

THE FLORIDA STATE UNIVERSITY

COLLEGE OF ENGINEERING

DESIGNING RELIABLE LARGE-SCALE STORAGE ARRAYS

By

EDWARD MICHAEL MCDONALD, III

A Thesis submitted to the
Department of Electrical and Computer Engineering
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Fall Semester, 2007

The members of the Committee approve the Thesis of Edward Michael McDonald, III defended on August 6, 2007.

Bruce A. Harvey
Professor Co-Directing Thesis

Lois Hawkes
Professor Co-Directing Thesis

Simon Y. Foo
Committee Member

Approved:

Victor DeBrunner, Chair, Department of Electrical and Computer Engineering

Ching-Jen Chen, Dean, FAMU-FSU College of Engineering

The Office of Graduate Studies has verified and approved the above named committee members.

ACKNOWLEDGMENTS

I would like to acknowledge the following sponsors for supporting my research: the Center for Ocean-Atmospheric Prediction Studies (COAPS) at the Florida State University, the Department of Defense (DoD), and the High-performance Computing and Simulation (HCS) Research Laboratory at the University of Florida (UF). I would also like to thank my committee members for their motivation and support: Dr. Bruce Harvey, Dr. Lois Hawkes, and Dr. Simon Foo.

TABLE OF CONTENTS

| | |
|---|-----------|
| LIST OF TABLES | vii |
| LIST OF FIGURES | viii |
| ABSTRACT..... | ix |
| 1. INTRODUCTION..... | 1 |
| 1.1. Motivation..... | 1 |
| 1.2. Thesis..... | 2 |
| 1.3. Summary..... | 2 |
| 2. LARGE-SCALE STORAGE ARRAYS..... | 4 |
| 2.1. What Are Storage Arrays? | 4 |
| 2.2. Unrecoverable Bit Errors..... | 7 |
| 2.3. Storage Demand and Its Projected Growth | 7 |
| 3. RAID SYSTEMS..... | 11 |
| 3.1. The Origins of RAID | 11 |
| 3.2. Base RAID Levels | 12 |
| <i>RAID 0</i>..... | 13 |
| <i>RAID 1</i>..... | 14 |
| <i>RAID 2</i>..... | 15 |
| <i>RAID 3</i>..... | 16 |
| <i>RAID 4</i>..... | 17 |
| <i>RAID 5</i>..... | 18 |
| <i>RAID 6</i>..... | 19 |
| 3.3. Nested RAID Levels..... | 20 |
| <i>RAID 0+1</i> | 21 |
| <i>RAID 10</i>..... | 22 |
| <i>RAID 50</i>..... | 22 |
| 3.4. Grouped RAID Levels..... | 23 |
| <i>GRAID 50 / 55 / 56</i>..... | 25 |
| <i>GRAID 60 / 65 / 66</i>..... | 25 |
| <i>GRAID 550 / 560</i>..... | 25 |
| <i>GRAID 650 / 660</i>..... | 26 |
| 3.5. RAID/GRAID Summary | 35 |
| 4. RAID METRICS..... | 37 |
| 4.1. Introduction..... | 37 |

| | | |
|----------|---|----|
| 4.2. | Variables | 37 |
| 4.3. | Storage Capacity and Storage Efficiency | 39 |
| 4.3.1. | Single-Level RAID | 39 |
| 4.3.2. | Dual-Level RAID/GRAID | 40 |
| 4.3.3. | Tri-Level GRAID | 42 |
| 4.4. | Availability vs. Reliability | 44 |
| 4.5. | Reliability Metrics | 45 |
| 4.5.1. | MTBF | 46 |
| 4.5.2. | MTTR | 47 |
| 4.5.3. | MTTDL | 49 |
| 4.5.3.1. | MTTDL Due to Disk Failure (DF) | 51 |
| 4.5.3.2. | MTTDL Due to Correlated Disk Failure (CDF) | 57 |
| 4.5.3.3. | MTTDL Due to Unrecoverable Bit Error (UBE) | 60 |
| 4.5.3.4. | Harmonic MTTDL | 64 |
| 4.6. | How Accurate Are MTBF Ratings? | 66 |
| 4.6.1. | Carnegie Mellon Case Study | 68 |
| 4.6.2. | Google Labs Case Study | 70 |
| 4.7. | Summary of Metrics | 72 |
| 5. | RELIABILITY ANALYSIS | 74 |
| 5.1. | RAID Experiments | 74 |
| 5.1.1. | Areca SATA RAID Controller | 74 |
| 5.1.2. | InforTrend EonStor SATA RAID Enclosure | 75 |
| 5.2. | GRAID Calculator | 76 |
| 5.2.1. | Menu Interface | 77 |
| 5.2.2. | Results and Figures | 78 |
| 5.3. | Reliability Anomalies | 80 |
| 5.3.1. | MTTDL Convergence | 80 |
| 5.3.2. | MTTDL Divergence | 83 |
| 5.4. | Recommended Designs | 84 |
| 5.5. | Analysis of Reliability Variables | 87 |
| 5.5.1. | Varying the MTBF | 88 |
| 5.5.2. | Varying the MTTR | 89 |
| 5.5.3. | Varying the BER | 92 |
| 5.5.4. | Varying the Disk Size | 93 |
| 5.5.5. | Varying the Enclosure Size | 93 |
| 5.6. | Methods for Improving/Upholding Reliability | 95 |
| 5.6.1. | Hot Spare Disks | 95 |
| 5.6.2. | Device Diversity | 96 |
| 5.6.3. | RAID-Z | 96 |

| | |
|--|-----|
| 6. CONCLUSION | 98 |
| APPENDIX A - Acronyms | 100 |
| APPENDIX B - MTTDL Derivations | 102 |
| APPENDIX C - GRAID Reliability Calculator | 106 |
| REFERENCES | 117 |
| BIOGRAPHICAL SKETCH | 121 |

LIST OF TABLES

| | |
|---|-----------|
| Table 1: Array Magnitudes and Minimum Number of Disks Required | 5 |
| Table 2: Percentage Difference in Decimal to Binary Conversion..... | 6 |
| Table 3: RAID Levels at a Glance [GUPT02]..... | 36 |
| Table 4: Grouped RAID Levels at a Glance | 36 |
| Table 5: RAID and GRAID Variables..... | 38 |
| Table 6: Special Case Single-Level RAID Storage Capacity and Storage Efficiency..... | 39 |
| Table 7: Storage Capacity and Storage Efficiency of RAID..... | 40 |
| Table 8: Special Case Dual-Level Storage Capacity and Storage Efficiency | 40 |
| Table 9: Dual-Level GRAID Storage Capacity and Storage Efficiency | 41 |
| Table 10: Tri-Level GRAID Storage Capacity and Storage Efficiency..... | 42 |
| Table 11: System Availability..... | 45 |
| Table 12: MTTDL Comparison [WINC06] | 51 |
| Table 13: Disk Type Parameters..... | 77 |
| Table 14: Maximum Array Sizes for GRAID 5x and 6x..... | 83 |
| Table 15: Recommended Configurations for GRAID 5x Levels..... | 85 |
| Table 16: Recommended Configurations for GRAID 6x Levels..... | 85 |
| Table 17: Recommended Configurations for GRAID 5x0 Levels..... | 86 |
| Table 18: Recommended Configurations for GRAID 6x0 Levels..... | 87 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1: Total Worldwide Digital Archive Capacity [MCKN06]..... | 8 |
| Figure 2: Longitudinal recording (left) and Perpendicular recording (right) [HITA06]..... | 9 |
| Figure 3: Data Striping Scheme | 11 |
| Figure 4: Data Mirroring Scheme..... | 11 |
| Figure 5: RAID 0 [ACNC07] | 13 |
| Figure 6: Relative Performance of SAS [ADAP07]..... | 14 |
| Figure 7: RAID 1 [ACNC07] | 15 |
| Figure 8: RAID 2 [ACNC07] | 16 |
| Figure 9: RAID 3 [ACNC07] | 17 |
| Figure 10: RAID 4 [ACNC07]..... | 17 |
| Figure 11: State Diagram for RAID 5 System | 18 |
| Figure 12: RAID 5 [ACNC07]..... | 19 |
| Figure 13: State Diagram for RAID 6 System | 19 |
| Figure 14: RAID 6 [ACNC07]..... | 20 |
| Figure 15: RAID 0+1..... | 21 |
| Figure 16: RAID 10 | 21 |
| Figure 17: RAID 50 [ACNC07]..... | 22 |
| Figure 18: GRAID 50 | 26 |
| Figure 19: GRAID 55 | 27 |
| Figure 20: GRAID 56 | 28 |
| Figure 21: GRAID 60 | 29 |
| Figure 22: GRAID 65 | 29 |
| Figure 23: GRAID 66 | 30 |
| Figure 24: GRAID 550 | 31 |
| Figure 25: GRAID 560 | 32 |
| Figure 26: GRAID 650 | 33 |
| Figure 27: GRAID 660 | 34 |
| Figure 28: Dual-Level GRAID Storage Efficiency | 41 |
| Figure 29: Tri-Level GRAID Storage Efficiency | 43 |
| Figure 30: Best Case (left) and Worst Case (right) Scenarios Before RAID 1 Data Loss..... | 50 |
| Figure 31: Elements of Harmonic MTTDL for RAID 5 (left) and RAID 6 (right) | 65 |
| Figure 32: Adjusted MTBF Due To POH [COLE00] | 68 |
| Figure 33: Failure Rate of Hard Drives over Expected Lifetime [YANG99]..... | 69 |
| Figure 34: ARR for Each Drive Type [SCHR07] | 70 |
| Figure 35: AFR by Age Group [PINH07] | 71 |
| Figure 36: AFR and Average Temperature by Age Group [PINH07] | 72 |
| Figure 37: Stages of the Menu Interface | 78 |
| Figure 38: Processing Flow Chart..... | 79 |
| Figure 39: Sample Results Provided In MATLAB Standard Output | 80 |
| Figure 40: MDDTL Convergence of RAID 5 and RAID 6 | 81 |
| Figure 41: MDDTL Convergence for GRAID 5x (left) and GRAID 6x (right)..... | 82 |
| Figure 42: MDDTL Divergence from 100 TiB (left) to 3 PiB (right)..... | 84 |
| Figure 43: Effect of MTBF on the MTTDL of GRAID 6x..... | 89 |
| Figure 44: Effect of MTTR on the MTTDL of GRAID 5x (left) and 6x (right)..... | 91 |
| Figure 45: Effect of the BER on MTTDL for GRAID 55 (left) and 66 (right)..... | 92 |
| Figure 46: Rack Enclosure Optimization..... | 95 |

ABSTRACT

Large-scale storage arrays are always in high demand by universities, government agencies, web search engines, and research laboratories. This unvarying need for more data storage has begun to push storage array magnitudes into an unknown stratum. As storage systems continue to outgrow the terabyte class and move into the petabyte range, these colossal arrays begin to show design limitations.

This thesis focuses primarily on disk drives as the building blocks of reliable large-scale storage arrays. As a feasibility baseline, the overall reliability of large-scale storage arrays should be greater than that of a single disk. However, petabyte- and exabyte-sized systems, requiring thousands to millions of disk drives, present a serious challenge in terms of reliability. Therefore, multi-level redundancy schemes must be used in order to slow these dwindling reliabilities.

This work, based upon the previous research of redundant arrays of independent disks (RAID) by Patterson et al., introduces the reliability analysis of dual- and tri-level Grouped RAID (GRAID) configurations. As storage arrays rapidly increase in size, the use of multi-level redundancy is essential. Design recommendations for various large-scale storage arrays, ranging from 100 Tebibytes (TiB) to 100 Exbibytes (EiB), can be generated using the custom reliability calculator tool written in MATLAB. The analysis of these design recommendations shows that dual-level GRAID configurations are only recommended for array magnitudes up to 5 PiB. Beyond this threshold, tri-level GRAID demonstrates feasibility for storage magnitudes up to 100 EiB and beyond.

CHAPTER 1

1. INTRODUCTION

1.1. Motivation

Since the early days of computing data storage has always been a critical system component. Over this time there have been numerous technology hurdles which needed to be overcome in order to meet the world's storage demands. A storage device is defined as an end device which physically stores data on a storage medium [SIMI03]. Common types of storage devices include: disk drives, physical memory, optical disks, and tape libraries. This thesis will focus primarily on disk drives as the building blocks of reliable large-scale storage arrays.

Today's mission critical data centers require 100% availability and highly reliable storage facilities. Large-scale storage arrays are in high demand by universities, government agencies, web search engines, and research laboratories. For example, at the CERN laboratory - the world's largest particle physics centre for nuclear research - the new Large Hadron Collider (slated for completion at the end of 2007) will generate 15 Petabytes (PB) of data annually [MOND03]. To accommodate this, its storage arrays need to be highly scalable and dependable in order to preserve access to this data. These storage arrays, requiring thousands to millions of disk drives, present a serious challenge in terms of reliability. As the number of components in any system increases, the overall reliability of said system will diminish [NARE66]. Therefore, redundancy in the form of extra disk drives must be used in order to slow the dwindling reliabilities of large-scale storage arrays. However, there exist limitations on the number of disk drives which should be used to safely and reliably store data. Preferably the overall reliability of large-scale storage arrays should be greater than that of a single disk. The problem is that large-scale storage arrays are approaching these limitations. When these limits are pushed, storage systems can become unstable. Organizations requiring a cost effective solution with high storage efficiency are beginning to realize the flaws in their current designs. One commonly encountered problem in these massive systems involves the failure of multiple disks during a rebuild. This is usually due to correlated environmental issues or the failure of other critical hardware, *e.g.*,

cables, storage controllers, disk enclosures. However, the increased frequency of unrecoverable bit errors (UBE) in large-scale storage arrays (amplified by the amount of data read during a rebuild) will prove to be one of the most critical issues in the design of reliable storage systems.

1.2. Thesis

This thesis is based upon previous research done by [PATT88] and [CHEN94]. Their original introduction and evaluation of redundant arrays of independent disks (RAID) in the late '80s was ground breaking for its time. However, with increases in disk sizes and the exponential increase in storage demand, no recent analysis on the reliability of large-scale systems has been performed. Therefore, this thesis will address the design issues and limitations involved with large-scale storage arrays using current and future storage technology. Design recommendations for various large-scale storage arrays - ranging from 100 Tebibytes (1 TiB = 2^{40} bits) to 100 Exbibytes (1 EiB = 2^{60} bits) - will be provided based on the analysis of these reliability metrics. These recommendations will be facilitated via a custom tool written in MATLAB, a numerical computation tool. This reliability calculator, specifically written for large-scale storage arrays, has many user changeable parameters such as: array size, disk type, disk size, and redundancy level(s). The dependency between each of these variables will be analyzed to provide better design recommendations. The analysis will also make use of recent case studies concerning the actual reliability of disk drives. These studies have discovered that manufacturer's reliability ratings are in some cases 3.4 times higher than those seen in actual storage environments. By using these updated variables to recalculate the reliability of modern large-scale storage arrays, their limitations become evident.

1.3. Summary

The above problems associated with designing large-scale storage arrays will be addressed in the following four chapters. Chapter 2 will provide an introduction to what a large-scale storage array is and why they are needed. Chapter 3 will focus on the various schemes used to provide redundancy in storage arrays. Chapter 4 will establish the various metrics to be used for

evaluating redundant storage arrays. Chapter 5 will evaluate the reliability metrics defined in Chapter 4 using analytical resources. Lastly, Chapter 6 will provide conclusions and recommendations for designing large-scale storage arrays.

CHAPTER 2

2. LARGE-SCALE STORAGE ARRAYS

2.1. What Are Storage Arrays?

A storage array is a structured group of disk drives that can collectively provide better data read/write performance and overall storage reliability than the collection individually [SIMI03]. By incorporating redundant disks into storage arrays, the overall reliability should ideally exceed that of a single disk. This redundant grouping of multiple disks is commonly referred to as a Redundant Array of Independent (formerly Inexpensive) Disks (RAID). With RAID, groups of disk drives can be managed collectively creating array magnitudes exponentially larger than the size of a Single Large Expensive Disk (SLED) [GUPT02]. This technology can also provide benefits in the form of: increased reliability, performance, or both [JEPS03]. When examining the read/write performance of a single disk drive, the data transfer rates are rather limited. For example, business class Serial Advanced Technology Attachment (SATA) and Serial Attached Small Computer System Interface (SAS) disk drives have maximum sustained throughputs of around 72 Megabytes per second (MB/s) and 100 MB/s, respectively [SEAG07a][SEAG07b]. By having multiple disk drives an aggregate throughput can be attained by reading and writing information to all the disks in parallel. The performance benefits gained with storage arrays is one of the key reasons for their popularity.

The definition of what is considered to be a large-scale storage array constantly changes with time. This paper will focus primarily on volume sizes ranging from pebibytes (PiB) to exbibytes (EiB). With current individual disk drives reaching capacities of 500 Gigabytes (GB) to 1 Terabyte (TB) in size, it is easy to construct multi-terabyte arrays with only a few disk drives. Next generation disk drives are beginning to show rapid growth in storage capacity. New methods in data storage, such as perpendicular recording, allow for high aerial density (or bit density). This would allow disk drives to store two to five times their current capacity [SEAG06]. However, even with a steady progression in disk size, pebibyte and exbibyte arrays still require a significant number of disk drives to be formed. Table 1 illustrates the **minimum**

number of disk drives (assuming a size of 500 GB) needed to achieve a desired array magnitude. By taking the array magnitude and dividing it by the disk size, the **minimum** number of disks required is produced. These arrays requiring thousands to millions of disk drives present a serious problem in terms of reliability. As the number of components in any system increases, the overall reliability of said system will diminish [PATT88]. More on reliability calculations of disk arrays can be found in Chapter 4.

Table 1: Array Magnitudes and Minimum Number of Disks Required

| Magnitude | Symbol | Binary Value | Minimum # of Disks |
|--|---------------|---------------------|---------------------------|
| <i>1 Kibibyte</i> | <i>KiB</i> | 2^{10} | <i>1</i> |
| <i>1 Mebibyte</i> | <i>MiB</i> | 2^{20} | <i>1</i> |
| <i>1 Gibibyte</i> | <i>GiB</i> | 2^{30} | <i>1</i> |
| <i>1 Tebibyte</i> | <i>TiB</i> | 2^{40} | <i>3</i> |
| 1 Pebibyte | PiB | 2^{50} | 2418 thousand |
| 1 Exbibyte | EiB | 2^{60} | 2.476 million |
| <i>1 Zebibyte</i> | <i>ZiB</i> | 2^{70} | <i>2.535 billion</i> |
| <i>1 Yobibyte</i> | <i>YiB</i> | 2^{80} | <i>2.596 trillion</i> |
| <i>Assuming maximum current disk size of DiskSize = 500 GB \approx 465.66 GiB</i> | | | |

When dealing with hard drives and storage, an important issue to be aware of is the difference between “marketed” capacity and actual capacity [DISH06]. Drive manufacturers prefer to use the SI (metric system) to advertise a drive’s storage capacity. This is due to the actual storage capacity (measured in binary) being less than the marketed SI capacity. It is most often stated on drive specification sheets as 1GB=1,000,000,000bytes (1×10^9). Even though this does not represent the actual storage capacity, manufacturers continue to follow this industry standard [DISH06]. The actual data storage is measured in binary using powers of 2, e.g., $2^{10} = 1,024$ bytes or $2^{30} = 1,073,741,824$ bytes= 1 Gibibyte (GiB). Most operating systems use this binary definition when referring to the size of files or a disk volume. Similar to hard disks, when bandwidth measurements are taken, these values are denoted using decimal powers of 10,

e.g., $10^3 = 1,000$ bytes $10^3 = 1000$ bytes = 1 KB) [SIMI03]. Take, for example, a disk drive marketed as 500 GB. To obtain the actual storage capacity seen by the operating system, multiply this size by the ratio of SI to binary units of a gigabyte ($10^9 / 2^{30}$, an approximation). Therefore, the actual number of gigabytes seen by the operating system will be 465.66 GiB ($500\text{GB} \times (10^9 / 2^{30}) = 465.66\text{GiB}$). This inconsistency plays a critical role in determining the “actual” number of disk drives needed to meet the storage requirements of large-scale storage arrays. Note that the information in Table 1 uses the actual disk capacity for determining the minimum number of disks required. When factors such as redundant disks and filesystem overhead are taken into consideration, the total number of disks required for each desired magnitude will be much greater. To further illustrate the importance of this issue, Table 2 shows the conversion ratio and percent difference between decimal and binary storage prefixes. Since these differences are logarithmic, the larger capacity arrays would have a significant discrepancy between expected and actual capacity. This binary storage convention will be used throughout this thesis when determining the reliability and recommended redundancy configurations. Disk drives need to first be converted from a decimal capacity to their binary equivalent. Since the desired array magnitudes are already in binary form, the correct number of disk required for each redundancy configuration will be produced.

Table 2: Percentage Difference in Decimal to Binary Conversion

| Name | Dec ÷ Bin | Example | Percentage Difference |
|----------------------|-----------|-------------------|-----------------------|
| kilobyte → kibibyte | 0.976 | 100 kB ≈ 97.6 KiB | -2.3% |
| megabyte → mebibyte | 0.954 | 100 MB ≈ 95.4 MiB | -4.6% |
| gigabyte → gibibyte | 0.931 | 100 GB ≈ 93.1 GiB | -6.9% |
| terabyte → tebibyte | 0.909 | 100 TB ≈ 90.9 TiB | -9.1% |
| petabyte → pebibyte | 0.888 | 100 PB ≈ 88.8 PiB | -11.2% |
| exabyte → exbibyte | 0.867 | 100 EB ≈ 86.7 EiB | -13.3% |
| zettabyte → zebibyte | 0.847 | 100 ZB ≈ 84.7 ZiB | -15.3% |
| yottabyte → yobibyte | 0.827 | 100 YB ≈ 82.7 YiB | -17.3% |

2.2. Unrecoverable Bit Errors

Every component in storage arrays has a reliability metric assigned by manufacturers. Disk drives have three primary metrics: Annual Failure Rate (AFR), Mean Time Between Failures (MTBF), and Bit Error Rate (BER). These metrics are commonly used to calculate the overall system reliability. The bit error rate is a ratio of the number of bits received in error to the total number of bits received [GUPT02]. Desktop class Parallel ATA (PATA) disk drives commonly have a BER of $1:10^{13}$ (1 bit in 10^{13} bits). Business class SATA and SAS disk drives usually have a BER of $1:10^{14}$ and $1:10^{15}$, respectively. Enterprise class SCSI and Fibre Channel (FC) disk drives ideally have a BER of $1:10^{16}$. When disk drives are constantly being read and written to, *e.g.*, most data centers, BER is the statistical chance of one bit being erroneously received during these continuous transactions. Data writes can circumvent these errors by marking the sector as bad, and then re-writing the data to another disk sector. However, bit errors are most problematic while reading data from a sector of a disk drive. When they do occur, the end result is an unrecoverable bit error (UBE). In order to correct a UBE, the data on this failed disk must be rebuilt from the other redundant disks in the storage array. In the event a second sector on another disk drive is unreadable, then all data could potentially be lost (unless additional levels of redundancy are in place). Previous research by [CHEN94] has shown that the probability of encountering a UBE during a rebuild is much higher than that of another single disk failing due to hardware issues. To compound this issue, arrays with more disks (and more importantly larger capacity disks) can become unreliable if not designed properly. Chapter 5 will analyze the impact of bit errors on storage arrays and provide design recommendations to compensate for this problem.

2.3. Storage Demand and Its Projected Growth

The constant need for more data storage has begun to push storage arrays into an unfamiliar stratum. As storage systems continue to grow in the pebibyte range, the reliability of such colossal arrays begins to show limitations in their intended design. Issues such as reduced overall system MTBF due to the sheer number of disk drives and longer rebuild times have resulted in unstable storage environments for important data.

There are many predictions of data growth which exist. Commonly used to estimate the exponential growth of the number of transistors on a chip is Moore's Law [PARH05]. Moore's law states that processor speeds will double every 18 months. In the past, predictions of storage growth have also applied Moore's Law. However, recent predictions have surfaced from a scientist at Seagate by the name of Dr. Mark Kryder. He estimates the rate of disk capacity increase will far exceed that of Moore's law. As a pivotal engineer and visionary at Seagate, he has pushed for bigger and better storage devices. A goal set by Kryder in 1998 suggested crowding 100 gigabits onto a single square inch. Within 7 years (2005) this goal was shattered, representing a 1000 fold capacity increase [WALT05].

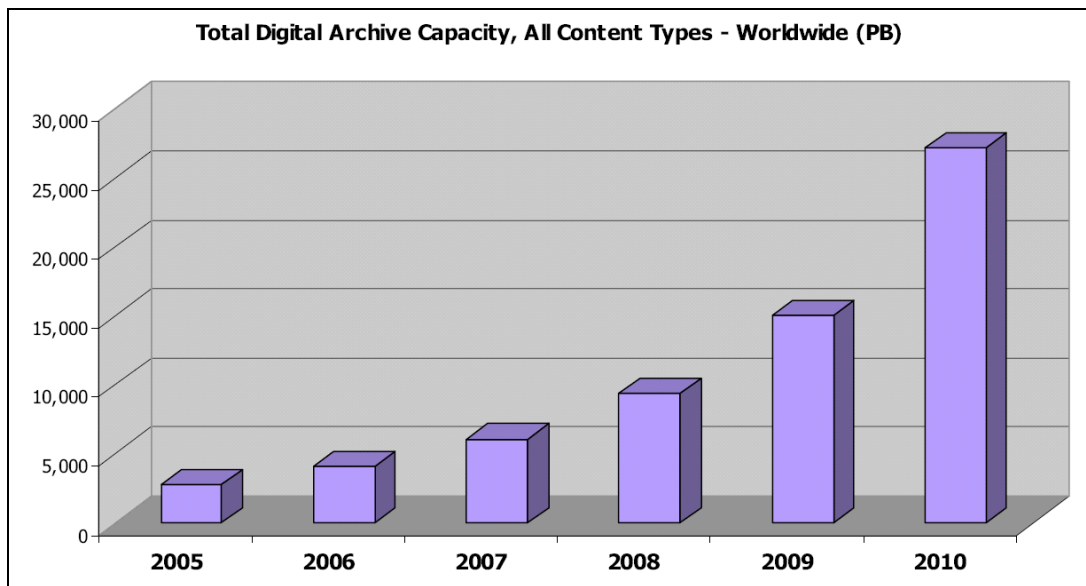


Figure 1: Total Worldwide Digital Archive Capacity [MCKN06]

A recent study by the Enterprise Strategy Group has shown that this demand for more storage is fueled by many factors. The study encompassed over 500 information technology, business, and management professionals at various public and private organizations. The data gathered showed that regulatory compliance, litigation support and records management are some of the key drivers for storage demand. Other factors such as email and database archiving are also requiring long-term storage. In today's legal world, courts and regulators are frequently demanding electronic e-mail records from many businesses. As a consequence, dropping disk prices and the need for quick and efficient regulatory compliance has forced most organizations

to migrate from the traditional tape system to a disk-based data retention system. Based on data gathered in this study, the total worldwide digital archive capacity will increase tenfold between 2005 and 2010 [MCKN06]. Figure 1 illustrates this anticipated growth in petabytes of data archiving worldwide.

Currently, disk storage growth is only seeing a 40% increase per year (in comparison with the 100% annual rates of the past). This is primarily due to how bits on disk platters are magnetized. The condensed longitudinal recording of bits (technology used for nearly 50 years) has reached a limitation where thermal energy is demagnetizing the bits (known as the superparamagnetism phenomenon) [SEAG06]. The current push by Seagate, Hitachi, and other storage companies is to adopt a perpendicular recording of data on disk drives. This recording method stores bits with their magnetism vertically aligned, allowing for more bits per square inch. Figure 2 illustrates the differences between longitudinal and perpendicular recording technology. It is predicted that perpendicular recording will allow up to 1 Tbps (Terabits per square inch) in next generation disk drives [KRYD03]. Modern disk drives using perpendicular recording have already reached capacities of 1 TB.

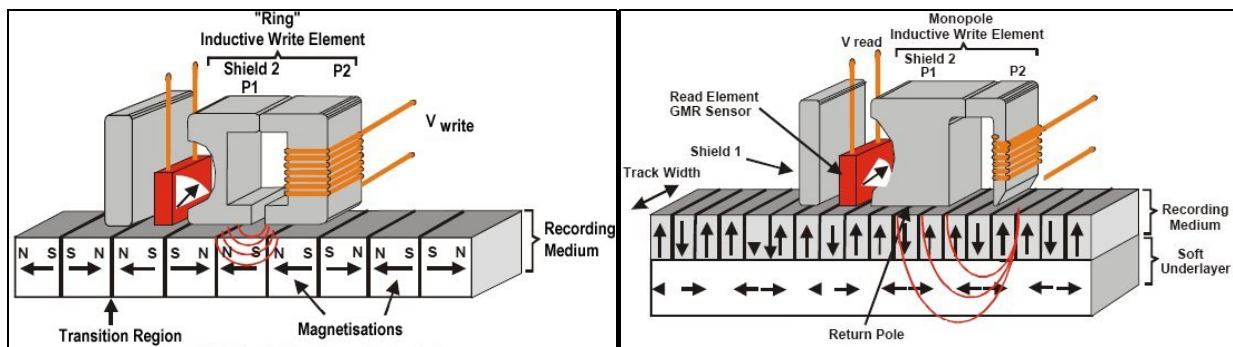


Figure 2: Longitudinal recording (left) and Perpendicular recording (right) [HITA06]

The method in which data is organized and addressed has also played a crucial role in the rate of size increase for storage volumes. For example, the FAT (File Allocation Table) file system used by the Windows OS throughout the 1990s had disk capacity limitations of around 4 GB. This temporarily hindered growth in storage systems. With the use of 64-bit addresses in NTFS (New Technology File System), HFS+ (Hierarchical File System Plus), and other modern file systems, this obstacle was temporarily overcome. Recently, the use of 128-bit addressing in Sun

Microsystems's Zettabyte File System (ZFS) has gained attention. ZFS has the capability of managing volume or single file sizes of around 16 EB [SUN04]. Nevertheless, how long before these current file system limitations are encountered? Present and future generations of scientists and engineers need to be kept aware of legacy limitations in storage design so that they can be prepared.

CHAPTER 3

3. RAID SYSTEMS

3.1. The Origins of RAID

The key component of any reliable and large-scale storage array is RAID technology [PATT88]. By using Redundant Arrays of Independent (formerly Inexpensive) Disks (RAID), groups of disk drives can be managed collectively. Array magnitudes exponentially larger than the size of a Single Large Expensive Disk (SLED) are achievable with RAID [GUPT02]. This technology can also provide benefits in the form of: increased reliability, performance, or both [JEPS03]. Data read/write performance improvements are achieved by striping chunks of data across multiple disks [BISC97]. This allows for parallel data transfers with both reads and writes, therefore achieving a combined throughput. Data striping also provides load balancing across all disks uniformly. This alleviates hot spots commonly found in concatenated disk arrays [CHEN94]. Figure 3 illustrates an example of data striping where blocks of data are written in stripes across multiple disk drives.

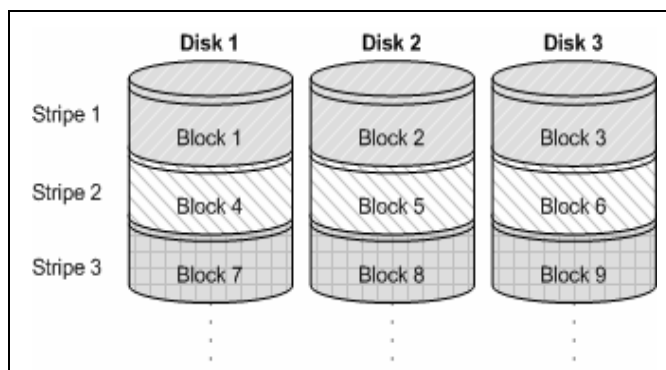


Figure 3: Data Striping Scheme

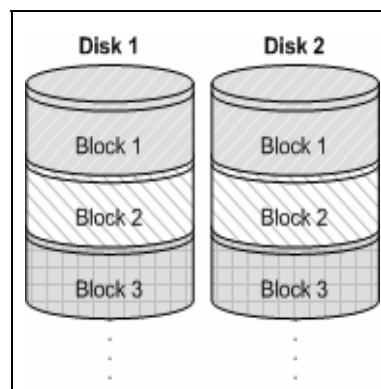


Figure 4: Data Mirroring Scheme

However, data striping across many disks produces a highly unstable and unreliable system. For example, a striped array with 100 disks is 100 times more likely to encounter a failure than a single disk [CHEN94]. To alleviate this issue, RAID systems utilize parity disks in each stripe for added redundancy. Parity is implemented as a simple binary “exclusive or” (XOR) of the bits in a data stripe. Another way of describing even (odd) parity is that the number of 1-bits in a

stripe is even (odd). For example, taking the XOR of each disk's first bit (bit 0 within the stripe) will produce the first bit on the parity disk, even (1) or odd (0). To improve performance, multiple XOR computations are performed in parallel for each bit of the stripe. Parity allows for disk arrays to achieve high storage efficiency while still maintaining some degree of reliability. Storage efficiency is defined as the degree to which a system or component performs its designated functions with minimum consumption of available storage [IEEE07]. Therefore, by limiting the number of redundant disks needed, the overall storage efficiency is improved.

Another method utilized in RAID's involves the mirroring of data blocks across two separate disk drives [BARK02]. Mirroring provides the highest degree of reliability and redundancy since data on one disk has a mirrored copy on another drive. The primary advantage to disk mirroring is that error correction is not necessary to rebuild data from a failed drive. Data is simply replicated from the active drive. Figure 4 illustrates an example of data mirroring where a single block of data is written to two different disk drives. The following sections provide an overview of the various base RAID levels, nested approaches, and finally a grouped structure to further improve redundancy and reliability.

3.2. Base RAID Levels

The initial RAID schemes established in [PATT88] were levels 1 through 5. These levels should be viewed as different configurations of disk arrays which provide various tradeoffs in reliability, performance and cost. RAID level 0 was not initially included in [PATT88] due to its lack of redundancy. Since RAID's beginning, there was a higher ordered RAID level 6 added in [CHEN94] to improve the overall reliability of large-scale storage arrays. The most commonly used RAID levels today are 0, 1, 3, 5, and 6. Some of the lesser-used RAID levels, *e.g.*, 2 and 4, did not gain as much of a following due to their inefficient distribution of parity information and lack of practicality. Redundancy is needed in RAID arrays to compensate for their lower overall reliability (see Chapter 4) [PATT96]. RAID Levels 3-5 compensate for this by using parity for data protection. Level 6 improves upon RAID 5's data protection by adding another separate parity scheme. The key benefit to using these parity schemes is that any corrupt data or disk failure can ideally be rebuilt from the other redundant disks in the array [PATT96]. The following provides an overview of each RAID level.

RAID 0

RAID level 0 stripes or spreads data across two to many drives in parallel. Stripes consist of data broken into blocks. For example, in Figure 5 the capital letters represent contiguous blocks of data striped across the four disk drives in the array. By doing so, this allows for the highest throughputs. However, the primary disadvantage to this level is that there is zero fault tolerance, *i.e.*, no parity. If one disk is lost in the array, then all data is lost [GUPT02].

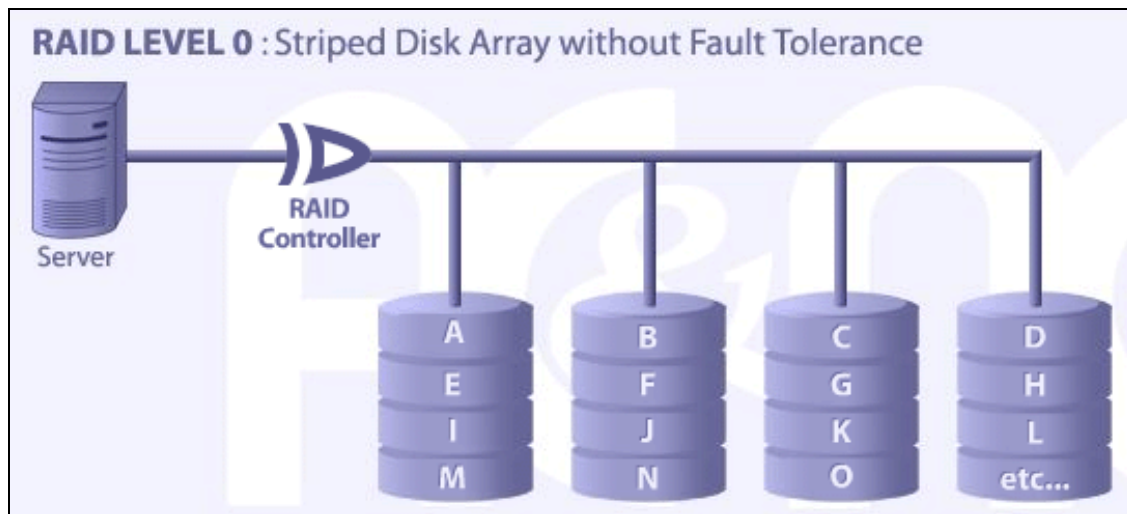


Figure 5: RAID 0 [ACNC07]

Level 0 was not considered in RAID's introduction [PATT88]. This was due to the fact that this level offers no level of redundancy at all. Therefore, how could level 0 be considered a "Redundant" Array of Independent Disks. In essence, it is not. However, the data read/write performance benefits of level 0 arrays far exceeds that of any single disk or higher ordered RAID level. This makes level 0 arrays ideal for temporary storage in super computing applications. For this reason, level 0 is still defined in almost every RAID text.

