Pós-Graduação em Ciência da Computação

Eric Rodrigues Borba

STOCHASTIC MODELING OF DATA STORAGE SYSTEMS FOR EVALUATING PERFORMANCE, DEPENDABILITY, AND ENERGY CONSUMPTION

Ph.D. Thesis

Federal University of Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE

2023

Federal University of Pernambuco
Center for Informatics
Graduate in Computer Science

Eric Rodrigues Borba

STOCHASTIC MODELING OF DATA STORAGE SYSTEMS FOR EVALUATING PERFORMANCE, DEPENDABILITY, AND ENERGY CONSUMPTION

A Ph.D. Thesis presented to the Center for Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Philosophy Doctor in Computer Science.

Advisor: Prof. Dr. Eduardo Antonio Guimarães Tavares
Co-Advisor: Prof. Dr. Paulo Romero Martins Maciel

RECIFE
2023

This thesis is dedicated to my family.

Acknowledgements

I want to express my gratitude to God for granting me the opportunity to experience and overcome the challenges encountered on this journey.

I extend my heartfelt appreciation to my advisor, Prof. Eduardo Tavares, who believed in my abilities and guided me in overcoming my shortcomings. His understanding and support were instrumental in the completion of this work.

I would also like to thank my co-advisor, Prof. Paulo Maciel, whose teachings have been invaluable since my days as an undergraduate student in computer engineering. I am grateful to my colleagues at MoDCS for their support and knowledge sharing.

I sincerely thank Prof. André Brinkmann for warmly welcoming me to his research group at Johannes Gutenberg-Universität Mainz and for sharing his knowledge with me. I'm also grateful to all the Efficient Computing and Storage group members at ZDV for their contributions.

A special thanks to my parents, Esdras and Marily, for their dedication, endless efforts, and unwavering belief in my abilities. Their constant support and investment in my education have been instrumental in bringing me this far. I am grateful to my wife, Rosiely Borba (and our Catucha), who has transformed my life by providing the necessary emotional support to achieve this goal since the moment I met her. Completing this stage in my life would not have been possible without her understanding and affection at every moment.

Many thanks to my incredible siblings, Cindel and Esdras, for always celebrating my wins like their own. I extend my gratitude to my extended family, who have been with me throughout my journey, even from a distance, believing in me and providing encouragement until the end, and to all other colleagues, friends, and relatives who have directly or indirectly contributed to the accomplishment of this goal.

This work has been supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq under grant 405224/2018-4, 302373/2018-7 and 202998/2019-3. Besides, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 88887.500766/2020-00.

This work was partially supported by the European Union's Horizon 2020 JTI-EuroHPC research and innovation programme and the BMBF/DLR under the "IO-SEA" project with grant agreement number: 955811.

Abstract

Improvements in data storage systems may be limited by the low performance of hard disk drives (HDDs) and the high cost per gigabyte of solid-state drives (SSDs). To mitigate these issues, several architectures based on hybrid storage systems have been proposed. However, energy consumption is usually neglected, and new approaches may not consider the impact on the mechanical components of HDDs, which can result in malfunctions and data loss. Similarly, the lifetime of SSDs can be reduced owing to their limited number of flash memory operations. This thesis presents an approach based on generalized stochastic Petri nets (GSPNs) to evaluate the performance and energy consumption of homogeneous (HDD and SSD) and hybrid storage systems. Two analytical models have been proposed to represent distinct workloads and estimate throughput, energy consumption, and response time. In addition, a performability model has been conceived using the GSPN and reliability block diagram (RBD) formalisms to evaluate the impacts of failures on the performance of storage systems. Hierarchical modeling approach has been adopted, and the proposed model can estimate the availability and response time. A benchmark tool is adopted in this study to generate workloads and collect data to characterize storage devices. Simultaneously, this investigation estimates the power demand of HDDs and SSDs from measurements. The results are utilized to validate the GSPN models using statistical analysis and experiments based on industry-standard benchmarks. A design of experiment (DoE) is performed to investigate the most important factors assumed in this study. An exploratory analysis was conducted using industry datasets from Alibaba and Backblaze to investigate the distinct effects of applications on storage failures. Results demonstrate the feasibility of the proposed models and provide important observations regarding storage solutions for different applications.

Keywords: Performance evaluation. Hybrid storage. Stochastic Petri nets. Cloud Computing. Data management. Energy consumption. Performability.

Resumo

O aperfeiçoamento de sistemas de armazenamento de dados pode ser limitado pelo baixo desempenho de dispositivos de disco rígido (HDDs) e pelo alto custo por gigabyte de dispositivos de estado sólido (SSDs). Para mitigar essas questões, diversas arquiteturas têm sido concebidas, baseadas em sistemas de armazenamento híbrido. No entanto, o consumo energético é geralmente negligenciado, e novas abordagens não consideram os impactos nos componentes mecânicos de HDDs, o que pode resultar em um mau funcionamento e perda de dados. Da mesma forma, os SSDs podem ter seu tempo de vida reduzido devido ao número limitado de operações em memórias *flash*. Esta tese apresenta uma abordagem baseada em redes de *Petri* estocásticas generalizadas (GSPN) para a avaliação de desempenho e consumo energético de sistemas de armazenamento homogêneos (HDD e SSD) e híbridos. Dois modelos analíticos são propostos para representar diferentes cargas de trabalho e estimar vazão, consumo energético e tempo de resposta. Além disso, um modelo de performabilidade foi concebido utilizando os formalismos GSPN e diagrama de blocos de confiabilidade (RBD) para avaliar o impacto de falhas no desempenho de sistemas de armazenamento. Uma abordagem de modelagem hierárquica foi adotada, e o modelo pode estimar disponibilidade e tempo médio de resposta. Uma ferramenta de *benchmark* foi adotada nesse estudo para gerar cargas de trabalho e coletar dados para a caracterização dos dispositivos de armazenamento. Simultaneamente, esta investigação estimou a potência demandada por HDDs e SSDs por meio de medições. Os resultados foram utilizados para validar os modelos GSPN através de técnicas estatísticas e experimentos baseados em *benchmarks* padrões da indústria. Um planejamento de experimento (DoE) foi realizado para investigar os fatores mais impactantes assumidos nesse estudo. Uma análise exploratória foi conduzida utilizando *datasets* das companhias Alibaba e Backblaze para investigar os diferentes efeitos de aplicações na falha de dispositivos de armazenamento de dados. Os resultados demonstram a viabilidade dos modelos propostos e fornecem importantes observações em relação a soluções de armazenamento de dados para diferentes aplicações.

Palavras-chave: Avaliação de Desempenho. Armazenamento Híbrido. Redes de *Petri* Estocásticas. Computação em Nuvem. Gerenciamento de dados. Consumo Energético. Performabilidade.

