta t) = \lambda \Delta t + o(\Delta t), \tag{A.14}$$

where the small-'o' notation is used to denote that the term $o(\Delta t)$ tends to zero faster than Δt as Δt tends to zero. Therefore, (A.10) reduces to

$$\begin{aligned} \Pr\{A^{(i)}_t \leq \Delta t, E^{(i)}_t > \tau, \nu^{(i)}_t = 1\} &= a^{(i)}_t \lambda \Delta t (1 - F_{\lambda,\Delta t}(\tau)) \\ &\quad + o(\Delta t). \end{aligned} \tag{A.15}$$

Using (A.2) and (A.5), the product term in (A.9) can be expanded as follows:

$$\begin{aligned} \prod_{\substack{j=1 \\ j\neq i}}^{n} \Pr\{E^{(j)}_t > \tau, \nu^{(j)}_t = 1\} &= \prod_{\substack{j=1 \\ j\neq i}}^{n} \Pr\{\nu^{(j)}_t = 1\} \Pr\{E^{(j)}_t > \tau | \nu^{(j)}_t = 1\} \\ &= \prod_{\substack{j=1 \\ j\neq i}}^{n} a^{(j)}_t (1 - \widetilde{F}^{(j)}_{\lambda,E_t}(\tau)). \end{aligned} \tag{A.16}$$

Substituting (A.15) and (A.16) into (A.9), we finally get

$$\begin{aligned} \Pr\{E(t,\Delta t,\tau)\} &= \lambda \Delta t (1 - F_{\lambda,\Delta t}(\tau)) \times \sum_{i=1}^{n} \left[a^{(i)}_t \prod_{\substack{j=1 \\ j\neq i}}^{n} a^{(j)}_t (1 - \widetilde{F}^{(j)}_{\lambda,E_t}(\tau)) \right] \\ &\quad + o(\Delta t). \end{aligned} \tag{A.17}$$

Similar to the calculations above for the probability $\Pr\{E(t,\Delta t,\tau\}$, the probability $\Pr\{E(t,\Delta t)\}$ can be computed by writing $E(t,\Delta t)$ as a disjoint union of events. The resulting expression is:

$$\Pr\{E(t,\Delta t)\} = \lambda \Delta t \times \sum_{i=1}^{n} \left[a^{(i)}_t \prod_{\substack{j=1 \\ j\neq i}}^{n} a^{(j)}_t \right] + o(\Delta t). \tag{A.18}$$

Substituting (A.17) and (A.18) into (3.11) and computing the limit as Δt tends to zero, we get

$$\begin{aligned} p_t(\tau) &= \lim_{\Delta t \to 0} \frac{\Pr\{E(t,\Delta t,\tau)\}}{\Pr\{E(t,\Delta t)\}} \\ &= (1 - F_{\lambda}(\tau)) \times \frac{\sum_{i=1}^{n} \left[a^{(i)}_t \prod_{\substack{j=1 \\ j\neq i}}^{n} a^{(j)}_t (1 - \widetilde{F}^{(j)}_{\lambda,E_t}(\tau)) \right]}{\sum_{i=1}^{n} \left[a^{(i)}_t \prod_{\substack{j=1 \\ j\neq i}}^{n} a^{(j)}_t \right]}. \end{aligned} \tag{A.19}$$

Using (A.3), (A.6), and (A.7), (A.19) yields

$$\begin{aligned}\lim_{t\to\infty} p_t(\tau) &= (1-F_\lambda(\tau))(1-\widetilde{F}_\lambda(\tau))^{(n-1)} \\ &= -\frac{1}{n\lambda}\frac{d}{d\tau}(1-\widetilde{F}_\lambda(\tau))^n, \end{aligned} \tag{A.20}$$

Thus,

$$T = \lim_{t\to\infty} T_t = \lim_{t\to\infty} \int_0^\infty p_t(\tau)d\tau. \tag{A.21}$$