With the use of new storage technology such as Serial Attached SCSI (SAS), overall data read/write performance of arrays can approximately scale linearly with additional disk drives. SAS has a unique serial interface which allows for point-to-point access to each disk drive. Therefore, each disk has a dedicated input and output (I/O) channel allowing for improved data transfers when sending and receiving data to an array of disks. RAID level 0 provides the

means for such an array of disks to be realized and utilized in high-performance environments. The linear scalability of SAS in comparison to other storage technology is represented in Figure 6. To alleviate the reliability issues and lack of redundancy associated with RAID level 0, a mirrored approach can be used.

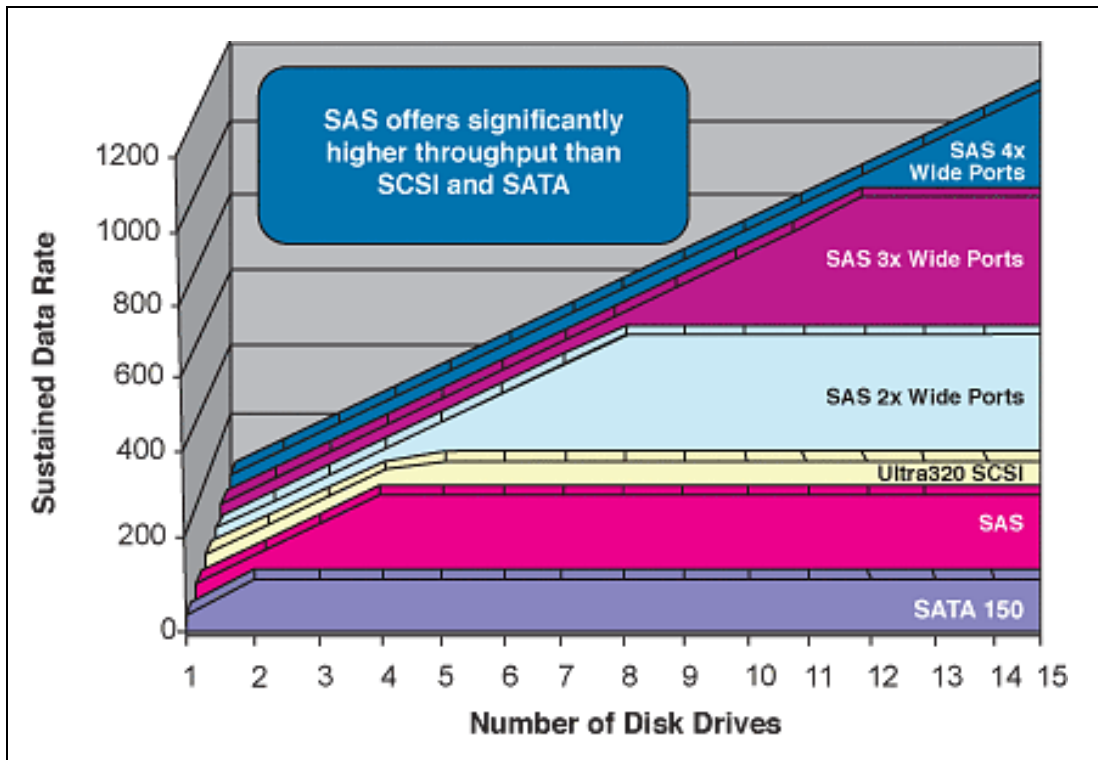


Figure 6: Relative Performance of SAS [ADAP07]

RAID 1

In comparison with the previous Level 0, Level 1's mirroring approach represents the complete opposite end of the RAID spectrum. On the one hand (level 0) you have 0% redundancy, and on the other (level 1) you have 100% redundancy. This mirroring scheme offers complete redundancy, where one drive is an exact copy of the other. In this configuration data read/write performance remains high; however 50% of the disk space is sacrificed to achieve 100% data redundancy. In comparison with all other RAID levels, disk mirroring has the highest overhead [GUPT02].

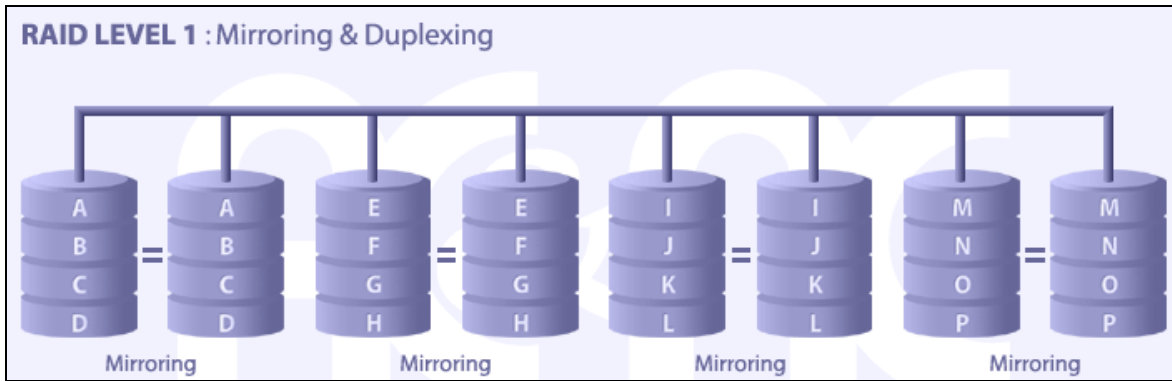


Figure 7: RAID 1 [ACNC07]

Another benefit to disk mirroring is a lower mean time to recover (MTTR). When a mirrored disk fails (assuming that a hot-spare is available) a direct disk-to-disk copy is performed using the good disk. No parity computations are required. Therefore, the time in which a mirrored array is in a degraded state is less than that of a RAID level with parity. Level 1 disk mirroring is best suited for database applications where small writes and high I/O transaction rates are needed. RAID 1 arrays on average have higher read rates compared with level 0. This is due to selective scheduling, where disks having a shorter seek and rotational delay can be used for data reads [CHEN94]. An example of an 8-disk RAID 1 array is shown in Figure 7.

RAID 2

Level 2 was initially designed to mimic the redundancy schemes used in memory systems. Striping is used with ECC (Error Correction Coding) to store the parity information. ECC is used with multiple redundant disks to recover data in the event of a disk failure. Similar to the memory systems which this level resembles, the key advantage to this scheme is that very high data transfer rates are attainable [GUPT02].

In the event of a drive failure, multiple redundant disks are needed to first determine which drive has failed. However, only one of the redundant disks is needed to recover the data. This is unnecessary in modern storage arrays since disk failures are easily identified by the storage controllers [CHEN94]. Therefore, this level is rarely implemented due to its high cost and

impracticality. Figure 8 represents a four disk RAID 2 arrays with three disks used for error correcting. An improvement upon level 2 in storage efficiency is level 3. This reduces the number of parity disks to only one, and relies on the storage controller to identify any failed disk drives.

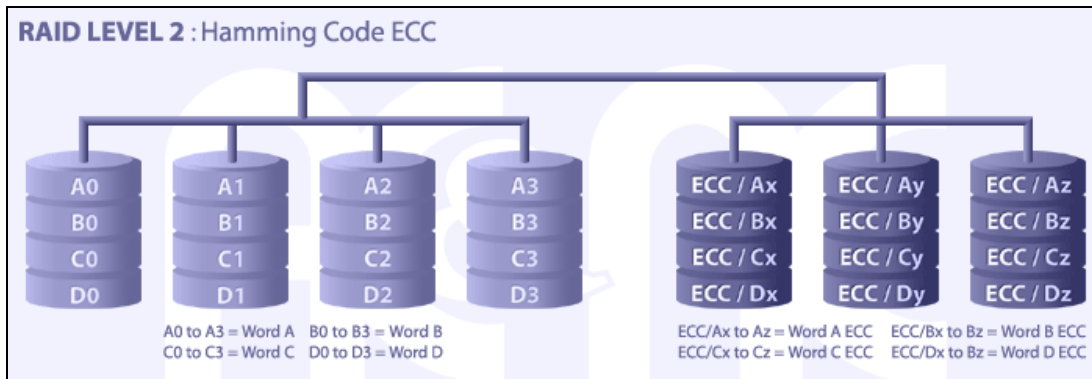


Figure 8: RAID 2 [ACNC07]

RAID 3

RAID level 3 improves upon the previous level by eliminating the extra disks required for identifying a failed disk drive. This reduction in disk overhead also produces a more reliable storage array [PATT88]. In this level, byte-interleaving parity is used and all parity information is stored on one disk drive. The parity information is generated using XOR to determine if the number of 1-bits in the stripe is even or odd. Then, the parity disk is written accordingly to give the desired overall parity. The primary disadvantage to this approach is that using only one parity disk creates a bottleneck in performance. Data reads will need to access all data disks and data writes will need to access all data disks plus the parity disk. For data reads the parity disk does not hold any data, therefore it cannot add to the performance. Therefore, only one transaction can be processed at a time since all disks must be accessed for each transaction. With a distributed parity approach, *e.g.*, RAID 5, this is no longer an issue [CHEN94]. Byte-interleaving works best with high-bandwidth applications requiring large data transfers. RAID 3 arrays are commonly paired with applications dealing with image manipulation using large graphic files and streaming video [GUPT02]. Figure 9 shows a four disk RAID 3 array with a single disk containing parity information of the data stripes.

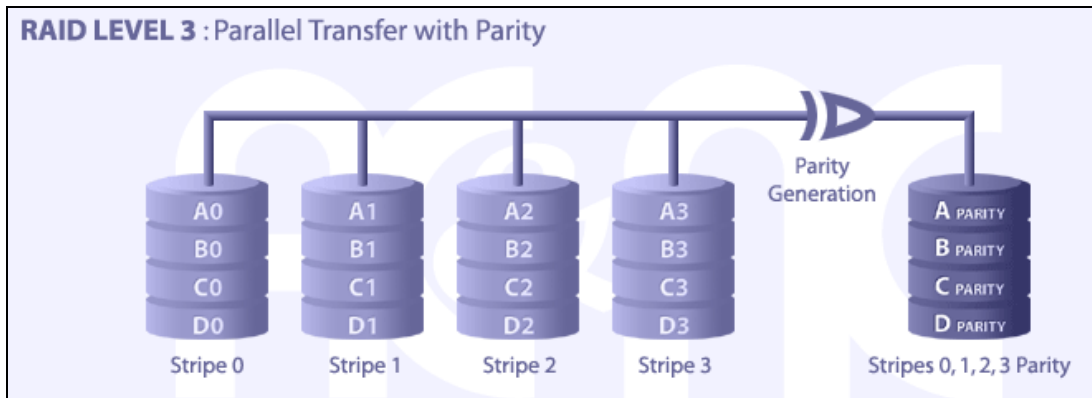


Figure 9: RAID 3 [ACNC07]

RAID 4

Dissimilar to Level 3, RAID 4 uses block rather than byte interleaving. Multiple arbitrary sized blocks are grouped to form a striping unit [CHEN90]. For small read requests which are less than the size of the striping unit, only a single data disk needs to be accessed. An advantage to this block approach is that read operations can be completed very fast with one read access [GUPT02]. On the other hand, small write operations will require four disk operations. The first is to write the new data to a disk. Next, the new parity must be computed by reading the old data and also reading the old parity. Finally, the newly computed parity is written to the parity disk. These actions are commonly known as *read-modify-write* procedures [PATT88]. Due to the bottleneck created by using a single parity disk (shown in Figure 10) and the high penalties for small writes, RAID 4 arrays are generally not implemented. The preferred means is to use a block-interleaved distributed-parity approach, realized with RAID 5.

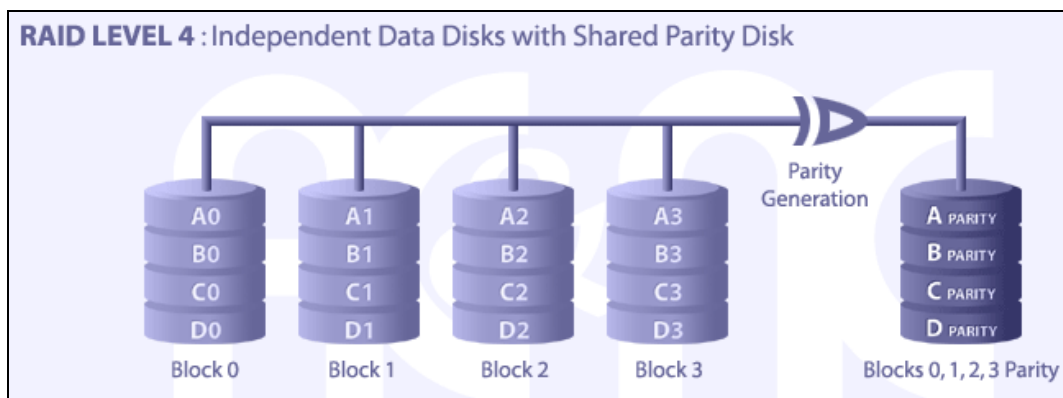


Figure 10: RAID 4 [ACNC07]

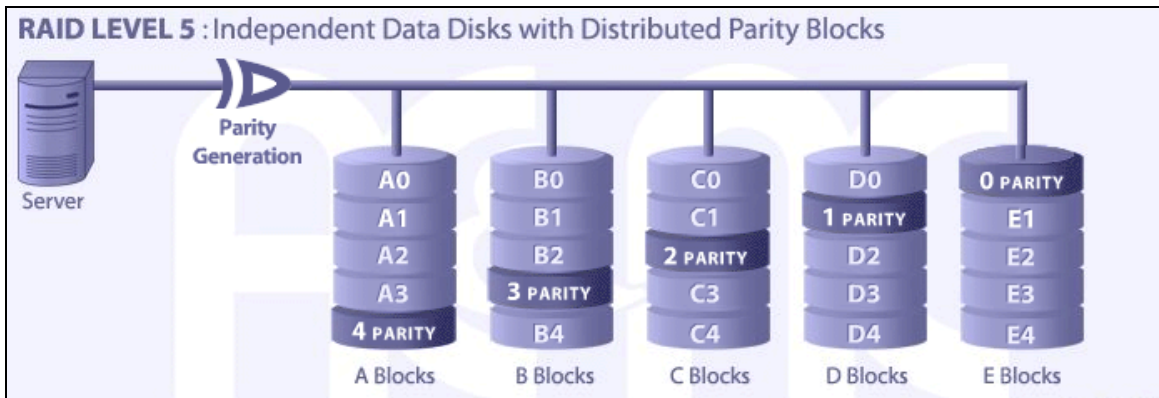


Figure 12: RAID 5 [ACNC07]

RAID 6

To meet the reliability demands of growing storage systems, a more redundant RAID scheme was needed. An important addition to the RAID family is Level 6. This level acts as an extension of RAID 5's parity scheme (P) by adding on a second independent parity scheme (Q), e.g., Adaptec, EVENODD, Reed-Solomon or X-Code encoding, creating dual parity (P+Q) for added fault tolerance. This dual-parity scheme allows for up to two disk failures to occur without risking any data loss.

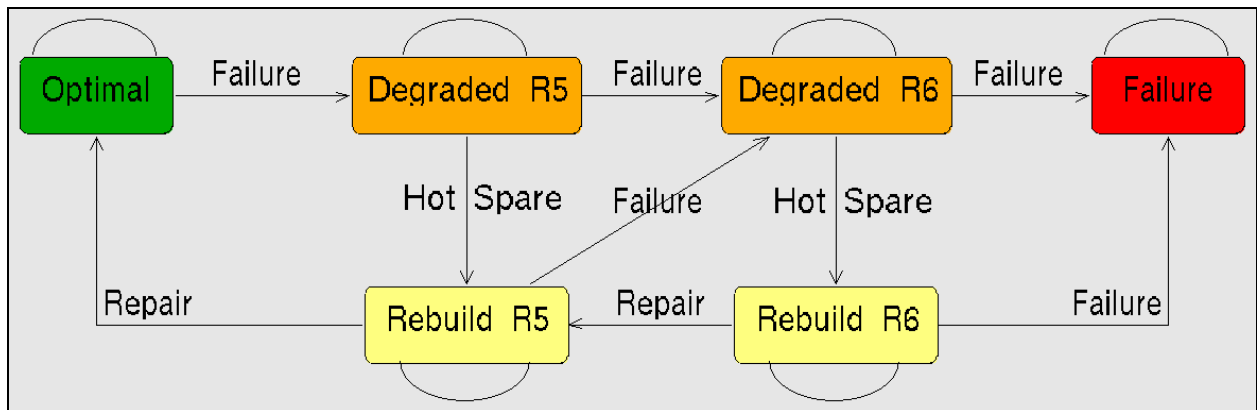


Figure 13: State Diagram for RAID 6 System

The six states commonly associated with a RAID 6 system is portrayed in Figure 13. The Reed-Solomon parity coding commonly used with level 6 arrays is very complex. The (Q) parity calculation uses bit strings rather than the single bits used with XOR parity (P). Reed-Solomon coding is based on the algebra of Galois fields which utilize basic algebra functions (addition,

subtraction, multiplication, and division) to calculate checksums. These bit strings are grouped into blocks which then have redundant checksum bits added to verify the block's integrity. Since RAID 6 uses two disks for redundancy, the Galois field operations limit the maximum number of disks to 255 ($2^8 - 1$) [SCHU06]. Luckily, this secondary (complex) parity is not needed for rebuilding a single disk failure [SIMI03]. As with RAID 5 arrays, the XOR parity is used for the first disk failure. Only in the event of a second disk failure will the secondary parity scheme (Q) be necessary. This is beneficial to small arrays with a low occurrence of encountering a second disk failure.

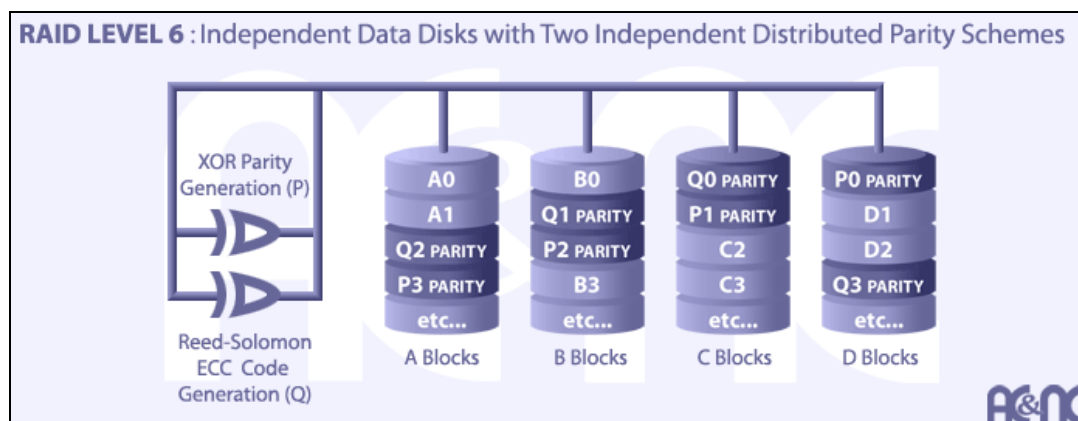


Figure 14: RAID 6 [ACNC07]

An example of a 4-disk RAID 6 array is represented in Figure 14. This figure shows what is required when the minimum number of disks is used. The benefits of using one encoding vs. another are beyond the scope of this thesis. The primary feature of RAID 6 is its double-disk failure protection.

3.3. Nested RAID Levels

Nested levels are formed from a combination of two RAID levels. Nested systems are also commonly referred to as dual-level arrays [TREA03]. The benefits of these configurations are: improved data read/write performance and added redundancy in some configurations. The need for nested levels was created by storage arrays requiring better performance from their already existing RAID 1 and RAID 5 arrays. Ideally RAID 0 should be used at the top layer. This will provide better performance with data striping and keep the number of drives required for a lower

layer rebuild small. Initially, these nested RAID implementations existed only in software. Logical volume managers (LVM) simplified this process with the creation of logical volumes. For example, each RAID 0 segment in Figure 15 would be treated as a logical volume (LV), each consisting of multiple physical drives. Combining these two logical volumes with RAID 1 mirroring forms a volume group (VG). The nested RAID levels 0+1, 10, and 50 in addition to standard base levels, are commonly integrated into most storage controllers using a similar approach.

RAID 0+1

This nested organization consists of a top level RAID 1 array with striped level 0 segments. I/O performance is higher than RAID 1 due to the striped segments and the overhead is equivalent to that of a RAID 1 system [SIMI03]. Illustrated in Figure 15 is a 4-disk RAID 0+1 array. This nested configuration is usually not favored due to its slightly lower reliability than a RAID 1 array. For example, if a single disk fails then the other half of the array essentially becomes a RAID 0 system. Therefore, the probability of another disk failing still must consider half the disks ($N/2$). Alternatively, with a RAID 10 approach only the probability of a single disk must be taken into consideration, *i.e.*, the other half of the mirrored pair. More on the reliability of RAID 0+1 in comparison to RAID 10 can be found in Chapter 4.

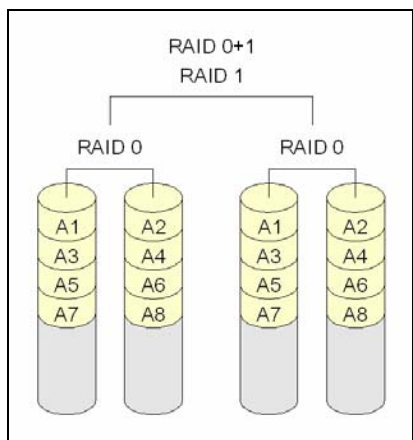


Figure 15: RAID 0+1

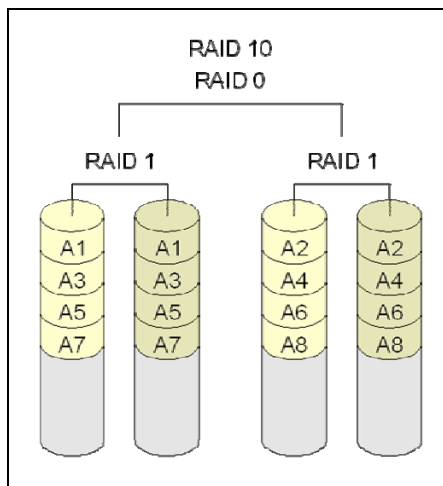


Figure 16: RAID 10

RAID 10

This nested approach consists of mirrored RAID 1 segments with top level RAID 0 striping. The end result is an array with an equivalent overhead and fault tolerance as a Level 1 system, but with improved I/O performance due to the data striping at the top level. For this reason, systems which previously used RAID 1 data mirroring are starting to migrate towards this nested approach [SIMI03]. An example of a 4-disk RAID 10 array is depicted in Figure 16. Similarly with RAID 1 arrays, this nested configuration is best suited for critical database applications. It is also the preferred method of nesting compared with RAID 0+1 due to the higher redundancy. This higher redundancy is due to mirroring at the lowest level. The probability of any disk in the array failing is the MTBF divided by the number of disks in the array (N). Since each disk has a mirrored pair, the probability of a second drive failing is simply the MTBF of the other drive divided by the remaining good disks in the group, *i.e.*, one. Conversely, with level 0+1 after the first drive failure occurs, the probability of a second disk failure is the MTBF divided by half the disks in the array ($N/2$). In addition to this, a much higher recovery time will be required depending on the size of the array. Therefore, it is ideal to keep lower levels of the array at the highest redundancy paired with the smallest repair time.

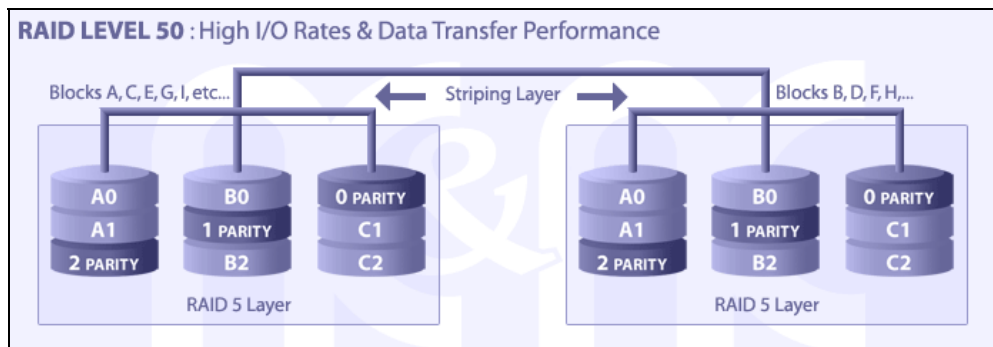


Figure 17: RAID 50 [ACNC07]

RAID 50

This nested approach consists of RAID 5 segments with top level RAID 0 striping. Similarly with levels 0+1 and 10, improved I/O performance is achieved through the data striping. Compared to a base level 5 system, this nested approach is more fault-tolerant. This is due to the grouping of two smaller level 5 arrays. A six disk example is represented in Figure 17. Each RAID 5 layer is considered a RAID group. In the best case scenario, up to two simultaneous disk

failures can occur (assuming they occur in separate groups). Still, if two disk failures occur in the same group, then data loss would occur. Inheriting the benefits of level 5 arrays, any small data requests can be performed very fast. However, since there are now two RAID 5 groups, the system is burdened with twice the parity overhead, *i.e.*, unique parity computations for each RAID 5 group [ACNC07].

3.4. Grouped RAID Levels

The logical sub-grouping of disks is not a new concept. This was initially introduced in [PATT88, GIBS88] with the use of groups in parity based arrays. With large-scale RAID arrays, the use of grouping is essential to upholding reliability. In [PATT88], the RAID reliability calculations for both grouped and non-grouped arrays were the same. If the number of groups was set to one (for small RAID arrays), this would produce the base levels shown in Figure 12 or Figure 14. However, most modern RAID definitions tend to always simplify these reliability calculations (for small scale arrays) so that the group size is equal to the number of useable data disks. To alleviate any further confusion these configurations will be classified as Grouped Redundant Arrays of Independent Disks (GRAID).

One of the key driving factors behind disk grouping involves their physical housing. In large-scale storage arrays, there are three main group classifications: disk enclosures, rack enclosures, and arrays. Disk enclosures are used to house the smallest grouping of drives usually ranging between 4 and 42 disks. These enclosures will typically have an integrated hardware RAID controller to handle the group's redundancy processes. The enclosure will also provide the backplane in which all drives in a group connect to. This backplane allows for failed disk drives to be hot-swapped without bringing down the whole enclosure. Enclosure size also plays a critical role in the physical space optimization of large-scale storage arrays. These sizes can range between 1 rack unit (1U) and 5U in height. This is the reason why enclosures can house variable number of disks, *i.e.*, correlates with the enclosure's rack unit height. Further analysis on the optimal selection of enclosure heights can be found in Chapter 5.

The next grouping of disks involves multiple enclosures to fill up a rack enclosure. A rack provides the physical housing for all its disk enclosure groups and distributes electricity to them.

The network switches used to facilitate external connectivity, *e.g.*, Fibre Channel, to the other disk enclosures in the storage array are also usually kept in rack enclosures. Racks enclosure heights can range from small (24U) models up to the larger (42U) enclosures. Ideally the largest height rack enclosure (42U) should be used in large storage arrays. This will allow for arrays to occupy the smallest footprint in server rooms, and also allow for storage growth in the future.

Lastly, the grouping of multiple rack enclosures forms a storage array. An array should ideally be contained within one site location to keep the I/O performance high and the latency low. The next logical grouping would involve the clustering of two or more storage arrays at different sites (geographically separated for added protection) to provide additional redundancy using failover technology. However, this is beyond the scope of this thesis. Future research should incorporate the reliability and failure probability of these site-to-site links.

The following defines six dual-level (50, 55, 56, 60, 65, 66) and four three-level (550, 560, 650 and 660) GRAID configurations. Each level consists of one of the base RAID levels 0, 5, or 6. For efficiency reasons, RAID 1 mirroring is not recommended for replicating data at single-site large storage arrays. Mirroring should be reserved for smaller critical data sets or for replicating data between multiple storage sites. In each of the following figures (Figure 18 through Figure 27), the use of P and Q parity disks does not represent a dedicated parity volume. These parity disks represent the same distributed parity scheme used with RAID 5 and 6. They are used to simply provide an understanding of what overhead is involved for each enclosure and the groups. The use of hot spare disks in each enclosure (denoted by HS) and in each sub-group is highly recommended. If possible, RAID 1 mirroring should be avoided in large-scale arrays due to its limiting storage efficiency (50%). The primary objective of grouping is to keep the enclosure sizes as large as possible (bounded by the maximum size of a large disk enclosure), while still maintaining sufficient reliability. This is done to maximize storage efficiency and to minimize the physical footprint of the storage array, *i.e.*, less overhead in the form of redundant disks. However, this maximization consequentially results in higher rebuild times, lowering the overall reliability. More on the tradeoffs between GRAID variables can be found in Chapter 5.

GRAID 50 / 55 / 56

A dual-level GRAID 50 array, illustrated in Figure 18, is constructed using multiple size-controlled RAID 5 enclosures together with RAID 0 striping at the top level. The top level data striping provides added read/write performance to the array. A dual-level GRAID 55 scheme, depicted in Figure 19, is created by combining multiple size-controlled RAID 5 enclosures together with RAID 5 distributed parity at the top level. This can induce a great deal of parity overhead as the number of enclosure increase. A dual-level GRAID 56 scheme, portrayed in Figure 20, is constructed by combining multiple size-controlled RAID 5 enclosures together with RAID 6 distributed dual-parity at the top level. This can also induce high parity overhead as the number of enclosures increase.

GRAID 60 / 65 / 66

A dual-level GRAID 60 scheme, represented in Figure 21, is derived by combining multiple size-controlled RAID 6 enclosures together with RAID 0 striping at the top level. The top level data striping provides added read/write performance to the array. A dual-level GRAID 65 scheme, illustrated in Figure 22, is constructed by combining multiple size-controlled RAID 6 enclosures together with RAID 5 distributed parity at the top level. This can induce a great deal of parity overhead as the number of enclosures increase. A dual-level GRAID 66 scheme, represented in Figure 23, is created by combining multiple size-controlled RAID 6 enclosures together with RAID 6 distributed dual-parity at the top level. This can cause extremely high parity overhead as the number of enclosures increase. However, this configuration does have the highest reliability of all other dual-level GRAID configurations.

GRAID 550 / 560

For these tri-level GRAID configurations it is best to find a balance between disk enclosure size and the number of enclosures in a rack. Rack optimization will be covered later in Chapter 5. A tri-level GRAID 550 scheme, illustrated in Figure 24, is constructed using multiple size-controlled RAID 5 disk enclosures in a rack enclosure with RAID 5 distributed parity across all the enclosures. At the top level, RAID 0 striping is performed across all rack enclosures, i.e., highly reliable GRAID 55 racks. A tri-level GRAID 560 scheme, depicted in Figure 25, is

created by combining multiple size-controlled RAID 5 disk enclosures in a rack with RAID 6 distributed dual-parity across all the enclosures. Again, at the top level RAID 0 striping is performed across all rack enclosures, i.e., highly reliable GRAID 56 racks.

GRAID 650 / 660

A tri-level GRAID 650 scheme, shown in Figure 26, is created by combining multiple size-controlled RAID 6 enclosures in a rack enclosure with RAID 5 distributed parity across all disk enclosures. At the top level, RAID 0 striping is performed across all rack enclosures, i.e., highly reliable GRAID 65 racks. A tri-level GRAID 660 scheme, depicted in Figure 27, is constructed by grouping multiple size-controlled RAID 6 disk enclosures in a rack enclosure with RAID 6 distributed dual-parity across all disk enclosures. Again, at the top level, RAID 0 striping is performed across all rack enclosures, i.e., highly reliable GRAID 66 racks. This configuration has the highest MTDL of all the tri-level GRAID configurations.