List of Figures

2.1 HDD components (own work (2023)). 25
2.2 SSD architecture and controller functionalities (own work (2023)). 28
2.3 Hybrid storage with a SSD as cache for HDD (BORBA; TAVARES, 2017). 31
2.4 Random data placement policy for hybrid storage (BORBA; TAVARES, 2017). . . 32
2.5 Dependability concepts (adapted from AVIZIENIS; LAPRIE; RANDELL (2001)). 35
2.6 CTMC example (own work (2023)). 39
2.7 Petri net elements (own work (2023)). 41
2.8 Firing of a transition (own work (2023)). 41
2.9 Petri net and corresponding reachability graph (own work (2023)). 42
2.10 Eliminating vanishing markings demonstrated by (a) a given GSPN, (b) equivalent ERG, (c) resulting RG, and (d) corresponding CTMC (adapted from BOLCH et al. (2006)). 47
2.11 *s-transition* (BORBA; TAVARES; MACIEL, 2022). 49
2.12 Erlang distribution (BORBA; TAVARES; MACIEL, 2022). 49
2.13 Hipoexponential distribution (BORBA; TAVARES; MACIEL, 2022). 49
2.14 Hiperexponenial distribution (BORBA; TAVARES; MACIEL, 2022). 49
2.15 RBD arrangement (BORBA; TAVARES, 2017). 51

4.1 UML elements adopted to illustrate the methodology proposed in this thesis (own work (2023)). 65
4.2 Supporting methodology for performance and energy consumption modeling of data storage systems (own work (2023)). 66
4.3 Supporting methodology for dependability modeling of data storage systems (own work (2023)). 71
4.4 Environment setting (BORBA; TAVARES; MACIEL, 2022). 73
4.5 Electrical circuit for measurement of HDD and SSD voltage values. 74

5.1 Single storage model (BORBA; TAVARES; MACIEL, 2022). 80
5.2 s-transition example (BORBA; TAVARES; MACIEL, 2022). 83
5.3 Hipoexponential subnet example (BORBA; TAVARES; MACIEL, 2022). 83

5.4 Exponential example (own work (2023)). 84
5.5 Erlang subnet example (own work (2023)). 84
5.6 Hiperexponential subnet example (own work (2023)). 85
5.7 Multiple storage model (BORBA; TAVARES; MACIEL, 2022). 86
5.8 RBD model for three storage node configurations (own work (2023)). 90
5.9 Performability model (own work (2023)). 90
5.10 Hierarchical modeling example (own work (2023)). 92

6.1 Experiment II - energy consumption (BORBA et al., 2020). 97
6.2 Experiment II - response time: (a) *object_size*; (b) *workers*; and (c) *operation*$*$*technology* (BORBA et al., 2020). 98
6.3 Experiment II - IOPS: (a) *object_size*; (b) *operation*$*$*technology*; and (c) *pattern* (BORBA et al., 2020). 100
6.4 *Random access* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022). 107
6.5 *Random access* - price/IOPS (BORBA; TAVARES; MACIEL, 2022). 107
6.6 *Sequential access* - IOPS/energy consumption x technology (BORBA; TAVARES; MACIEL, 2022). 108
6.7 *Sequential access* - price/IOPS (BORBA; TAVARES; MACIEL, 2022). 108
6.8 *Read operations* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022).109
6.9 *Read operations* - price/IOPS (BORBA; TAVARES; MACIEL, 2022). 109
6.10 *Mixed* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022). 110
6.11 *Mixed* - price/IOPS (BORBA; TAVARES; MACIEL, 2022). 111
6.12 SSD annual failure rate per application (own work (2023)). 114
6.13 Attributes and SSD failures over the number of written blocks (own work (2023)). . 115
6.14 Attributes and HDD failures over number of blocks written (own work (2023)). . . 117

C.1 Performability model execution - initial marking (own work (2023)). 140
C.2 Performability model execution - all storage nodes available (own work (2023)). . . 141
C.3 Performability model execution - ready for processing (own work (2023)). 142
C.4 Performability model execution - processing request (own work (2023)). 142
C.5 Performability model execution - ready for communicating (own work (2023)). . . 143
C.6 Performability model execution - storage node unavailable (own work (2023)). . . . 143
C.7 Performability model execution - repairing storage node (own work (2023)). 144

List of Tables

3.1 Comparison between this thesis and related work. 62

4.1 Experiment components. 74

4.2 SMART attributes adopted for HDD and SSD analysis (✳ means an attribute is included in the respective technology). 75

5.1 Transition attributes - single storage model. 81

5.2 GSPN metrics - single storage model. 82

5.3 Transition attributes - multiple storage model. 85

5.4 GSPN metrics - multiple storage model. 87

6.1 Factors and levels. 95

6.2 Experiment I - mean values. 96

6.3 Experiment II - ANOVA two-way analysis. 96

6.4 Experiment III - Composite desirability. 101

6.5 Moment matching - HDD and SSD. 102

6.6 Mean power values. 102

6.7 Validation results - single storage model. 103

6.8 Validation results - multiple storage model. 103

6.9 Validation using Fio tool - multiple storage model. 104

6.10 Screening - factors and levels. 105

6.11 Rank of main and interaction effects. 105

6.12 Experimental results. 106

6.13 Case study - chosen configurations. 111

6.14 Case study results summary - composite desirability. 112

6.15 Scalability evaluation - storages. 113

6.16 Scalability evaluation - workers. 113

A.1 SSD dataset - applications overview. 138

B.1 HDD failures and attributes overview (✳ means that a specific attribute has been found on logs from the respective HDD model). 139

List of Acronyms

AWS *Amazon Web Service* 56

CC *Cloud Computing* 16

CTMC *Continuous-Time Markov Chain* 39

DTMC *Discrete-Time Markov Chain* 39

DC *Data Center* 16

DoE *Design of Experiment* 67

HPC *High-Performance Computing* 16

VM *Virtual Machine* 59

KVM *Kernel-Based Virtual Machine* 59

HDD *Hard Disk Drive* 17

P/E *Program/Erase Cycle* 17

IaaS *Infrastructure as a Service* 18

IOPS *Input/Output per Second* 18

ERG *Extended Reachability Graph* 46

PN *Petri Net* 18

QoS *Quality of Service* 19

RAM *Random Access Memory* 19

RBD *Reliability Block Diagram* 18

SSD *Solid-State Drive* 17

SMART *Self-Monitoring Analysis and Reporting Technology* 56

SLA *Service Level Agreement* 17

SPN *Stochastic Petri Net* 44

GSPN *Generalized Stochastic Petri Net* 18

SPC *Storage Performance Council* 68

SATA *Serial Advanced Technology Attachment* 55

RAID *Redundant Array of Independent Disks* 57

UML *Unified Modeling Language* 64

DAE *Data Analytics Engine* 75

DB *Database* 75

NAS *Network Attached Storage* 75

RM *Resource Management* 75

SS *SQL Services* 75

WPS *Web Proxy Services* 75

WS *Web Services* 75

WSM *Web Service Management* 75

Contents

1 INTRODUCTION . . . 16
1.1 CONTEXT . . . 16
1.2 MOTIVATION . . . 18
1.3 OBJECTIVES . . . 20
1.4 CONTRIBUTIONS . . . 20
1.5 OUTLINE . . . 22

2 BACKGROUND . . . 24
2.1 DATA STORAGE SYSTEMS . . . 24
2.1.1 **Hard disk drives** . . . 24
2.1.1.1 Performance . . . 25
2.1.1.2 Reliability . . . 27
2.1.2 **Solid-state drives** . . . 27
2.1.2.1 Performance . . . 28
2.1.2.2 Reliability . . . 29
2.2 HYBRID STORAGE SYSTEMS . . . 30
2.2.1 **SSD as Cache** . . . 30
2.2.2 **SSD for Random Requests** . . . 31
2.3 PERFORMANCE EVALUATION . . . 32
2.4 DEPENDABILITY EVALUATION . . . 34
2.5 CONTINUOUS MARKOV CHAINS . . . 38
2.6 STOCHASTIC PETRI NETS . . . 40
2.6.1 **Petri nets** . . . 40
2.6.1.1 Petri nets properties . . . 42
2.6.2 **Generalized stochastic Petri nets** . . . 44
2.6.3 **Phase-type distributions** . . . 48
2.7 RELIABILITY BLOCK DIAGRAMS . . . 50
2.8 SUMMARY . . . 51