From (A.19), it can be seen that $p_t(\tau) \leq 1 - F(\tau)$. As $1 - F(\tau)$ is integrable, by the dominated convergence theorem, the limit and the integral can be exchanged in the above equation. Therefore,

$$\begin{aligned} T &= \int_0^\infty \lim_{t\to\infty} p_t(\tau)d\tau \\ &= \int_0^\infty -\frac{1}{n\lambda}\frac{d}{d\tau}(1-\widetilde{F}_\lambda(\tau))^n = \frac{1}{n\lambda}. \end{aligned} \tag{A.22}$$

Direct Path Approximation

B

Let $q_{j\to r}$, $j = 1, 2, \cdots, r-1$, denote the probability that, once the system has entered exposure level j, it goes to exposure level r prior to going to exposure level $j-1$. Note that the probability of the direct path to data loss following the first node failure is then equal to $q_{1\to r}$. Let the conditional probability of transition from exposure level j to $j+1$ be equal to ϵ_j. For generally reliable systems, (4.25) reveals that $\epsilon_j \ll 1$ (see Remark 4.1).

We now proceed to derive $q_{j\to r}$, by conditioning on the subsequent transition given that the system is at exposure level j. It follows that

$$q_{j\to r} = \epsilon_j \, h_{(j+1)\to r} + (1-\epsilon_j)\, 0 \text{ , for } j = 1, \cdots, r-1, \tag{B.1}$$

where $h_{(j+1)\to r}$ denotes the probability that once the system has entered exposure level $j+1$, it goes to exposure level r prior to going to exposure level $j-1$. This probability is derived by conditioning on which of the two exposure levels j and r is subsequently entered first, that is, for $j = 1, \cdots, r-1$,

$$h_{(j+1)\to r} = q_{(j+1)\to r} + (1 - q_{(j+1)\to r})\, q_{j\to r}. \tag{B.2}$$

The first term of the summation accounts for the event that exposure level r is entered first, whereas the second term accounts for the event that exposure level j is entered first. In the latter case, the probability that the exposure level r is subsequently entered prior to entering exposure level $j-1$ is given by $q_{j\to r}$, according to its definition. Combining (B.1) and (B.2) yields, for $j = 1, \cdots, r-1$,

$$q_{j\to r} = \epsilon_j (q_{(j+1)\to r} + (1 - q_{(j+1)\to r})\, q_{j\to r}). \tag{B.3}$$

Solving (B.3) for $q_{j\to r}$ yields the recursive relation

$$q_{j\to r} = \frac{\epsilon_j \, q_{(j+1)\to r}}{1 - \epsilon_j \, (1 - q_{(j+1)\to r})} \text{ , for } j = 1, \cdots, r-1. \tag{B.4}$$

In particular, for $\epsilon_j \ll 1$, it follows that

$$q_{j\to r} \approx \epsilon_j \, q_{(j+1)\to r} , \quad \text{for } j = 1, \cdots, r-1. \tag{B.5}$$

Consequently, repeatedly applying (B.5) yields

$$q_{j\to r} \approx \prod_{i=j}^{r-1} \epsilon_i , \quad \text{for } j = 1, \cdots, r-1. \tag{B.6}$$

Note that the product on the right hand side of the above equation is equal to the probability of occurrence of the direct path $j \to j+1 \to \cdots \to r$ from exposure level j to data loss. Thus, for $j = 1$, Eq. (B.6) leads to the result sought:

$$P_{DL} = q_{1\to r} \approx \prod_{i=1}^{r-1} \epsilon_i, \text{ for } \epsilon_i \ll 1, \; i = 1, \cdots, r-1, \tag{B.7}$$

namely, for a highly reliable system, the probability that, once the system has entered exposure level one, it goes to exposure level r prior to reaching exposure level zero, is equal to the probability of the direct path $1 \to 2 \to \cdots \to r$ to data loss for a highly reliable system.

Data not Rebuilt during an Exposure Level Transition

C

Consider a storage system under rebuild which entered a particular exposure level, say e, at time t. Let R be the time required to rebuild the most-exposed data and return to the previous exposure level, $e-1$. Also, let R have mean $1/\mu$ and distribution G_μ that satisfies condition (2.11). Suppose there are $\tilde{n}$ nodes whose failure during rebuild can cause the system to enter the next exposure level, $e+1$. Let the times to failures of these nodes at time t by $E_t^{(i)}$, $i = 1, \cdots, \tilde{n}$. According to Lemma 2 and the node-failure independence assumption $E_t^{(i)}$ are independent and identically distributed according to $\widetilde{F}_\lambda$ in the stationary period of the system. Let

$$F := \min_{i \in \{1, \cdots, \tilde{n}\}} E_t^{(i)} \tag{C.1}$$

denote the time taken for a node failure to occur that can cause the system to enter the next exposure level, $e+1$.