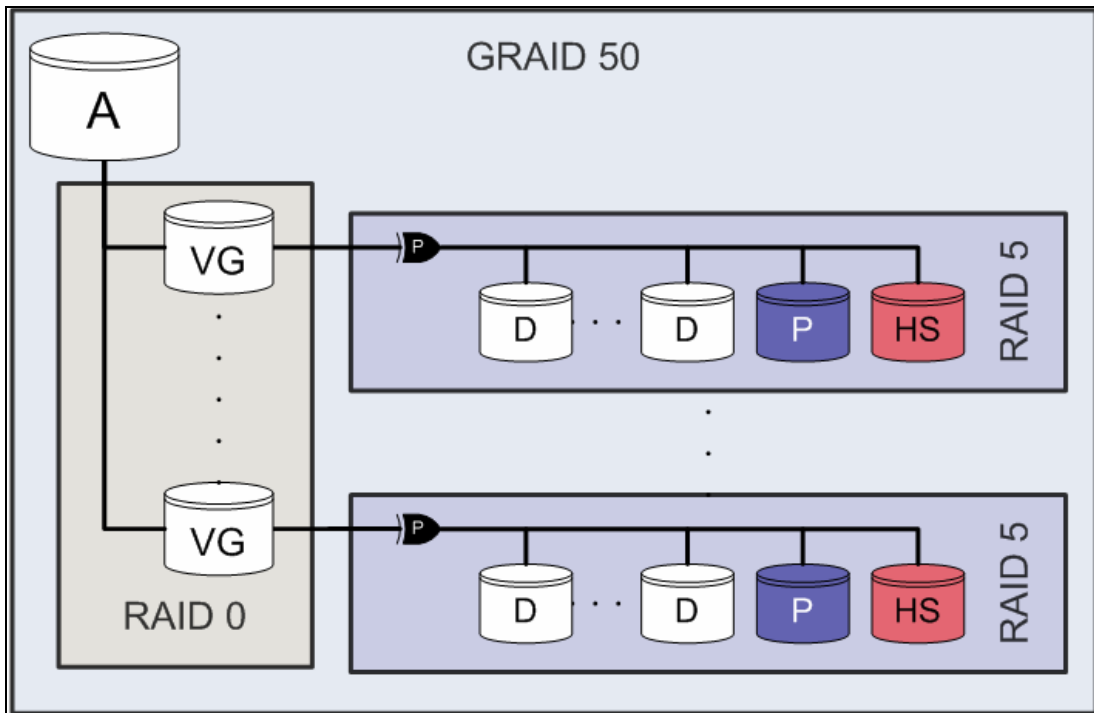


Figure 18: GRAID 50

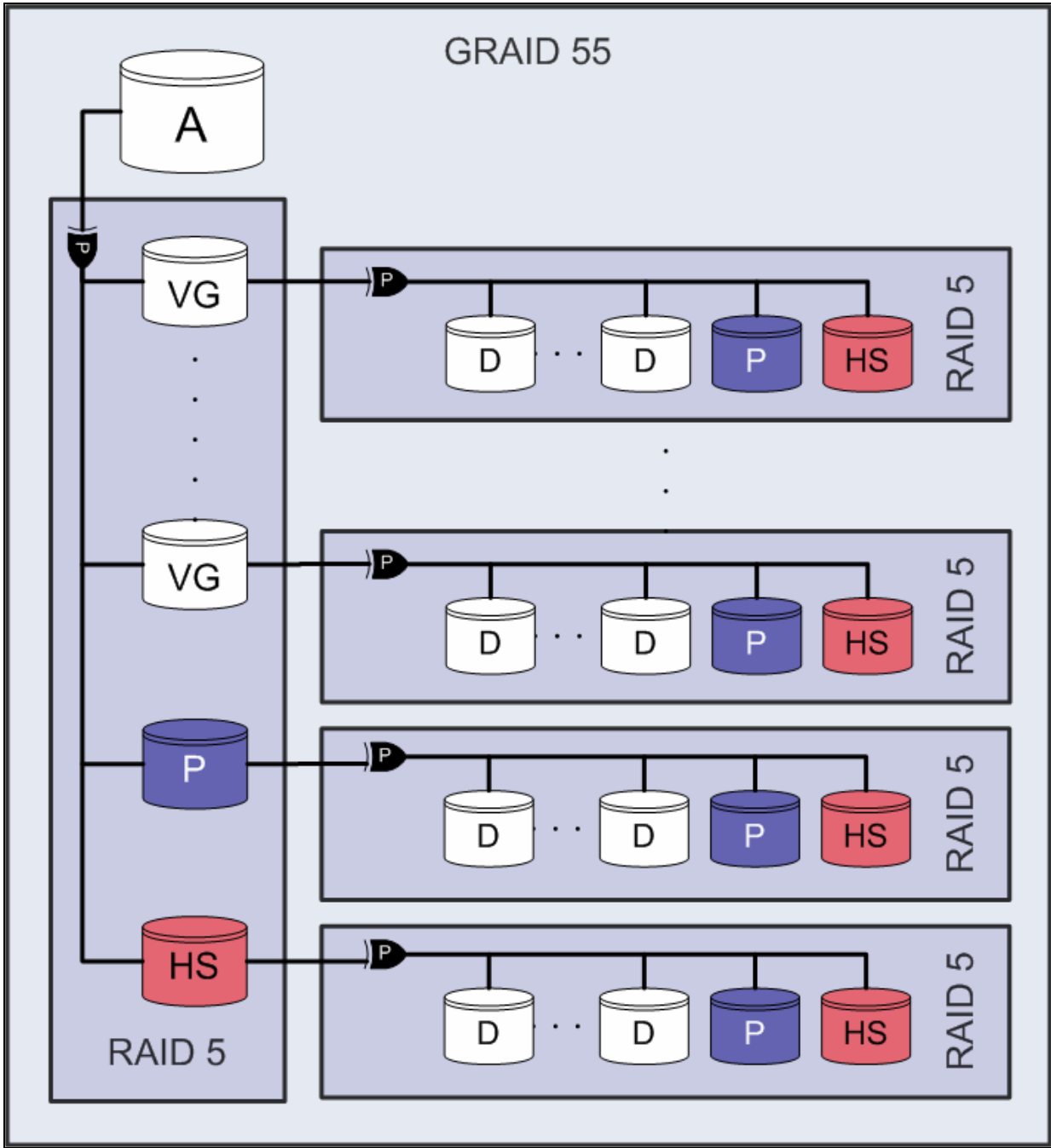


Figure 19: GRAID 5

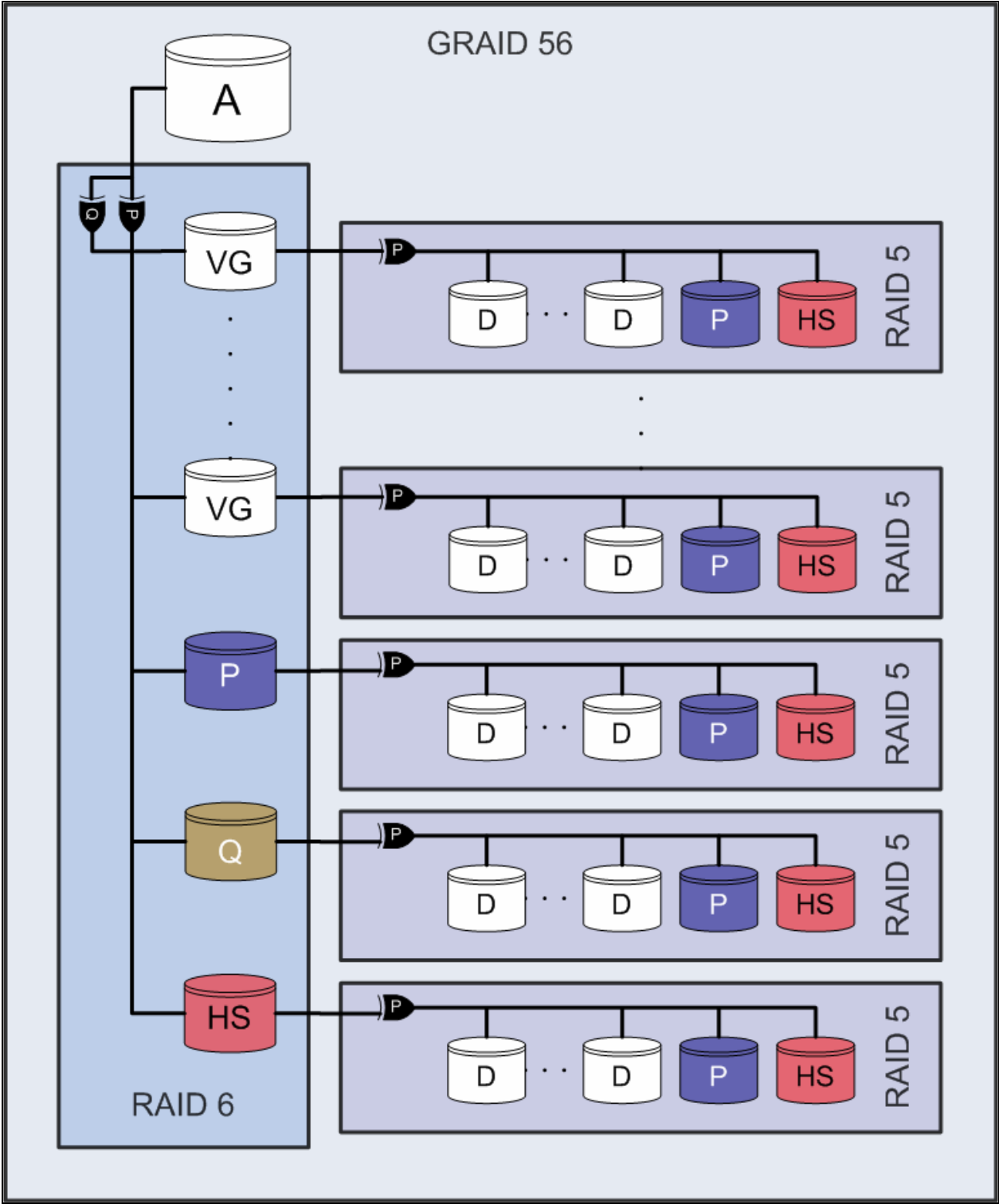


Figure 20: GRAID 56

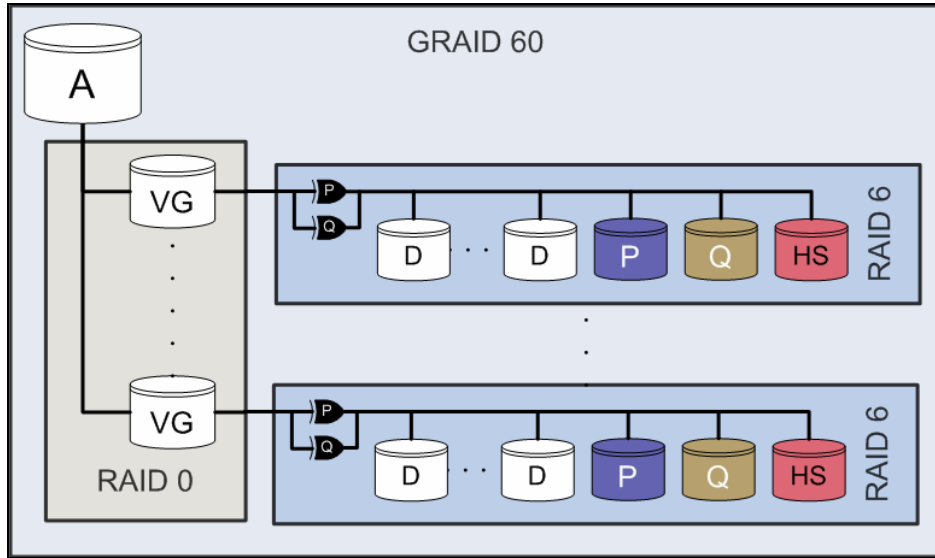


Figure 21: GRAID 60

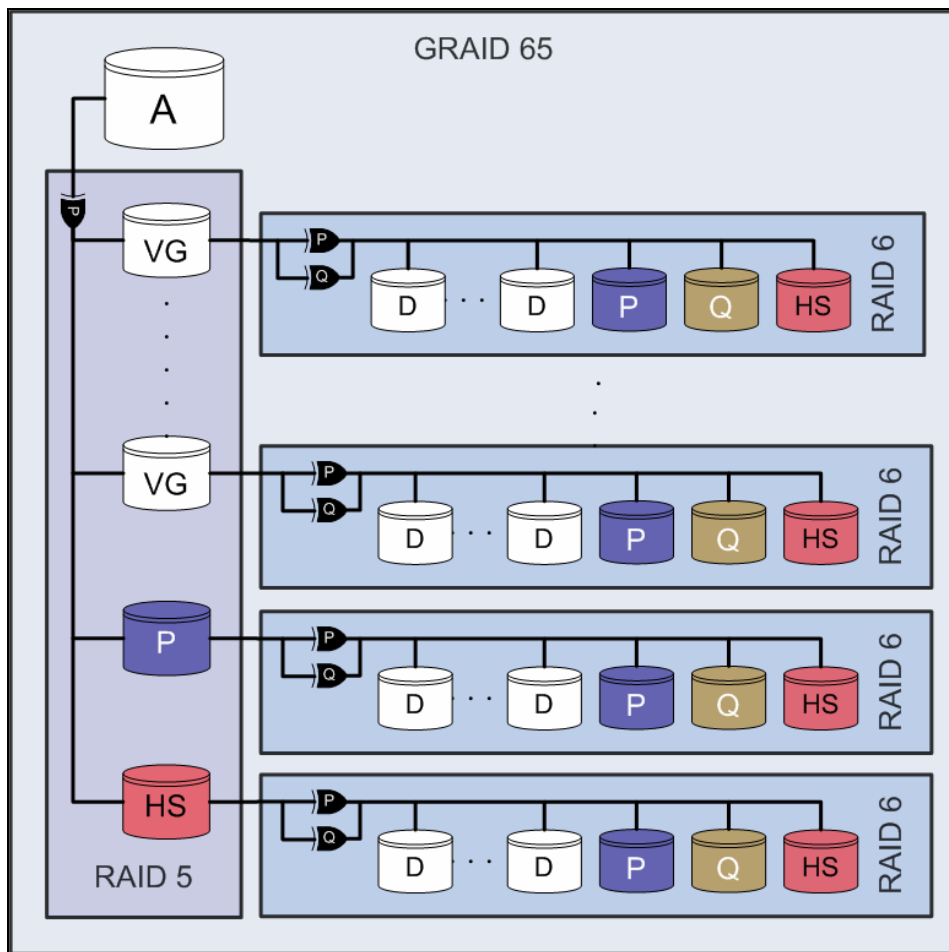


Figure 22: GRAID 65

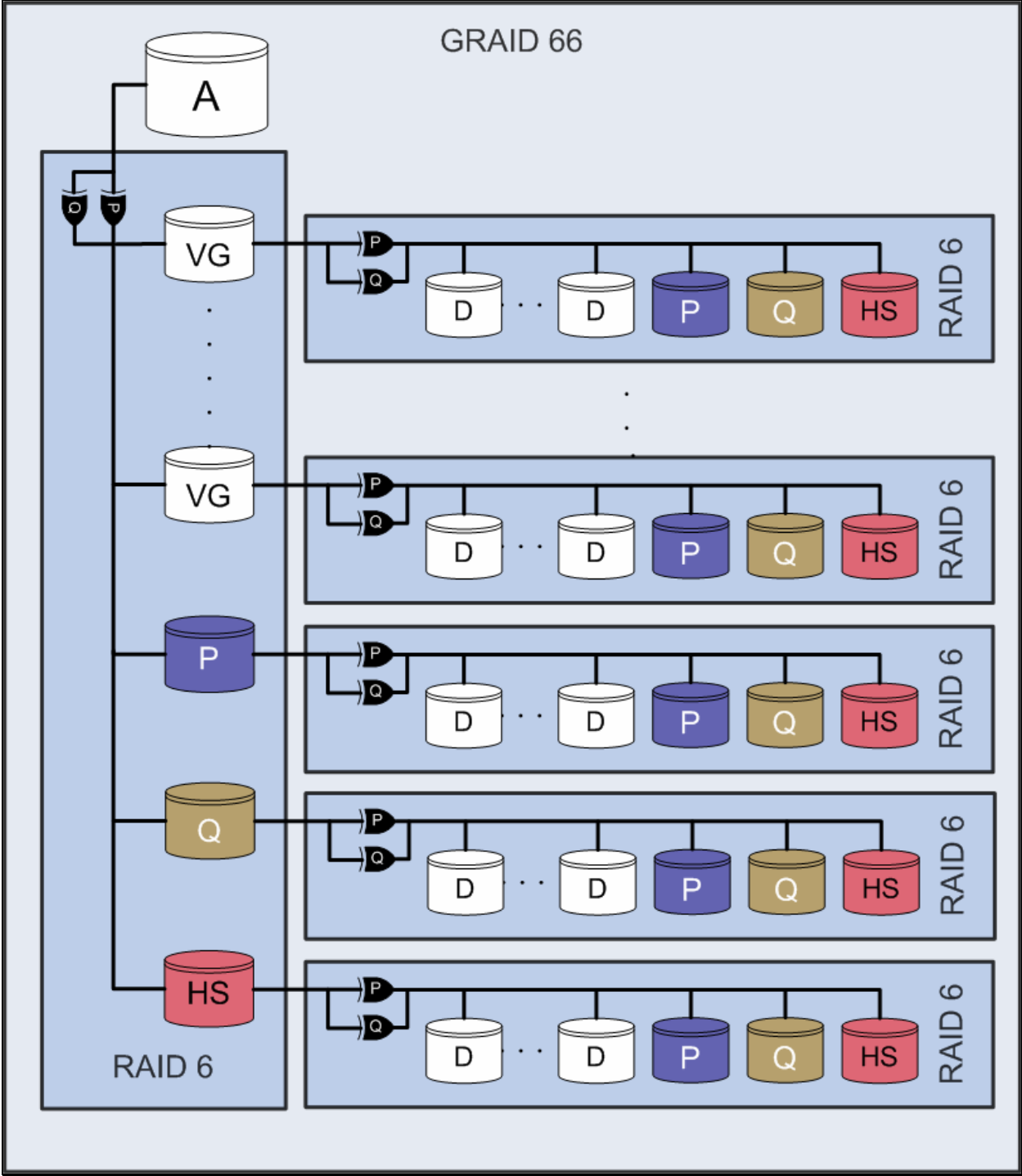


Figure 23: GRAID 66

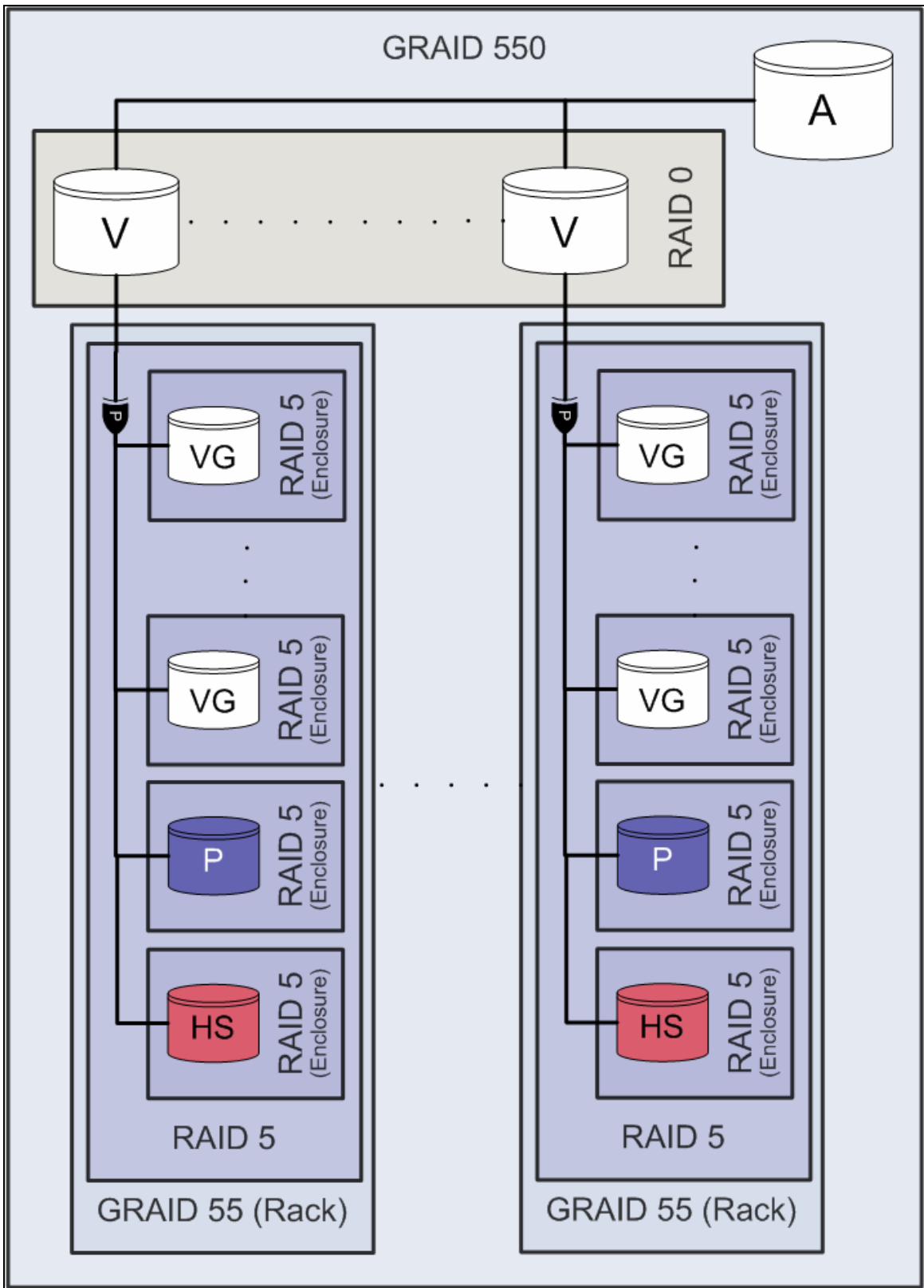


Figure 24: GRAID 550

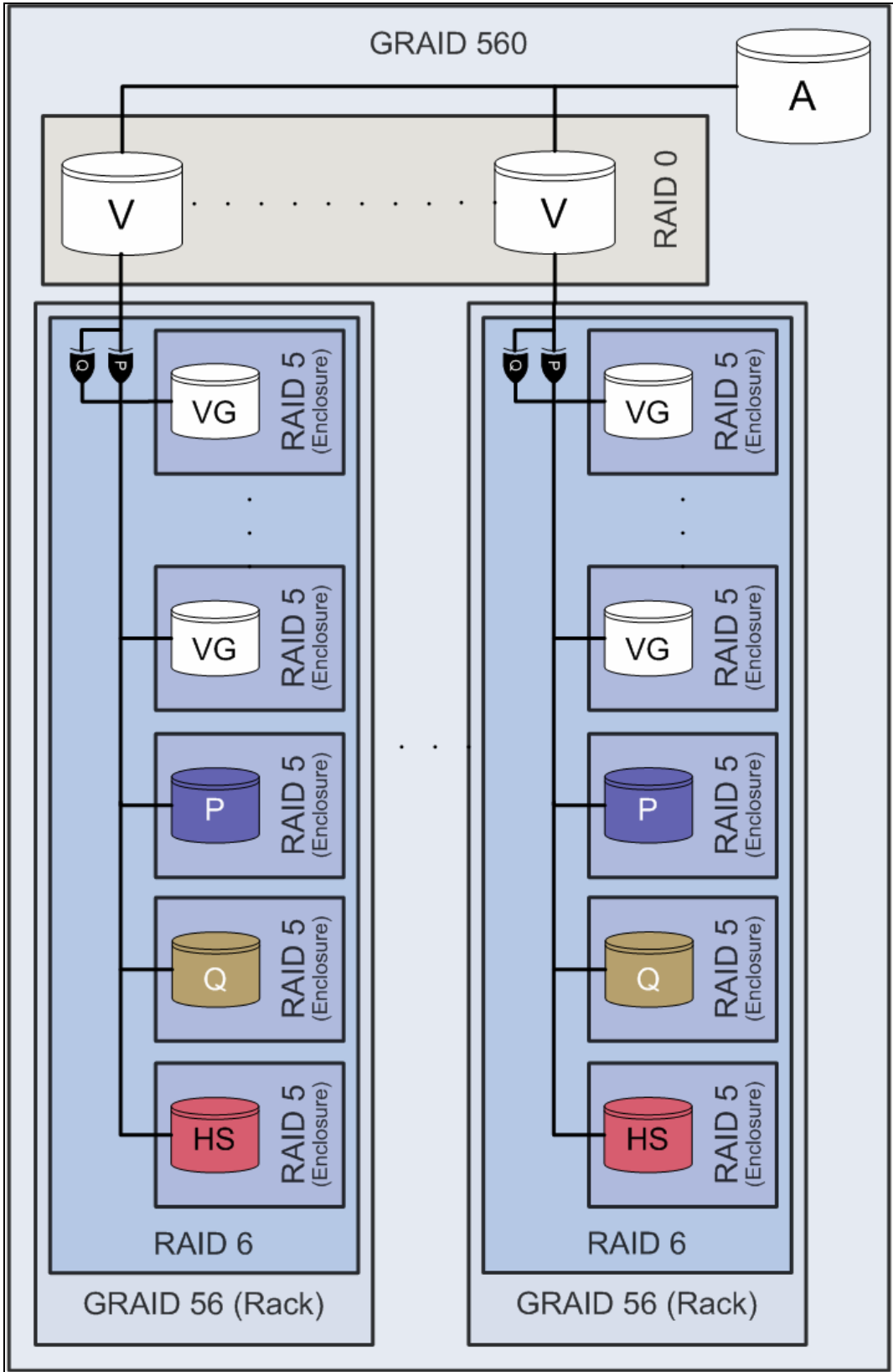


Figure 25: GRAID 560

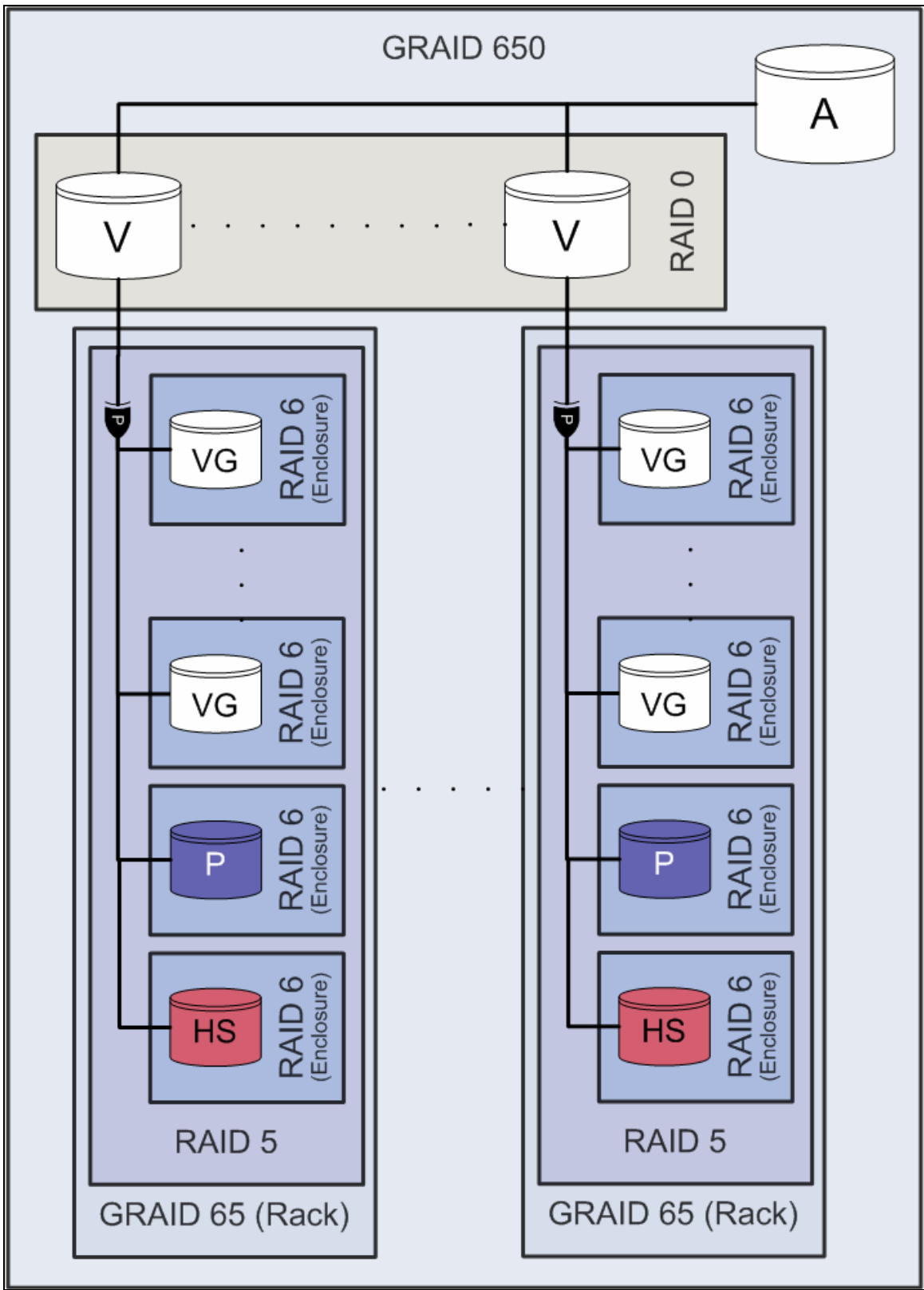


Figure 26: GRAID 650

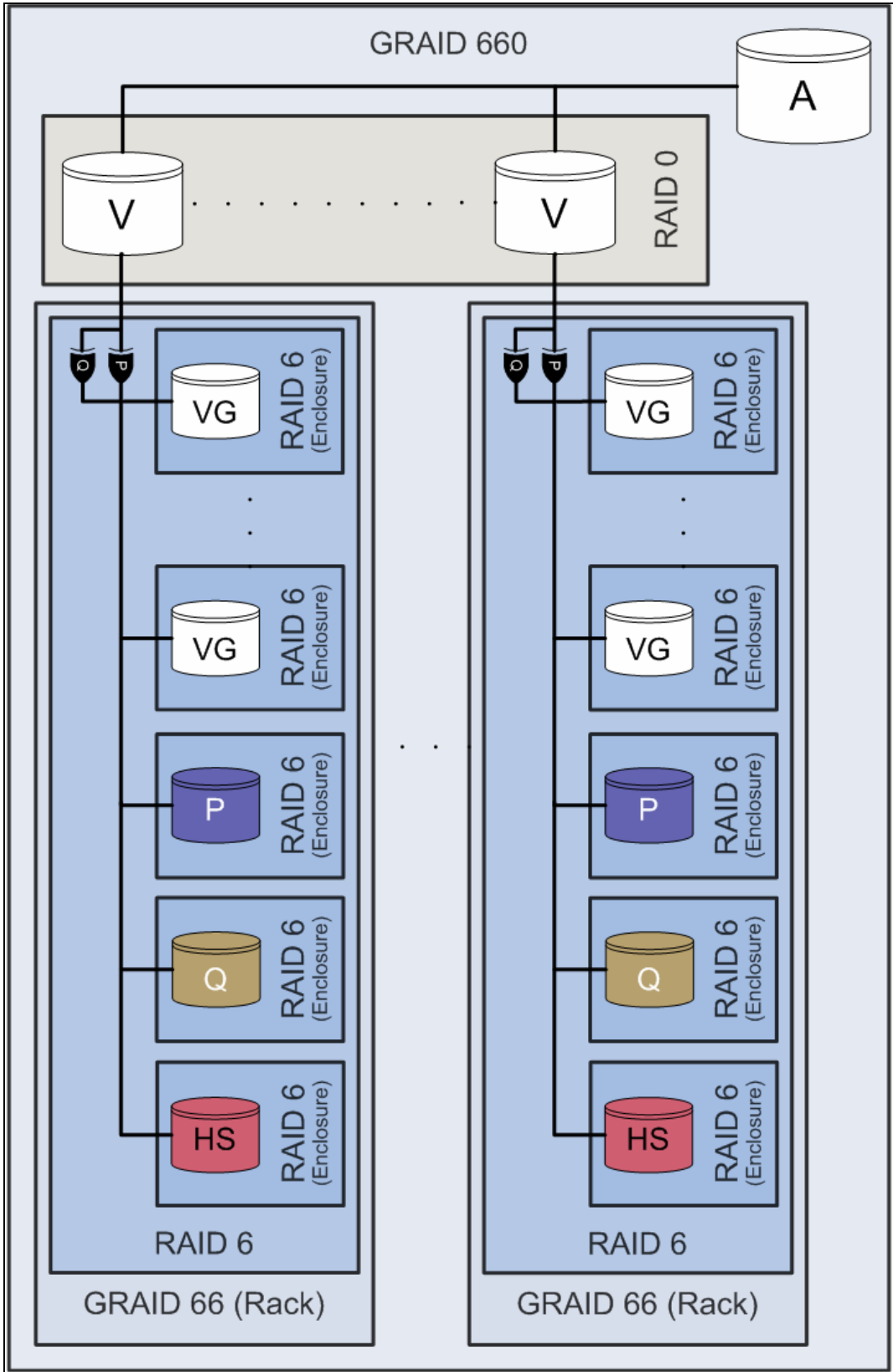


Figure 27: GRAID 660

3.5. RAID/GRAID Summary

Large-scale arrays require thousands to millions of disk drives to implement. This causes the probability of multiple disk failures to rapidly increase. Therefore, double parity schemes such as RAID 6 are ideal for small scale arrays. If cost is not an issue and the best overall read/write performance is desired, then a nested RAID level 10 will provide the highest level of fault-tolerance. However, for large-scale arrays ranging from 100 TiB to 100 EiB, a dual-level or tri-level GRAID approach is needed to provide sufficient reliability. A more in-depth reliability analysis can be found in Chapter 4. To summarize the RAID and GRAID levels outlined in this Chapter, Table 3 and Table 4 provide the advantages and disadvantages of each configuration.

Since the origins of RAID [PATT88], there have been some levels which had conceptual merit, but in the end did not prove to be useful. RAID levels 2 and 4 are rarely implemented due to their inefficient designs. Since all modern storage controllers can detect and identify failed disk drives, the technology employed in RAID 2 is superfluous. RAID 4's design flaw was the bottleneck associated with its single parity disk. The superior distributed parity scheme used with RAID 5 and 6 has become the preferred choice over level 4. Although RAID 3 still has its niche in audio/video and photography applications, this configuration option is not commonly found in storage controllers. The nested level 0+1 has also been pushed aside in favor of RAID 10. This was due to the inefficient use of mirroring at the top level, verses the preferred data striping. The above eliminations have reduced the storage controller's hardware cost and allowed the most frequently utilized RAID levels to be available for use (0, 1, 5, 6, 10, 50, and 60).

Nesting two RAID levels has been successfully implemented in small scale storage arrays. However, their translation to large-scale storage arrays requires further explanation. The proposed GRAID naming convention should be used when dealing with large-scale storage systems. By using multiple enclosures with multi-levels of redundancy, the overall reliability will greatly improve while still providing a high level of storage efficiency. In order to not incur excess parity overhead, enclosure sizes in GRAID configurations should be kept as large as possible. However, as larger capacity storage arrays are demanded, the number of disks will increase and necessitate more enclosures with smaller sizes. More on the tradeoffs of GRAID can be found in Chapter 5.

Table 3: RAID Levels at a Glance [GUPT02]

| RAID Level | Technique Employed | Advantages | Disadvantages |
|------------|---------------------------------|--|--|
| 0 | Striping | Highest data transfer rates and efficiency | No redundancy! |
| 1 | Mirroring | High-performance and fast write operations | High cost and overhead |
| 0+1 | Mirroring & Striping | Improved I/O rates with striping | Can result in high MTTR with mirroring at top level |
| 10 | Striping & Mirroring | Ideal for Database Applications, Improved I/O rates with striping | High cost and overhead |
| 2 | <i>Hamming-code parity</i> | <i>High redundancy and data availability</i> | <i>Extra disks used to locate failed disk. Impractical!</i> |
| 3 | <i>Byte-level Parity</i> | <i>Ideal for image and video applications, Easy to implement and high error recoverability</i> | <i>Low performance for small data reads</i> |
| 4 | <i>Block-level parity</i> | <i>High redundancy and better performance than Level 3</i> | <i>Read-modify-write overhead for small writes</i> |
| 5 | Distributed Parity | High-performance, less expensive than Level 4, partially eliminates write related bottlenecks | Low performance with small-sized write operations |
| 6 | Distributed Double parity (P+Q) | Higher redundancy, better availability than level 5, double disk failure protection | Requires another parity dimension creating additional overhead |

Table 4: Grouped RAID Levels at a Glance

| GRAID Level | Techniques Employed |
|-------------|--|
| 50 | Striping with Distributed Disk Parity |
| 55 | Distributed Group Parity with Distributed Disk Parity |
| 56 | Distributed Double-Group Parity (P+Q) with Distributed Disk Parity |
| 60 | Striping with Distributed Double Parity (P+Q) |
| 65 | Distributed Group Parity with Distributed Double Disk Parity |
| 66 | Distributed Double-Group Parity with Distributed Double-Disk Parity |
| 550 | Striping with Distributed Group Parity and Distributed Disk Parity |
| 560 | Striping with Distributed Double-Group Parity and Distributed Disk Parity |
| 650 | Striping with Distributed Group Parity and Distributed Double Disk Parity |
| 660 | Striping with Distributed Double-Group Parity and Distributed Double-Disk Parity |

CHAPTER 4

4. RAID METRICS

4.1. Introduction

The following chapter will focus on defining the various metrics needed to analyze and interpret the reliability of each GRAID configuration. First, the variables used with each redundancy configuration will be defined. Second, the storage capacity and efficiency equations are defined. With large-scale storage arrays, it is important to keep the cost-per-gigabyte ratio low by maintaining a high storage efficiency rating. Next, the various reliability metrics for each GRAID configuration are defined. An in depth analysis of these reliability metrics can be found in Chapter 5. Lastly, an overview of two case studies dealing with the questionable accuracy of manufacturer's MTBF ratings is provided.

4.2. Variables

When only considering the single-level RAID schemes, the number of variables required for computing the various metrics is quite simple. This is due to the singular dimension of these base RAID levels. However, with the dual- and tri-level GRAID configurations, additional variables are needed to define the various sub-components, *e.g.*, enclosures and racks. The variables represented in Table 5 were adapted from [PATT88, CHEN94] and updated to incorporate the sub-grouping of enclosures and racks for this thesis. Note that some variables are only valid for the dual- or tri-level redundancy configurations. The addition of *FullRack*, *RackSize*, and N_R were necessary for the tri-level GRAID calculations. The N variable represents the total number of disks (data, redundant, and hot-spare) in the storage array. In addition, the variables *DiskSize*, *RackSize*, and *ArraySize* are used for calculating the storage capacity, storage efficiency, and MTTR for each RAID/GRAID level. The storage capacity variables represent a binary magnitude.

Table 5: RAID and GRAID Variables

| Variable | Definition |
|----------------------|--|
| <i>Enclosure</i> | Total number of drives per disk enclosure $Enclosure = N_D + HS_D$ |
| <i>FullRack</i> | Total number of disk enclosures per rack enclosure (<i>tri-level variable</i>) $FullRack = N_{E,3} + HS_E$ |
| C_D | Number of check (or redundant) disks per disk enclosure |
| HS_D | Number of hot spare disks per disk enclosure |
| N_D | Number of reliability-dependent drives per disk enclosure $N_D = Enclosure - HS_D$ |
| C_E | Number of check (or redundant) enclosures per rack |
| HS_E | Number of hot spare disk enclosures per rack |
| $N_{E,1}$ | Number of reliability-dependent enclosures for single-level RAID $N_{E,1} = \mathbf{ceil}(ArraySize / EnclosureSize)$ |
| $N_{E,2}$ | Number of reliability-dependent enclosures for dual-level GRAID $N_{E,2} = \mathbf{ceil}(ArraySize / EnclosureSize) + C_E$ |
| $N_{E,3}$ | Number of reliability-dependent enclosures per rack for tri-level GRAID $N_{E,3} = FullRack - HS_E$ |
| N_R | Total number of racks in the array (<i>tri-level variable</i>) $N_R = 1$, for single- and dual-level GRAID |
| N | Total number of disks in the array $N = (N_D + HS_D) \times (N_{E,\langle 1,2,3 \rangle} + HS_E) \times N_R$ |
| <i>DiskSize</i> | Individual disk storage capacity $DiskSize = Mfgr. Size \times (10^9 / 2^{30})$ |
| <i>EnclosureSize</i> | Useable Storage Capacity per disk enclosure $EnclosureSize = DiskSize \times (N_D - C_D)$ |
| <i>RackSize</i> | Useable Storage Capacity per rack enclosure (<i>tri-level variable</i>) $RackSize = EnclosureSize \times (N_{E,3} - C_E)$ |
| <i>ArraySize</i> | Total storage capacity of array $ArraySize = [DiskSize \times (N_D - C_D)] \times (N_{E,\langle 1,2,3 \rangle} - C_E) \times N_R$ |

4.3. Storage Capacity and Storage Efficiency

When considering the various redundancy levels available, the total storage capacity desired will often play a critical role in the selection process. Storage arrays with higher storage efficiency allow for a lower price-per-gigabyte ratio. Storage efficiency is the ratio of useable data storage (excluding redundant and hot spare disks) over the total storage quantity (including redundant and hot spare disks). When dealing with large-scale arrays, even the smallest increase in efficiency will either result in more storage available or a less costly storage system.