3 RELATED WORKS . . . 53
3.1 OVERVIEW . . . 53
3.2 MODELS . . . 54
3.3 ARCHITECTURES . . . 55
3.4 DEPENDABILITY EVALUATION . . . 56
3.5 ENERGY CONSUMPTION . . . 58
3.6 DATA MANAGEMENT . . . 59
3.7 FLASH MEMORY MANAGEMENT . . . 60
3.8 COMPARISON . . . 61
3.9 SUMMARY . . . 62

4 METHODOLOGY AND TOOLS . . . 64
4.1 PRELIMINARIES . . . 64
4.2 MODELING STORAGE SYSTEMS FOR PERFORMANCE AND ENERGY CONSUMPTION EVALUATION . . . 65
4.2.1 **Conception of performance and energy consumption models** . . . 65
4.2.2 **Measurement and investigation of storage performance and energy consumption** . . . 67
4.2.3 **Model validation, refinement, and solving** . . . 68
4.2.4 **Experiments utilizing the models** . . . 68
4.3 MODELING STORAGE SYSTEMS FOR DEPENDABILITY EVALUATION . . . 70
4.4 TOOLS AND ENVIRONMENT SETTING . . . 72
4.4.1 **HDDs and SSDs failure logs** . . . 74
4.4.1.1 SMART logs . . . 74
4.4.1.2 Datasets . . . 75
4.5 SUMMARY . . . 76

5 MODELS . . . 77
5.1 PERFORMANCE MODELING . . . 77
5.1.1 **Single storage model** . . . 79
5.1.1.1 Phase-type distribution example . . . 83
5.1.2 **Multiple storage model** . . . 84

5.2 DEPENDABILITY MODELING . . . 88
5.2.1 **Availability model** . . . 89
5.2.2 **Performability model** . . . 89
5.2.2.1 Hierarchical modeling example . . . 92
5.3 SUMMARY . . . 93

6 EXPERIMENTS . . . 94
6.1 MEASUREMENT EXPERIMENT - EXPLORATORY ANALYSIS . . . 94
6.1.1 **Experiment I: screening** . . . 95
6.1.2 **Experiment II: hybrid storage evaluation** . . . 96
6.1.2.1 Evaluation of the impact of factors on energy consumption . . . 97
6.1.2.2 Evaluation of the impact of factors on response time . . . 97
6.1.2.3 Evaluation of the impact of factors on IOPS . . . 99
6.1.3 **Experiment III: composite desirability** . . . 101
6.2 PERFORMANCE MODEL VALIDATION . . . 101
6.3 EXPERIMENTAL RESULTS . . . 104
6.3.1 **Experiment I: screening** . . . 104
6.3.2 **Experiment II: random accesses** . . . 106
6.3.3 **Experiment III: sequential accesses** . . . 107
6.3.4 **Experiment IV: read operations** . . . 109
6.3.5 **Experiment V: mixed** . . . 110
6.3.6 **Case study** . . . 111
6.3.7 **Scalability** . . . 112
6.4 HDDs AND SSDs FAILURES ANALYSIS . . . 113
6.4.1 **SSD analysis** . . . 114
6.4.2 **HDD analysis** . . . 116
6.5 SUMMARY . . . 118

7 CONCLUSION . . . 119
7.1 CONTRIBUTIONS . . . 120
7.2 LIMITATIONS . . . 122
7.3 PUBLICATIONS . . . 123
7.3.1 **Journals** . . . 123

7.3.2 **Conferences** 123
7.4 FUTURE WORKS 124

References 126

Appendix 137

A SSD dataset - applications overview 138

B HDD dataset - SMART attributes overview 139

C Example of the performability model execution 140

1

INTRODUCTION

This chapter provides a comprehensive overview of the research conducted in this thesis. Section 1.1 outlines the context of the proposed approach. Section 1.2 highlights the motivation for this study. Section 1.3 discusses the general and specific objectives of the research to clarify the intended outcomes. The contributions of this study are presented in Section 1.4. Finally, Section 1.5 details the thesis structure, providing readers with an overview of the chapters and their respective contents.

1.1 CONTEXT

Energy consumption in *Data Center*s (DCs) is a critical and challenging issue, which has motivated many studies to reduce operational costs. For instance, reports indicate that the cost of energy consumed by a server (during its lifetime) will exceed the hardware costs if the current demand continues to increase (BHARANY et al., 2022). Estimates mention that in 2016, approximately 416.2 billion kWh of energy was consumed by computer servers, which is more than the total energy consumed in the entire United Kingdom (ZHOU et al., 2022).

The massive growth in structured and unstructured data requires significant computational capabilities (VEF et al., 2020). As a result, information technology-related services currently consume approximately 7% of global electricity, and this is expected to increase to 13% by 2030 (ZENG et al., 2022). By 2050, the energy consumption of DCs is expected to grow twelvefold, with a fivefold increase projected by 2025 (ZHAO; ZHOU, 2022). Therefore, several efforts have been made to maximize the energy efficiency of DCs, particularly in the context of *High-Performance Computing* (HPC) and *Cloud Computing* (CC) (TULI et al., 2022).

CC has been widely adopted since this paradigm reduces operational costs and improves computational resources utilization. For instance, the United States Library of Congress has moved its digital content to a cloud storage provider, and Netflix has adopted the Amazon S3 platform to store its videos (PALACIOS CHAVARRO et al., 2022). Nevertheless, the energy consumption of cloud computing systems also needs to be addressed, as the amount of stored data and applications using this paradigm continues to steadily increases (KATAL; DAHIYA; CHOUDHURY, 2023). Of all data worldwide, 90% have been generated over the last few years (YANG et al., 2022), and computer-stored data are predicted to reach 163 zettabytes by 2025 (BERISHA; MËZIU; SHABANI, 2022; SINGHAL et al., 2018). For example, Facebook generates approximately 10 petabytes per month of log data and Google processes over 100 petabytes in the same period. (SIAPOUSH; JAMALI; BADIRZADEH, 2023; MANOGARAN; THOTA; LOPEZ, 2022).

Existing solutions for increasing the energy efficiency and performance of storage devices do not consider the adverse impacts on reliability (YIN et al., 2018a). For example, owing to their excellent performance results, several approaches suggest concentrating intense workloads on *Solid-State Drive*s (SSDs). However, the durability of flash memory chips is directly associated with the number of *Program/Erase Cycle*s (P/Es). Therefore, intense write operations can compromise the reliability of solid-state devices (ELYASI et al., 2018; WANG et al., 2022; SALKHORDEH et al., 2021). The high energy consumption of *Hard Disk Drive*s (HDDs) has also motivated the development of new techniques. Although existing approaches obtain significant results, they commonly involve frequent changes in the rotation of magnetic disks, which can cause failures in their internal components and consequently decrease the HDD lifetimes (YIN et al., 2018a). Consequently, storage failures cause downtime and disrupt system operations (FRANK et al., 2019), which may result in financial penalties due to *Service Level Agreement*s (SLAs). On the other hand, high availability usually entails equipment redundancy, which increases infrastructure costs.

Solid-state drives provide faster read operations than magnetic hard disks (WANG et al., 2022). However, for some workloads, SSDs may not provide better sequential access results than HDDs. As an alternative, hybrid approaches have been proposed. Hybrid storage systems may perform better than HDD storage at an affordable cost, making them a promising solution for many systems, such as those based on cloud computing (BOUKHELEF et al., 2019; WANG et al., 2022). Consequently, research on storage architectures has been conducted (WANG et al.,

2022).