Given that a node failure occurs before the completion of rebuild causing the system to enter the next exposure level, we are interested in the fraction α of rebuild time left when that node failure occurs, that is, we are interested in

$$\alpha := \frac{R-F}{R} \Bigg| F < R. \tag{C.2}$$

This fraction is a random variable whose distribution depends on the distributions of F and R. The distribution function of α is computed as follows. For

$x \in (0, 1]$,

$$\begin{aligned}
\Pr\{\alpha \le x\} &= \Pr\left\{ \frac{R-F}{R} \le x \middle| F < R \right\} && \text{(C.3)}\\
&= \Pr\{F \ge R(1-x) | F < R\} && \text{(C.4)}\\
&= \frac{\Pr\{F \ge R(1-x), F < R\}}{\Pr\{F < R\}} && \text{(C.5)}\\
&= \frac{\Pr\{R(1-x) \le F < R\}}{\Pr\{F < R\}} && \text{(C.6)}\\
&= \frac{\Pr\{F < R\} - \Pr\{F < R(1-x)\}}{\Pr\{F < R\}} && \text{(C.7)}\\
&= 1 - \frac{\Pr\{F < R(1-x)\}}{\Pr\{F < R\}} && \text{(C.8)}\\
&= 1 - \frac{\Pr\{\min_{i\in\{1,\cdots,\tilde{n}\}} E_t^{(i)} < R(1-x)\}}{\Pr\{\min_{i\in\{1,\cdots,\tilde{n}\}} E_t^{(i)} < R\}} && \text{(C.9)}\\
&= 1 - \frac{1 - \Pr\{\min_{i\in\{1,\cdots,\tilde{n}\}} E_t^{(i)} \ge R(1-x)\}}{1 - \Pr\{\min_{i\in\{1,\cdots,\tilde{n}\}} E_t^{(i)} \ge R\}} && \text{(C.10)}\\
&= 1 - \frac{1 - \Pr\{E_t^{(i)} \ge R(1-x)\, \forall\, i \in \{1,\cdots,\tilde{n}\}\}}{1 - \Pr\{E_t^{(i)} \ge R \forall\, i \in \{1,\cdots,\tilde{n}\}\}} && \text{(C.11)}\\
&= 1 - \frac{1 - \prod_{i=1}^{\tilde{n}} \Pr\{E_t^{(i)} \ge R(1-x)\}}{1 - \prod_{i=1}^{\tilde{n}} \Pr\{E_t^{(i)} \ge R\}} && \text{(C.12)}\\
&= 1 - \frac{1 - (\Pr\{E_t^{(1)} \ge R(1-x)\})^{\tilde{n}}}{1 - (1 - \Pr\{E_t^{(1)} < R\})^{\tilde{n}}} && \text{(C.13)}\\
&= 1 - \frac{1 - (1 - \Pr\{E_t^{(1)} < R(1-x)\})^{\tilde{n}}}{1 - (1 - \Pr\{E_t^{(1)} < R\})^{\tilde{n}}}. && \text{(C.14)}
\end{aligned}$$

Here, (C.5) follows from Bayes' theorem, (C.9) follows by substituting (C.1) in (C.8), (C.12) follows from the fact that $E_t^{(i)}$, $i = 1, \cdots, \tilde{n}$, are independent, and (C.13) follows from the fact that $E_t^{(i)}$, $i = 1, \cdots, \tilde{n}$, are identically distributed. It is shown in Appendix D that

$$\begin{aligned}
\Pr\{E_t^{(1)} < R\} &= \frac{\lambda}{\mu} + o\left(\frac{\lambda}{\mu}\right), && \text{(C.15)}\\
\Pr\{E_t^{(1)} < R(1-x)\} &= (1-x)\frac{\lambda}{\mu} + o\left((1-x)\frac{\lambda}{\mu}\right). && \text{(C.16)}
\end{aligned}$$