4.3.1. Single-Level RAID

There exist two special RAID cases which do not exhibit any change in their storage efficiency. RAID 0, which only uses data striping, does not incorporate any form of redundancy. Therefore its storage efficiency is always 100%. Similarly with mirroring in RAID 1, the storage efficiency is always 50%. Data striping in RAID 0 utilizes no check disks or any hot spare disks. Therefore, the total number of disks in the array is simply, N_D . In the case of RAID 1 mirroring, for every data disk there is a redundant check disk. Therefore, the number of useable data disks is represented by $N_D / 2$. The formulas shown in Table 6 generate the storage capacity and storage efficiency for each of these special cases.

Table 6: Special Case Single-Level RAID Storage Capacity and Storage Efficiency

| RAID Level | Assumptions & Disk Requirements | Storage Capacity (<i>ArraySize</i>) | Storage Efficiency (%) |
|------------|---------------------------------|---------------------------------------|------------------------|
| 0 | $N_D \geq 2$ | $DiskSize \times N_D$ | 1 = 100% |
| 1 | $N_D \geq 2$ and N_D is even | $DiskSize \times (N_D / 2)$ | $\frac{1}{2} = 50\%$ |

For all parity-based RAID configurations, *e.g.*, RAID 3 through 6, the storage efficiency is no longer static as with RAID 0 and 1. As the number of disks in the array increases, its storage efficiency will also increase. The formulas shown in Table 7 produce the storage capacity and storage efficiency for each parity-based single-level RAID configuration. The equation for the

total number of disks in the array is $N = (N_D + HS_D)$, and the equation for calculating the array size is $ArraySize = [DiskSize \times (N_D - C_D)]$.

Table 7: Storage Capacity and Storage Efficiency of RAID

| RAID Level(s) | Assumptions & Disk Requirements | Storage Capacity (<i>ArraySize</i>) | Storage Efficiency (%) |
|--------------------|---------------------------------|---------------------------------------|------------------------------|
| 3, 4, and 5 | $C_D = 1$ and $N_D \geq 3$ | $DiskSize \times (N_D - 1)$ | $[(N_D - 1) / N] \times 100$ |
| 6 | $C_D = 2$ and $N_D \geq 4$ | $DiskSize \times (N_D - 2)$ | $[(N_D - 2) / N] \times 100$ |

4.3.2. Dual-Level RAID/GRAID

Similarly with the previous mirroring special case, RAID levels 0+1 and 10 do not exhibit any variation in their storage efficiency, *i.e.*, always 50%. The number of useable data disks is represented by $N_D / 2$. The formulas shown in Table 8 generate the storage capacity and storage efficiency for each of these special case dual-level RAID configurations.

Table 8: Special Case Dual-Level Storage Capacity and Storage Efficiency

| RAID Level(s) | Assumptions & Disk Requirements | Storage Capacity (<i>ArraySize</i>) | Storage Efficiency (%) |
|----------------|---------------------------------|---------------------------------------|------------------------|
| 0+1, 10 | $N_D \geq 2$ and N_D is even | $DiskSize \times (N_D / 2)$ | $\frac{1}{2} = 50\%$ |

The formulas shown in Table 9 produce the storage efficiency and storage capacity for each dual-level GRAID configuration defined in Chapter 2. Note that the base level RAID 5 and 6 assumptions and requirements defined in Table 7 are still relevant. The general equation for the total number of disks in the array is $N = (N_D + HS_D) \times (N_{E,2} + HS_E)$, and the general equation for calculating the array size is $[DiskSize \times (N_D - C_D)] \times (N_{E,2} - C_E)$.

Table 9: Dual-Level GRAID Storage Capacity and Storage Efficiency

| GRAID Level | Assumptions & Requirements | Storage Capacity (<i>ArraySize</i>) | Storage Efficiency (%) |
|-------------|---|--|---|
| 50 | $C_D = 1, C_E = 0,$ $N_D \geq 3, N_{E,2} \geq 2$ | $[DiskSize \times (N_D - 1)] \times (N_{E,2})$ | $\frac{N_D - 1}{N_D + HS_D} \times 100$ |
| 55 | $C_D = 1, C_E = 1,$ $N_D \geq 3, N_{E,2} \geq 3$ | $[DiskSize \times (N_D - 1)] \times (N_{E,2} - 1)$ | $\frac{N_D - 1}{N_D + HS_D} \times \frac{N_{E,2} - 1}{N_{E,2} + HS_E} \times 100$ |
| 56 | $C_D = 1, C_E = 2,$ $N_D \geq 3, N_{E,2} \geq 4$ | $[DiskSize \times (N_D - 1)] \times (N_{E,2} - 2)$ | $\frac{N_D - 1}{N_D + HS_D} \times \frac{N_{E,2} - 2}{N_{E,2} + HS_E} \times 100$ |
| 60 | $C_D = 2, C_E = 0,$ $N_D \geq 4, N_{E,2} \geq 2$ | $[DiskSize \times (N_D - 2)] \times (N_{E,2})$ | $\frac{N_D - 2}{N_D + HS_D} \times 100$ |
| 65 | $C_D = 2, C_E = 1,$ $N_D \geq 4, N_{E,2} \geq 3$ | $[DiskSize \times (N_D - 2)] \times (N_{E,2} - 1)$ | $\frac{N_D - 2}{N_D + HS_D} \times \frac{N_{E,2} - 1}{N_{E,2} + HS_E} \times 100$ |
| 66 | $C_D = 2, C_E = 2,$ $N_D \geq 4, N_{E,2} \geq 4$ | $[DiskSize \times (N_D - 2)] \times (N_{E,2} - 2)$ | $\frac{N_D - 2}{N_D + HS_D} \times \frac{N_{E,2} - 2}{N_{E,2} + HS_E} \times 100$ |

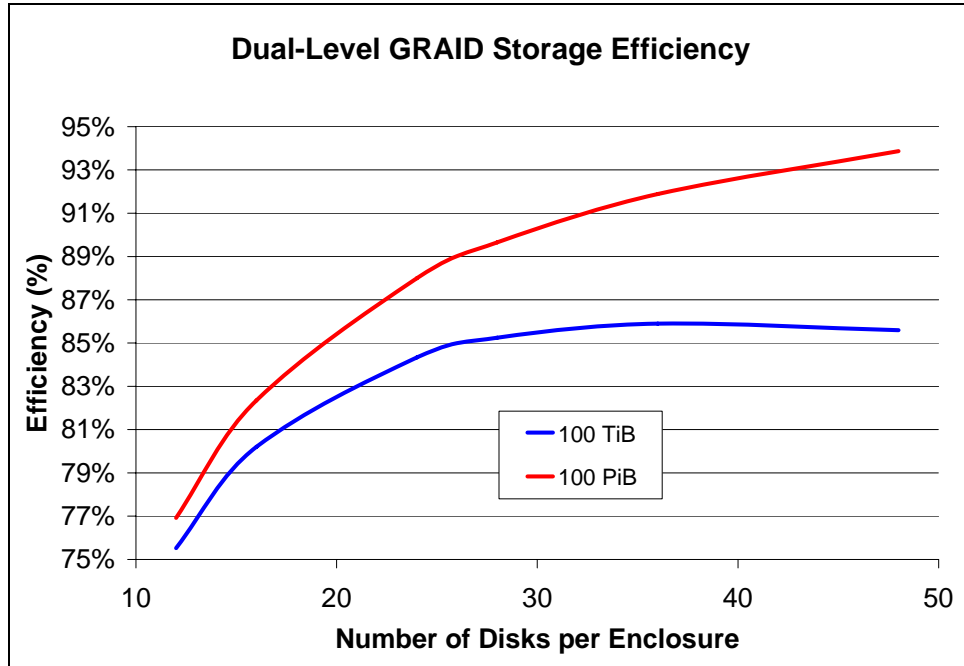


Figure 28: Dual-Level GRAID Storage Efficiency

The results in Figure 28 illustrate how more disks per enclosure will improve the storage efficiency for dual-level RAID configurations. As the ratio of useable data disks to the total number of disks per enclosure (data, parity, and hot spare) increases, the overall storage efficiency improves. For all of the RAID levels outlined in Table 9, the variance in efficiency between each is minimal. This is due to the large number of disk enclosures required for terabyte- to petabyte-sized storage arrays. Therefore, only the storage efficiency of RAID 60 is shown in the results.

Table 10: Tri-Level RAID Storage Capacity and Storage Efficiency

| RAID Level | Assumptions & Requirements | Storage Capacity (ArraySize) | Storage Efficiency (%) |
|-------------------|--|--|---|
| 550 | $C_D = 1, C_E = 1,$ $N_D \geq 3, N_{E,3} \geq 3,$ $N_R \geq 2$ | $[DiskSize \times (N_D - 1)]$ $\times (N_{E,3} - 1) \times N_R$ | $\frac{N_D - 1}{N_D + HS_D} \times \frac{N_{E,3} - 1}{N_{E,3} + HS_E} \times 100$ |
| 560 | $C_D = 1, C_E = 2,$ $N_D \geq 3, N_{E,3} \geq 4,$ $N_R \geq 2$ | $[DiskSize \times (N_D - 1)]$ $\times (N_{E,3} - 2) \times N_R$ | $\frac{N_D - 1}{N_D + HS_D} \times \frac{N_{E,3} - 2}{N_{E,3} + HS_E} \times 100$ |
| 650 | $C_D = 2, C_E = 1,$ $N_D \geq 4, N_{E,3} \geq 3,$ $N_R \geq 2$ | $[DiskSize \times (N_D - 2)]$ $\times (N_{E,3} - 1) \times N_R$ | $\frac{N_D - 2}{N_D + HS_D} \times \frac{N_{E,3} - 1}{N_{E,3} + HS_E} \times 100$ |
| 660 | $C_D = 2, C_E = 2,$ $N_D \geq 4, N_{E,3} \geq 4,$ $N_R \geq 2$ | $[DiskSize \times (N_D - 2)]$ $\times (N_{E,3} - 2) \times N_R$ | $\frac{N_D - 2}{N_D + HS_D} \times \frac{N_{E,3} - 2}{N_{E,3} + HS_E} \times 100$ |

4.3.3. Tri-Level RAID

The highest number of redundancy levels considered for large-scale storage arrays is three. Although the use of dual-level redundancy would seem to be sufficient for the reliable storage of data, there are other factors driving the need for three levels. The third (top) level used in the following four cases does not add any redundant disks or groups to the array. Instead, these schemes aim to keep the redundancy high by keeping the groups small. This subsequently keeps the MTTR low by requiring fewer disks during a rebuild. The intended implementation of these tri-level schemes was designed with rack enclosures in mind. First, an enclosure size is chosen so that the number of disks and enclosures housed in a rack is optimized (this will be covered in

more detail in Chapter 5). Each disk enclosure forms the first level of redundancy, *e.g.*, highly reliable RAID 5 or 6 arrays. Next, grouping all the disk enclosures in a single rack forms the second level of redundancy, *e.g.*, highly reliable GRAID 55, 56, 65, or 66 arrays. Lastly, all of these rack enclosures are tied together using RAID 0 striping at the top level. The end result is an efficient configuration with very good reliability. The formulas shown in Table 10 produce the storage efficiency and storage capacity for each tri-level GRAID configuration. The general equation for the total number of disks in the array is, $N = (N_D + HS_D) \times (N_{E,3} + HS_E) \times N_R$, and the general equation for calculating the array size is, $ArraySize = [DiskSize \times (N_D - C_D)] \times (N_{E,3} - C_E) \times N_R$.

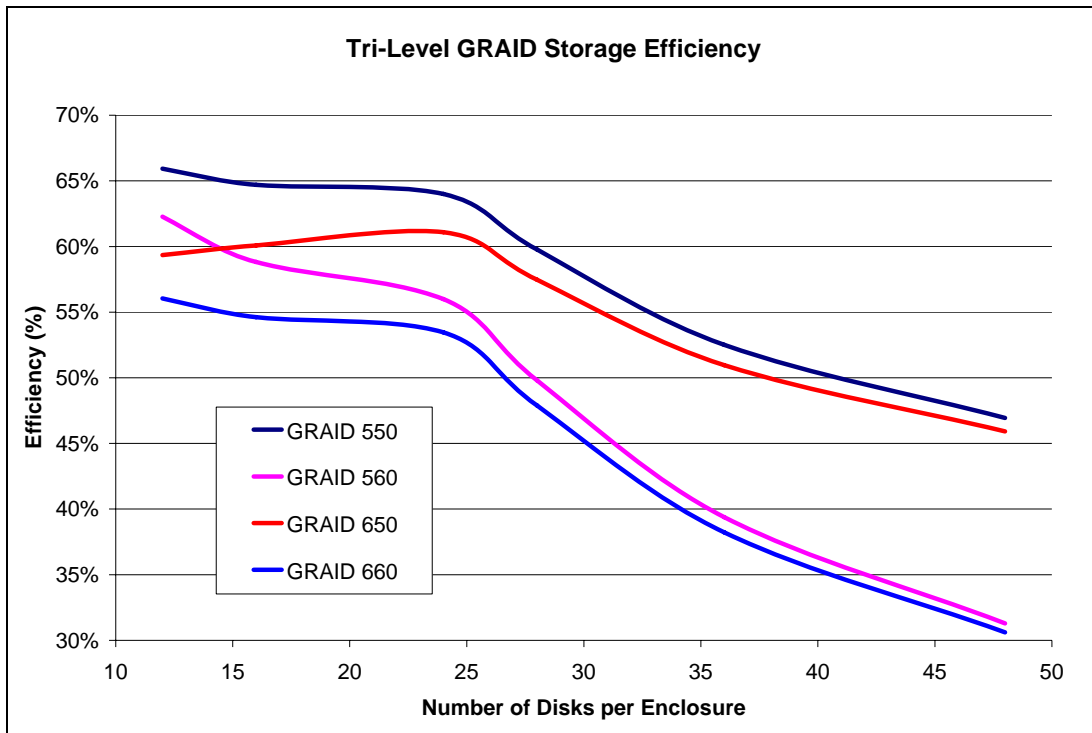


Figure 29: Tri-Level GRAID Storage Efficiency

The results in Figure 29 represent the storage efficiency of tri-level GRAID configurations. Since the proposed design of tri-level GRAID is confined to single rack enclosures, smaller disk enclosure sizes result in better storage efficiency. As disk enclosure sizes increase, the number of enclosures per rack must decrease proportionally. Therefore, it is more efficient to use smaller disk enclosures with tri-level GRAID.

4.4. Availability vs. Reliability

“Reliability - The ability of a system or component to perform its required functions under stated conditions for a specified period of time.” [IEEE07]

By incorporating additional components in storage arrays, the overall system reliability decreases. In an ideal storage array the overall reliability of the system should be high enough that it never causes any data loss. However, due to “Mother Nature”, economic and other constraints, this is hardly ever achievable in large-scale storage arrays. This is primarily due to their sheer size. Improvements in reliability are achieved by utilizing redundant disks, hot swap disks, and optimized parity schemes. Additionally, by quickly regenerating data on a failed disk drive from the other disks in the array (MTTR), this will improve both reliability and availability. When an array is in the degraded state due to a disk failure, this is the window of time which the condition of the system is most critical. Commonly encountered in large-scale arrays, during this rebuild process there will be a UBE encountered. This could potentially cause the system to lose data and fail unless a mirrored design or secondary parity scheme has been implemented.

“Availability - The degree to which a system or component is operational and accessible when required for use.” [IEEE07]

How does availability differ from reliability? By not allowing any data loss to occur, while still providing access to the data, the availability of the system is theoretically 100%. With RAID systems utilizing mirrored, single-parity, and double-parity redundancy, this allows for multiple failures to occur, while still making the data available. To classify systems which achieve near constant uptime, a categorization of nines is used. For example, the industry defines a highly available (HA) system as five nines (99.999%). This means that the system will only suffer downtime for less than 5.256 minutes per year. An enumeration of the various availability ratings are shown in Table 11. To calculate the availability of RAID systems, the general equation (1) is used. However, the MTTR used for availability is not necessarily the same MTTR used with RAID reliability calculations. This MTTR is the system repair time required

after a storage array has exhausted all redundancy methods and fails. This is usually the time necessary to replace multiple failed disk drives or storage enclosures and then restore from nearline storage or tape backup. For HA systems, this MTTR is commonly the failover time needed for a secondary storage site to assume an active status. The key thing to note is that a single system cannot provide high availability. In order to realistically achieve this, multiple storage systems must be integrated. The use of multiple storage resources allows for failover to standby systems or load balancing between all storage systems.

$$A = \frac{MTBF}{MTBF + MTTR} \quad (1)$$

Table 11: System Availability

| Alias | Uptime (%) | Maximum Downtime |
|-------------------|----------------|------------------------|
| One nine | 90% | 876 hours (36.5 days) |
| Two nines | 99% | 87.6 hours (3.65 days) |
| Three nines | 99.9% | 8.76 hours |
| Four nines | 99.99% | 52.56 minutes |
| Five nines | 99.999% | 5.256 minutes |
| Six nines | 99.9999% | 31.536 seconds |

4.5. Reliability Metrics

In order to better compare the different levels of redundancy in RAID systems, there must be common metrics. For data storage systems, the basic reliability metrics are: Mean Time Before Failure (MTBF), Mean Time To Repair (MTTR), and Mean Time To Data Loss (MTTDL). If these reliability metrics are bounded by the statistical assumption that all disk failures are independent of one another, reliability equations are simplified [CHEN94]. However, correlated disk failures also need to be taken into consideration. Correlated disk failures take into account the higher probability that a second or third disk drive will subsequently fail after the first. These failures are commonly based on environmental, (*e.g.*, earthquakes, power dips/surges), and manufacturing factors, (*e.g.*, early/late failures in disk drives) [CHEN94]. Also meriting

importance is the probability of encountering a UBE. As stated before, during the rebuild process of large-scale storage systems, there is a high likelihood that another disk will fail due to a bit error. In the subsequent sections, all of these factors will be taken into account to provide a better estimation of overall system reliability.

4.5.1. MTBF

Most hard disk drives have an MTBF rating which is assigned by the manufacturer. The mean time between failures is used to convey the mean (average) life of a disk drive in hours. This value is based on the frequency of failures. The three primary classes of disk drives are desktop, business, and enterprise. Each class represents an improvement in reliability with desktop drives having the lowest reliability and enterprise class drives achieving the highest. Desktop class PATA disk drives commonly have a MTBF of 1 million hours (MTBF=1e6). Business class SATA and SAS disk drives usually have an MTBF of 1.2 million hours (MTBF=1.2e6) and 1.4 million hours (MTBF=1.4e6), respectively. Enterprise class SCSI/FC drives will ideally have an MTBF rating of 1.6 million hours (MTBF=1.6e6). Essentially, drives with higher MTBF ratings are more reliable [LIOT03]. The calculation of a system MTBF with multiple unique devices is shown in equation (2). Alternatively, if each device has the same MTBF, *e.g.*, a storage array with thousands of disk drives, then the equation simplifies to the following (3). However, case studies have shown that these MTBF ratings provided by manufacturers can be misleading and are not good estimations of reliability. More on these discrepancies is discussed in section 4.6.

$$MTBF_{System} = \frac{1}{\left(\frac{1}{MTBF_1} + \frac{1}{MTBF_2} + \dots + \frac{1}{MTBF_N} \right)} \quad (2)$$

$$MTBF_{System} = \frac{MTBF_{Component}}{N} \quad (3)$$

Another common representation of a disk's reliability is its Annualized Failure Rate (AFR). This rating is simply another representation of MTBF based on the average disk usage per year. Therefore, if a disk was running constantly for an entire year, which is assumed for large-scale storage systems, then the total number of power-on hours (POH) would be 8760

($365\text{ days} \times 24\text{ hours/day} = 8760\text{ hours}$). First, divide the MTBF by the number of annual POH. Then, taking the reciprocal of this produces the AFR. Equation (4) shows an example calculation of the AFR for a SATA disk drive. Hence, on average 0.73% of a disk drive population is expected to fail annually.

$$AFR = \frac{MTBF}{POH} = \left(\frac{1,200,000}{8760} \right)^{-1} \times 100\% = 0.73\% \quad (4)$$

4.5.2. MTTR

To convey the total time in which a storage array is in the degraded state, MTTR is often utilized. The MTTR is the total time in hours required for a system to return to its optimal state. The basic representation of MTTR is shown in equation (5). The first time component of MTTR is the time (H_T) required diagnosing the problem and replacing the failed device. This can vary depending on how quickly a technician can respond to an incident or if a hot spare disk drive is already waiting in an array to take over for a failed device. This response time (ranging between 4 and 24 hours) will depend on the client's service contract with a vendor. In the hot spare case, the H_T time component is assumed to be zero. In the event of a failed drive, the enclosure's hot spare disk will immediately take its place, and initiate the rebuild process. The second time component (R_T) is the time required for a complete data rebuild of a disk drive. This rebuild time must take into account multiple factors. First, the storage controller has a maximum sustainable media rate (M) established by the disk/interface technology used. For example, SATA technology has a media rate of 150MB/sec ($M=150e6$), where SAS operates at 300MB/sec ($M=300e6$). However, all of this bandwidth is not generally made available to the rebuild process. The rebuild priority (P) only allocates a certain percentage of the controller's resources for the rebuild. MTTR also depends on the redundancy scheme used. A mirrored rebuild is not the same as the parity rebuild. The rebuild time for parity-based RAID levels also depends on reading all data disks in the group, computing the parity, and then writing the new value. HP has calculated this to be approximately an N:1 inefficiency ratio for large storage systems [HP05b]. Therefore, depending on the number of disks in a group, parity-based rebuild times are significantly higher than a mirrored disk-to-disk copy. The MTTR used for evaluating RAID

levels 1 and 10 disk mirroring is illustrated in equation (7). For parity-based RAID levels, the MTTR is shown in equation (8).

Mean Time To Repair:
$$MTTR = H_T + R_T \tag{5}$$

Previous calculations [TREA03] made the assumption that MTTR was a fixed time for any number of disk drives. This estimated MTTR was calculated using the total capacity of a single disk (*DiskSize*) divided by the quantity, media rate (*M*) divided by 3 (the number of disk accesses required for one rebuild cycle: read data, compute parity, and then write data) [TREA03]. However, as the number of disk drives in large-scale storage systems increase, the MTTR will also rise. Therefore, a better inefficiency ratio for an enclosure of disks (less than 50) would be to use an exponential factor, instead of the static value of 3. The empirical ratio used is based on data from experiment observations in Chapter 5. This empirical ratio was determined to be $I = e^{(0.08 \cdot [N_D - C_D])}$ in two different experiments. When rebuilding an enclosure, an empirical inefficiency ratio of N:1 should be used for the enclosure MTTR of large-scale arrays. This empirical inefficiency ratio is based upon the experimental observations in [HP05b].

Incorporating the empirical inefficiency ratio (*I*) into parity-based MTTR equations provides a more realistic rebuild time. This also proves to be a critical component that greatly affects the overall reliability. The empirical inefficiency ratio is defined by the number of data disks needed to be read from during a rebuild. This excludes any parity or check disks.

Rebuild Time (R_T):
$$R_T = \frac{DiskSize}{R_R}, \quad R_R = \frac{M}{I} \times P, \quad I = e^{(0.08 \cdot [N_D - C_D])}$$

Generic MTTR (with empirical Inefficiency Ratio *I*): (6)

$$MTTR = H_T + \frac{DiskSize}{\left(\frac{M}{I} \times P\right)}$$

Currently, the time required for a mirrored disk to be rebuilt is negligible in comparison to a disk's MTBF and the other factors involved with determining the reliability of mirrored systems. Because of this, most MTTDL equations for mirrored systems omit this variable. To remain consistent, this variable should be incorporated into reliability metrics so that it is not forgotten. As disk capacities increase faster than their bus speeds do, this will cause rebuild times to gradually increase. If this trend continues, rebuilt times for mirrored disks can no longer be neglected in reliability calculations.

Mean Time To Repair for Mirrored systems (in hours): (7)

$$MTTR_{mirror} = H_T + R_T = H_T + \frac{DiskSize}{M \times P \times 3600sec}$$

Mean Time To Repair a Disk for Parity systems (in hours): (8)

$$MTTR_{Disk} = H_T + R_T = H_T + \frac{DiskSize}{\frac{M}{e^{(0.08[N_D - C_D])}} \times P \times 3600sec}$$

For multi-level RAIDs, an equivalent MTTR (9) is used for the rebuilding of an entire failed enclosure. However, for these cases, the *DiskSize* is replaced by *EnclosureSize*. The media rate will also be dependent on the interconnection between disk enclosures rather than the disk's end interface, *e.g.*, SATA, SCSI. The most commonly used linkage is Fibre Channel (FC) due to its high bandwidth. Fibre Channel can support speeds from 1 Gbps up to 6 Gbps.

Mean Time To Repair a Enclosure for Parity systems (in hours): (9)

$$MTTR_{Enclosure} = H_T + R_T = H_T + \frac{EnclosureSize}{\frac{M}{N_{E,\{1,2,3\}} - C_E} \times P \times 3600sec}$$

4.5.3. MTTDL

MTTDL utilizes the previous two metrics (MTBF and MTTR) in order to determine the average time in which the loss of data will occur in a given RAID configuration. When

considering large-scale arrays, the MTDDL is particularly useful since only the reliability of all disk drives must be taken into consideration. The reliability of other system components (such as controllers, cables, enclosures, servers, and software) can be disregarded for the purposes of this thesis. The following sections represent the MTDDL values for non-redundant, mirroring, single-parity, and double-parity schemes. These equations are based on standard failure analysis models and derived from the following sources [PATT88, CHEN94, TREA03].

Calculating the MTDDL of RAID systems is easier to understand if the worse case scenario is considered, *i.e.*, what is the shortest failure path needed to produce data loss. For example, in a 6-disk RAID 1 array there are essentially 3 disks which hold redundant mirrored data. Therefore, the best case scenario would be if all three redundant disks failed first and then any fourth disk failure would result in data loss, *i.e.*, four disk failures. However, with MTDDL, the shortest path needed to result in data loss would be if one disk failed and then its mirrored copy subsequently failed. Therefore, in a worse case scenario, only two disk failures are required to cause data loss in any mirrored array. An example of these two scenarios is illustrated in Figure 30. In the best case scenario (left), all redundant disks must fail first, followed by any of the others before any data are lost. In the worst case scenario (right), if two disks in the same mirrored pair fail then data loss occurs.

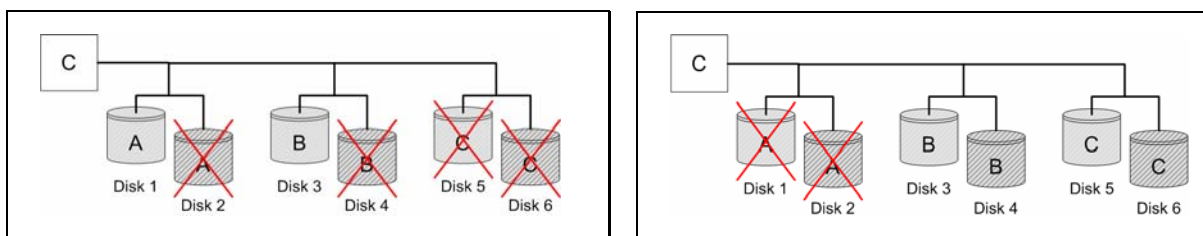


Figure 30: Best Case (left) and Worst Case (right) Scenarios Before RAID 1 Data Loss

Additional background information and the complete derivation of MTDDL for parity based RAID systems can be found in [PATT88]. However, for the purposes of this thesis, a more user friendly way of understanding MTDDL is to use the above thought process. MTDDL is essentially the compound probability of each failure in a worse case scenario [WINC06]. These multiple scenarios are combined through multiplication to form the overall MTDDL. Take for example, the series of events needed to result in data loss for a RAID 6 array. For each disk

failure the total number of drives factored into the next probability scenario decreases. This is beneficial in keeping the MTTDL high. As the numerator's MTBF value is squared and then cubed, the denominator's number of disks for each scenario is decreasing.

MTTDL (RAID 6) Compound Probability:

$$\begin{aligned}
 \text{MTTDL}_{DF} = & \frac{\text{MTBF}_{Disk}}{N_D} * \left(\frac{1}{\left(\frac{\text{MTTR}_{Disk}}{\text{MTBF}_{Disk} / (N_D - 1)} \right)} \right) * \left(\frac{1}{\left(\frac{\text{MTTR}_{Disk}}{\text{MTBF}_{Disk} / (N_D - 2)} \right)} \right) \\
 & \text{Probability of 1 in } N_D \text{ disks failing} \quad \text{Probability of 1 in } N_D - 1 \text{ disks failing including the rebuild time.} \quad \text{Probability of 1 in } N_D - 2 \text{ disks failing including the rebuild time.}
 \end{aligned}$$

A key benefit to calculating MTTDL is for comparing different RAID levels. Table 12 better illustrates the usefulness of MTTDL in RAID comparisons. Here a RAID 5 and RAID 6 system are compared based on the size of the array. The improved reliability of RAID 6 is clearly evident when looking at the reliability increase ratio.

Table 12: MTTDL Comparison [WINC06]

| Array Capacity | RAID 5 MTTDL (months) | RAID 6 MTTDL (months) | Reliability Increase Ratio |
|----------------|-----------------------|-----------------------|----------------------------|
| 1 TB | 100 | 3,000,000 | 30,000x |
| 2 TB | 50 | 500,000 | 10,000x |
| 5 TB | 10 | 20,000 | 2,000x |
| 10 TB | 7 | 7,000 | 1,000x |
| 20 TB | 2 | 1,000 | 500x |

Source: Intel Corp (Assuming SATA disk drives with BER 10¹⁴).

4.5.3.1. MTTDL Due to Disk Failure (DF)

The commonly used metric for evaluating the reliability of RAID levels is the MTTDL due to disk failure(s). This analysis does not take into account any other failures, so the

equations are less complex. Recall, an easier way to understand MTTDL is to assume the shortest path to failure. An example of this was illustrated in Figure 30. A more detailed derivation of this MTTDL equation can be found in Appendix B. The equations (10) through (14) are adapted from the following sources [PATT88, CHEN94, and TREA03]. The remaining equations (15) through (24) are derived using the preceding reliability fundamentals.

For non-redundant systems (RAID 0) only one failure is needed to cause data loss. Note that this equation is the base component for all other MTTDL calculations. It is the probability of the first disk failure out of all the other (N_D) disk drives.

Non-Redundant (RAID 0):
$$MTTDL_{DF} = \frac{MTBF_{Disk}}{N_D} \quad (10)$$

With a mirrored approach (RAID 1 or 10) the MTTDL is if any of the (N) disks in the array fail followed by its mirrored copy. Since mirroring produces the smallest group size with only one data disk and one check disk, this results in the highest reliability of all RAID schemes.

Mirroring (RAID 1 / 10):
$$MTTDL_{DF} = \frac{MTBF_{Disk}^2}{N_D \times MTTR_{Mirror}} \quad (11)$$

The nested RAID 0+1, has a lower MTTDL resulting from the higher probability of encountering a second disk failure. This is due to mirroring at the top level producing two potentially large RAID 0 arrays. For this reason, RAID 10 is preferred over a RAID 0+1 implementation.

Nested Mirroring (RAID 0+1):
$$MTTDL_{DF} = \frac{MTBF_{Disk}^2}{N_D^2 / 2 \times MTTR_{Mirror}} \quad (12)$$

For single-parity ($C_D = 1$), data loss will occur if any of the (N_D) disks fail followed by a second disk in the same parity group of size ($N_D - 1$).

Single-Parity (RAID 3 / 4 / 5):
$$MTTDL_{DF} = \frac{MTBF_{Disk}^2}{N_D \times (N_D - 1) \times MTTR_{Disk}} \quad (13)$$

The MTTDL of a dual-parity ($C_D = 2$) scheme is derived from the same conditions as the above single-parity scheme. However, the second and third disk failures each occur in the same diminishing group (N_D). After the first disk failure there are still $(N_D - 1)$ disks in the parity group susceptible to failure. Similarly, after the second failure, there are still $(N_D - 2)$ data disks remaining to potentially fail.

Dual-Parity (RAID 6):
$$MTTDL_{DF} = \frac{MTBF_{Disk}^3}{N_D \times (N_D - 1) \times (N_D - 2) \times MTTR_{Disk}^2} \quad (14)$$

The following six dual-level equations (15) through (20) define the MTTDL due to multiple disk failures (DF). The MTTDL for a dual-level GRAID 50 scheme (15) is derived by combining multiple size-controlled RAID 5 arrays together with RAID 0 striping at the top level. The top level data striping provides added data read/write performance to the array.

Dual-Level (GRAID 50):
$$MTTDL_{DF} = \frac{MTBF_{Disk}^2}{N_D \times N_{E,2} \times (N_D - 1) \times MTTR_{Disk}} \quad (15)$$

The MTTDL for a dual-level GRAID 55 scheme (16) is derived by combining multiple size-controlled RAID 5 arrays together with RAID 5 distributed group parity at the top level. This can generate a great deal of parity overhead as the number of groups increase.

Dual-Level (GRAID 55): (16)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^4}{N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (N_{E,2} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}}$$

The MTTDL for a dual-level GRAID 56 scheme (17) is derived by combining multiple size-controlled RAID 5 arrays together with RAID 6 distributed group dual-parity at the top level. This can produce high parity overhead as the number of groups increase. Similar to the RAID 0+1 example, it is better to use the highest level of redundancy at the lowest level, *i.e.*, GRAID 65 instead of GRAID 56, or RAID 10 instead of RAID 0+1.

Dual-Level (GRAID 56): (17)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^6}{N_D^3 \times N_{E,2} \times (N_D - 1)^3 \times (N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2}$$

The MTTDL for a dual-level GRAID 60 scheme (18) is derived by combining multiple size-controlled RAID 6 arrays together with RAID 0 striping at the top level. The top level data striping provides added data read/write performance to the array.

Dual-Level (GRAID 60): (18)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^3}{N_D \times N_{E,2} \times (N_D - 1) \times (N_D - 2) \times MTTR_{Disk}^2}$$

The MTTDL for a dual-level GRAID 65 scheme (19) is derived by combining multiple size-controlled RAID 6 arrays together with RAID 5 distributed group parity at the top level. This can create a great deal of parity overhead as the number of groups increase.

Dual-Level (GRAID 65): (19)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^6}{N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (N_D - 2)^2 \times (N_{E,2} - 1) \times MTTR_{Disk}^4 \times MTTR_{Enclosure}}$$

The MTTDL for a dual-level GRAID 66 scheme (20) is derived by combining multiple size-controlled RAID 6 arrays together with RAID 6 distributed group dual-parity at the top level. This can generate high parity overhead as the number of groups increase. However, this configuration has the highest MTTDL of all the dual-level GRAID configurations.

Dual-Level (GRAID 66):

(20)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^9}{N_D^3 \times N_{E,2} \times (N_D - 1)^3 \times (N_D - 2)^3} \times \frac{1}{(N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Disk}^6 \times MTTR_{Enclosure}^2}$$

The following four tri-level equations (21 through 24) define the MTTDL due to multiple disk failures (DF). By keeping the enclosure size (N_D) small and the number of disk enclosures ($N_{E,2}$) small, a high MTTDL is produced. The MTTDL for a tri-level GRAID 550 scheme (21) is derived by combining multiple size-controlled RAID 5 disk enclosures collectively in a rack enclosure with RAID 5 distributed parity across all the disk enclosures in each individual rack. At the top level RAID 0 striping is performed across all rack enclosures, *i.e.*, highly reliable GRAID 55 racks.

Tri-Level (GRAID 550):

(21)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^4}{N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times (N_{E,3} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}}$$

The MTTDL for a tri-level GRAID 560 scheme (22) is derived by combining multiple size-controlled RAID 5 enclosures collectively in a rack enclosure with RAID 6 distributed dual-parity across all disk enclosures in each individual rack. At the top level, RAID 0 striping is performed across all rack enclosures, *i.e.*, highly reliable GRAID 56 racks. Once again, this configuration also does not use the highest level of redundancy at the lowest level. Therefore, a GRAID 650 configuration is recommended.