Although many studies have evaluated storage system behavior, few approaches rely on formal models and concomitantly consider *Input/Output per Second* (IOPS), availability, and response time. *Petri Net*s (PNs) provide a mathematical foundation and, unlike simulators, such a formalism supports the analysis/verification of quantitative (behavioral) and qualitative (structural) properties (MURATA, 1989). Therefore, performance and dependability models are quite important (MACIEL et al., 2011; OLIVEIRA et al., 2019), as different data-placement policies and architectures can be assessed before implementing a real system.

This thesis presents an approach based on *Generalized Stochastic Petri Net*s (GSPNs) and *Reliability Block Diagram*s (RBDs) for the evaluation of the performance, dependability, and energy consumption of homogeneous (i.e., HDDs or SSDs only) and hybrid storage systems. The proposed models can represent different workloads and architectures. Furthermore, the models can estimate the throughput, response time, availability, and energy consumption. Experimental results based on industry-standard benchmarks demonstrate the feasibility of the proposed approach.

1.2 MOTIVATION

The performance of HDDs must be improved to meet the current demand for systems that require high throughput, low latency, and reduced power consumption. For example, in data centers that provide *Infrastructure as a Service* (IaaS), the use of HDDs in cloud storage infrastructure has become a bottleneck for applications that demand progressively higher performance levels (MIAO et al., 2022; CHIKHAOUI; BOUKHALFA; BOUKHOBZA, 2018). The inherent characteristics of these devices, such as the need for mechanical movement of some of their components (e.g., the platter, spindle, and actuator arm), hinder more significant progress (MEI et al., 2019). This observation becomes particularly evident, especially when considering workloads composed of random requests, which demand access to data in different sectors, resulting in physical displacement to extreme points on the disk.

Despite these factors, the average response time for data access on magnetic disks has reached a 15% reduction percentage per year (PARK; LEE; KIM, 2017). The decrease in seek time (8%) and the increase in rotation speed (9%) are examples of factors responsible for this improvement, which were obtained due to techniques such as caching, write buffering, prefetching, request scheduling, and parallel I/O (YANG et al., 2020). However, the pace of evolution

of HDD technology in terms of performance does not match that of other system components, thus, limiting the speed of data access and distribution to the storage device's performance. For example, one can cite an increase in the speed of processors, which have an annual evolution of up to 60%, in addition to the decrease in access time in *Random Access Memory* (RAM), which is 50% faster than the corresponding growth in magnetic disk performance (PARK; LEE; KIM, 2017).

Because they have no mechanical components and are based on flash memory, SSDs are suitable substitutes, given their significant throughput rates, shorter average response times, and low power consumption (LI et al., 2019). However, their exclusive adoption in data centers is not possible, given that SSDs still need to meet all performance and reliability requirements. For example, the low capacity of solid-state disks is a significant drawback, particularly for data centers where large volumes of data are stored (MEI et al., 2018). Moreover, when subjected to the same workload, the lifetime of SSDs is significantly reduced compared to that of HDDs. This limitation results from the endurance limit of flash memory, which is directly related to the number of programming and erasure cycles (P/E cycles) that can be tolerated, usually less than 100000 (ZHANG et al., 2018; MEI et al., 2019). The wear on flash memory chips results in the generation of bad blocks, which leads to a decrease in the lifetime of SSDs; for example, in a data center, 30–80% of SSDs develop bad blocks during their lifetimes (HAN et al., 2018).

According to the literature (YU et al., 2018; LI et al., 2019; WU et al., 2018), the combination of SSDs and HDDs represents a possible solution to enhance the *Quality of Service* (QoS) provided by clouds. However, achieving this improvement requires a balance among aspects such as energy consumption, dependability, and performance. Neglecting these considerations can render the entire system economically unviable due to maintenance costs. An example of this is the annual loss of US$1.7 trillion incurred by cloud service providers due to long response times and downtimes (HUANG et al., 2019).

To improve the utilization of storage drives, it is crucial to develop a solution that enables cost-effective analysis and exploitation of their characteristics. This requires implementing suitable storage architectures and data placement policies to facilitate the effective integration of different technologies. By doing so, organizations can enhance performance and reduce costs while ensuring the availability and durability of their stored data.

1.3 OBJECTIVES

This thesis presents a modeling approach based on GSPNs and RBDs to evaluate the performance, availability, and energy consumption of data storage systems under different workloads. The specific objectives are as follows:

- Provide performance and power measurement values collected from data storage devices (SSDs and HDDs) under different operations, access patterns (random or sequential), numbers of workers (i.e., clients), and object sizes;
- Propose formal models based on generalized stochastic Petri nets to estimate the performance and energy consumption of homogeneous and hybrid data storage systems. These models should be able to represent different operations, access patterns, and object sizes;
- Perform experiments to demonstrate the feasibility of using the proposed models to represent storage devices. The performance and energy consumption of HDDs and SSDs are investigated under various workloads. Furthermore, this study aims to provide valuable insights into the impact of workload characteristics on hybrid storage devices;
- Provide an exploratory analysis of industry logs containing information from HDD and SSD sensors regarding the wear and utilization of such devices. This investigation aims to provide insights into the effects of applications on storage health;
- Propose availability and performability models based on GSPN and RBD formalisms for analyzing the impact of failures on the performance of data storage systems (i.e., the impact on the overall system performance when storage components occasionally become non-operational). The proposed model adopts a hierarchical modeling approach to provide a feasible solution for planning storage architectures and data management strategies.

1.4 CONTRIBUTIONS

The main contribution of this thesis lies in the development of models based on the GSPN and RBD mathematical formalisms. Specifically, these models allow the design of data

storage systems (e.g., capacity planning and technology evaluation) that may be adopted for purposes such as cloud computing services. Additionally, the proposed models may be utilized to investigate different storage arrangements, considering cost and SLA constraints. The additional contributions of this study can be summarized as follows:

- **Measurement values.** Performance and power values have been collected while operating storage devices under different workloads. The collected values have been used to validate and demonstrate the feasibility of the proposed models. However, this contribution is not limited to this approach, as the acquired information can also be utilized, for instance, to determine the appropriate storage technology for a given service.

- **Workload-driven storage technologies study.** An investigation has been conducted to assess the performance and energy consumption of a solid-state drive, various hard disk drives, and a hybrid storage system. This study provides insights into the impact of storage technology and workload characteristics on adopted metrics. To demonstrate the practical applicability of this study, data-placement policies are suggested by optimizing a set of metrics, thereby, identifying an optimal combination of storage technologies and workloads.

- **Workload-driven storage failures study.** An exploratory analysis of storage failures has been conducted using datasets from two representative companies. This study provides insight into the failure rates of SSDs under commonly used workloads. In the case of HDDs, wear and failure evolution were analyzed by studying the data collected from various sensors to investigate their behavior during utilization. The results can be utilized, for instance, in planning storage architectures and devising data placement strategies to mitigate the wear and failure of storage technologies.

- **Storage technology analyses utilizing industry-based benchmarks.** Experiments have been performed to demonstrate the feasibility of the proposed models. This study provides the main factors that impact the performance and energy consumption of the storage devices adopted in this thesis. Results also demonstrate the behavior of different storage technologies under workloads encountered in the industry. Such insights can be employed for planning workload-driven data-management strategies

considering storage technology characteristics. A storage-system architecture study demonstrates the benefit of the proposed approach.

1.5 OUTLINE

This section briefly presents the contents of the remaining chapters of this thesis, which are organized as follows.