Therefore, substituting (C.15) and (C.16) in (C.14), we get

$$\Pr\{\alpha \le x\} = 1 - \frac{\tilde{n}(1-x)\lambda/\mu + o((1-x)\lambda/\mu)}{\tilde{n}\lambda/\mu + o(\lambda/\mu)} \approx x. \qquad \text{(C.17)}$$

This means that, for systems with generally reliable nodes that satisfy (2.10) and (2.11), the fraction α of most-exposed data not rebuilt due to an exposure level transition is uniformly distributed between zero and one.

Probability of Node Failure during Rebuild

D

According to (3.8) and (A.7), it holds that

$$\begin{aligned}\Pr\{E_t < R\} &= \int_{\tau=0}^{\infty} \widetilde{F}_\lambda(\tau) dG_\mu(\tau) \\ &= \int_{\tau=0}^{\infty} \lambda \int_{t=0}^{\tau} (1 - F_\lambda(t)) dt dG_\mu(\tau).\end{aligned}$$

Changing the order of integrals, yields after some manipulations

$$\Pr\{E_t < R\} = \frac{\lambda}{\mu}\left(1 - \mu \int_{t=0}^{\infty} F_\lambda(t)(1 - G_\mu(t))dt\right).$$

In the last step above, we used the fact that integrating the complementary cumulative distribution function $1-G_\mu(t)$ gives the mean $1/\mu$. As the functions F_λ and G_μ satisfy (2.22) and (2.23) respectively, it can be seen that the second term inside the parentheses is $o(1)$. Therefore,

$$\Pr\{E_t < R\} = \lambda/\mu + o(\lambda/\mu).$$

Similarly the following can also be shown for any $x \in (0, 1)$:

$$\Pr\{E_t < Rx\} = x\lambda/\mu + o(x\lambda/\mu).$$

Bibliography

[1] International Data Corporation, "The diverse and exploding digital universe," White Paper, 2008.

[2] E. Pinheiro, W.-D. Weber, and L. A. Barroso, "Failure trends in a large disk drive population," in *Proc. 5th USENIX conference on File and Storage Technologies (FAST'07)*, 2007, pp. 17–28.

[3] B. Schroeder and G. A. Gibson, "Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you?" *ACM Transactions on Storage*, vol. 3, no. 3, pp. 1–31, October 2007.

[4] S. Ramabhadran and J. Pasquale, "Analysis of long-running replicated systems," in *Proc. 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, 2006, pp. 1–9.

[5] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-area cooperative storage with CFS," in *Proc. 18th ACM Symposium on Operating Systems Principles (SOSP'01)*, 2001, pp. 202–215.

[6] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "OceanStore: an architecture for global-scale persistent storage," *SIGPLAN Notices*, vol. 35, no. 11, pp. 190–201, November 2000.

[7] A. Muthitacharoen, R. Morris, T. Gil, and B. Chen, "Ivy: A read/write peer-to-peer file system," in *Proc. 5th USENIX Symposium on Operating Systems Design and Implementation (OSDI '02)*, December 2002.

[8] A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly durable, decentralized storage despite massive correlated failures," in *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI'05)*, May 2005.

[9] M. Leslie, J. Davies, and T. Huffman, "A comparison of replication strategies for reliable decentralised storage," *Journal of Networks*, vol. 1, no. 6, pp. 36–44, December 2006.

[10] A. Thomasian and M. Blaum, "Mirrored disk organization reliability analysis," *IEEE Transactions on Computers*, vol. 55, pp. 1640–1644, December 2006.

[11] K. M. Greenan, E. L. Miller, and J. Wylie, "Reliability of flat XOR-based erasure codes on heterogeneous devices," in *Proc. 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'08)*, June 2008, pp. 147–156.

[12] Q. Xin, E. L. Miller, T. Schwarz, D. D. E. Long, S. A. Brandt, and W. Litwin, "Reliability mechanisms for very large storage systems," in *Proc. 20th IEEE / 11th NASA Goddard Conference on Mass Storage Systems and Technologies (MSS'03)*, 2003, pp. 146–156.