Tri-Level (GRAID 560):

(22)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^6}{N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3} \times \frac{1}{(N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2}$$

The MTTDL for a tri-level GRAID 650 scheme (23) is derived by combining multiple size-controlled RAID 6 enclosures collectively in a rack enclosure with RAID 5 distributed parity across all disk enclosures in each individual rack. At the top level, RAID 0 striping is performed across all rack enclosures, *i.e.*, highly reliable GRAID 65 racks.

Tri-Level (GRAID 650): (23)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^6}{N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times (N_D - 2)^2} \times \frac{1}{(N_{E,3} - 1) \times MTTR_{Disk}^4 \times MTTR_{Enclosure}}$$

The MTTDL for a tri-level GRAID 660 scheme (24) is derived by combining multiple size-controlled RAID 6 disk enclosures collectively in a rack enclosure with RAID 6 distributed dual-parity across all disk enclosures in each individual rack. At the top level, RAID 0 striping is performed across all rack enclosures, *i.e.*, highly reliable GRAID 66 racks. This configuration has the highest MTTDL of all the tri-level GRAID configurations.

Tri-Level (GRAID 660): (24)

$$MTTDL_{DF} = \frac{MTBF_{Disk}^9}{N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3 \times (N_D - 2)^3} \times \frac{1}{(N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^6 \times MTTR_{Enclosure}^2}$$

However, the probability of encountering multiple disk failure is much lower than the combination of other failures frequently encountered [CHEN94]. Therefore, a better representation of the MTTDL would be to consider the likelihood of a disk failure occurring followed by multiple correlated disk failures.

4.5.3.2. MTTDL Due to Correlated Disk Failure (CDF)

Correlated disk failures (CDF) are an important issue to consider when looking at RAID reliability. As stated before, these failures are commonly based on environmental, (*e.g.*, earthquakes, power dips/surges) and manufacturing factors, (*e.g.*, early/late failures in disk drives) [CHEN94]. By taking into account the higher probability of a second or third disk drive failure, this provides a more realistic mean time to data loss due to a CDF.

A simple method for representing the likelihood of a second drive failing is one-tenth the MTBF of the first drive ($MTBF_{Disk}/10$), *i.e.*, each successive disk failure is ten times more likely to occur than the previous [CHEN04]. Similarly, the general representation of a third correlated failure is one-tenth the previous ($(MTBF_{Disk}/10)/10 = MTBF_{Disk}/100$) [CHEN94]. Equations (25) through (28) are adapted from the following sources [CHEN94, TREA03]. The remaining equations (29) through (38) are derived using the preceding reliability fundamentals with the incorporation of correlated failures. Substituting the correlated probability of a second disk failure for the mirrored and single parity schemes is shown in equations (25) through (27). Likewise, taking into account the CDF probability of a second and third disk failure for the dual-parity scheme is shown in equation (28).

Mirroring (RAID 1 / 10): (25)

$$MTTDL_{CDF} = \frac{MTBF_{Disk} \times \left(\frac{MTBF_{Disk}}{10} \right)}{N_D \times MTTR_{mirror}} = \frac{MTBF_{Disk}^2}{10 \times N_D \times MTTR_{mirror}}$$

Nested Mirroring (RAID 0+1): (26)

$$MTTDL_{CDF} = \frac{MTBF_{Disk} \times \left(\frac{MTBF_{Disk}}{10} \right)}{N_D^2 / 2 \times MTTR_{mirror}} = \frac{MTBF_{Disk}^2}{5 \times N_D^2 \times MTTR_{mirror}}$$

Single-Parity (RAID 3 / 4 / 5): (27)

$$MTTDL_{CDF} = \frac{MTBF_{Disk} \times \left(\frac{MTBF_{Disk}}{10} \right)}{N_D \times (N_D - 1) \times MTTR_{Disk}} = \frac{MTBF_{Disk}^2}{10 \times N_D \times (N_D - 1) \times MTTR_{Disk}}$$

Dual-Parity (RAID 6): (28)

$$\begin{aligned} MTTDL_{CDF} &= \frac{MTBF_{Disk} \times \left(\frac{MTBF_{Disk}}{10} \right) \times \left(\frac{MTBF_{Disk}}{100} \right)}{N_D \times (N_D - 1) \times (N_D - 2) \times MTTR_{Disk}^2} \\ &= \frac{MTBF_{Disk}^3}{10^3 \times N_D \times (N_D - 1) \times (N_D - 2) \times MTTR_{Disk}^2} \end{aligned}$$

Dual-level MTTDL equations are constructed using the base level components above and the original reliability equations defined in the previous section.

Dual-Level (GRAID 50): (29)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^2}{1e1 \times N_D \times N_{E,2} \times (N_D - 1) \times MTTR_{Disk}}$$

Dual-Level (GRAID 55): (30)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^4}{1e2 \times N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (N_{E,2} - 1) \times MTTR_{Disk}^2 \times MTTR_{Group}}$$

Dual-Level (GRAID 56): (31)

$$\begin{aligned} MTTDL_{CDF} &= \frac{MTBF_{Disk}^6}{1e3 \times N_D^3 \times N_{E,2} \times (N_D - 1)^3} \\ &\times \frac{1}{(N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2} \end{aligned}$$

Dual-Level (GRAID 60): (32)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^3}{1e3 \times N_D \times N_{E,2} \times (N_D - 1) \times (N_D - 2) \times MTTR_{Disk}^2}$$

Dual-Level (GRAID 65): (33)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^6}{1e6 \times N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (N_D - 2)^2} \times \frac{1}{(N_{E,2} - 1) \times MTTR_{Disk}^4 \times MTTR_{Enclosure}}$$

Dual-Level (GRAID 66): (34)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^9}{1e9 \times N_D^3 \times N_E \times (N_D - 1)^3 \times (N_D - 2)^3} \times \frac{1}{(N_E - 1) \times (N_E - 2) \times MTTR_{Disk}^6 \times MTTR_{Enclosure}^2}$$

Tri-level MTTDL equations are constructed using base dual-level GRAID components above and the original reliability equations defined in the previous section.

Tri-Level (GRAID 550): (35)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^4}{1e2 \times N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times (N_{E,3} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}}$$

Tri-Level (GRAID 560): (36)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^6}{1e3 \times N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3} \times \frac{1}{(N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2}$$

Tri-Level (GRAID 650): (37)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^6}{1e6 \times N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times (N_D - 2)^2} \times \frac{1}{(N_{E,3} - 1) \times MTTR_{Disk}^4 \times MTTR_{Enclosure}}$$

Tri-Level (GRAID 660):

(38)

$$MTTDL_{CDF} = \frac{MTBF_{Disk}^9}{1e9 \times N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3 \times (N_D - 2)^3} \times \frac{1}{(N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^6 \times MTTR_{Enclosure}^2}$$

Once again, there is another potential failure which often goes overlooked. The chance of encountering bit errors in large-scale arrays plays a key role in causing data loss when enough rebuild time and redundancy are not put into place. Therefore, an even better representation of MTTDL would be to consider the likelihood of encountering an UBE.

4.5.3.3. MTTDL Due to Unrecoverable Bit Error (UBE)

The BER is the probability of encountering one bit error in X-bits. Variable X fluctuates based on the class of the disk drive. Enterprise class disk drives usually have a BER of $1:10^{15}$ where business class drives can have a BER of $1:10^{14}$. With large-scale storage arrays ranging in size from petabytes to exabytes, the probability of encountering these types of errors are heightened. This BER gap between classes of drives is critical in determining the overall reliability. Since data on disk drives are read one sector at a time, the sector error rate can be deduced by dividing the BER by the number of bits per sector (512 bytes in a sector, 8 bits in a byte, 4096 bits/sector) [TREA03]. Equations (39) through (42) were adapted from following source [TREA03].

Sector Error Rate:
$$SER = \frac{BER}{512 \times 8} = \frac{BER}{4096} \quad (39)$$

Calculating the number of sectors per disk drive is also a trivial task. The disk capacity (in bytes) divided by the number of bytes per sector (512), produces the number of sectors in a disk, rounded to the nearest whole sector. This places an upper bound on the probability of a sector error. The process of converting the bit error rate to a sector error rate has its benefits to understanding the process. For example, when a bit error occurs on a disk sector, the bit does not

get marked as bad, the sector does. Therefore, using a sector error rate is considered useful. In addition, the number of sectors per disk is much smaller than the number of bytes. Consequentially, this keeps the probability exponent smaller when performing these computations in calculators and computers.

Sectors per Disk:
$$Sectors = \left\lceil \frac{Disk\ Capacity}{Bytes\ per\ Sector} \right\rceil = \left\lceil \frac{DiskSize}{512} \right\rceil \quad (40)$$

By incorporating the above variables, the probability of being able to successfully read all sectors on a disk drive can be computed. This probability assumes that all errors are random [TREA03]. With the disk probability in hand, the probability of encountering a sector error in the rebuild group can be calculated based on the number of data disks per group.

Probability of Successfully Reading All Disk Sectors: (41)

$$P_{Disk} = \left(1 - \frac{1}{SER} \right)^{Sectors}$$

Probability of Encountering a Sector Error in a Disk Enclosure: (42)

$$P_{Enclosure} = 1 - P_{Disk}^{N_D - C_D}$$

Now that the group probability is known, this can be substituted into the preceding MTTDL due to CDF equations, (25) through (38). This replaces the last disk failure needed in each group to result in data loss. In the case of mirroring only, the probability of a single disk is required. However, with RAID 0+1 the probability of encountering a UBE is much greater since a rebuild must read from $(N_D / 2)$ disks. This reiterates the pitfalls of using RAID 0+1, and why RAID 10 is the preferred choice. Equations (43) through (46) were adapted from following sources [TREA03, CHEN94]. The remaining equations (47) through (56) are derived using the preceding reliability fundamentals with the incorporation of unrecoverable bit errors.

Mirroring (RAID 1 / 10): (43)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}}{N_D \times (1 - P_{Disk}^1)}$$

Nested Mirroring (RAID 0+1): (44)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}}{N_D \times \left(1 - P_{Disk}^{N_D/2}\right)}$$

For both the single and dual-parity schemes, the probability of encountering a UBE in the last disk is based on the remaining data disks in the group, *i.e.*, it is assumed that all check disks have failed.

Single-Parity (RAID 3 / 4 / 5): (45)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}}{N_D \times \left(1 - P_{Disk}^{(N_D-1)}\right)}$$

Dual-Parity (RAID 6): (46)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^2}{10 \times N_D \times (N_D - 1) \times \left(1 - P_{Disk}^{(N_D-2)}\right) \times MTTR_{Disk}}$$

Dual-Level (GRAID 50): (47)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}}{N_D \times N_{E,2} \times \left(1 - P_{Disk}^{(N_D-1)}\right)}$$

Dual-Level (GRAID 55): (48)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^2}{N_D^2 \times N_{E,2} \times \left(1 - P_{Disk}^{(N_D-1)}\right) \times (N_{E,2} - 1) \times MTTR_{Enclosure}}$$

Dual-Level (GRAID 56): **(49)**

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^3}{N_D^3 \times N_{E,2} \times (1 - P_{Disk}^{(N_D-1)})^3 \times (N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Enclosure}^2}$$

Dual-Level (GRAID 60): **(50)**

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^2}{1e1 \times N_D \times N_{E,2} \times (N_D - 1) \times (1 - P_{Disk}^{(N_D-2)}) \times MTTR_{Disk}}$$

Dual-Level (GRAID 65): **(51)**

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^4}{1e2 \times N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (1 - P_{Disk}^{(N_D-2)})^2} \times \frac{1}{(N_{E,2} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}}$$

Dual-Level (GRAID 66): **(52)**

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^6}{1e3 \times N_D^3 \times N_{E,2} \times (N_D - 1)^3 \times (1 - P_{Disk}^{(N_D-2)})^3} \times \frac{1}{(N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2}$$

Tri-Level (GRAID 550): **(53)**

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^2}{N_D^2 \times N_{E,3} \times N_R \times (1 - P_{Disk}^{(N_D-1)})^2 \times (N_{E,3} - 1) \times MTTR_{Enclosure}}$$

Tri-Level (GRAID 560): **(54)**

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^3}{N_D^3 \times N_{E,3} \times N_R \times (1 - P_{Disk}^{(N_D-1)})^3 \times (N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Enclosure}^2}$$

Tri-Level (GRAID 650): (55)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^4}{1e2 \times N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times \left(1 - P_{Disk}^{(N_D-2)}\right)^2} \times \frac{1}{(N_{E,3} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}}$$

Tri-Level (GRAID 660): (56)

$$MTTDL_{UBE} = \frac{MTBF_{Disk}^6}{1e3 \times N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3 \times \left(1 - P_{Disk}^{(N_D-2)}\right)^3} \times \frac{1}{(N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2}$$

4.5.3.4. Harmonic MTTDL

Analyzing each of the three previous failure scenarios (disk failure, correlated disk failure, or unrecoverable bit error) individually can be useful in determining which type of failure results in the lowest system reliability. However, this has already been evaluated in [CHEN94]. Their conclusion was that a disk failure followed by a bit error has a much lower MTTDL than an array encountering two correlated disk failures. Therefore, by taking the harmonic mean of the above MTTDL equations for each respective configuration, a better representation of the expected reliability is produced [TREA03, CHEN94]. The harmonic mean is a type of average/approximation commonly used to evaluate the average of rates. It is derived by taking the number of elements (n) and dividing it by the sum of the reciprocals of each element (a_n). Equation (57) is the basic representation of a harmonic mean.

Harmonic Mean: (57)

$$H = \frac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \dots + \frac{1}{a_n}}$$

The MTTDL results attained for each scenario in the previous sections are rates of failure. By taking the harmonic mean of these rates, the correct average is produced. The harmonic mean is ideal for calculating the MTTDL since it produces more conservative rates by favoring the

lesser values. This is due to the harmonic mean's tendency to exaggerate the impact of small values (MTTDL due to UBE) and lessen the effect of much larger ones (MTTDL due to DF and CDF). In this situation the number of variables (n) is equal to three (MTTDL due to DF, CDF, and UBE). Dividing the number of elements by the sum of the reciprocals of each value produces the harmonic MTTDL shown in equation (58). This representation will be used in Chapter 5 to determine what effect system variables have on overall reliability for each redundancy level.

Harmonic MTTDL:

$$MTTDL = \frac{3}{\frac{1}{MTTDL_{DF}} + \frac{1}{MTTDL_{CDF}} + \frac{1}{MTTDL_{UBE}}} \quad (58)$$

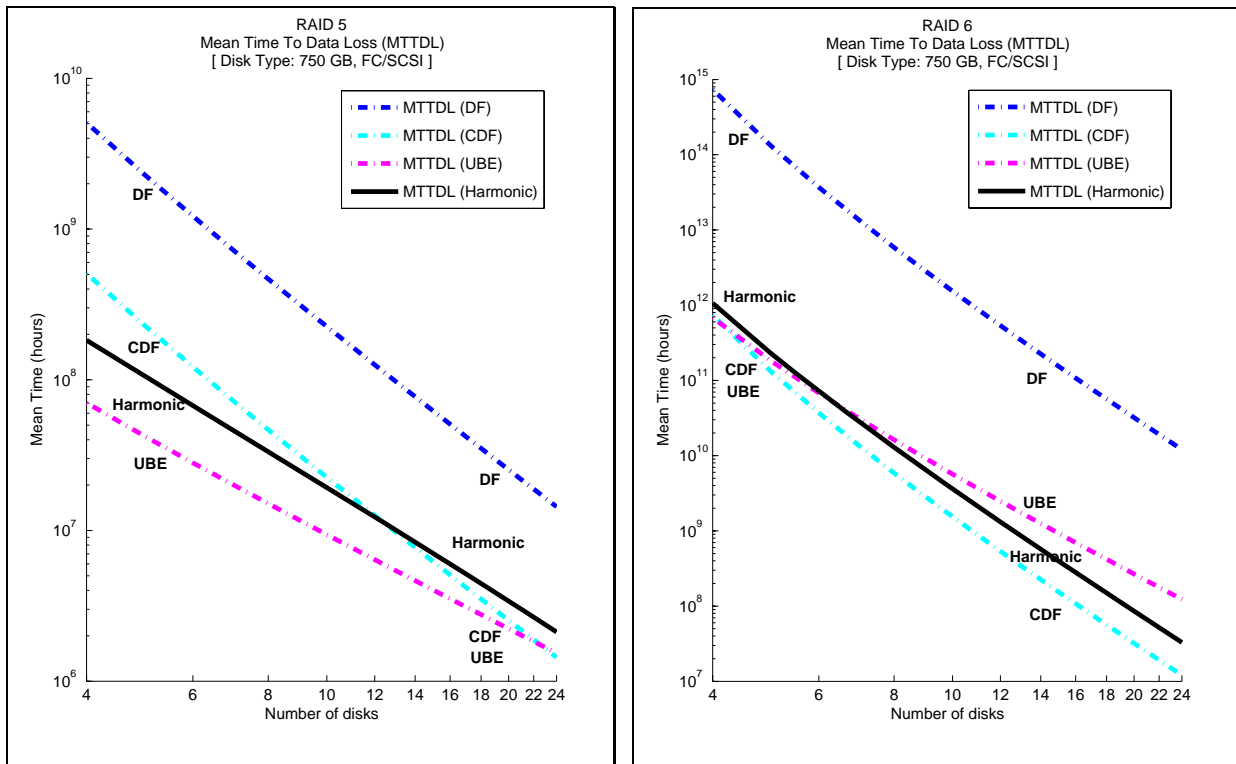


Figure 31: Elements of Harmonic MTTDL for RAID 5 (left) and RAID 6 (right)

An example of what each element in a harmonic MTTDL looks like is illustrated in Figure 31. The left figure shows the MTTDL for RAID 5 due to disk failure (DF), correlated disk failure (CDF), and unrecoverable bit error (UBE). In the right figure, the equivalent for a

RAID 6 array is represented. The harmonic MTTDL for each respective figure is represented by the solid black line. Note how the harmonic mean tends to represent the lesser MTTDL trends (CDF and UBE) rather than the larger (DF) values. Another interesting point to make involves the variance between RAID 5 and 6 MTTDL values. For a small RAID 5 array with only 4 disks the variance between each of the MTTDL values is wider than an array with 24 disks. Conversely, a RAID 6 array with 4 disks has a tightly clustered grouping of each MTTDL and gradually diverges as the number of disks increases.

4.6. How Accurate Are MTBF Ratings?

Back in the early days of RAID, a manufacturer's MTBF rating was considered to be an acceptable estimation of its reliability. However, storage arrays which existed in the 90's are small by today's standards. Failure rates recorded by modern data centers - housing thousands of disk drives - conflict with those provided by the manufacturers. Most modern data centers continually collect data on the quantity and frequency of hardware failures in their systems. This information is then used to better estimate the yearly expenses needed for maintaining large-scale storage systems. Even though hardware manufacturers already provide these ratings in the form of MTBF and AFR estimations, end users are finding discrepancies in the actual failure rates vs. those of the manufacturer. Recently, in February 2007 at the 5th USENIX conference on File and Storage Technologies (FAST), independent studies by researchers at Carnegie Melon [SCHR07] and Google Labs [PINH07] gathered evidence indicating that these reliability ratings do not correlate with real world failure statistics. Each study independently came to the conclusion that current MTBF and AFR ratings of disk drives by manufacturers are not good estimations of their expected failure rate. To better understand why these ratings are currently under scrutiny, an overview of the various methods in which manufacturers use to derive their reliability ratings is provided.

The basic flaw in calculating these rates comes from statistical estimations which are then projected over time. Manufacturers commonly evaluate their disk drives in a testbed which consists of hundreds to thousands of disk drives running in parallel for many months. The reliability-demonstration test (RDT) period occurs after the disks have been properly burned in

to rule out any premature hardware failures. During a RDT the disks are put through an accelerated stress test to emulate extreme disk workloads and operating conditions [COLE00]. After this accelerated test period, the number of collective hours which all disk drives have been “in service” can approach several million hours. Take for example, a testbed containing 1000 disks running in parallel for 100 days. Even though each disk has only been operating for 2400 hours ($100\text{ days} \times 24\text{ hours/day} = 2400\text{ hours}$), by taking into consideration all 1000 disks, a manufacturer has essentially accumulated 2.4 million hours of disk “in service” time ($1,000\text{ disks} \times 2,400\text{ hours} = 2,400,000\text{ hours}$) [NRS07]. Finally this projected disk in service time divided by the number of disk failures encountered during the RDT results in the expected MTBF, e.g., $2,400,000\text{ hours} / 2\text{ failures} = 1,200,000\text{ hours}$. Bear in mind that example is simplified and manufacturers will often incorporate additional statistical data to better determine a drive’s MTBF rating. However, the pitfall with this approach is that projected MTBF ratings are usually generous estimations of a device’s expected failure rate. A more conservative approach would be to use the Mil-Spec method [NRS07].

Although the Mil-Spec method is not usually utilized by disk manufacturers, the process takes into consideration several variables, thus providing a more conservative MTBF. The prediction model outlined in the Military Handbook for Reliability Prediction of Electronic Equipment (MIL-HDBK-217) (a.k.a. Mil-Spec) is the reliability rating system frequently used by the military for determining failure rates of printed circuit board (PCB) and microprocessor designs [NRS07]. This method takes into consideration the reliability of each individual component. Therefore, it traditionally provides a more conservative MTBF preferred by the military when determining the reliabilities of high availability systems. However, for large-scale storage systems these MTBF ratings are too moderate. Most reliable failure ratings are those generated by the customers, who consume an estimated 300 million disk drives annually [SCHR07].

By using failure rates based on consumer experience, this has allowed drive manufacturers to provide more accurate MTBF ratings. These improved rates are derived from the demonstrated AFR recorded in the field. Dividing the total number of hours in a year (8760 hours) by the customer’s demonstrated AFR ($\sim 3\%$) [SCHR07] produces more realistic

MTBF results ($8760 \text{ hours} / 0.03 = 292,000 \text{ hours}$). Another problem with current MTBF ratings is that they are based on the assumption that a constant failure rate exists over its useful lifetime. Research at Seagate has shown that various usage levels of disk drives will require an adjustment of the manufacturer's MTBF. The MTBF specification multiplier based on expected power-on hours (POH) is shown in Figure 32. Based on this information, disks which are operating year round can expect to have their MTBF ratings approximately halved. However, recent studies show actual disk failure rates to be much higher. The following two independent case studies by researchers at Carnegie Melon and Google Labs provide recommendations on how to better represent a disk drive's MTBF and AFR.

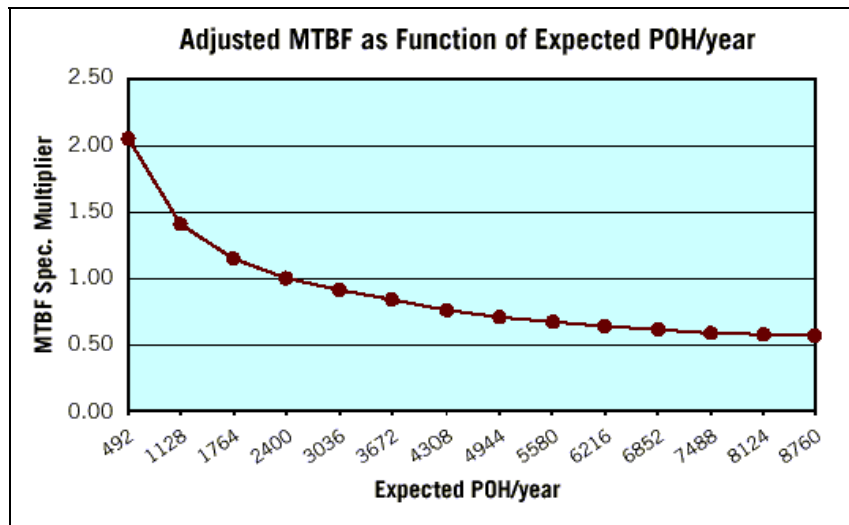


Figure 32: Adjusted MTBF Due To POH [COLE00]

4.6.1. Carnegie Melon Case Study

For many years hard disk reliability estimations have assumed that once a device has been properly burned in, it will exhibit a much lower and constant rate of failure over its useful lifetime. After this period (around the time drive warranties expire) the failure rate of disk drives will rapidly increase (its wear out period). An illustration of a disk's expected failure rate over its expected lifetime, commonly referred to as the "bathtub curve", is shown in Figure 33. Most hardware manufacturers perform their own burn-in of new disk drives before shipping them off to customers. Therefore, most premature disk failures are caught before ever reaching the

consumer. However, this assumption of a constant failure rate during a disk's useful-life period is being disputed by researchers at Carnegie Melon University (CMU).

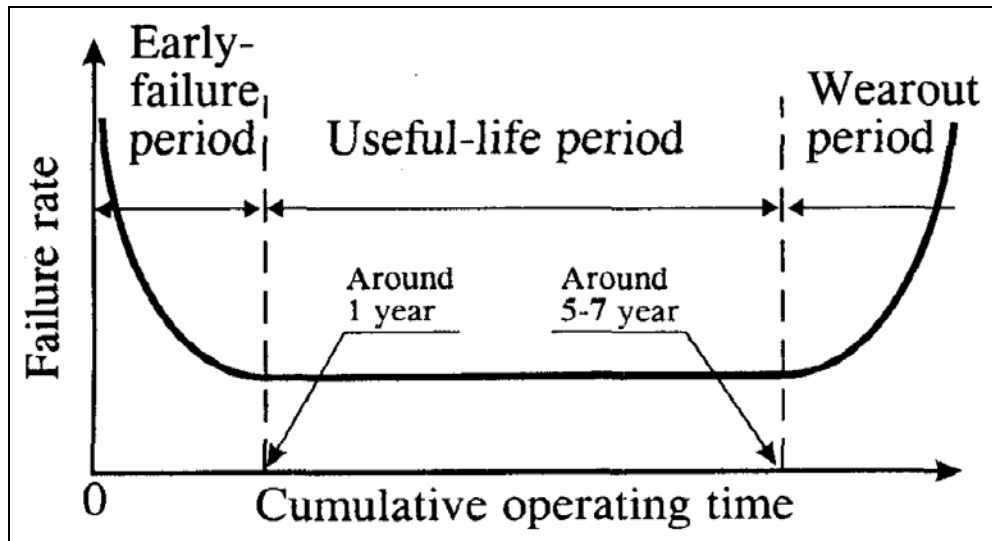


Figure 33: Failure Rate of Hard Drives over Expected Lifetime [YANG99]

The study by CMU used failure statistics gathered from 7 different large-scale data centers, forming a collective drive population of more than 100,000 disks. More than four different drive vendors and most of the common classes of disk drives (SATA, SCSI, and FC) were represented in the study. Based on drive manufacturer's MTBF ratings (ranging from 1 to 1.5 million hours) the AFR of these disk drives should typically fall between 0.58% and 0.88%. However, CMU's findings show that real world annual disk replacement rates are commonly between 2% and 4%. The weighted averages of the AFR rates were 3.5 times larger than manufacturer's rates [SCHR07]. The annual replacement rates (ARR) for each site's drive types is depicted in Figure 34. The dotted line at ~3% represents the average ARR across all disks. Also noted in their case study was that failure rates grew constantly after the second year of operation. This differs from the constant failure rate expected by the bathtub curve in a drive's useful lifespan. It also raises concern since the International Disk drive Equipment and Materials Association (IDEMA) is currently in the process of forming a new standard for specifying disk reliability. Their reliability criteria (basing on the bathtub curve) would require vendors to provide four different MTBF ratings for incremental ranges of time in a disk's lifetime. However, this new proposed standard is still based on the assumption that failure rates remain constant during a disk's useful lifespan

[SCHR07]. Hopefully researchers at CMU and others in the community will voice their concerns to the IDEMA regarding this issue before finalizing their decision.

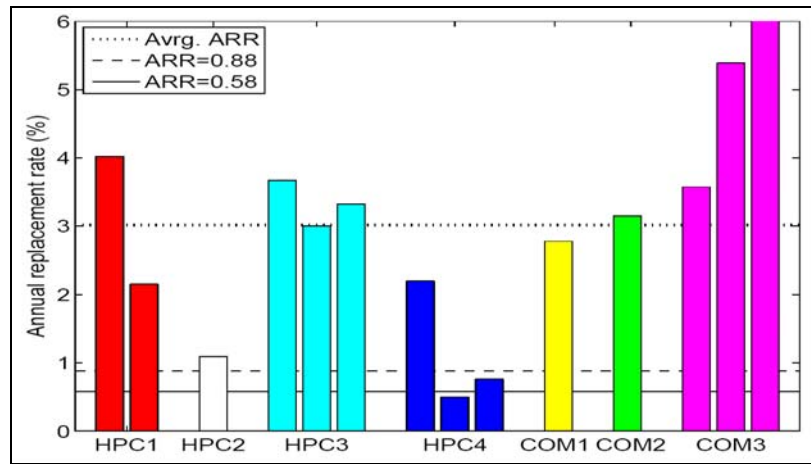


Figure 34: ARR for Each Drive Type [SCHR07]

Another important finding was the lack of difference between SATA, SCSI, and FC replacement rates. Even though the HPC4 site (shown in Figure 34) consisted entirely of business class SATA disk drives, it had some of the lowest replacement rates in the entire study. In addition to higher performance, the lower failure rates of enterprise class disk drives are their key selling point. This study, which showed no significant reliability advantage between business and enterprise class disk drives, should cause drive vendors to start re-examining their drive's reliability classes. This discovery provides an indication that disk-independent factors are resulting in the higher failure rates. However, the study by Google Labs disagrees with this finding. Their data show no direct correlation between environmental variances and drive failures.

4.6.2. Google Labs Case Study

Researchers at Google Labs recently presented their findings on failure trends in large disk drive populations at the same FAST'07 conference as CMU. However, Google's advantage was their own massive population of disk drives to gather data from (more than 100,000) [PINH07]. Their findings agree with CMU's regarding higher failure rates in disk drives (between 2% and

8% in some circumstances). The AFRs of Google’s disk drives, broken down by age group, are shown in Figure 35. This corresponds with CMU’s conclusion that disk failure rates do not remain constant during their useful lifetime (1 → 5 years).

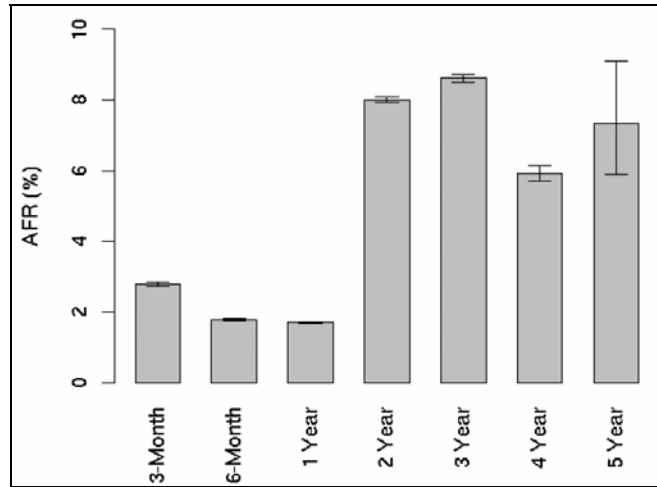


Figure 35: AFR by Age Group [PINH07]

Conversely, Google’s study disagrees with CMU’s conclusion that disk-independent variables such as temperature and humidity play a critical role in the failure rates of disk drives. It also disagrees with the findings in [COLE00] which showed how expected MTBF rates will increase as the ambient temperature drops. Google’s study finds no correlation between increases in temperature causing more failures (with the exception of extremely high temperatures). In fact, just the opposite was concluded. Disk drives (less than 3 years old) seem to suffer from higher failure rates when the average temperature decreases [PINH07]. Figure 36 shows this relationship between AFR and a drive’s average temperature broken down by age group. However, what was lacking from Google’s study is a more diverse sample of disk drives. Google’s analysis only contained data from consumer grade ATA disk drives used in their search engine’s nodes. Therefore, the claim made by CMU that failure rates are independent of drive type cannot be confirmed nor denied.

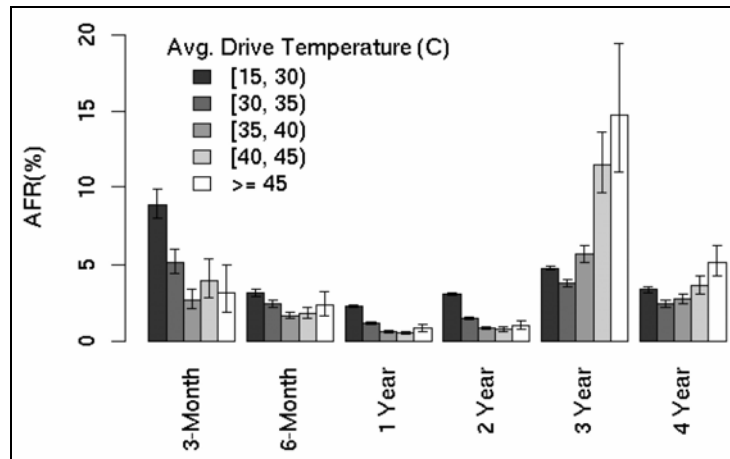


Figure 36: AFR and Average Temperature by Age Group [PINH07]

4.7. Summary of Metrics

The MTBF ratings provided by manufacturers are not always good estimations of failure rates. Case studies [SCHR07] show that during a disk's useful lifespan (1 → 5 years), failure rates are 3.4 times larger than actual manufacturer rates. And in the wear our period of disk drives (5 → 8 years) AFR rates are 30 times greater. In order to obtain more realistic failure rates of disk drives, large-scale storage centers need to start sharing their actual failure rates with vendors and the community. Doing so will allow for a better representation of failure rates when multiple sources are amassed together. However, the primary reason for this lack of information sharing is due to the reluctance of large-scale storage centers, to release their failure statistics [SCHR07]. Many businesses consider this information proprietary and critical to maintaining a competitive edge in the industry. Another reason could be due to a lack of any failure statistics maintained at some data centers.

Now that all of the necessary GRAID metrics have been derived, the next step is to thoroughly evaluate each redundancy scheme to better understand its reliability. However, an examination of mirroring schemes will not be considered in Chapter 5. The reliability of mirroring schemes has already been established in [PATT88, CHEN94]. Plus, another consideration is also due to the financial limitations of building and maintaining a large-scale storage array with mirroring. When dealing with thousands to millions of disk drives, having

50% storage efficiency and 100% redundancy is simply not feasible. As stated previously, mirroring schemes (RAID 1/10) are best suited for small scale arrays which house frequently accessed critical data. Once data become infrequently accessed, it then should be migrated to a large-scale system with much higher storage efficiency. The mirroring of geographically separated data centers is something to consider if more system components were incorporated into reliability calculations. However, this thesis is focused towards the reliability of single-site storage systems. The reliability of the dual-level and tri-level configurations will be analyzed in the subsequent Chapter 5.

CHAPTER 5

5. RELIABILITY ANALYSIS

5.1. RAID Experiments

The following experiments were conducted at the Center for Ocean Atmospheric Prediction Studies in July of 2007 using newly acquired storage disks and storage enclosures. Each of these configurations was tested to validate the proposed revision to the MTTR formula, previously assumed to be a fixed empirical value. Instead, these experiments show how the number of disks in a group will greatly affect the rebuild time required to return the storage system to an optimal state.

5.1.1. Areca SATA RAID Controller

An experimental RAID system was implemented using an Areca 1160 (ARC-1160) SATA RAID controller with sixteen (16) 750 GB SATA-II disk drives. The rack enclosure was 3U in height (maximum capacity of 16 drives). These disks were using the latest perpendicular recording technology. The storage array was configured using RAID Level 6, resulting in a total useable volume size of 9.8 TiB ($N_D = 16$, $C_D = 2$, $DiskSize = 750 \text{ GB} = 698.49 \text{ GiB}$, $ArraySize = 698.49 \text{ GiB} \times (16-2) = 9.8 \text{ TiB}$). The rebuild priority was set to 20% ($P=0.2$) and the media rate of SATA-II is 300 MB/s ($M=300e6$).