Chapter 2 presents the basic concepts necessary to understand this thesis. First, the concepts of the studied storage devices (HDD and SSD) are discussed. Next, it explains the concept of hybrid storage in the existing literature and presents two storage policies aimed at hybrid devices, which researchers have suggested several times. Subsequently, it introduces concepts that have been adopted in this study to guide the metrics used to evaluate the storage devices. The mathematical formalisms GSPN and RBD are also discussed to present the concepts necessary to better understand the solution proposed in this thesis.

Chapter 3 describes previous works related to this study to highlight solutions and gaps in the existing literature. This chapter is divided into analytical models, architectures, dependability evaluation, energy consumption, data management, and flash memory management, and a comparison is made between this thesis and previous solutions.

Chapter 4 describes the methodology used in this study. First, it describes the steps necessary to define the problem, design and validate the analytical models, and plan the experiments. In addition, the evaluation methodology assumed for planning the experiments is clarified in detail. The statistical methods and techniques used to analyze the experimental results are also discussed. Finally, the tools and environment for the measurement and data collection are presented.

Chapter 5 introduces the GSPN and RBD models conceived in this study. This chapter first presents the considerations and concepts assumed in this thesis. Furthermore, it explains the metrics of interest and notations for each model. Subsequently, it describes the proposed analytical models in detail.

Chapter 6 shows the results of the experiments conducted for this thesis. First, it shows the validation of the proposed analytical models. The results and analysis of the experiments performed using the conceived GSPN models and industry-based standards are then presented. Finally, a case study is conducted to confirm the feasibility of the proposed solution.

Finally, Chapter 7 concludes this thesis by discussing its contributions. In addition, this

chapter details possible future works and the limitations of the proposed solution.

2

BACKGROUND

This chapter introduces the basic concepts necessary to understand this thesis. Section 2.1 describes the storage devices used in the proposed solution. Section 2.2 discusses hybrid storage technologies and presents two commonly suggested architectures for didactic purposes. The concepts of performance and dependability are explained in Sections 2.3 and 2.4, respectively. Finally, Sections 2.5, 2.6 and 2.7 present the mathematical formalisms adopted in this thesis to conceive the proposed analytical models.

2.1 DATA STORAGE SYSTEMS

This section introduces the basic concepts of the storage devices adopted in this thesis (HDD and SSD). In addition, it explains aspects of the architecture of each technology and the operation of their internal components. In addition, this section describes the performance and reliability characteristics of SSDs and HDDs and identifies the advantages and disadvantages of each of these technologies.

2.1.1 **Hard disk drives**

HDDs are essential for personal computers and large data processing systems (WANG et al., 2019). Since their production began in 1956, the industry has fostered outstanding innovations in design and manufacturing, reaching levels of evolution similar to those of the semiconductor revolution (AL MAMUN; GUO; BI, 2006). Furthermore, in the domain of magnetic storage systems, HDDs are the dominant devices (WANG et al., 2019), as far as industrial production is concerned, owing to their large storage space, low cost per gigabyte, and

broad productions infrastructure (WU et al., 2018).

With regard to the computer architecture, HDDs are situated between RAM and removable drives. In this way, HDDs provide direct access to large amounts of non-volatile data; therefore, no power is required for data preservation. Despite the existence of modern technology (e.g., SSDs), HDDs are still widely used for data storage in DCs (WANG et al., 2019; YU et al., 2018). In addition, HDDs have essential attributes for cloud computing platforms, such as low cost, reasonable performance, and a long lifetime.

The proper functioning of hard-disk devices is directly related to the current states of their various components. In HDDs, wear on mechanical and electronic components can result in a loss of performance and reliability for the entire data storage system. The components (Figure 2.1) of HDDs are classified into four categories (AL MAMUN; GUO; BI, 2006): magnetic components (disk and head slider), mechanical component (pivot), electromechanical components (spindle motor, actuator arm, voice coil motor), and electronic components (integrated circuits and interface).

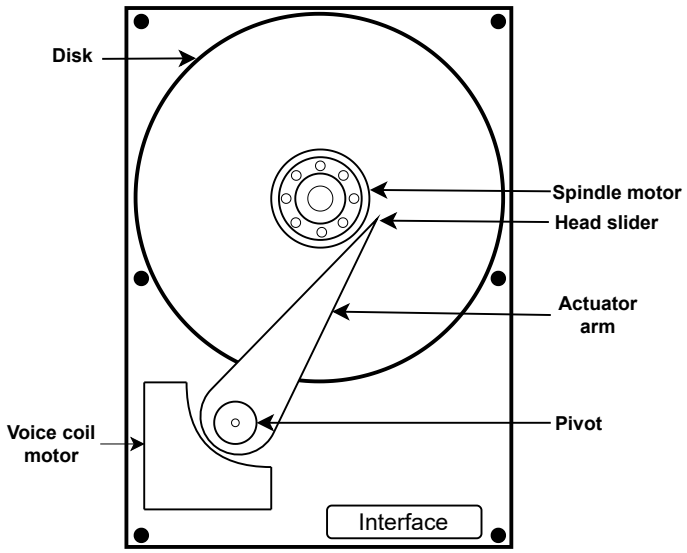


Figure 2.1: HDD components (own work (2023)).

2.1.1.1 Performance

Systems that demand many accesses operations (e.g., cloud computing platforms) require high throughput and low average response times to meet the performance requirements commonly set in SLAs. Small delays in request delivery can significantly impair the data processing and program execution (WU et al., 2018). Therefore, it is necessary to develop new techniques to

improve the components that directly affect the performance of hard-disk devices.

The following parameters can be considered for evaluating the performance of HDDs (WANG; TARATORIN, 1999):

- **Rotational latency:** is the time required for the displacement of read and write heads between the data storage sectors;
- **Access time:** is the sum of the seek time, time for the read head to cease vibrations at the end of the search for the requested data (head setting time), and rotational latency time;
- **Average response time:** is directly related to the access time, in addition to the execution of the requested operation;
- **Throughput:** represents the number of bits per unit of time that the read and write heads can process.

The rotational latency of an HDDs is directly related to the speed of the spindle motor. Consequently, increasing the rotational speed reduces the access time to the data storage sectors (WU et al., 2018) for applications that require a low mean response time. Similarly, advances in actuators and read and write heads have increased the transfer rate of bits in hard-disk devices (WU et al., 2018).

Although the number of revolutions per minute (RPM) of spindle motors has increased over the years, there are still limitations that prevent the further advancement of the HDDs (WU et al., 2018) performance. For example, as the speed of the axis motors increases, exact synchronization with the actuator is required to ensure compliance with the written or read bits. Furthermore, an increase in the mechanical motion can result in HDDs with higher energy consumption (YIN et al., 2018a).

Improvements in the aforementioned components have optimized the performance of HDDs. However, compared with other technologies (e.g., SSDs), hard disk devices still have difficulties in handling requests for small and random objects (WU et al., 2018). Decreasing the time required to move the mechanical components between disk sectors remains challenging for industry and researchers. Thus, HDDs are typically recommended for data storage systems characterized by sequential requests (LI et al., 2019).

This thesis considers only the metrics of throughput and average response time to compare the performance of different data storage technologies (e.g., HDDs and SSDs). Both

metrics have been commonly adopted in the literature to evaluate data storage systems (WU et al., 2023; LEE; MOON; PARK, 2009).

2.1.1.2 Reliability

The reliability of storage devices is a significant concern for high-performance computing systems and cloud service providers (MEI et al., 2018). Failure in storage devices can lead to the unavailability of data in DCs, which can lead to huge financial losses as a result of fines set in SLAs (WANG et al., 2019). In addition, the progressive increase in data generation may result in the exhaustion of storage systems, resulting in more storage device failures.