[13] Q. Xin, E. L. Miller, and T. J. E. Schwarz, "Evaluation of distributed recovery in large-scale storage systems," in *Proc. 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04)*, 2004, pp. 172–181.

[14] Q. Lian, W. Chen, and Z. Zhang, "On the impact of replica placement to the reliability of distributed brick storage systems," in *Proc. 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005, pp. 187–196.

[15] V. Venkatesan, I. Iliadis, X.-Y. Hu, R. Haas, and C. Fragouli, "Effect of replica placement on the reliability of large-scale data storage systems," in *Proc. 18th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'10)*, 2010, pp. 79–88.

[16] D. A. Patterson, G. Gibson, and R. H. Katz, "A case for redundant arrays of inexpensive disks (RAID)," in *Proc. ACM International Conference on Management of Data (SIGMOD'88)*, 1988, pp. 109–116.

[17] J. Elerath and M. Pecht, "A highly accurate method for assessing reliability of redundant arrays of inexpensive disks (RAID)," *IEEE Transactions on Computers*, vol. 58, pp. 289–299, 2009.

[18] V. Venkatesan, I. Iliadis, C. Fragouli, and R. Urbanke, "Reliability of clustered vs. declustered replica placement in data storage systems," in *Proc. 19th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'11)*, 2011, pp. 307–317.

[19] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Transactions on Information Theory*, vol. 56, 2010.

[20] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A survey on network coding for distributed storage," *Proc. IEEE*, vol. 99, no. 3, 2011.

[21] Y. Kim, A. Gupta, B. Urgaonkar, P. Berman, and A. Sivasubramaniam, "HybridStore: A cost-efficient, high-performance storage system combining SSDs and HDDs," in *Proc. IEEE 19th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'11)*, 2011, pp. 227–236.

[22] D. Ford, F. Labelle, F. I. Popovici, M. Stokely, V.-A. Truong, L. Barroso, C. Grimes, and S. Quinlan, "Availability in globally distributed storage systems," in *Proc. 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI'10)*, 2010, pp. 61–74.

[23] W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky, "Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics," *ACM Transactions on Storage*, vol. 4, no. 3, pp. 1–25, November 2008.

[24] K. M. Greenan, J. S. Plank, and J. J. Wylie, "Mean time to meaningless: MTTDL, Markov models, and storage system reliability," in *Proc. USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage)*, 2010, pp. 1–5.

[25] A. Thomasian and M. Blaum, "Higher reliability redundant disk arrays: Organization, operation, and coding," *ACM Trans. Storage*, vol. 5, no. 3, pp. 1–59, 2009.

[26] B. Gnedenko, I. Beliaev, A. Solovev, and R. Barlow, *Mathematical methods of reliability theory*, ser. Probability and mathematical statistics. Academic Press, 1969.

[27] P. M. Chen, E. K. Lee, G. A. Gibson, R. H. Katz, and D. A. Patterson, "RAID: high-performance, reliable secondary storage," *ACM Computing Surveys*, vol. 26, no. 2, pp. 145–185, June 1994.

Curriculum Vitae

Education

École Polytechnique Fédérale de Lausanne

- *Doctoral Program in Computer, Communication, and Information Sciences* — *Sep. 2008 - Sep. 2012*
  - Thesis: Reliability Analysis of Data Storage Systems
  - Advisors: Prof. Rüdiger Urbanke (EPFL), Prof. Christina Fragouli (EPFL), Dr. Ilias Iliadis (IBM Research)
  - Specialized in reliability modeling of large-scale data storage systems

Eidgenössische Technische Hochschule Zürich

- *M.Sc., Electrical Engineering and Information Technology* — *Oct. 2006 - Apr. 2008*
  - Thesis: On Low Power Capacity of the Poisson Channel
  - Advisors: Prof. Amos Lapidoth, Dr. Ligong Wang
  - Specialized in information theory

- **Indian Institute of Technology Madras**
  B.Tech., Electrical Engineering — *Aug. 2002 - Jul. 2006*
  - Thesis: Cricket Video Analysis: Detection of Chucking Using Machine Learning Techniques
  - Advisor: Prof. R. Aravind
  - Specialized in communication systems and signal processing, minored in Physics