Before the experiment began, the array was at an optimal state and no additional traffic was accessing the storage volume. To simulate a disk failure, a drive was physically removed from the array. While in a degraded state, random large data files were written to the array, so that upon re-insertion of the “failed” disk, an entire disk rebuild would be needed. All disk activity was then suppressed and the “failed” drive was re-inserted into the array. The rebuild process began immediately and completed after 35666 seconds (~9.9 hours). Knowing this actual MTTR value and the related array variables, the expected MTTR can be determined for verification.

$$\begin{aligned}
MTTR_{disk} &= H_T + R_T = H_T + \frac{DiskSize}{\frac{M}{e^{(0.08 \cdot [N_D - C_D])}} \times P \times 3600 \text{sec}} \\
&= 0 + \frac{750e9 \times (10^9 / 2^{30})}{\frac{300e6}{e^{(0.08 \cdot [16-2])}} \times 0.2 \times 3600 \text{sec}} = 9.9 \text{ hours}
\end{aligned} \tag{59}$$

The empirical inefficiency ratio $e^{(0.08 \times [N_D - C_D])}$ was determined by solving the MTTR equation with the known variables above for the variable x represented in (60), ($x=3.06368$). An exponential ratio seemed to better represent the increase in rebuild time as the number of disks in an enclosure increases. Therefore, solving for the variable y in (61) results in the skew value ($y=0.0799$), rounded up to (0.08). Using this new empirical inefficiency ratio, which takes into consider the number of disks in an enclosure, is a better representation of a disk's MTTR.

$$35666 = \frac{750e9 \times (10^9 / 2^{30})}{\frac{300e6}{x} \times 0.2}, \quad \text{solving for } x, \quad x = 3.06368 \tag{60}$$

$$3.06368 = e^{(y \cdot [16-2])}, \quad \text{solving for } y, \quad y = 0.0799 \cong 0.08 \tag{61}$$

5.1.2. InforTrend EonStor SATA RAID Enclosure

A second independent RAID system was setup using an InforTrend EonStor SATA RAID enclosure which can house twenty-four (24) 750 GB SATA-II disk drives. The rack enclosure was 4U in height (maximum capacity of 24 drives, minus one for hot spare). These disks were using the latest perpendicular recording technology. The storage array was configured using RAID Level 6, resulting in a total useable volume size of 14.6 TiB ($N_D = 23$, $C_D = 2$, $DiskSize = 750 \text{ GB} = 698.49 \text{ GiB}$, $ArraySize = 698.49 \text{ GiB} \times (23-2) = 14.6 \text{ TiB}$). The rebuild priority was set to a maximum value of 20%, however this volume was part of a clustered filesystem. Therefore, a minimal amount of disk traffic is always generated while operational. To take into account this variance, a rebuild priority of 19% was used ($P=0.19$). The media rate was 300 MB/s ($M=300e6$).

The same setup and failure procedure was used for this experiment as the previous. The rebuild process completed after 66005 seconds (~18.3 hours). Knowing this actual MTTR value and the related array variables, the expected MTTR can be determined for verification.

$$\begin{aligned}
 MTTR_{disk} &= H_T + R_T = H_T + \frac{DiskSize}{\frac{M}{e^{(0.08 \cdot [N_D - C_D])}} \times P \times 3600 \text{sec}} \\
 &= 0 + \frac{750e9 \times (10^9 / 2^{30})}{\frac{300e6}{e^{(0.08 \cdot [23 - 2])}} \times 0.19 \times 3600 \text{sec}} = 18.3 \text{hours}
 \end{aligned} \tag{62}$$

The empirical inefficiency ratio $e^{(0.08 \cdot [N_D - C_D])}$ can be verified by solving the MTTR equation with the known variables above for the variable x represented in (63), ($x=5.38629$). Once again, an exponential ratio seems to better represent the increase in rebuild time as the number of disks in an enclosure increases. Therefore, solving for the variable y in (64) results in the skew value ($y=0.0801$), rounded down to (0.08). This verifies that using this new empirical inefficiency ratio, which takes into consideration the number of disks in an enclosure, is a fairly accurate representation of a disk's MTTR.

$$66005 = \frac{750e9 \times (10^9 / 2^{30})}{\frac{300e6}{x} \times 0.19}, \quad \text{solving for } x, \quad x = 5.38629 \tag{63}$$

$$5.38629 = e^{(y \cdot [23 - 2])}, \quad \text{solving for } y, \quad y = 0.0801 \cong 0.08 \tag{64}$$

5.2. GRAID Calculator

Determining the recommended configuration for multiple array magnitudes can be a difficult undertaking. Therefore, to simplify this task and to assist in the reliability analysis, the GRAID Reliability Calculator was implemented in MATLAB. This reliability calculator, specifically written for large-scale storage arrays, has many user selectable/changeable

parameters, such as: Array Size, Disk Type, Disk Size, and GRAID Level. The following two sections outline the main menu components of this tool and what the expected output(s) will entail.

5.2.1. Menu Interface

The interface to the GRAID Reliability Calculator is driven by multiple simple menu selections. The first menu provides a mixture of array magnitudes ranging from 100 TiB up to 100 EiB. After this, the user has four different disk types to select from. The four main types of disks include: PATA, SATA, SAS, and FC/SCSI. All of the associated parameters for each are automatically utilized based on the type selection. These parameters are detailed in Table 13. The next parameter selection involves the disk size. This can range from the smallest 80 GB drive up to the largest disk drive on the market today, 1 TB. The last menu selection is for the desired GRAID level(s) needing consideration. Each of the single GRAID levels can be chosen individually, a grouping of levels can be chosen, *e.g.*, GRAID 5x, or ALL GRAID levels can be analyzed. It is important to note that all of the above parameters are changeable in the code if desired. However, a menu interface with the commonly used components in storage arrays make calculating different configuration quick and easy. Figure 37 illustrates the overall flow of the menus.

Table 13: Disk Type Parameters

| | |
|---------------------------------------|---------------------------------------|
| Disk Type: SATA | |
| Mfgr. MTBF = 1200 khrs | |
| Actual MTBF = 352.941 khrs | |
| Annualized Failure Rate (AFR) = 2.48% | |
| Media Rate (M) = 300 MB/s | |
| Bit Error Rate (BER) = 1e+14 | |
| Disk Type: SAS | Disk Type: FC/SCSI |
| Mfgr. MTBF = 1400 khrs | Mfgr. MTBF = 1600 khrs |
| Actual MTBF = 411.765 khrs | Actual MTBF = 470.588 khrs |
| Annualized Failure Rate (AFR) = 2.13% | Annualized Failure Rate (AFR) = 1.86% |
| Media Rate (M) = 300 MB/s | Media Rate (M) = 320 MB/s |
| Bit Error Rate (BER) = 1e+15 | Bit Error Rate (BER) = 1e+16 |

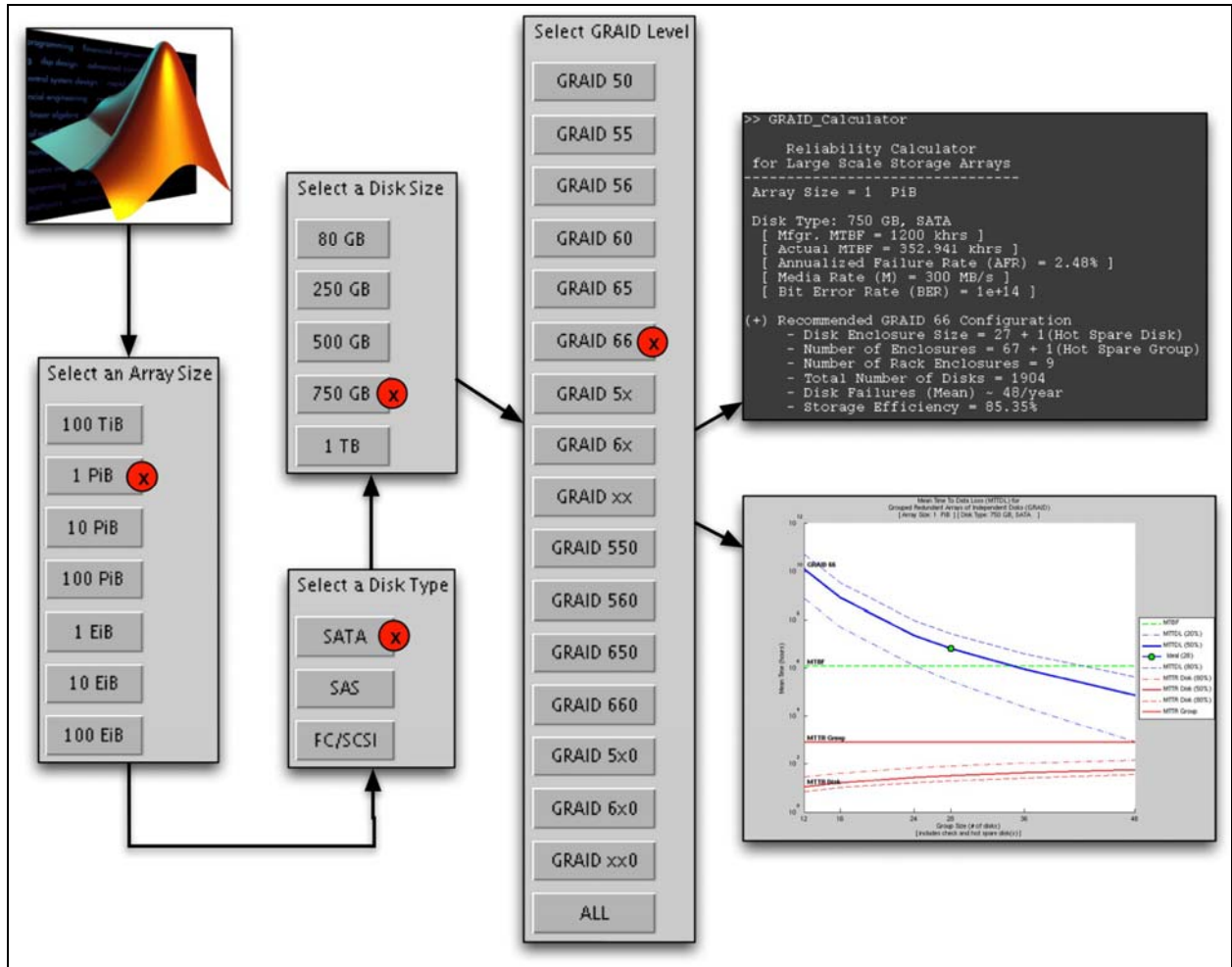


Figure 37: Stages of the Menu Interface

5.2.2. Results and Figures

Once all of the parameters have been selected from the menus, the next stage calculates the recommended configuration for the GRAID level(s) and then plots the results in an easy to read figure. The background information processing is very complex, and can involve multiple processing iterations depending on the number of GRAID levels selected. A flow diagram of the processing code is represented in Figure 38. To provide the user with as much feedback as possible, there are two sources of information regarding each recommended design configuration. In the textual standard output of MATLAB the following parameters are provided: Disk Enclosure Size, Number of Disk Enclosures, Number of Rack Enclosures, Total

Number of Disks, Mean Disk Failures (per year/day/hour), and the Storage Efficiency. In the figure, the MTTDL trend(s) is/are plotted with respect to the enclosure size. As guidelines, the manufacturer's MTBF rating and the MTTR will be displayed (depending on the number of levels selected). MATLAB code for the GRAID Reliability Calculator is provided in Appendix C.

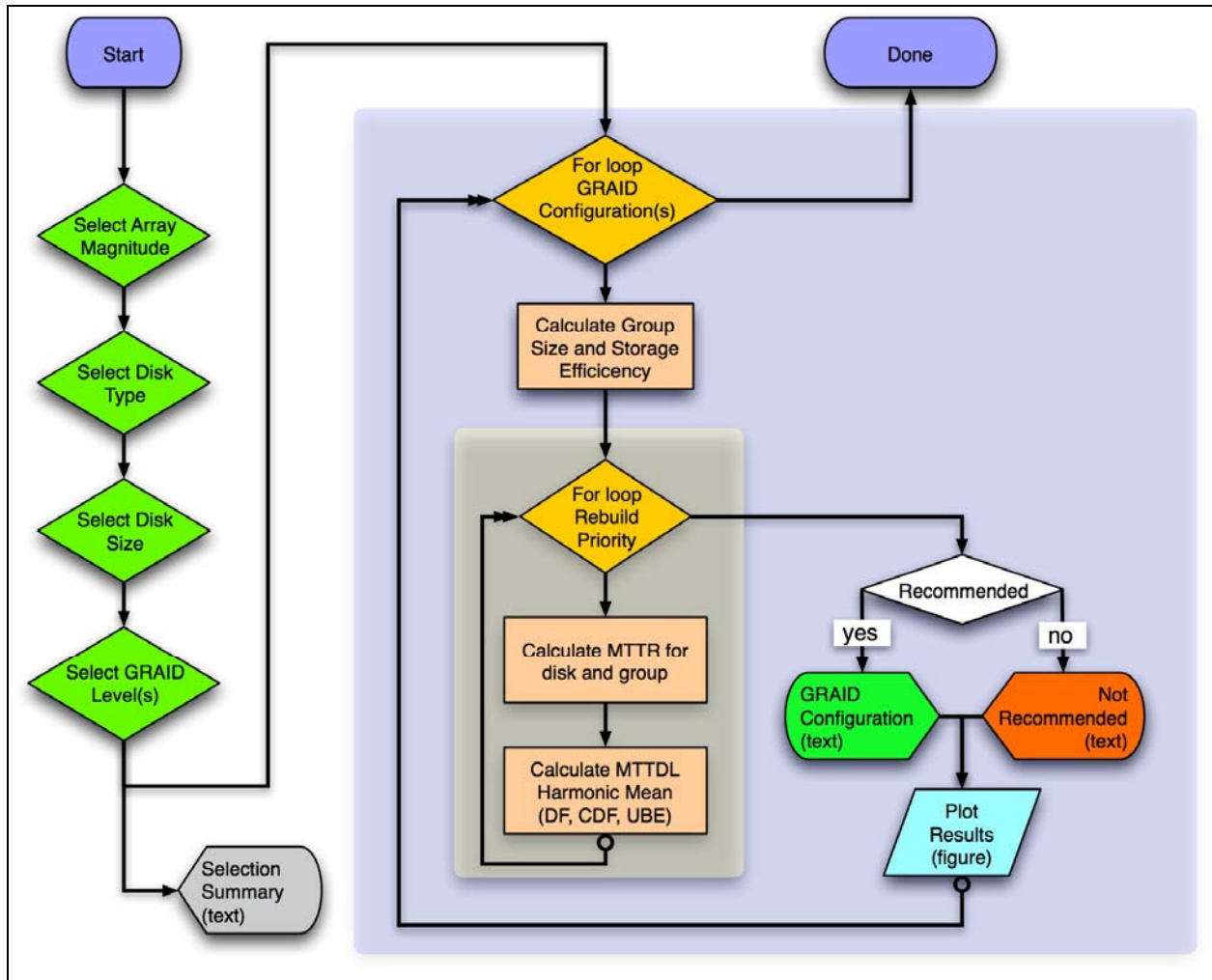


Figure 38: Processing Flow Chart

```
Reliability Calculator
for Large-scale Storage Arrays
-----
Array Size = 10 PiB

Disk Type: 500 GB, FC/SCSI
[ Mfgr. MTBF = 1600 khrs ]
[ Actual MTBF = 470.588 khrs ]
[ Annualized Failure Rate (AFR) = 1.86% ]
[ Media Rate (M) = 320 MB/s ]
[ Bit Error Rate (BER) = 1e+16 ]

(+) Recommended GRAID 66 Configuration
- Disks Per Enclosure = 27 + 1(Hot Spare Disk)
- Number of Disk Enclosures = 970 + 1(Hot Spare Enclosure)
- Number of Rack Enclosures = 122
- Total Number of Disks = 27188
- Disk Failures (Mean) ~ 2/day
- Storage Efficiency = 89.01%
```

Figure 39: Sample Results Provided In MATLAB Standard Output

5.3. Reliability Anomalies

During the analysis and testing phase, there were several anomalies encountered which need to be addressed. The first discrepancy entails a convergence of two or more MTTDL values, resulting in a breakdown of the MTTDL reliability analysis beyond certain array magnitudes. The second involves the MTTDL divergence of GRAID 5x and 6x groups as array magnitudes increase. The following sections will discuss these two occurrences and give reasons for their potential cause.

5.3.1. MTTDL Convergence

An MTTDL convergence occurs when a grouping of GRAID 5x, 6x, or simply two single-levels cross paths with one another. Beyond this point the MTTDL reliability analysis appears to break down and is no longer valid. This was to some extent anticipated since an assumption

made in [PATT88] was for the MTTDL to be much greater than the MTTR, i.e., $[MTTDL/(N_D+C_D)] \gg MTTR$. One potential source behind this convergence may possibly be due to the growing MTTR as the group size increases. This addition represents a more realistic MTTR value in these large-scale arrays. Past assumptions used a static MTTR value, no matter what the group size was.

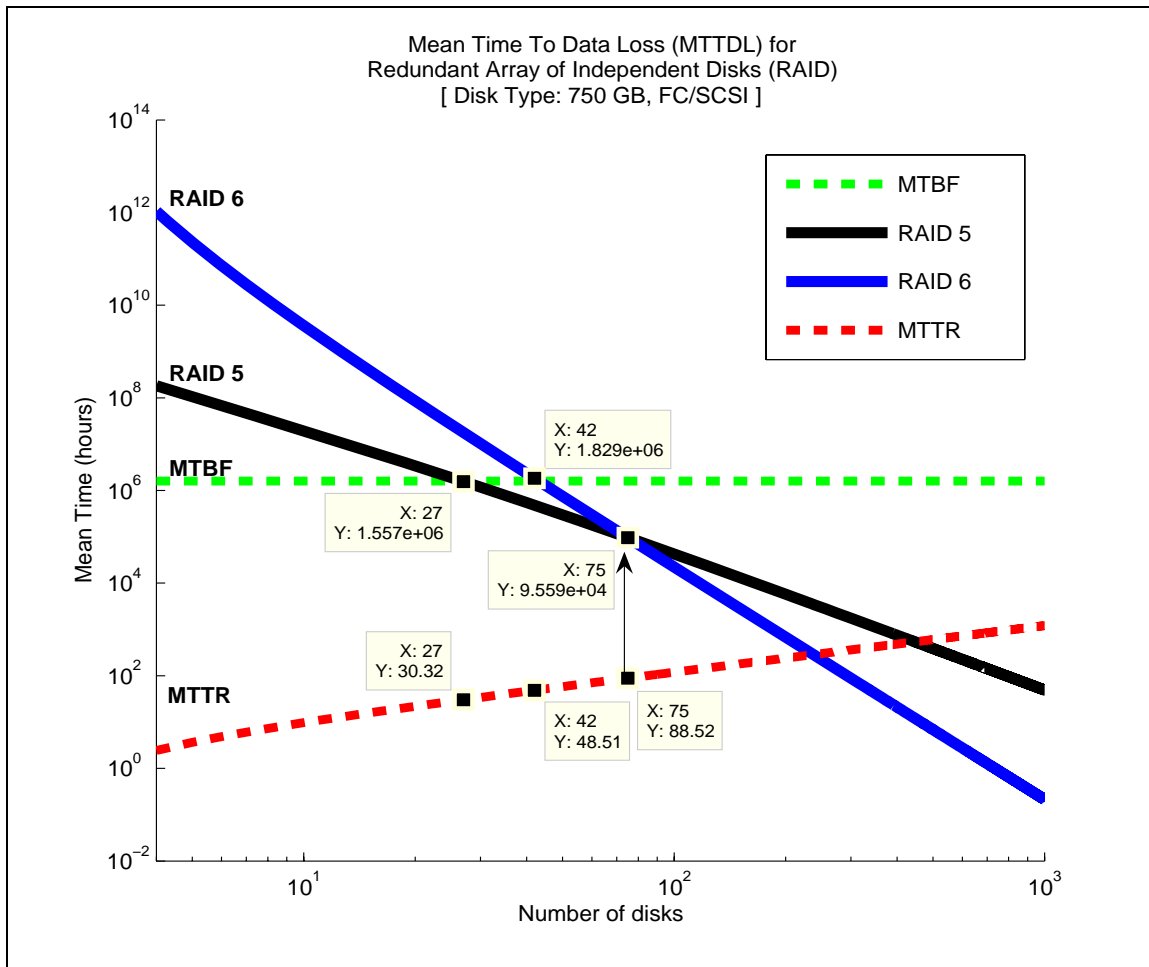


Figure 40: MDDTL Convergence of RAID 5 and RAID 6

Take for example, the case of two single-level arrays, RAID 5 and RAID 6, represented in Figure 40. Here the MTTDL junction occurs when the number of drives in each level reaches approximately 75 disks. Beyond this point the MTTDL for a RAID 5 array is deemed more reliable than a RAID 6 array. However, this could never be the case and it represents a clear example of where the MTTDL analysis is no longer valid. At this point the MTTDL for 75 disks

in a RAID 5 or RAID 6 array is roughly 95.5 thousand hours and the MTTR is approximately 88.5 hours (~3.6 days). Here the MTTDL is greater than the MTTR by a factor of about 1000, and is lower than the manufacturer's MTBF (1.6 million hours for FC/SCSI disk) by a factor of around 16. Even though this seems like a wide margin, it is minuscule in comparison to the MTTDL of a 4 disk RAID 6 array ($MTTDL=10^{12}$). In this example the MTTDL is larger than its MTTR (2.5 hours) by a factor of 400 billion.

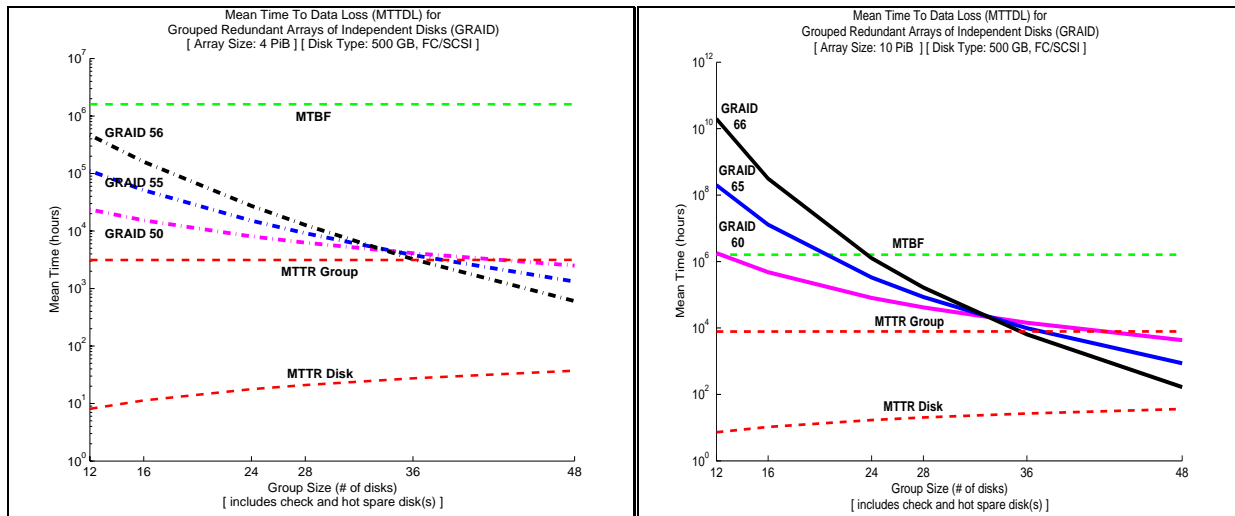


Figure 41: MDDTL Convergence for GRAID 5x (left) and GRAID 6x (right)

The convergence between the GRAID 5x or 6x levels is illustrated in Figure 41. For the GRAID 5x levels (left) this convergence begins to occur at array sizes of approximately 2.5 PiB. In this example an array magnitude of 4 PiB was chosen so that the anomaly occurs within the variable range of the enclosure size. Similarly with GRAID 6x, the convergence of all three dual-levels begins to occur for array sizes of roughly 4.5 PiB. Varying the disk size or the disk type in each GRAID configuration can slightly raise and lower this threshold in which the reliability analysis breaks down. Approximations for when the MTTDL results become invalid are represented in Table 14.

Table 14: Maximum Array Sizes for GRAID 5x and 6x

| GRAID 5x | <i>500 GB Disk</i> | | GRAID 6x | <i>500 GB Disk</i> |
|------------------|--------------------|--|-------------------|--------------------|
| Disk Type | Array Size | | Array Size | Array Size |
| SATA | ~450 TiB | | SATA | ~2.0 PiB |
| SAS | ~1.1 PiB | | SAS | ~3.5 PiB |
| FC/SCSI | ~2.5 PiB | | FC/SCSI | ~4.5 PiB |

5.3.2. MTTDL Divergence

For small arrays, *e.g.*, less than 100 TiB, the ranking of GRAID levels has a logical ordering. From the lowest MTTDL to the highest MTTDL, dual-level GRAID ordering is as follows: 50, 60, 55, 56, 65, and 66. An example of this ordering for a 100 TiB array magnitude is illustrated in Figure 42 (left). However, as the array size increases, a divergence from this initial ordering occurs. What occurs is the GRAID levels begin to cluster based on their lowest level of redundancy, *i.e.*, GRAID 5x and 6x. The GRAID 6x levels retain higher MTTDL magnitudes while the GRAID 5x levels rapidly decrease in their reliability. An example of this divergence for a 3 PiB array is represented in Figure 42 (right). Shortly beyond this array magnitude the validity of the reliability analysis breaks down at the MTTDL juncture of all three GRAID 5x levels. However, at the 3 PiB scale, the new ranking of the GRAID levels (lowest to highest) is as follows: 50, 55, 56, 60, 65, and 66.

One of the potential causes for this divergence could be associated with the breakdown in the reliability analysis. For the 3 PiB array size, the MTTDL for each level has crossed below the MTBF of a single disk and rapidly approaches the group's MTTR threshold. As stated in the previous section, GRAID 5x levels encounter a MTTDL convergence well before the GRAID 6x levels do. Therefore, this divergence could be directly related to the breakdown in validity of the GRAID 5x levels. Another probable reason for the divergence of GRAID 5x and 6x levels could be related to the base levels of each, *i.e.*, 5 and 6, and how the number of disks required for an array magnitude can drastically effect the MTTDL of each. Since all three GRAID 5x configurations share a common RAID 5 weak link, this results in the MTTDL for each to decrease at a greater rate than GRAID 6x levels do. Consequently, a divergence is created.

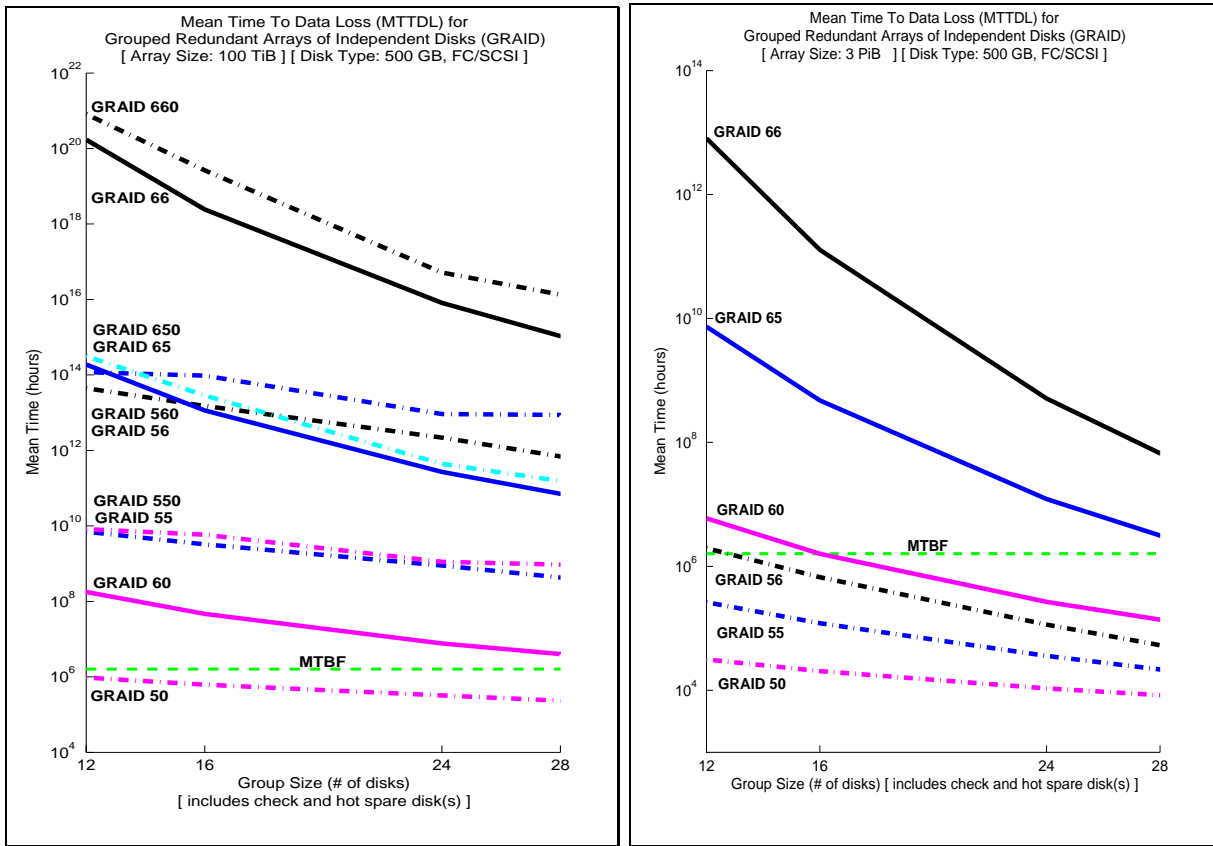


Figure 42: MDDTL Divergence from 100 TiB (left) to 3 PiB (right)

5.4. Recommended Designs

The following results show the recommended designs for array magnitudes ranging from 100 TiB to 100 EiB. If the MTTDL for a particular GRAID level is below that of an individual disk's manufacturer rated MTBF, then the configuration is not recommended (indicated by an x). The GRAID levels have been broken up into logical base level groupings, *i.e.*, 5x, 6x, 5x0, and 6x0. Only a single disk size (500 GB) and type (FC/SCSI) were used for the following recommendations. There are just too many potential combinations to list them all.

In Table 15 the recommended configurations for GRAID 5x are listed. Note that GRAID 50 does not satisfy the reliability requirements for any of the potential array magnitudes. As the desired array size grows beyond 1 PiB, there are no recommended configurations using these GRAID 5x levels. Therefore a GRAID 6x level must be utilized.

Table 15: Recommended Configurations for GRAID 5x Levels

| | 100 TiB | | | 1 PiB | | |
|----------------|----------|----------|----------|----------|----------|-------|
| | 50 | 55 | 56 | 50 | 55 | 56 |
| Enclosure Size | x | 48 | 48 | x | 16 | 36 |
| # Enclosures | x | 8 | 9 | x | 175 | 75 |
| # Racks | x | 2 | 2 | x | 14 | 13 |
| # Disks | x | 384 | 432 | x | 2800 | 2700 |
| Mean Failures | x | 8/yr | 9/yr | x | 53/yr | 51/yr |
| Efficiency (%) | x | 71.88 | 63.89 | x | 86.50 | 90.67 |
| | 10 PiB | 100 PiB | 1 EiB | 10 EiB | 100 EiB | |
| | 50/55/56 | 50/55/56 | 50/55/56 | 50/55/56 | 50/55/56 | |
| Enclosure Size | x | x | x | x | x | |
| # Enclosures | x | x | x | x | x | |
| # Racks | x | x | x | x | x | |
| # Disks | x | x | x | x | x | |
| Mean Failures | x | x | x | x | x | |
| Efficiency (%) | x | x | x | x | x | |

In Table 16 the recommended configurations for GRAID 6x are listed. As the desired array size grows beyond 10 PiB, there are no recommended configurations using these GRAID 6x levels. Therefore a tri-level GRAID configuration must be utilized.

Table 16: Recommended Configurations for GRAID 6x Levels

| | 100 TiB | | | 1 PiB | | | 10 PiB | | |
|----------------|----------|----------|----------|----------|-------|-------|--------|-------|-------|
| | 60 | 65 | 66 | 60 | 65 | 66 | 60 | 65 | 66 |
| Enclosure Size | 28 | 48 | 48 | 16 | 36 | 48 | 12 | 16 | 16 |
| # Enclosures | 10 | 8 | 9 | 186 | 76 | 57 | 2687 | 1862 | 1863 |
| # Racks | 2 | 2 | 2 | 15 | 13 | 12 | 135 | 144 | 144 |
| # Disks | 280 | 384 | 432 | 2976 | 2736 | 2736 | 32244 | 29792 | 29808 |
| Mean Failures | 6/yr | 8/yr | 9/yr | 56/yr | 51/yr | 51/yr | 2/day | 2/day | 2/day |
| Efficiency (%) | 89.29 | 70.31 | 62.50 | 81.25 | 89.25 | 88.82 | 75.00 | 81.16 | 81.12 |
| | 100 PiB | 1 EiB | 10 EiB | 100 EiB | | | | | |
| | 60/65/66 | 60/65/66 | 60/65/66 | 60/65/66 | | | | | |
| Enclosure Size | x | x | x | x | | | | | |
| # Enclosures | x | x | x | x | | | | | |
| # Racks | x | x | x | x | | | | | |
| # Disks | x | x | x | x | | | | | |
| Mean Failures | x | x | x | x | | | | | |
| Efficiency (%) | x | x | x | x | | | | | |

In Table 17 the recommended configurations for GRAID 5x0 are listed. Note that GRAID 550 is only recommended for array magnitudes up to 100 PiB. On the other hand, all the GRAID 560 configurations are recommended for array sizes up to 100 EiB, the maximum evaluated size. However, the validity of this recommendation should not go unquestioned. The physical space requirements and monetary funds required to first build and then maintain such a colossal array is almost inconceivable. To increase the expected reliability even further, a tri-level GRAID 6x0 configuration can be utilized.

Table 17: Recommended Configurations for GRAID 5x0 Levels

| | 100 TiB | | 1 PiB | | 10 PiB | | 100 PiB | |
|------------------------|---------|---------|--------|----------|---------|-----------|---------|-----------|
| | 550 | 560 | 550 | 560 | 550 | 560 | 550 | 560 |
| Enclosure Size | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Enclosures/Rack | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| # Racks | 2 | 2 | 14 | 15 | 135 | 143 | 1344 | 1423 |
| # Disks | 480 | 480 | 3360 | 3600 | 32400 | 34320 | 322560 | 341520 |
| Mean Failures | 9/yr | 9/yr | 63/yr | 68/yr | 2/day | 2/day | 17/day | 18/day |
| Efficiency (%) | 75.00 | 70.83 | 75.00 | 70.83 | 75.00 | 70.83 | 75.00 | 70.83 |
| | 1 EiB | | 10 EiB | | 100 EiB | | | |
| | 550 | 560 | 550 | 560 | 550 | 560 | 550 | 560 |
| Enclosure Size | x | 12 | x | 12 | x | 12 | x | 12 |
| Enclosures/Rack | x | 20 | x | 20 | x | 20 | x | 20 |
| # Racks | x | 14565 | x | 145641 | x | 1456401 | x | 1456401 |
| # Disks | x | 3495600 | x | 34953840 | x | 349536240 | x | 349536240 |
| Mean Failures | x | 8/hr | x | 75/hr | x | 743/hr | x | 743/hr |
| Efficiency (%) | x | 70.83 | x | 70.83 | x | 70.83 | x | 70.83 |

In Table 18 the recommended configurations for GRAID 6x0 are listed. Note that all the GRAID 6x0 configurations are deployed for array sizes up to 100 EiB, the maximum evaluated size. Once again, the validity of these 100 EiB recommendations seems far fetched with current technology. Based on the mean number of disks which are expected to fail annually (AFR), approximately \$300 thousand dollars will be spent, every hour, to maintain a constant supply of disk drives to replace the failed ones. However, as disk drives become more reliable and the demand for storage rises, Exbibyte sized storage arrays will eventually become practical.

Table 18: Recommended Configurations for GRAID 6x0 Levels

| | 100 TiB | | 1 PiB | | 10 PiB | | 100 PiB | |
|----------------------------|---------|-------|-------|-------|--------|-------|---------|--------|
| | 650 | 660 | 650 | 660 | 650 | 660 | 650 | 660 |
| Enclosure Size | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| Enclosures per Rack | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| # Racks | 2 | 2 | 15 | 16 | 150 | 159 | 1493 | 1581 |
| # Disks | 480 | 480 | 3600 | 3840 | 36000 | 38160 | 358320 | 379440 |
| Mean Failures | 9/yr | 9/yr | 68/yr | 72/yr | 2/day | 2/day | 19/day | 20/day |
| Efficiency (%) | 67.50 | 63.75 | 67.50 | 63.75 | 67.50 | 63.75 | 67.50 | 63.75 |

| | 1 EiB | | 10 EiB | | 100 EiB | |
|----------------------------|---------|---------|----------|----------|-----------|-----------|
| | 650 | 660 | 650 | 660 | 650 | 660 |
| Enclosure Size | 12 | 12 | 12 | 12 | 12 | 12 |
| Enclosures per Rack | 20 | 20 | 20 | 20 | 20 | 20 |
| # Racks | 15284 | 16183 | 152833 | 161823 | 1528322 | 1618223 |
| # Disks | 3668160 | 3883920 | 36679920 | 38837520 | 366797280 | 388373520 |
| Mean Failures | 8/hr | 9/hr | 78/hr | 83/hr | 780/hr | 826/hr |
| Efficiency (%) | 67.50 | 63.75 | 67.50 | 63.75 | 67.50 | 63.75 |

5.5. Analysis of Reliability Variables

As the magnitude of storage systems becomes larger and larger, the required number of disks will proportionally increase. One interesting finding is that as the enclosure size increases, the MTDDL will frequently intersect with the MTTR. In some circumstances, e.g., large magnitude arrays composed of lower class PATA and SATA drives, the MTDDL is always lower than the MTTR. Any system design which operates below this crossover point ($MTDDL \leq MTTR$) should be avoided at all costs. These configurations are unstable and would theoretically never be able to successfully rebuild a degraded disk before another failure is expected to occur. Therefore, large-scale arrays must be designed using enclosure groups to keep the MTTR low and drive the MTDDL up, ideally resulting in a more reliable system, *i.e.*, $MTDDL \gg MTTR$ and $MTDDL \geq MTBF$. The following sections analyze the various components of an array which are variable, and how this variance affects the overall reliability.