HDD failures can occur under various operating conditions. For example, factors such as temperature, humidity, distinct workloads, and operating hours can affect the same model differently (MEI et al., 2018). Internal factors are also sources of disturbance and errors. In this case, the control systems must achieve an exact level of regulation concerning servo mechanisms.

The lifetime of an HDDs is primarily related to the wear of its electromechanical components. Spindle motors and actuators are components that may suffer premature wear if the workload subjected to HDDs requires numerous movements (e.g., workloads composed mainly of random requests) (ZHANG et al., 2019a). Despite this, the average time to failure reported by HDDs manufacturers (i.e., the average time to replace a hard-disk device) is generally longer than that for solid-state devices.

By contrast, HDDs have a shorter average time to data loss than SSDs (LI et al., 2019). Although they contain fairly robust mechanical components with long lifespans HDDs, they are still subject to failures of another natures, such as error sources that can affect the transfer of bits during the processing of a given request (WANG et al., 2019). The primary sources of errors that can affect the reliability of data transfer in HDDs are vibrations, external shocks, imprecision of the reading and programming heads, and mechanical resonances in the actuator and disk.

2.1.2 **Solid-state drives**

SSDs incorporate solid-state memory for nonvolatile data storage (MICHELONI; MARELLI; ESHGHI, 2012). The evolution of SSDs represents a significant change in storage systems because such devices can achieve excellent throughput and mean response time values, particularly when subjected to workloads composed mostly of random requests (AGRAWAL et al., 2008).

In enterprise environments, SSDs have become essential resources for improving the performance of data storage systems (WU et al., 2018). For instance, migrating services to cloud computing platforms has expanded the number of parallel applications that can be run on a single DC. Consequently, transmission bottlenecks in the data storage system can lead to significant financial losses; furthermore, random access accounts for a large proportion of online transfers by large companies (WU et al., 2018).

With regard to computer architecture, SSDs typically have an interface-integrated controller (e.g., PCI-Express, Serial Attached SCSI, or Serial Advanced Technology) to physically connect to the host server physically. In solid-state devices, data management is performed using an SSD controller. This component is responsible for wear leveling, garbage collection, bad block management, and the mapping of logical blocks to physical blocks. These mechanisms constitute a flash translation layer (EL MAGHRAOUI et al., 2010). Specific hardware executes an error correction code (ECC) for error identification and repair, which is usually shared among multiple flash channels (MICHELONI; MARELLI; ESHGHI, 2012; WOO; KIM, 2013). Figure 2.2 shows the architecture and functionality of SSD components.

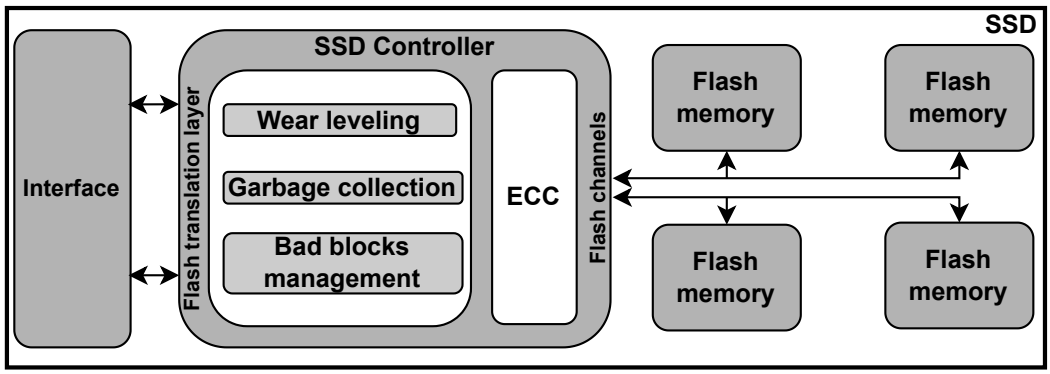


Figure 2.2: SSD architecture and controller functionalities (own work (2023)).

2.1.2.1 Performance

In SSDs, data storage is stored in flash NAND (Not And) memories. A NAND memory chip consists of several blocks, usually 64 to 128 pages (MAO et al., 2012). A page is the standard granularity for writing data to solid-state devices, and is usually 4KB.

Although they provide excellent speed for random data access (when compared to HDDs), flash memory has some limitations. For example, writing data to flash memory requires deleting the entire memory block in which the data will be stored. Consequently, relevant stored infor-

mation must be rewritten in another memory space (erase-before-write) (WOO; KIM, 2013). Excessive deletion operations are a performance bottleneck for flash memory (RICHTER, 2013) and can amplify the internal fragmentation of data storage devices (BREWER; GILL, 2011).

Unlike in SSDs, the time required for data access in HDDs depends on the speed at which the mechanical components can be moved (MEI et al., 2018). For example, the maximum data throughput in HDDs is dictated by the rotational speed of the spindle motor in addition to the transmission capacity of the read and program heads. Thus, although HDDs obtains significant performance results when processing sequential requests, the limitations of their mechanical components remains the predominant reason for their poor results when dealing with random requests (MEI et al., 2018).

Compared with HDDs, solid-state devices perform better in terms of power consumption, mean response time, impact resistance, and IOPSs for random read requests (MEI et al., 2019). For example, the average response time in SSDs for processing random read requests can be on the order of microseconds, whereas HDDs requires milliseconds. However, despite the increased production and reduced price of flash memory, its high cost per gigabyte still makes it infeasible to exclusively adopt solid-state devices (WANG et al., 2022).

2.1.2.2 Reliability

The execution of several writes and rewrite operations during the lifetime of flash memory will eventually damage the nonvolatile cell components. Inevitably, the injection and removal of electrons (storing or deleting data) causes irreversible damage. The degradation of flash memory cells is estimated according to the number of write/delete cycles, averaging 100K cycles for single-level cells (SLC) (WU et al., 2018). The limited programmability of flash memory cells is the main reason why the lifespan of HDDs is longer than that of solid-state devices (LI et al., 2019).

To increase SSDs capacity, multilevel cells (MLC) have been used instead of single-level cells. This occurs because SLCs allow the storage of only one bit, whereas MLCs can double this capacity (LI et al., 2019). However, the MLC approach has disadvantages in terms of the lifetimes of the flash memory cells. This is because MLCs tolerate approximately 10K P/Es per block for faults to arise, whereas SLCs support approximately 100 K P/Es (LI et al., 2019; MEI et al., 2019).

The wear-leveling mechanism aims to minimize and uniformly utilize NAND memory

blocks. Therefore, the maximum number of P/Es per block is estimated and considered in the execution of this technique. This data distribution entails running a garbage collection mechanism to identify data that can be discarded (LI et al., 2019). Starting at a limit value of free blocks (set by the SSD manufacturer), the garbage collection starts checking for existing copies of the same file, then deletes the duplicates. This activity can harm memory performance; therefore, the garbage collectors typically operates in the background (MEI et al., 2019).

The balance between these two techniques (wear leveling and garbage collection) can delay the lifetime of SSDs, and thus, the emergence of unused blocks. However, owing to programming limitations in flash memory cells, the appearance of bad blocks is inevitable. The unused block management module identifies and maps unused blocks. For this purpose, a new blocks table (bad blocks table) is created upon the first initialization of the memory (MEI et al., 2019), which contains a list of the bad blocks present in the factory test and is subsequently updated during the use of the solid-state device.