Work Experience

- **IBM Research - Zurich** — Rüschlikon, Switzerland
  Pre-doc, Storage Systems Group — *Oct. 2008 - Sep. 2012*
- **Information Processing Group, EPFL** — Lausanne, Switzerland
  Teaching Assistant — *Sep. 2010 - Jan. 2011*
- **IBM Research - Zurich** — Rüschlikon, Switzerland
  Intern, Tape Storage Group — *Apr. 2008 - Sep. 2008*

- **Computer Engineering and Networks Laboratory (TIK), ETH Zürich** — Zürich, Switzerland
  Research Assistant — *Oct. 2006 - Jun 2007*
- **Midas Communication Technologies** — Chennai, India
  Intern — *May 2005 - Aug. 2005*

Publications

- **Journals**

· The Discrete-time Poisson Channel at Low Input Powers
Amos Lapidoth, Jeffrey H. Shapiro, **Vinodh Venkatesan**, and Ligong Wang
IEEE Transactions on Information Theory, Vol. 57, No. 6, June 2011

- **Conferences**

· A General Reliability Model for Data Storage Systems
Vinodh Venkatesan and Ilias Iliadis
accepted at International Conference on Quantitative Evaluation of Systems (QEST) 2012

· Reliability of Data Storage Systems under Network Rebuild Bandwidth Constraints
Vinodh Venkatesan, Ilias Iliadis, and Robert Haas
accepted at IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) 2012

· Reliability of Clustered vs. Declustered Replica Placement in Data Storage Systems (**Best Paper Award**)
Vinodh Venkatesan, Ilias Iliadis, Christina Fragouli, and Rüdiger Urbanke
in IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) 2011, Singapore

· Effect of Replica Placement on the Reliability of Large-Scale Data Storage Systems
Vinodh Venkatesan, Ilias Iliadis, Xiao-Yu Hu, Robert Haas, and Christina Fragouli
in IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS) 2010, Miami Beach FL, USA

· On the Characterization of Write-Equalized Magnetic Recording Channels
Sedat Oelcer and **Vinodh Venkatesan**
in IEEE International Magnetics Conference (INTERMAG) 2009, Sacramento CA, USA

· The Poisson Channel at Low Input Powers
Amos Lapidoth, Jeffrey H. Shapiro, **Vinodh Venkatesan**, and Ligong Wang
in IEEE Convention of Electrical and Electronics Engineers in Israel (IEEEI) 2008, Eilat, Israel

- **Posters**

· Coding for Large-Scale Data Storage Systems
Vinodh Venkatesan
in 2nd Annual North American School of Information Theory 2009, Evanston IL, USA

- **Theses**

· Reliability Analysis of Data Storage Systems (Doctoral Thesis, 2012)
Communication Theory Lab., EPFL, Switzerland
Advisors: Prof. Rüdiger Urbanke and Prof. Christina Fragouli

· On Low Power Capacity of the Poisson Channel (Masters Thesis, 2008)
Signal and Information Processing Lab., ETH Zürich, Switzerland
Advisors: Prof. Amos Lapidoth and Dr. Ligong Wang

· Optimality of Gaussian Inputs for a Multi-Access Achievable Rate-Region (Semester Thesis, 2007)
Signal and Information Processing Lab., ETH Zürich, Switzerland
Advisors: Prof. Amos Lapidoth and Prof. Michèle Wigger

· Cooperation Required Between Destinations in a Two-Source Two-Destination Network to Achieve Full Multiplexing Gain (Semester Thesis, 2007)
Communication Theory Group, ETH Zürich, Switzerland
Advisor: Dr. Jatin Thukral

· Cricket Video Analysis: Detection of Chucking Using Machine Learning Techniques (Bachelors Thesis, 2006)
Telecommunications and Computer Networking Group, IIT Madras, India
Advisor: Prof. R. Aravind

- **Preprints**

· Effect of Codeword Placement on the Reliability of Erasure Coded Data Storage Systems
Vinodh Venkatesan and Ilias Iliadis
submitted to IEEE International Conference on Computer Communications (INFOCOM) 2013