5.5.1. Varying the MTBF

The variance of the MTBF is primarily based on the class of disk drive, *i.e.*, PATA, SATA, SAS, or FC/SCSI. Each has their own corresponding MTBF rating provided by the manufacturer which should not be taken literally. Looking back at the case study findings of Google Labs, their results indicated that the drive type has little effect on the expected MTBF. Instead all disks will generally have around the same MTBF rating, on average ~292 khrs (AFR=3%). However, the source of their results only included data for two classes of disk drives, all of which could be considered as desktop class drives, *i.e.*, PATA and SATA. On the other hand, the Carnegie Mellon case study collected data on a much broader range of disk drive types. However, their findings did not back this conclusion by Google in which all drives are equal, no matter what their class. The one common finding between the two studies was that manufacturer ratings are almost never accurate. On average, most of the failure rates were 3.4 times higher than expected. Therefore, this 3.4 factor is included in the design of the GRAID Calculator tool and used in determining the overall MTTDL for each GRAID level.

The primary way to vary the MTBF in different GRAID configurations is to simply select a different disk type. The manufacturer's MTBF rating starts at 1 million hours for PATA drives and increments by 200 khrs for each subsequent disk class, up to 1.6 million hours for a FC/SCSI disk. As the array magnitude increases, the effect of a change in MTBF can greatly lower or increase the expected reliability. An illustration of this variance is portrayed in Figure 43 for the GRAID 6x levels. Here the MTBF of a FC/SCSI disk is varied by ± 20 and $\pm 40\%$. Note how the variance in the MTBF affects the MTTDL outcome greater as the enclosure size increases. This is partially due to the growing disk MTTR rate with larger enclosure sizes. From the analysis, it is clear that selecting a higher class disk drive not only brings with it a higher reliability rating, but also provides faster media rates lessening the MTTR. In addition, the higher BER rating of enterprise FC/SCSI disks can improve the expected reliability of the storage array.

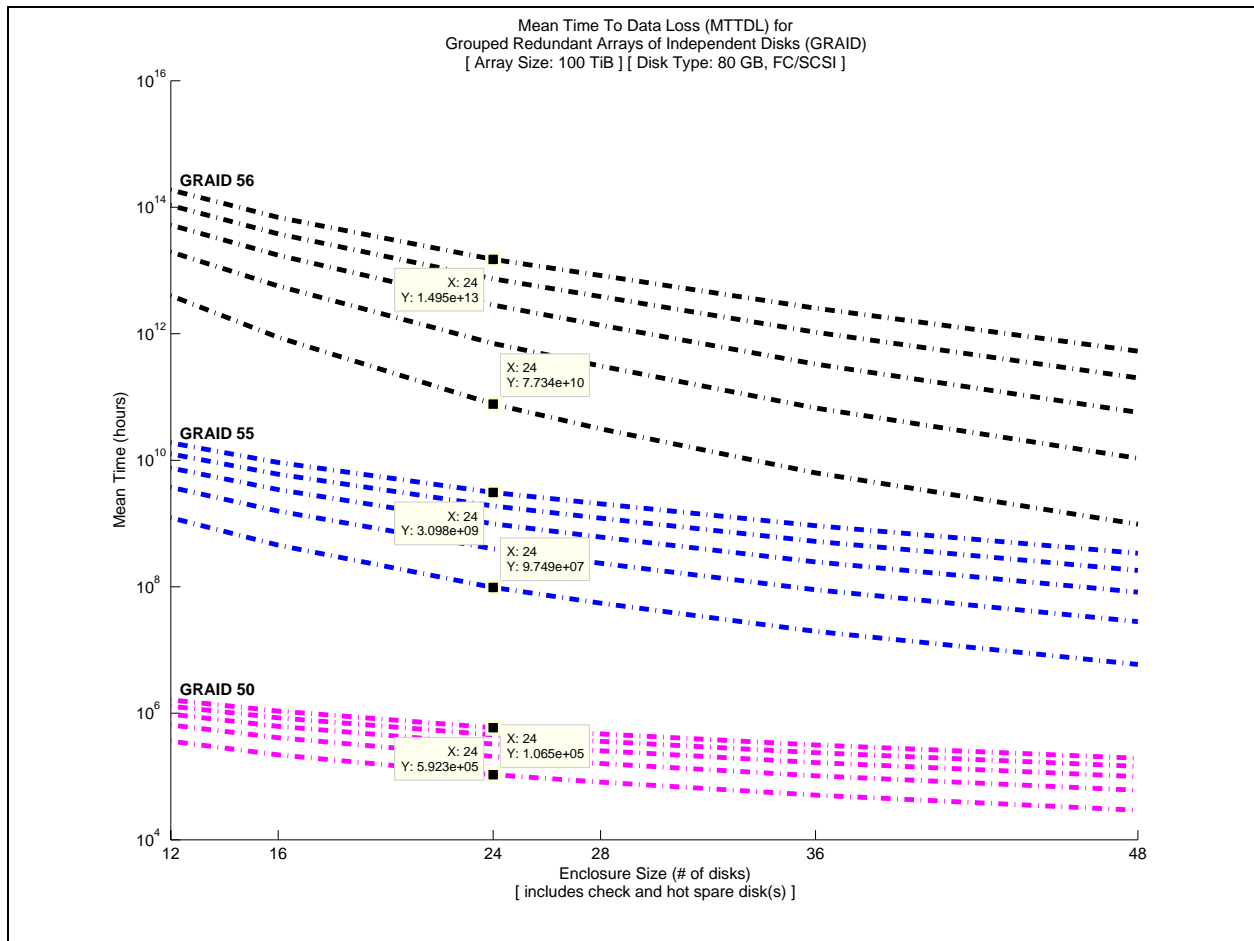


Figure 43: Effect of MTBF on the MTTDL of GRAID 6x

5.5.2. Varying the MTTR

Large-scale arrays with disk enclosures and multiple enclosures results in high rebuild times. The use of greater capacity disk drives also results in more time needed to rebuild data if one drive fails. Since storage arrays can suffer from heavy processor loading during the rebuild process, it's highly likely that the failure of another drive during these extended rebuild times will occur.

One of the primary factors to consider with rebuild times is the rebuild priority (P). Three commonly used priorities used in RAID systems are 20% (low), 50% (medium), or 80% (high). Rebuild priorities less than 20% are usually not recommended. This is because any prolonged

time that a storage system is in a state of degradation, the higher the probability that another failure will occur, possibly resulting in data loss. In storage systems with constant I/O activity, the rebuild process would proceed at a negligible rate. With a low priority rebuild, a RAID controller will serve any host I/O activities first and foremost [HP05b]. If the RAID system were to rebuild at a higher priority (80%), this will ideally allow the system to quickly return to an optimal state. However, this higher prioritization allocates system resources to the rebuild process foremost. Consequently, this takes away from any host I/O requests needed in highly used storage systems [HP05b]. As the baseline for analysis, the rebuild priority of 50% is used for all the multi-GRAID analysis results. When selecting just a single GRAID level in the Reliability tool, then all three priority values are used to show how it affects the MTDDL. A rebuild priority of 50% would allow the array to be rebuilt at a modest rate while still catering to any host I/O requests.

Previous MTTR calculations used by [PATT88], [TREA03], and others found that this mean time variable affected the MTDDL outcome by a negligible amount. Therefore, the MTTR value was usually set to be a fixed time in hours. This value reflected the service response time, e.g., 4 to 24 hours; required by a technician to replace the failed drive and regenerate the missing data. When the array size under evaluation is small scale, *i.e.*, only a few terabytes in magnitude, this still holds true. However, for large-scale storage systems this mean time to recovery, previously considered to be insignificant, becomes a critical factor in determining the overall reliability. The MTTR is better represented if the number of disks in an enclosure, needed to rebuild the failed disk, were taken into consideration. This new MTTR element has been proven effective at better estimating the recovery times necessary for a given enclosure size.

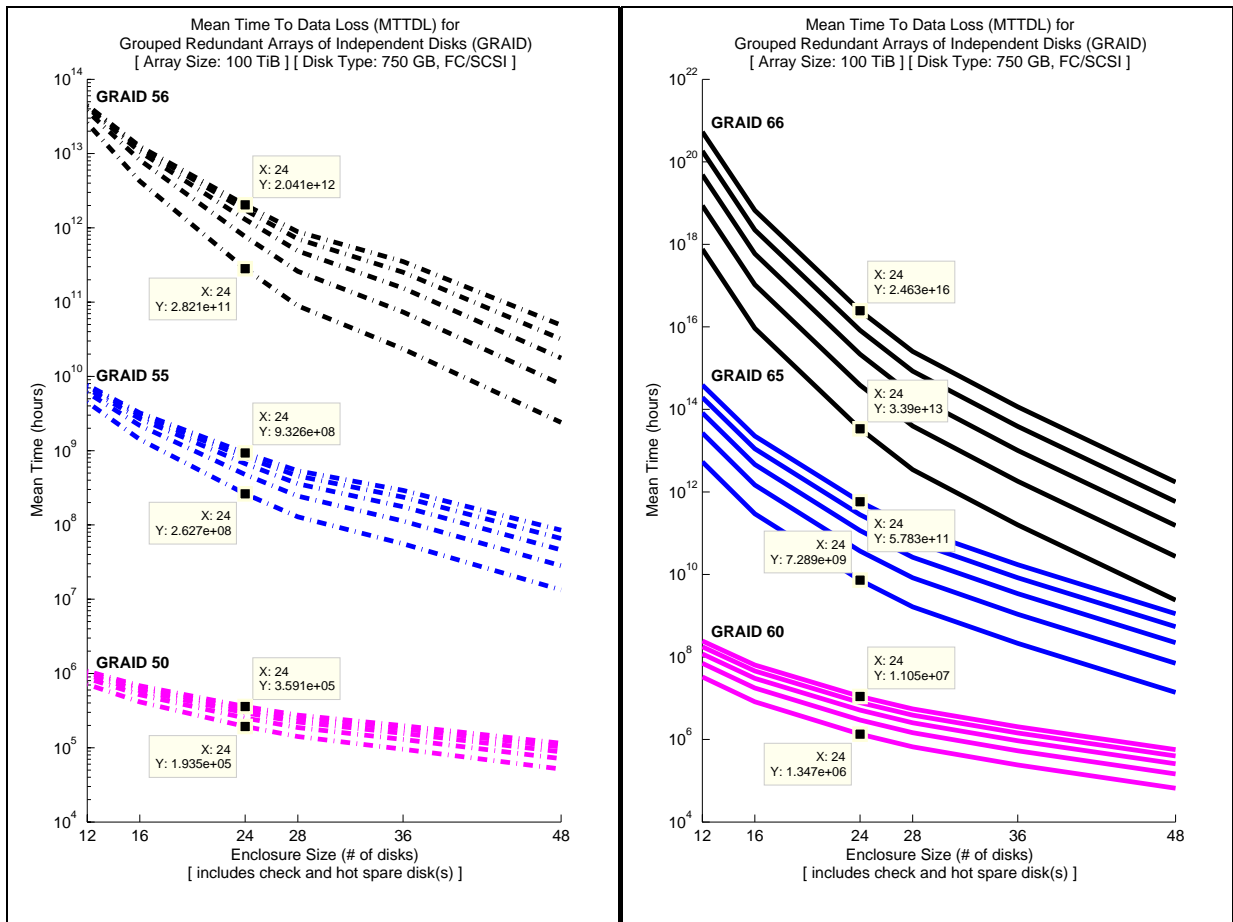


Figure 44: Effect of MTTR on the MTTDL of GRAID 5x (left) and 6x (right)

The results in Figure 44 illustrate how the MTTR can affect the reliability of the system. In these examples, the media rate M of the SCSI/FC disks were varied by ± 20 and $\pm 40\%$. Note how the variance in MTTR for the higher reliable GRAID 66 level is greatly affected by this variance. As the reliability for each of the levels decreases, the effect of MTTR of the MTTDL is greatly muted. As stated above in the MTBF analysis, selecting a higher class disk drive not only brings with it a higher reliability rating, but also provides faster media rates which reduce the MTTR. As disk drives become larger and larger, the media rates need to keep up. Otherwise, rebuild times will exponentially grow with the rate of storage increase.

5.5.3. Varying the BER

The bit error rate of disk drives is often overlooked in the reliability analysis of disk arrays. Therefore, as array magnitudes increase, the impact of unrecoverable bit errors is greatly magnified. Depending on the rate in which data is being read/written to a storage array, it is not uncommon for these errors to frequently occur somewhere in a system [XIN03]. It's a reality that must be taken seriously or the potential likelihood of data loss will be heightened. The disk class will determine what the BER will be and it is primarily driven by the quality of the media used. Since higher end disk drives, demanded by enterprise customers, need to be highly reliable and fail less frequently than the lower class drives, higher quality components are typically incorporated into these disks, resulting in higher costs.

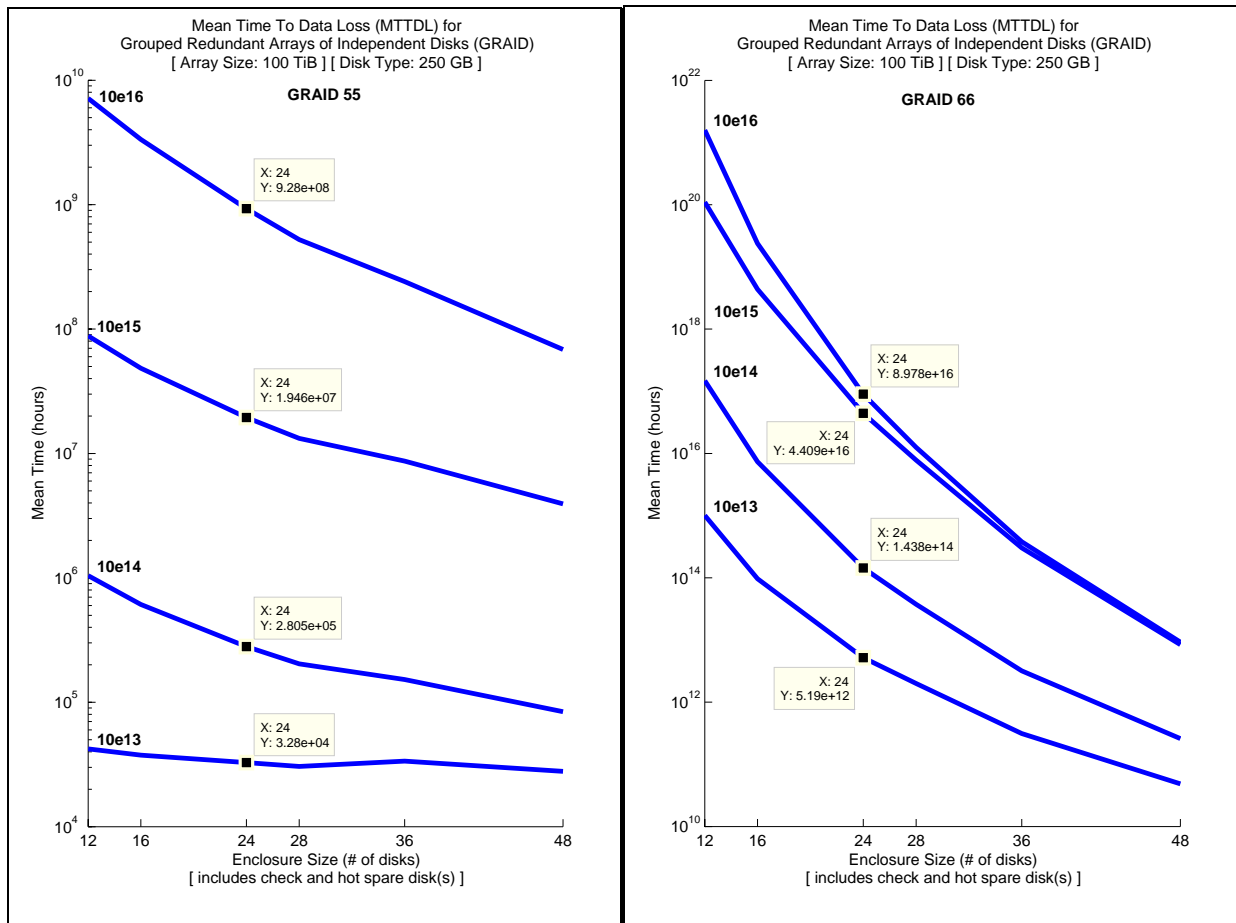


Figure 45: Effect of the BER on MTTDL for GRAID 55 (left) and 66 (right)

In the following example, shown in Figure 45, a RAID 55 and RAID 66 configuration are compared side-by-side to demonstrate the impact in which the BER has on expected reliability. In each case, the BER ranges from $1:10^{13}$ (PATA class disk drives) up to $1:10^{16}$ (FC/SCSI class disk drives). For the RAID 55 case, the difference between MTDL values, if using 24-disk enclosures, is 1 billion (1×10^9). Similarly in the RAID 66 example, variance between high and low bit error rates can be as large as 9×10^{16} . This exceedingly larger variance in the RAID 66 case is primarily due to its extraordinarily high MTDL ratings. As the array magnitudes increase, the variances between these results will gradually diminish.

5.5.4. Varying the Disk Size

One key advantage to using higher capacity disk drives over smaller ones is the lower number of failures expected per year, on average. This is derived from the AFR and the total number of disks required to implement a given storage array magnitude (includes data, check (parity), and hot spare disks). As expected, the use of larger disks allow for storage arrays to be implemented using fewer disks, compared with an equivalent sized array using smaller disks. Take for example two equivalent capacity arrays of size 100TiB, “Array A” and “Array B”. Array A is constructed using 250GB disk drives and Array B using 500GB disks. By using disks in Array B which are double the size of those in Array A, this approximately halves the expected number of disk failures per year in Array B. This shows why it is important to periodically upgrade large-scale systems with higher capacity disks. Doing so will produce a more reliable system, increase the storage capacity, and introduce device diversity.

5.5.5. Varying the Enclosure Size

Varying the enclosure size between its smallest configuration and a maximum practical value - *i.e.*, the largest enclosure capacity - is one of the best ways of visualizing the ideal enclosure range. Since a redundant array should ideally have a higher reliability than that of a single disk drive, this introduces a reliability threshold. When the MTDL of a RAID configuration crosses below the MTBF of a single disk, enclosure sizes larger than this crossover

value are not recommended. This is not a hard limit, but rather a guideline. Choosing an enclosure size lower than the point where the MTDDL intersects with the disk manufacturer's MTBF is recommended. Doing so will increase the reliability of the system. However, increasing the enclosure size beyond the MTBF threshold is not recommended either. The reliability is drastically reduced with these additions.

The second reason for varying the enclosure size, and probably the most important one, is to accomplish rack optimization. When dealing with storage arrays in the pebibyte and exbibyte range, physical space optimization should play a big role in decision making. Take for example the five rack enclosures portrayed in Figure 46. From right to left, the disk enclosure height is incremented by one rack unit (1U) up to height of 5U. A rack unit (U) is equivalent to 1.65 inches in height. This unit of measurement is standardized by the Electronics Industries Alliance (EIA) to quantify equipment heights in rack enclosures. The standard width for rack enclosures is 19 inches. Taking into account these fixed dimensions for disk and rack enclosures, and the fixed dimension of the individual disk drive, there are certain U sizes which can optimally accommodate more disks than others.

In Figure 46, the rack enclosure on the far right is constructed using 40 x 1U disk enclosures, each capable of housing up to 4 drives. Therefore, a total of 160 disk drives can be housed in a single rack using 1U enclosures. The same principle was applied to each incremental enclosure height up to a maximum height of 5U. Disk enclosures 6U and 8U in height are not commonly manufactured. However, these can be formed by grouping together two 3U or two 4U enclosures, respectively. All of the 42U height racks in this example have 2U of overhead reserved for miscellaneous equipment, *e.g.*, power conditioners or network switches. Looking at the total number of disk drives possible in each configuration, the even U rack heights, *i.e.*, 2U and 4U, can support the most number of drives (240 disks per rack). With four drives across and three to six drives in height, depending on a 2U or 4U enclosure, this arrangement of drives can optimally fill out a disk and rack enclosure. These enclosure sizes are used with the GRAID Calculator tool so that a full enclosure size is selected for a given array magnitude. Therefore, preference should be given to enclosure sizes of 12 disks or 24 disks, due to their rack optimization.

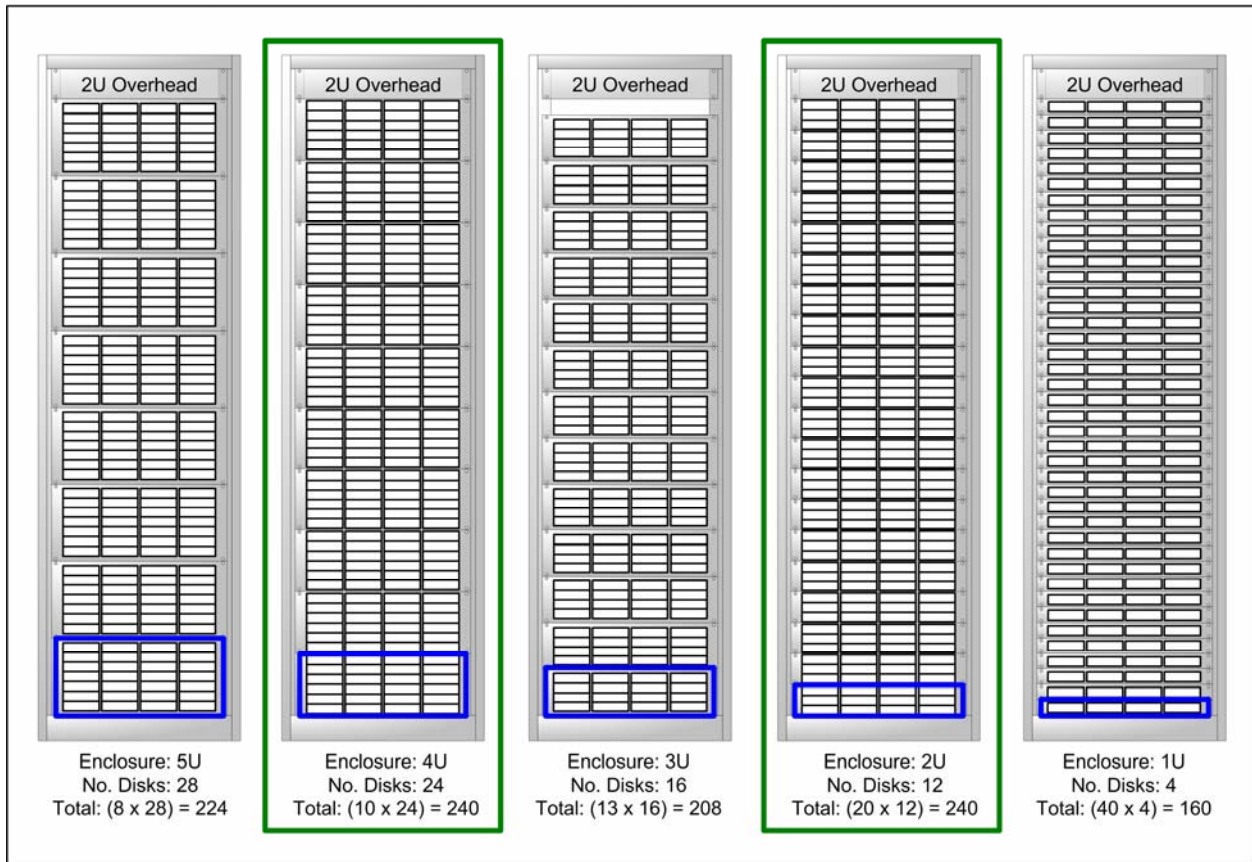


Figure 46: Rack Enclosure Optimization

5.6. Methods for Improving/Upholding Reliability

The various GRAID configurations just analyzed can all provide different levels of reliability. However, with every improvement there are consequential tradeoffs associated with each. Nevertheless, for any new or existing storage array there are numerous options available to improve or promote overall storage reliability. The following section will provide a brief overview on each of the following methods: use of hot spare disks, introduction of device diversity, and the use of RAID-Z under the Zettabyte Filesystem.

5.6.1. Hot Spare Disks

The use of hot spare disks in large-scale storage arrays is essential. Without them, a technician, or team of technicians, would be running around the server room non-stop replacing failed disk drives left and right. This would also inadvertently introduce the worse element in

any system's reliability, the human element. It is very common for technicians to eject working disks rather than failed ones while attempting to fix the problem. In turn, they just create another by potentially failing a system, unless added redundancy is in place. By using hot spares, the enclosures can detect any failure and immediately activate the hot spare to begin the rebuild process. The added benefit of this is a reduction in MTTR, i.e., no lengthy response time for the technician to physically replace a drive. This also reduces the window of time in which an array is in the degraded state, which consequentially improves reliability. Therefore until a failure in an enclosure occurs, the hot spare disk is powered on just like the others, but it does not actively participate in any data storage. It is also important to note that hot spare drives should be taken into account when calculating the mean number of disk drives which are expected to fail annually (based on the AFR). Although hot spare disks are mostly waiting for other disks to fail, the likelihood of itself failing is about the same as any other active disk in the array. Disk drive access pattern tests performed by Google Lab's case study showed that infrequently accessed disks are not immune from failure. In some cases those inactive disks failed more frequently.

5.6.2. Device Diversity

The introduction of device diversity in storage systems can be very beneficial to its overall reliability [PARI06]. Device diversity involves the building of storage enclosures using disk drives from different production batches and/or different manufacturers. Doing so can protect against encountering common defects frequently found in batches of disk drives. These common defects can frequently cause correlated disk failures. By incorporating four different batches of disk drives into a storage array, the probability of encountering correlated disk failures, due to a common media defect can be greatly reduced. Even the integration of two different disk batches can modestly lower the probability of encountering correlated disk failures. Therefore, device diversity should be implemented in any large-scale storage array to defend against any batch-correlated disk failures [PARI06].

5.6.3. RAID-Z

As mentioned before, filesystems play a big part in the growth and expansion of large-scale storage systems. The existence of a 100 EiB array in the above analysis could not exist under any

current filesystems format. The Zettabyte File System (ZFS) being developed and maintained by Sun Microsystems uses 128-bit addressing yet still only has the capability of managing volume or single file sizes of around 16 EiB [SUN04]. An array magnitude measured in EiB seems so immense by today's standards. However, as the demand for more storage grows, the number of bits necessary to address these storage pools will need to be 256-bits.

One interesting aspect of ZFS is its intelligent integration of multiple redundancy schemes. Under the current implementation of ZFS there exist equivalents to RAID 5 and RAID 6, aptly named RAID-Z and RAID-Z2, respectively. These single and double parity implementations under ZFS are equivalent to that of a RAID 5 or RAID 6 configuration, only better. In the case of RAID-Z, this design uses a copy-on-write strategy instead of the typical RAID 5 read-modify-write policy. Doing so avoids the "write hole", frequently encountered when old data is overwritten with new data. The copy-on-write strategy used with RAID-Z will write new data to an empty space and then automatically update the pointer to its new location. Therefore, small writes which previously required a read-modify-write operation, will instead be processed as a full-stripe write. To further improve efficiency and performance, ZFS can intelligently mirror small blocks instead of wasting resources with parity calculations [SUN04]. This uniform integration of filesystem and redundancy make ZFS a promising technology for use in present and future large-scale storage systems.

CHAPTER 6

6. CONCLUSION

The reliable storage of data will always be an essential component in mission-critical environments. With the use of multi-level, highly fault-tolerant GRAID configurations, system downtime due to disk failures can be significantly minimized. The need for calculating overall reliability is considered essential when comparing different GRAID configurations. As shown in the analysis, doing so brought to attention one of the fundamental limitations for MTTDL $[(MTTDL/(N_D+C_D)] \gg MTTR$). Any reliability comparison beyond a MTTDL convergence point no longer holds any validity. However, as long as the GRAID designs perform above this threshold, the results still provide an excellent representation of just how reliable one approach is compare to another.

Large-scale storage arrays will continually be needed by universities, government agencies, web search engines, and research laboratories. As research laboratories rapidly generate petabytes of annual data, their storage arrays will need to be highly scalable and dependable in order to preserve continuous access to a dynamic stream of data. Designing storage arrays, requiring thousands to millions of disk drives, still represents a serious challenge in terms of reliability. As a guiding principle, the overall reliability of large-scale storage arrays should be greater than that of a single disk, *i.e.*, its MTBF. If a MTTDL trend slopes below this threshold, storage systems can become unstable very quickly. The failure of multiple disks during a rebuild also poses a serious threat to massive storage systems. However, with the use of hot spare disk drives and the integration of device diversity, many correlated disk failures can be avoided. The increased frequency of unrecoverable bit errors in large-scale storage arrays (amplified by the amount of data read during a rebuild) can be one of the most limiting factors of a GRAID's reliability. By decreasing the BER, *i.e.*, improving the disk class, storage arrays can greatly benefit from this added reliability.

This thesis addresses the design issues and limitations involved with large-scale storage arrays using current and future storage technology. It is partially based upon previous research of RAID reliability calculations done by [PATT88] and [CHEN94]. However, the addition of dual-

and tri-level RAID configurations is necessary in the design of large-scale arrays. Various design recommendations for array magnitudes (ranging from 100 TiB up to 100 EiB) are provided based on the analysis of these MTDL reliability metrics. These recommendations can be quickly generated using the custom RAID Reliability Calculator tool written in MATLAB. This tool, specifically written for the large-scale storage arrays, has many user selectable/changeable parameters, including: the array size, disk type, disk size, and redundancy level(s).

The dependency between each of the variables which affect the reliability of data storage was thoroughly analyzed and then incorporated into the reliability calculator tool. In doing so, better design recommendations are consequentially provided. Recent case studies concerning the actual reliability of disk drives by Google Labs and Carnegie Mellon show that manufacturer's reliability ratings are on average 3.4 times higher than those seen in actual storage environments. This reduction factor is also incorporated into the reliability calculations to better reflect the expected disk failure rate. The last variable incorporated into the new reliability analysis is the more realistic MTTR. This was previously assumed to be a fixed value. A better representation was implemented to take into account the number of drives needed to be read from during a disk rebuild, *i.e.*, the enclosure size.

It is important to understand that the design recommendations provided by the RAID Reliability Calculator tool should not be taken literally. Since the reliability analysis uses the MTDL, which is based upon the statistical assumption that all disk failures are independent, it allows for a much simpler analysis. Alternatively, as the other system components are taken into consideration, the system complexity increases rapidly. The reliability analyses of MTDL for the RAID levels outlined in this thesis are meant to establish the feasibility of creating such massive storage arrays. Based on the analysis, it shows that repair time, disk class, and the sheer number of disks necessitated in these arrays are the critical limiting factors. Some of the main components required in the construction of storage arrays include: disk enclosures, network switches, power distribution blocks, and data access servers. For future research on this topic, these other critical system components should be incorporated into the overall system reliability. Doing so will provide even more accurate design configurations.

APPENDIX A - Acronyms

| | |
|-----------------|--|
| AFR | Annualized Failure Rate |
| ARR | Annual Replacement Rate |
| ATA | Advanced Technology Attachment |
| BER | Bit Error Rate |
| CDF | Correlated Disk Failure |
| DF | Disk Failure |
| EB | Exabyte |
| ECC | Error Correction Coding |
| EiB | Exbibyte (exa binary byte) |
| EIA | Electronics Industries Alliance |
| FAT | File Allocation Table |
| GB | Gigabyte |
| GiB | Gibibyte (giga binary byte) |
| GRAID | Grouped Redundant Arrays of Independent Disks |
| HA | Highly Available |
| HFS+ | Hierarchical File System Plus |
| I/O | Input/Output |
| KB | Kilobyte |
| KiB | Kibibyte (kilo binary byte) |
| LV | Logical Volume |
| LVM | Logical Volume Manager |
| MB | Megabyte |
| MiB | Mebibyte (mega binary byte) |
| MIL-HDBK | Military Handbook for Reliability Prediction of Electronic Equipment |
| MTBF | Mean Time Between Failures |
| MTTDL | Mean Time To Data Loss |
| MTTR | Mean Time To Recovery |
| NTFS | New Technology File System |
| OS | Operating System |

| | |
|--------------|--|
| PATA | Parallel ATA |
| PB | Petabyte |
| PiB | Pebibyte (pe ta bi nary by te) |
| POH | Power-On Hours |
| RAID | Redundant Array of Independent Disks |
| RDT | Reliability-Demonstration Test |
| SAS | Serial Attached SCSI |
| SATA | Serial ATA |
| SCSI | Small Computer System Interface |
| SLED | Single Large Expensive Disk |
| TB | Terabyte |
| Tbpsi | Terabits per square inch |
| TiB | Tebibyte (te ra bi nary by te) |
| UBE | Unrecoverable Bit Error |
| VG | Volume Group |
| YB | Yottabyte |
| YiB | Yobibyte (yo ttta bi nary by te) |
| ZB | Zettabyte |
| ZFS | Zettabyte File System |
| ZiB | Zebibyte (ze ttta bi nary by te) |

APPENDIX B - MTTDL Derivations

The following equations represent the basic derivations of the Mean Time to Data Loss due to a Disk Failure for dual- and tri-level RAID equations. The simplified representation of there equations was introduced in Section 4.5.3.1 and is used in the RAID Reliability Calculator.