2.2 HYBRID STORAGE SYSTEMS

Solid-state devices can be used in conjunction with magnetic disk devices because they have an input and output interface similar to that of HDDs (LIN et al., 2017). Various studies have devised new techniques for developing hybrid storage systems (SALKHORDEH; BRINKMANN, 2019; WOO; KIM, 2013); however, improvement attempts have usually focused on the software layer (i.e., the storage controller). To provide a better understanding of this topic, this section presents two representative architectures for hybrid storage systems proposed by several researchers (XIE et al., 2018; NAKASHIMA; KON; YAMAGUCHI, 2018; WU et al., 2018).

2.2.1 **SSD as Cache**

Owing to the low performance of magnetic disks in handling random requests and the high cost of traditional (RAM-based) cache memories, SSDs have, in principle, become a suitable solution for improving the throughput and response time of computer systems. As modifications to this approach are usually minimal, several studies have adopted SSDs as a caching mechanism (LEE; MIN; EOM, 2015; WU et al., 2015; BU et al., 2012).

Figure 2.3 depicts a hybrid storage system using an SSD as the cache and an HDD for

persistent storage. This architecture adopts the write-back storage policy, which is a common data-management mechanism performed by a storage controller (LEE; MIN; EOM, 2015; APPUSWAMY; MOOLENBROEK; TANENBAUM, 2012). This policy is characterized by directing all write operations to the cache and periodically to the primary disk to reduce the response time, access the most recent requests, and leverage the substantial reliability of the magnetic disk systems.

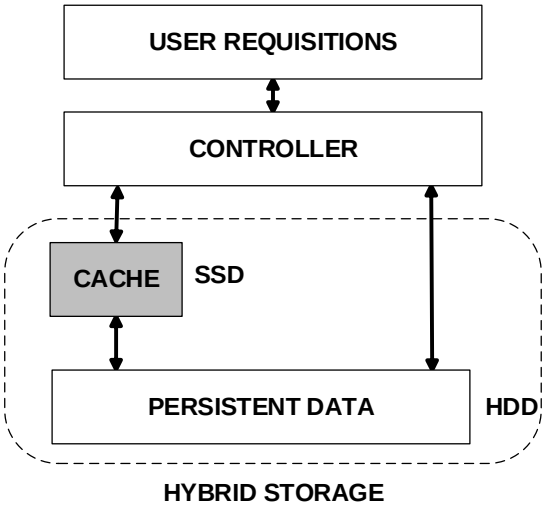


Figure 2.3: Hybrid storage with a SSD as cache for HDD (BORBA; TAVARES, 2017).

2.2.2 SSD for Random Requests

Small-object requests in a storage system can cause significant performance degradation (CHEN; KOUFATY; ZHANG, 2011), and in this context, metadata may have a considerable impact. Metadata blocks contain attributes related to each stored file, such as its location on the drive and its size. Thus, metadata must be stored in memory before a file can be manipulated, which significantly increases the number of input and output requests (MAO; WU; JIANG, 2015). These blocks, although generally small (STRUNK, 2012), account for 99% of I/O operation time (CARNS et al., 2011). In HDDs, metadata manipulation significantly affects the performance because of the rotations required to access the metadata and the data it refers to, as both are usually in different chunks.

Because data searches usually follow a random-access pattern, several authors have suggested storing metadata blocks on SSDs (CHEN; KOUFATY; ZHANG, 2011; APPUSWAMY; MOOLENBROEK; TANENBAUM, 2012; WU et al., 2015). According to the results of these

studies, SSDs are a suitable mechanism for reducing delays in random access to these blocks. However, using SSDs exclusively for this purpose (metadata) can detract from the possible benefits that can be achieved when storing other types of data on this device. Therefore, performance improvements must consider factors such as the number of metadata blocks, operation type (write or read), and access pattern (sequential or random). Furthermore, it is important to state that such intense use of random requests may negatively affect the endurance of an SSD; thus, this aspect should not be neglected.

Figure 2.4 illustrates a system in which SSDs are used for storing metadata blocks and other random data accesses operations, whereas HDDs are responsible for storing sequential data. Pattern recognition, that is, whether the workload is random or sequential, is a possible procedure using both software- and hardware-based approaches (NIJIM et al., 2011; JOO et al., 2014). For example, system calls in the Linux operating system kernel and the firmware of drives (e.g., HDD, SDD, and hybrid) (JOO et al., 2014; NIJIM et al., 2011; CHEN; DING; JIANG, 2009) can detect whether a given read or write operation has a sequential pattern by observing the request size, frequency, and distance between blocks.

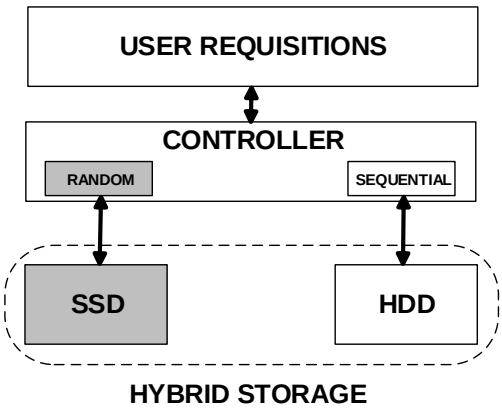


Figure 2.4: Random data placement policy for hybrid storage (BORBA; TAVARES, 2017).

2.3 PERFORMANCE EVALUATION

Performance evaluation is a prerequisite for every stage in the life of a computer system, from design to manufacturing, and for possible future enhancements (JAIN, 1990). Computer systems appear in many areas and in various forms, such as embedded systems in cars, online

banks, data centers, and smartphones. The widespread adoption of such systems demands that both developers and users pay attention to the performance of the adopted equipment (LILJA, 2005).

When applied to computer science and engineering experiments, a performance analysis should be conducted according to a combination of measuring, interpreting, and reporting the studied metrics of a given computer system. However, it is often necessary to analyze only a small independent portion of a system, such as a storage devices. Unfortunately, some components can have very complex interactions that may constitute a challenge in making decisions regarding the techniques, workload, and tools to be used.

In principle, for a performance evaluation, it is typically necessary to represent the characteristics of the applications to be run on the system to be evaluated. Then, a real workload can be collected by observing the system under normal operating conditions. However, these conditions in a real system may be unlikely to be repeated and may even take a long time. Therefore, a synthetic workload may be suitable depending on the experiment in question. In addition to being similar to real workloads, synthetic workloads enable the investigation to be repeated in a controlled manner, thereby allowing for a more precise analysis of the system parameters. For example, the following workloads can be used to compare computer systems: instruction addition, write and read operations, instruction mixing, kernel operations, synthetic programs, and comparative applications.

The measurement of a real system, simulation, and analytical modeling are the three fundamental techniques for evaluating the performance of systems (JAIN, 1990). The measurement technique involves collecting information from a real system regarding the specific aspects under investigation. Although this approach can provide reliable information regarding a system, one desirable performance evaluation characteristic is the tracking of behavioral differences as the settings change, which this technique may not adequately address. Therefore, evaluating the impact of modifying only one component may cause in a complex system can prove challenging. In addition, measurements in real systems can be time-consuming and costly, as such equipment may need to be purchased and observed for long periods to conduct a proper study (JAIN, 1990).

Analytical modeling techniques and simulations are not significantly affected by the abovementioned disadvantages regarding the measurement of real systems (JAIN, 1990). The simulation of a computer system is performed by using a program designed to model the essential features to be analyzed (LILJA, 2005). The program can be modified to study the impact of

changes in the simulated components. Depending on the level of detail of the simulated system, the cost and time required for the analysis can be significantly reduced compared to experiments on real machines. However, the difficulty in covering all the details and the reduced time required to develop the simulator and run the simulator may limit the accuracy of the results.