Dual-Level (RAID 50): (65)

$$\begin{aligned}
 MTTDL_{DF} &= \frac{MTBF_{Disk}^2}{N_D \times N_{E,2} \times (N_D - 1) \times MTTR_{Disk}} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,2}} \right\rangle
 \end{aligned}$$

Dual-Level (RAID 55): (66)

$$\begin{aligned}
 MTTDL_{DF} &= \frac{MTBF_{Disk}^4}{N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (N_{E,2} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,2}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,2} - 1) \times MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Dual-Level (RAID 56): (67)

$$\begin{aligned}
 MTTDL_{DF} &= \frac{MTBF_{Disk}^6}{N_D^3 \times N_{E,2} \times (N_D - 1)^3 \times (N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,2}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,2} - 1) * MTTR_{Enclosure}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,2} - 2) * MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Dual-Level (GRAID 60):

(68)

$$\begin{aligned}
 MTTDL_{DF} &= \frac{MTBF_{Disk}^3}{N_D \times N_{E,2} \times (N_D - 1) \times (N_D - 2) \times MTTR_{Disk}^2} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,2}} \right\rangle
 \end{aligned}$$

Dual-Level (GRAID 65):

(69)

$$\begin{aligned}
 MTTDL_{DF} &= \frac{MTBF_{Disk}^6}{N_D^2 \times N_{E,2} \times (N_D - 1)^2 \times (N_D - 2)^2 \times (N_{E,2} - 1) \times MTTR_{Disk}^4 \times MTTR_{Enclosure}} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,2}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,2} - 1) \times MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Dual-Level (GRAID 66):

(70)

$$\begin{aligned}
 MTTDL_{DF} &= \frac{MTBF_{Disk}^9}{N_D^3 \times N_{E,2} \times (N_D - 1)^3 \times (N_D - 2)^3} \\
 &\quad \times \frac{1}{(N_{E,2} - 1) \times (N_{E,2} - 2) \times MTTR_{Disk}^6 \times MTTR_{Enclosure}^2} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,2}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,2} - 1) \times MTTR_{Enclosure}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,2} - 2) \times MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Tri-Level (GRAID 550):

(71)

$$\begin{aligned}
 MTDL_{DF} &= \frac{MTBF_{Disk}^4}{N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times (N_{E,3} - 1) \times MTTR_{Disk}^2 \times MTTR_{Enclosure}} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,3}} \right\rangle * \left\langle \frac{1}{N_R} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,3} - 1) \times MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Tri-Level (GRAID 560):

(72)

$$\begin{aligned}
 MTDL_{DF} &= \frac{MTBF_{Disk}^6}{N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3 \times (N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^3 \times MTTR_{Enclosure}^2} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,3}} \right\rangle * \left\langle \frac{1}{N_R} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,3} - 1) * MTTR_{Enclosure}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,3} - 2) * MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Tri-Level (GRAID 650):

(73)

$$\begin{aligned}
 MTDL_{DF} &= \frac{MTBF_{Disk}^6}{N_D^2 \times N_{E,3} \times N_R \times (N_D - 1)^2 \times (N_D - 2)^2 \times (N_{E,3} - 1) \times MTTR_{Disk}^4 \times MTTR_{Enclosure}} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,3}} \right\rangle * \left\langle \frac{1}{N_R} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,3} - 1) \times MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

Tri-Level (GRAID 660):

(74)

$$\begin{aligned}
 &MTTDL_{DF} \\
 &= \frac{MTBF_{Disk}^9}{N_D^3 \times N_{E,3} \times N_R \times (N_D - 1)^3 \times (N_D - 2)^3} \\
 &\quad \times \frac{1}{(N_{E,3} - 1) \times (N_{E,3} - 2) \times MTTR_{Disk}^6 \times MTTR_{Enclosure}^2} \\
 &= \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{N_{E,3}} \right\rangle * \left\langle \frac{1}{N_R} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,3} - 1) \times MTTR_{Enclosure}} \right\rangle \\
 &* \left[\frac{MTBF_{Disk}}{N_D} * \frac{MTBF_{Disk}}{(N_D - 1) \times MTTR_{Disk}} * \frac{MTBF_{Disk}}{(N_D - 2) \times MTTR_{Disk}} \right] * \left\langle \frac{1}{(N_{E,3} - 2) \times MTTR_{Enclosure}} \right\rangle
 \end{aligned}$$

APPENDIX C - RAID Reliability Calculator

MATLAB Version Requirement: > v7.0.x

MATLAB Code:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Reliability Calculator for Large Scale Storage Arrays
% Using Grouped Redundant Arrays of Independent Disks (RAID)
%
% By: Michael McDonald
% Date: June 1, 2007
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
close all
clear all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf('\n')
fprintf('      Reliability Calculator      \n')
fprintf(' for Large-scale Storage Arrays    \n')
fprintf('----- \n')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k = menu('Select an Array Size',...
        '100 TiB', '1 PiB', '10 PiB', '100 PiB', '1 EiB', ...
        '10 EiB', '100 EiB');
A_name = ['100 TiB'; '1 PiB '; '10 PiB '; '100 PiB'; '1 EiB  '; ...
        '10 EiB  '; '100 EiB'];
A_size = [100*2^40, 1*2^50, 10*2^50, 100*2^50, 1*2^60, ...
        10*2^60, 100*2^60];

ArrayName = A_name(k,:);
ArraySize = A_size(k);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k = menu('Select a Disk Type',...
        'SATA', 'SAS', 'FC/SCSI');
D_type = ['SATA  '; 'SAS  '; 'FC/SCSI']; % type of disk drive
D_mtbf = [1.2e6, 1.4e6, 1.6e6]; % MTBF in (hours)
D_m = [300e6, 300e6, 320e6]; % media rate in (B/s)
D_ber = [10^14, 10^15, 10^16]; % bit error rate (1 in X bits)

POH = 8760; % Power on hours 365 days * 24 hours/day
DiskType = D_type(k,:); % type of disk drive
MTBF_mfgr = D_mtbf(k); % mfgr MTBF in (hours)
MTBF = D_mtbf(k) ./ 3.4; % actual MTBF in (hours)
AFR = 1 / ( MTBF / POH ); % annualized failure rate (AFR)
M = D_m(k); % media rate in (B/s)
BER = D_ber(k); % bit error rate (1 in X bits)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k = menu('Select a Disk Size',...
        '80 GB', '250 GB', '500 GB', '750 GB', '1 TB');
```

```

D_name = [80,      250,      500,      750,      1000]; % disk size in GB

DiskName = D_name(k);
% convert (GB) to binary value (GiB)
DiskSize = DiskName .* ( (10^9)^2 / 2^30 );
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rebuild Priority (ii loop)
P_name = ['Low ( 20% ) '; 'Med ( 50% ) '; 'High ( 80% )'];
P_value = [      0.2,      0.5,      0.8      ];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k = menu('Select GRAID Level',...
        'GRAID 50','GRAID 55',...
        'GRAID 56','GRAID 60',...
        'GRAID 65','GRAID 66',...
        'GRAID 5x','GRAID 6x',...
        'GRAID xx',...
        'GRAID 550','GRAID 560',...
        'GRAID 650','GRAID 660',...
        'GRAID 5x0','GRAID 6x0',...
        'GRAID xx0','ALL');
levels = {1,2,3,4,5,6,...
          [1:3],[4:6],[1:6],...
          11,12,13,14,...
          [11:12],[13:14],...
          [11:14],[1:6,11:14]};
level = levels{k};
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Selection Summary
fprintf(' Array Size = %s \n\n', ArrayName);
fprintf(' Disk Type: %i GB, %s \n', DiskName, DiskType);
fprintf(' [ Mfgr. MTBF = %g khrs ] \n', MTBF_mfgr/1e3);
fprintf(' [ Actual MTBF = %g khrs ] \n', MTBF/1e3);
fprintf(' [ Annualized Failure Rate (AFR) = %1.3g%% ] \n', AFR*100);
fprintf(' [ Media Rate (M) = %i MB/s ] \n', M/1e6);
fprintf(' [ Bit Error Rate (BER) = %1.0e ] \n', BER);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% varying enclosure size and corresponding rack sizes
% for 3-level GRAIDS
Enclosure = [12, 16, 24, 28, 36, 48]; % # of disks per enclosure
FullRack = [20, 13, 10, 8, 6, 5]; % # of enclosures per rack
Dmin=min(Enclosure);
Dmax=max(Enclosure);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure('Position',[1 1 800 600]);
hold on;
legendindex = 1;

de = 1; % disk/enclosure index
MTTR_disk = 0;
MTTR_disk(de,max(size(Enclosure)))=0;

```

```

MTTR_disk(:,:)=0; % initialize array for storing disk MTTR
MTTR_enclosure = 0;
MTTR_enclosure(de,max(size(Enclosure)))=0;
MTTR_enclosure(:,:)=0; % initialize array for storing enclosure MTTR

% Change axis properties
set(gca,'yscale','log');
xlim([Dmin Dmax]);
set(gca, 'XTick', Enclosure)

% Plot MTBF
h = plot([Dmin,Dmax],[MTBF_mfgr,MTBF_mfgr],'--green','LineWidth',2);
text(Enclosure(1),MTBF_mfgr,...
     strcat(['\bf MTBF']),...
     'VerticalAlignment', 'bottom');
mylegend(legendindex) = {strcat(['MTBF'])};
legendindex = legendindex + 1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Loop through all GRAID levels 50/60/55/56/65/66/550/560/650/660
for graid = level,

    % Variables
    switch graid
        case 1 %% GRAID 50 %%
            lvl_name = 50;
            style = ['-magenta,'];
            Ce = 0; % check enclosures
            HSe = 0; %hot spare enclosures
            Cd = 1; % check disks
            HSd = 1; % hot spare disks
        case 2 %% GRAID 55 %%
            lvl_name = 55;
            style = ['-blue'];
            Ce = 1; % check enclosures
            HSe = 1; %hot spare enclosures
            Cd = 1; % check disks
            HSd = 1; % hot spare disks
        case 3 %% GRAID 56 %%
            lvl_name = 56;
            style = ['-black'];
            Ce = 2; % check enclosures
            HSe = 1; %hot spare enclosures
            Cd = 1; % check disks
            HSd = 1; % hot spare disks
        case 4 %% GRAID 60 %%
            lvl_name = 60;
            style = ['-magenta'];
            Ce = 0; % check enclosures
            HSe = 0; %hot spare enclosures
            Cd = 2; % check disks
            HSd = 1; % hot spare disks
        case 5 %% GRAID 65 %%
            lvl_name = 65;
            style = ['-blue'];
            Ce = 1; % check enclosures
            HSe = 1; %hot spare enclosures
            Cd = 2; % check disks
            HSd = 1; % hot spare disks
    end
end

```

```

case 6 %% GRAID 66 %%
    lvl_name = 66;
    style = ['-black'];
    Ce = 2; % check enclosures
    HSe = 1; %hot spare enclosures
    Cd = 2; % check disks
    HSd = 1; % hot spare disks
case 11 %% GRAID 550 %%
    lvl_name = 550;
    style = ['-magenta'];
    Ce = 1; % check enclosures
    HSe = 1; %hot spare enclosures
    Cd = 1; % check disks
    HSd = 1; % hot spare disks
case 12 %% GRAID 560 %%
    lvl_name = 560;
    style = ['-blue'];
    Ce = 2; % check enclosures
    HSe = 1; %hot spare enclosures
    Cd = 1; % check disks
    HSd = 1; % hot spare disks
case 13 %% GRAID 650 %%
    lvl_name = 650;
    style = ['-cyan'];
    Ce = 1; % check enclosures
    HSe = 1; %hot spare enclosures
    Cd = 2; % check disks
    HSd = 1; % hot spare disks
case 14 %% GRAID 660 %%
    lvl_name = 660;
    style = ['-black'];
    Ce = 2; % check enclosures
    HSe = 1; %hot spare enclosures
    Cd = 2; % check disks
    HSd = 1; % hot spare disks
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% immediately begin rebuild if using hot spare disk
if HSd >= 1, Htd = 0;
    % assume average response time for manual disk replacement
else, Htd = 4; end

% immediately begin rebuild if using hot spare enclosure
if HSe >= 1, Hte = 0;
    % average response time for manual enclosure replacement
else, Hte = 12; end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Calculate number of reliability-dependent disks per enclosure
Nd = Enclosure - HSd; % remove hot spare disks from calculations
EnclosureSize = DiskSize .* (Nd-Cd); % Storage per enclosure

if (graid <= 6), % dual-level GRAIDs
    Ne = ceil( ArraySize ./ EnclosureSize ) + Ce ; % # of enclosures
    Nr = ceil( (Ne+HSe) ./ FullRack ); % # of Racks
    N = (Nd+HSd) .* (Ne+HSe); % # of Disk Drives

    % Storage Efficiency
    Efficiency = ((Nd-Cd)./(Enclosure)) .* ((Ne-Ce)./(Ne+HSe)) .* 100;

```

```

elseif (graid >= 11), % tri-level GRAIDs
    Ne = FullRack - HSe; % number of enclosures per rack
    RackSize = EnclosureSize .* (Ne-Ce); % Storage per Rack
    Nr = ceil( ArraySize ./ RackSize ); % Number of Racks
    N = (Enclosure) .* (FullRack) .* Nr; % Total Number of Disk Drives

    % Storage Efficiency
    Efficiency = ((Nd-Cd)./(Enclosure)) .* ((Ne-Ce)./(FullRack)) .* 100;
end

% Reset MTTR and MTDDL arrays for Priority loop
MTTR=0;
MTTR(3,max(size(Nd)))=0;
MTTR(:,:)=0;

MTDDL=0;
MTDDL(3,max(size(Nd)))=0;
MTDDL(:,:)=0;

ii=1; % Rebuild Priority loop
while(ii <= 3)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Priority = P_name(ii,:);
    P = P_value(ii);

    %% MTTR for Disk

    % Rebuild Rate (MB/sec)
    Rr = (M * P) ./ (exp(2/25 .* (Nd-Cd))+(1e-128));
    % Rebuild Time (hours)
    Rt = DiskSize ./ Rr ./ 3600;
    % Mean Time To Repair (hours)
    MTTRd = (Htd + Rt);

    %% The media rate between each enclosure is bounded
    %% by its interconnection (assuming 4Gb/s Fibre Channel).
    Me = 4e9/8;

    % When an enclosure fails it is critical that the
    % rebuild process has the highest priority.
    Pe = 0.8;

    %% MTTR for Enclosure
    Rr = (Me * Pe) ./ (Ne-Ce); % Rebuild Rate (MB/sec)
    Rt = EnclosureSize ./ Rr ./ 3600; % Rebuild Time (hours)
    MTTRe = (Hte + Rt); % Mean Time To Repair (hours)

    %% Save each MTTR for disk/enclosure
    if max(size(level)) == 1,
        MTTR_disk(de,:) = MTTRd;
        MTTR_enclosure(de,:) = MTTRe;
        de = de + 1;
    else
        if(ii == 2),
            MTTR_disk(de,:) = MTTRd;
            MTTR_enclosure(de,:) = MTTRe;
            de = de + 1;
        end
    end
end
end

```

```

%% Mean Time To Data Loss due to Disk Failure
switch RAID
case 1 %% RAID 50 %%
    MTTDL_DF = MTBF^2 ./ (Nd .* Ne .* (Nd-1) .* ...
        MTTRd );
case 2 %% RAID 55 %%
    MTTDL_DF = MTBF^4 ./ ...
        (Nd.^2 .* Ne .* (Nd-1).^2 .* (Ne-1) .* ...
        MTTRd.^2 .* MTTRe.^1);
case 3 %% RAID 56 %%
    MTTDL_DF = MTBF^6 ./ ...
        (Nd.^3 .* Ne .* (Nd-1).^3 .* (Ne-1) .* (Ne-2) .* ...
        MTTRd.^3 .* MTTRe.^2);
case 4 %% RAID 60 %%
    MTTDL_DF = MTBF^3 ./ (Nd .* Ne .* (Nd-1) .* (Nd-2) .* ...
        MTTRd.^2 );
case 5 %% RAID 65 %%
    MTTDL_DF = MTBF^6 ./ ...
        (Nd.^2 .* Ne .* (Nd-1).^2 .* (Nd-2).^2 .* (Ne-1) .* ...
        MTTRd.^4 .* MTTRe.^1);
case 6 %% RAID 66 %%
    MTTDL_DF = MTBF^9 ./ ...
        (Nd.^3 .* Ne .* (Nd-1).^3 .* (Nd-2).^3 .* (Ne-1) .* ...
        (Ne-2) .* MTTRd.^6 .* MTTRe.^2);

case 11 %% RAID 550 %%
    MTTDL_DF = MTBF^4 ./ ...
        (Nd.^2 .* Ne .* Nr .* (Nd-1).^2 .* (Ne-1) .* ...
        MTTRd.^2 .* MTTRe.^1);
case 12 %% RAID 560 %%
    MTTDL_DF = MTBF^6 ./ ...
        (Nd.^3 .* Ne .* Nr .* (Nd-1).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^3 .* MTTRe.^2);
case 13 %% RAID 650 %%
    MTTDL_DF = MTBF^6 ./ ...
        (Nd.^2 .* Ne .* Nr .* (Nd-1).^2 .* (Nd-2).^2 .* ...
        (Ne-1) .* MTTRd.^4 .* MTTRe.^1);
case 14 %% RAID 660 %%
    MTTDL_DF = MTBF^9 ./ ...
        (Nd.^3 .* Ne .* Nr .* (Nd-1).^3 .* (Nd-2).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^6 .* MTTRe.^2);
end

```

```

%% Mean Time To Data Loss due to Correlated Disk Failure
switch RAID
case 1 %% RAID 50 %%
    MTTDL_CDF = MTBF^2 ./ (10 .* ...
        Nd .* Ne .* (Nd-1) .* MTTRd);
case 2 %% RAID 55 %%
    MTTDL_CDF = MTBF^4 ./ (1e2 .* ...
        Nd.^2 .* Ne .* (Nd-1).^2 .* (Ne-1) .* ...
        MTTRd.^2 .* MTTRe.^1);
case 3 %% RAID 56 %%
    MTTDL_CDF = MTBF^6 ./ (1e3 .* ...
        Nd.^3 .* Ne .* (Nd-1).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^3 .* MTTRe.^2);
case 4 %% RAID 60 %%
    MTTDL_CDF = MTBF^3 ./ (1e3 .* ...
        Nd .* Ne .* (Nd-1) .* (Nd-2) .* MTTRd.^2 );
end

```

```

case 5 %% GRAID 65 %%
    MTTDL_CDF = MTBF^6 ./ (1e6 .* ...
        Nd.^2 .* Ne .* (Nd-1).^2 .* (Nd-2).^2 .* ...
        (Ne-1) .* MTTRd.^4 .* MTTRe);
case 6 %% GRAID 66 %%
    MTTDL_CDF = MTBF^9 ./ (1e9 .* ...
        Nd.^3 .* Ne .* (Nd-1).^3 .* (Nd-2).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^6 .* MTTRe.^2);

case 11 %% GRAID 550 %%
    MTTDL_CDF = MTBF^4 ./ (1e2 .* ...
        Nd.^2 .* Ne .* Nr .* (Nd-1).^2 .* ...
        (Ne-1) .* MTTRd.^2 .* MTTRe.^1);
case 12 %% GRAID 560 %%
    MTTDL_CDF = MTBF^6 ./ (1e3 .* ...
        Nd.^3 .* Ne .* Nr .* (Nd-1).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^3 .* MTTRe.^2);
case 13 %% GRAID 650 %%
    MTTDL_CDF = MTBF^6 ./ (1e6 .* ...
        Nd.^2 .* Ne .* Nr .* (Nd-1).^2 .* (Nd-2).^2 .* ...
        (Ne-1) .* MTTRd.^4 .* MTTRe.^1);
case 14 %% GRAID 660 %%
    MTTDL_CDF = MTBF^9 ./ (1e9 .* ...
        Nd.^3 .* Ne .* Nr .* (Nd-1).^3 .* (Nd-2).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^6 .* MTTRe.^2);

end

%% Unrecoverable Bit Errors
SER = BER ./ 4096; % Sector Error Rate
Sectors = ceil(DiskSize ./ 512); % Sectors per disk

%% Probability of Successfully Reading All Disk Sectors
P_Disk = ( 1 - SER ./ SER )^Sectors;

%% Probability of Encountering a Sector Error in the Enclosure
P_Enclosure = ( 1 - P_Disk.^(Nd-Cd) );

%% Mean Time To Data Loss due to Unrecoverable Bit Error
switch graid
case 1 %% GRAID 50 %%
    MTTDL_UBE = MTBF ./ ...
        (Nd .* Ne .* (P_Enclosure));
case 2 %% GRAID 55 %%
    MTTDL_UBE = MTBF^2 ./ ...
        (Nd.^2 .* Ne .* (P_Enclosure).^2 .* ...
        (Ne-1) .* MTTRe.^1);
case 3 %% GRAID 56 %%
    MTTDL_UBE = MTBF^3 ./ ...
        (Nd.^3 .* Ne .* (P_Enclosure).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRe.^2);
case 4 %% GRAID 60 %%
    MTTDL_UBE = MTBF^2 ./ (10 .* ...
        Nd .* Ne .* (Nd-1) .* (P_Enclosure) .* MTTRd);
case 5 %% GRAID 65 %%
    MTTDL_UBE = MTBF^4 ./ (1e2 .* ...
        Nd.^2 .* Ne .* (Nd-1).^2 .* (P_Enclosure).^2 .* ...
        (Ne-1) .* MTTRd.^2 .* MTTRe);
case 6 %% GRAID 66 %%

```

```

        MTTDL_UBE = MTBF^6 ./ (1e3 .* ...
        Nd.^3 .* Ne .* (Nd-1).^3 .* (P_Enclosure).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^3 .* MTTRe.^2);

    case 11 %% GRAID 550 %%
        MTTDL_UBE = MTBF^2 ./ ...
        (Nd.^2 .* Ne .* Nr .* (P_Enclosure).^2 .* ...
        (Ne-1) .* MTTRe.^1);
    case 12 %% GRAID 560 %%
        MTTDL_UBE = MTBF^3 ./ ...
        (Nd.^3 .* Ne .* Nr .* (P_Enclosure).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRe.^2);
    case 13 %% GRAID 650 %%
        MTTDL_UBE = MTBF^4 ./ (1e2 .* ...
        Nd.^2 .* Ne .* Nr .* (Nd-1).^2 .* (P_Enclosure).^2 .* ...
        (Ne-1) .* MTTRd.^2 .* MTTRe);
    case 14 %% GRAID 660 %%
        MTTDL_UBE = MTBF^6 ./ (1e3 .* ...
        Nd.^3 .* Ne .* Nr .* (Nd-1).^3 .* (P_Enclosure).^3 .* ...
        (Ne-1) .* (Ne-2) .* MTTRd.^3 .* MTTRe.^2);

end

%% Compute the Harmonic MTTDL
MTTDL(ii,:) = 3 ./ (1./MTTDL_DF + 1./MTTDL_CDF + 1./MTTDL_UBE);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ii=ii+1;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DRAW FIGURE %%%%%%%%%
c=1;
if (MTTDL(2,1) > MTBF_mfgr), % only show recommended results
    while(c <= max(size(MTTDL)))
        if( MTTDL(2,c) < MTBF_mfgr ) || ( c == max(size(MTTDL)) ),

            if( MTTDL(2,c) > MTBF_mfgr ) && ( c == max(size(MTTDL)) )
                c = c;
            else
                c = c-1;
            end

            if (graid >= 11), % tri-level GRAIDs
                c = 1; % smaller disk enclosures have better efficiency
            end

fprintf('\n(+) Recommended GRAID %i Configuration \n', lvl_name);
fprintf(' - Disks Per Enclosure = %i + %i(Hot Spare Disk) \n', ...
        Nd(c),HSd);
fprintf(' - Number of Enclosures = %i + %i(Hot Spare Enclosure) \n', ...
        Ne(c),HSe);
fprintf(' - Number of Racks = %i \n', Nr(c));
fprintf(' - Total Number of Disks = %i \n', N(c));

        MeanFailures = (N(c).*AFR);
        if (MeanFailures > POH),
            Failures = strcat([num2str(ceil(MeanFailures/POH)), '/hour']);
        elseif (MeanFailures > 365),
            Failures = strcat([num2str(ceil(MeanFailures/365)), '/day']);
        else

```

```

        Failures = strcat([num2str(ceil(MeanFailures)), '/year']);
    end

    fprintf(' - Disk Failures (Mean) ~ %s \n', Failures);
    fprintf(' - Storage Efficiency = %2.2f%% \n', Efficiency(c));

    break,
end

Recommend=1;
c = c + 1;
end
else
    fprintf('\n(-) GRAID %i Not Recommended! \n', lvl_name);
    Recommend=0; % less than ideal
    c=1;
end

%% Plot the results
if max(size(level)) == 1,
    %% MTTDL
    h = plot(Enclosure,MTTDL(1,:), '-.blue', 'LineWidth',1); % P=20%
    mylegend(legendindex) = {strcat(['MTTDL (20%)'])};
    legendindex = legendindex + 1;

    h = plot(Enclosure,MTTDL(2,:), '-blue', 'LineWidth',3); % P=50%
    text(Enclosure(1),MTTDL(2,1),...
        strcat(['\bf GRAID ', num2str(lvl_name)]),...
        'VerticalAlignment', 'bottom');
    mylegend(legendindex) = {strcat(['MTTDL (50%)'])};
    legendindex = legendindex + 1;

    if Recommend,
        % Plot ideal enclosure size
        h = plot(Enclosure(c), MTTDL(2,c), '-blueo',...
            'LineWidth',2,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','g',...
            'MarkerSize',11);
        mylegend(legendindex) = ...
            {strcat([' Ideal (' ,num2str(Enclosure(c)),')'])};
        legendindex = legendindex + 1;
    end

    h = plot(Enclosure,MTTDL(3,:), '--blue', 'LineWidth',1); % P=80%
    mylegend(legendindex) = {strcat(['MTTDL (80%)'])};
    legendindex = legendindex + 1;

    %% MTTR Disk
    h = plot(Enclosure,MTTR_disk(1,:), '-.red', 'LineWidth',1);
    mylegend(legendindex) = {strcat(['MTTR Disk (80%)'])};
    legendindex = legendindex + 1;

    h = plot(Enclosure,MTTR_disk(2,:), '-red', 'LineWidth',2);
    text(Enclosure(1),MTTR_disk(2,1),...
        strcat(['\bf MTTR Disk']),...
        'VerticalAlignment', 'bottom');
    mylegend(legendindex) = {strcat(['MTTR Disk (50%)'])};
    legendindex = legendindex + 1;

```

```

h = plot(Enclosure,MTTR_disk(3,:), '--red', 'LineWidth', 1);
mylegend(legendindex) = {strcat(['MTTR Disk (80%)'])};
legendindex = legendindex + 1;

%% MTTR Enclosure
MTTRe = 0;
MTTRe = ( MTTR_enclosure(1,:) + MTTR_enclosure(2,:) + ...
          MTTR_enclosure(3,:) ) ./ 3;

h = plot(Enclosure,MTTRe, '-red', 'LineWidth', 2);
text(Enclosure(1),MTTRe(1),...
      strcat(['\bf MTTR Enclosure']),...
      'VerticalAlignment', 'bottom');
mylegend(legendindex) = {strcat(['MTTR Enclosure'])};
legendindex = legendindex + 1;

else
  % Multi-Level MTDDL
  h = plot(Enclosure,MTDDL(2,:), style, 'LineWidth', 3);
  text(Enclosure(1),MTDDL(2,1),...
        strcat(['\bf GRAID ', num2str(lvl_name)]),...
        'VerticalAlignment', 'bottom');
  mylegend(legendindex) = {strcat(['GRAID ', num2str(lvl_name)])};
  legendindex = legendindex + 1;

  if Recommend,
    % Plot ideal enclosure size
    h = plot(Enclosure(c), MTDDL(2,c),strcat([style,'o']),...
            'LineWidth', 2,...
            'MarkerEdgeColor', 'k',...
            'MarkerFaceColor', 'g',...
            'MarkerSize', 11);
    mylegend(legendindex) = ...
      {strcat([' Ideal (', num2str(Enclosure(c)), ')'])};
    legendindex = legendindex + 1;
  end
end
end

if( max(size(level)) ~= 1 && max(size(level)) < 4),
  % Plot MTTR for disk and enclosure
  MTTRd = 0;
  MTTRe = 0;

  ddee=1;
  while(ddee < de)
    MTTRd = MTTRd + MTTR_disk(ddee,:);
    MTTRe = MTTRe + MTTR_enclosure(ddee,:);
    ddee = ddee + 1;
  end

  MTTRd = MTTRd ./ (ddee-1);
  MTTRe = MTTRe ./ (ddee-1);

  % Enclosure
  h = plot(Enclosure,MTTRe, '--red', 'LineWidth', 2);
  text(Enclosure(1),MTTRe(1),...
        strcat(['\bf MTTR Enclosure']),...
        'VerticalAlignment', 'bottom');
  mylegend(legendindex) = {strcat(['MTTR Enclosure'])};

```

```

    legendindex = legendindex + 1;

    % Disk
    h = plot(Enclosure,MTTRd,'--red','LineWidth',2);
    text(Enclosure(1),MTTRd(1),...
        strcat(['\bf MTTR Disk']),...
        'VerticalAlignment', 'bottom');
    mylegend(legendindex) = {strcat(['MTTR Disk'])};
    legendindex = legendindex + 1;
end

T1 = strcat(['Mean Time To Data Loss (MTTDL) for']);
T2 = strcat(['Grouped Redundant Arrays of Independent Disks (GRAID)']);
T3 = strcat(['[ Array Size: ', ArrayName, ...
    ' ] [ Disk Type: ', num2str(DiskName), ' GB, ', DiskType, ' ]']);
title({T1; T2; T3});

legend(mylegend,'Location', 'EastOutside');

xlabel({'Enclosure Size (# of disks)';...
    '[ includes check and hot spare disk(s) ]'});
ylabel('Mean Time (hours)');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% End FIGURE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fprintf('\n');
return

```

REFERENCES

- [ACNC07] AC&NC, "Strategic and Innovative RAID Solutions," 2007, <URL: <http://www.acnc.com>>.
- [ADAP07] Adaptec, "Serial Attached SCSI Performance," Education Center, 2007, <URL: <http://graphics.adaptec.com/SASperformance.gif>>.
- [BARK02] R. Barker and P. Massiglia, Storage Area Network Essentials: A Complete Guide to Understanding and Implementing SANs, John Wiley & Sons, New York, New York, 2002.
- [BISC97] J. Bischoff and T. Alexander, Data Warehouse: Practical Advice from the Experts, Prentice-Hall, Upper Saddle River, New Jersey, 1997.
- [CHEN90] P. Chen, G. Gibson, R. Katz, and D. Patterson, "An evaluation of redundant arrays of disks using an Amdahl 5890," SIGMETRICS Perform. Eval. Rev. 18, pp 74-85, April 1990, <URL: <http://doi.acm.org/10.1145/98460.98509>>.
- [CHEN94] P. Chen, E. Lee, G. Gibson, R. Katz, and D. Patterson, "RAID: high-performance, reliable secondary storage," ACM Computing Surveys 26, pp 145-185, June 1994, <URL: <http://doi.acm.org/10.1145/176979.176981>>.
- [COLE00] G. Cole, "Estimating drive reliability in desktop computers and consumer electronics systems" Seagate Technical Paper, TP-338.1, 2000.
- [DICT07a] "availability." The Free On-line Dictionary of Computing, 2007. <URL: <http://dictionary.reference.com/browse/availability>>.
- [DICT07b] "reliability." The Free On-line Dictionary of Computing, 2007. <URL: <http://dictionary.reference.com/browse/reliability>>.
- [DISH06] J. Disher, "I lost a Terabyte!," Adaptec Storage Advisors, March 2006, <URL: <http://storageadvisors.adaptec.com/2006/03/03/i-lost-a-terabyte>>.
- [DWIV06] H. Dwivedi, Securing Storage: A Practical Guide to SAN and NAS Security, Pearson Education, Upper Saddle River, New Jersey, 2006.
- [GIBS88] G. Gibson, L. Hellerstein, R. Karp, R. Katz, and D. Patterson, "Coding Techniques for Handling Failures in Large Disk Arrays," Technical Report. UMI Order Number: CSD-88-477., University of California at Berkeley, 1988.
- [GRAY05] J. Gray and C. van Ingen, "Empirical Measurements of Disk Failure Rates and Error Rates," MSR-TR-2005-166, December 2005.

- [GUPT02] M. Gupta, Storage Area Network Fundamentals, Cisco Systems, Indianapolis, Indiana, 2002.
- [HITA06] Hitachi, "Perpendicular Magnetic Recording Technology," White Paper, 2006, <URL: <http://www.hitachigst.com>>.
- [HP05a] Hewlett-Packard, "RAID 6 with HP Advanced Data Guarding Technology," Technology Brief TC050604TB, June 2005, <URL: <http://www.hp.com>>.
- [HP05b] Hewlett-Packard, "RAID 5 Rebuild Performance in ProLiant," Technology Brief TC050702TB, July 2005, <URL: <http://www.hp.com>>.
- [IEEE07] IEEE Computer Society, "IEEE SE Definitions," IEEE Software Engineering Online Glossary, 2007, <URL: <http://www.computer.org/portal/pages/seportal/subpages/sedefinitions.html>>.
- [JEPS03] T. C. Jepsen, Distributed Storage Networks: Architecture, Protocols and Management, John Wiley & Sons, Hoboken, New Jersey, 2003.
- [KRYD03] M. Kryder, "Future trends in magnetic storage technology," Joint NAPMRC 2003. Digest of Technical Papers [Perpendicular Magnetic Recording Conference 2003], January 2003, <URL: <http://ieeexplore.ieee.org/iel5/8392/26434/01177072.pdf>>.
- [LIOT03] M. Liotine, Mission-Critical Network Planning, Artech House, Norwood, Massachusetts, 2003.
- [MCKN06] J. McKnight, "Digital Archiving: End-User Survey & Marketing Forecast 2006-2010," Enterprise Strategy Group, Research Report, January 2006, <URL: <http://www.enterprisestrategygroup.com/ESGPublications/ReportDetail.asp?ReportID=591>>.
- [MOND03] R. Mondardini and W. Rueden, "The Large Hadron Collider (LHC) Data Challenge," IEEE Technical Committee on Scalable Computing, 2003, <URL: <http://www.ieeetcs.org/newsletters/2003-01/mondardini.html>>.
- [NARE66] J. J. Naresky, "Reliability and maintainability research in the United States Air Force", Proceedings of 5th Reliability and Maintainability Conference, pp 769-87, 1966.
- [NRS07] North River Solutions, "Calculating the True Reliability of RAID," White Paper, 2007, <URL: <http://www.northriversolutions.com>>.
- [PARH05] B. Parhami, Computer Architecture: From Microprocessors to Supercomputers, Oxford University Press; New York, New York, 2005.
- [PARI06] J. Pâris, and D. Long, "Using device diversity to protect data against batch-correlated disk failures," In Proceedings of the Second ACM Workshop on

Storage Security and Survivability. ACM Press, New York, NY, pp 47-52, 2006, <URL: <http://doi.acm.org/10.1145/1179559.1179568>>.

- [PATT88] D. Patterson, G. Gibson, and R. Katz. "A Case for Redundant Arrays of Inexpensive Disks (RAID)," In International Conference on Management of Data (SIGMOD), pp 109-116, June 1988.
- [PATT96] D. Patterson and J. Hennessy, Computer Architecture A Quantitative Approach 2nd Edition, Morgan Kaufmann, San Francisco, California, 1996.
- [PINH07] E. Pinheiro, W. Weber, and L. Barroso, Google Inc., "Failure Trends in a Large Disk Drive Population," 5th USENIX Conference on File and Storage Technologies, pp 17–28, 2007, <URL: <http://www.usenix.org/events/fast07/tech/pinheiro.html>>.
- [SCHR07] B. Schroeder and G. Gibson, Carnegie Mellon University, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?" 5th USENIX Conference on File and Storage Technologies, pp 1–16, 2007, <URL: <http://www.usenix.org/events/fast07/tech/schroeder.html>>.
- [SCHU06] M. Schuster, "For Example, RAID 6," Transtec AG, 2006, <URL: <http://www.transtec.de>>.
- [SEAG06] Seagate, "Perpendicular Recording: Powering New Levels of Disc Drive Capacity," Whitepaper TP-549, February 2006, <URL: <http://www.seagate.com>>.
- [SEAG07a] Seagate, "Barracuda ES," Data Sheet, 2007, <URL: http://www.seagate.com/docs/pdf/datasheet/disc/ds_barracuda_es.pdf>.
- [SEAG07b] Seagate, "Cheetah 15K.5," Data Sheet, 2007, <URL: http://www.seagate.com/docs/pdf/datasheet/disc/ds_cheetah_15k_5.pdf>.
- [SIMI03] H. Simitci, Storage Network Performance Analysis, Wiley Publishing; Indianapolis, Indiana, 2003.
- [SUN04] Sun Microsystems, "ZFS: the last word in file systems," Sun News Archive, 2004, <URL: <http://www.sun.com/2004-0914/feature>>.
- [TREA03] T. Treadway, "Enterprise SATA," Adaptec, version 2, draft 3, June 2003, <URL: http://treadway.us/SA_Images/Enterprise%20SATA.pdf>.
- [WALT05] C. Walter, "Kryder's Law," Scientific American, August 2005.
- [WINC06] Winchester Systems, "Enterprise RAID 6," Technology Update White Paper, 2006, <URL: <http://www.winsys.com/whitepapers>>.

- [XIAO02] L. Xiao-Guang, W. Gang, and L. Jing, "A research on multi-level networked RAID based on cluster architecture," Proceedings of 5th International Conference on Algorithms and Architectures for Parallel Processing, pp 226-229, 2002, <URL: <http://ieeexplore.ieee.org/iel5/8379/26368/01173578.pdf>>.
- [XIN03] Q. Xin, E. Miller, T. Schwarz, D. Long, S. Brandt, and W. Litwin, "Reliability mechanisms for very large storage systems," 20th IEEE/11th NASA Goddard Conference on Mass Storage Systems and Technologies, pp 146- 156, April 2003, <URL: <http://ieeexplore.ieee.org/iel5/8502/26874/01194851.pdf>>.
- [YANG99] J. Yang and F. Sun, "A comprehensive review of hard-disk drive reliability," In Proceedings of the Annual Reliability and Maintainability Symposium, 1999, <URL: <http://ieeexplore.ieee.org/iel4/6006/16055/00744151.pdf>>.

BIOGRAPHICAL SKETCH

Edward Michael McDonald, III was born in Stuart, Florida, in 1982. He graduated from Martin County High School in 2000 and promptly began his collegiate career at the Florida State University. In the summer of 2005, he graduated from Florida State with a Bachelors of Science degree in Computer Engineering and a Bachelors of Science degree in Electrical Engineering. Immediately following this Mr. McDonald began pursuing his Masters Degree in Electrical Engineering at his alma mater. Since 2003 he has served as the System and Network Administrator at the Center for Ocean-Atmospheric Prediction Studies (COAPS) at the Florida State University, under the direction of Dr. James J. O'Brien. During this time he was also involved as a research assistant at the High-Performance Computing and Simulation (HCS) Research Laboratory at the FAMU-FSU College of Engineering. His research has been sponsored by the joint HCS lab at the University of Florida and by the U.S. Department of Defense for topics including high-performance computing, storage area networks, and advanced technologies for mission assurance. Mr. McDonald's other research interests include digital forensics and network security.