Analytical models may be a simpler, more accurate, and less costly solution for evaluating system performance (JAIN, 1990). An analytical model is a mathematical model with a closed-form solution used to describe a system (LILJA, 2005). Usually, analytical modeling provides a better understanding of the effects of the system parameters and their interactions. In addition, it can help to validate the results produced by a simulator or the measured values in a real system.

Queueing theory is an important analytical modeling technique for systems. Many tasks in computer systems do not share resources such as processor cores, disks, and network interfaces. Nevertheless, when considering a system with only one resource for each piece of equipment, the tasks must be executed individually, thereby generating queues. One of the purposes of queue theory is to precisely determine the time spent on tasks, from the time in the queue to their processing, that is, the time within the system.

One of the most common theorems used in the context of queue theory is Little's law, which is adopted in this thesis to analyze important aspects of the processing time of requests in storage systems. The average response time of the system can be obtained using Equation 2.1 (JAIN, 1990).

$$R = \frac{L}{\lambda} \tag{2.1}$$

where R is the average response time, L is the average number of requests, and λ is the task arrival rate. This relationship can be applied to a system in which the number of incoming tasks is equal to the number of completed tasks, i.e., the system is not overloaded.

2.4 DEPENDABILITY EVALUATION

Dependability is characterized as the ability of a system to provide services reliably (MACIEL et al., 2011). Typically, the concept of dependability covers the following metrics: availability, reliability, security, confidentiality, integrity, and maintainability. The criteria established for these attributes can be qualitatively evaluated in systems (BERNARDI; MERSEGUER; PETRIU, 2012). Figure 2.5 shows the systematic organization of the concepts related to dependability.

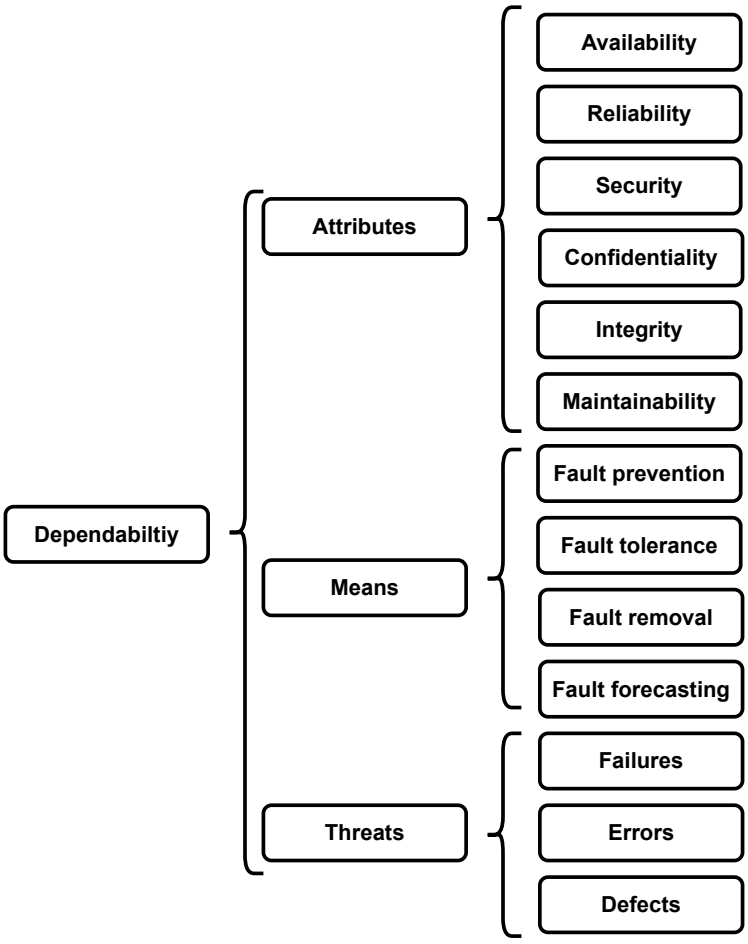


Figure 2.5: Dependability concepts (adapted from AVIZIENIS; LAPRIE; RANDELL (2001)).

Dependability attributes define the ability of a system to provide the specific functionality for which it was designed. However, certain *threats* can cause a system to behave differently than expected. Specifically, a fault can be defined as the cause of an error, which is a part of the system state that can lead to system failure. Therefore, an error can be defined as an intermediate stage between a fault and failure (BERNARDI; MERSEGUER; PETRIU, 2012). A fault can be considered a failure if it refers to a specific component of a system (MACIEL et al., 2011).

Four techniques (*means*) can be used to define the reliability of a system within the context of dependability (BERNARDI; MERSEGUER; PETRIU, 2012). *Fault prevention* concerns the methods employed during a system's design and production phases to prevent undesirable future occurrences (AVIZIENIS et al., 2001). *Fault removal* occurs during the development and operation phases. Therefore, there are three stages: verification, diagnosis, and correction. However, despite the initial planning, *fault tolerance* strategies must be applied to preserve the service offered, even in the presence of failures. Considering the planning aspects, that is, predicting possible undesired behavior (*fault forecasting*), the evaluation during the system operation seeks to identify whether the dependability attributes are satisfied in advance.

The **reliability** attribute represents the probability that a system will perform the intended functions for which it is designed within a specific time without the occurrence of failures (MACIEL et al., 2011). This relationship is mathematically expressed by Equation 2.2, where T is a continuous random variable representing the time to failure of a system. For a given value of t, $R(t)$ is the probability that the time to failure is greater than or equal to t (EBELING, 2004).

$$R(t) = P\{T \geq t\}, T \geq 0 \tag{2.2}$$

Therefore, if we consider $P\{T < t\}$, the failure probability up to instant t can be obtained. Equation 2.3 shows this relationship, in which $F(t)$ represents the cumulative distribution function of the failure distribution (EBELING, 2004).

$$F(t) = 1 - R(t) = P\{T < t\}, T \geq 0 \tag{2.3}$$

Availability is the probability that a given system is in a working condition (MACIEL et al., 2011). In particular, steady-state availability can be expressed as a function of the mean time to failure (MTTF) and mean time to repair (MTTR) (Equation 2.4) (KANOUN; SPAINHOWER, 2008).

$$Availability = \frac{MTTF}{MTTF + MTTR} = \frac{uptime}{uptime + downtime} \tag{2.4}$$

where *uptime* represents the system uptime and *downtime* is related to the downtime period.

The mean time to failure specifies how long a given system or subsystem will function correctly (MODARRES; KAMINSKIY; KRIVTSOV, 2009); that is, it represents the expected time for a failure to be observed (BERNARDI; MERSEGUER; PETRIU, 2012). Estimating the MTTF requires knowledge of the statistical distribution of the time-to-failure values (*failure distribution function*) (KAPUR; PECHT, 2014). For example, in the case of the exponential distribution, which has a constant failure rate (EBELING, 2004), the MTTF calculation follows the Equation 2.5, where $f(t)$ represents the probability density function, and λ is the failure rate.

$$MTTF = \int_0^\infty R(t)dt = \int_0^\infty tf(t)dt = \int_0^\infty e^{-\lambda t}dt = \frac{1}{\lambda} \tag{2.5}$$

Maintainability is the probability that a failing system or component will be restored or repaired to a specific condition within a specified period of time (EBELING, 2004). Similar to reliability, maintainability is characterized by a probability distribution; however, in this case, it considers the time to repair. Maintainability is described by Equation 2.6, which represents the probability that the repair will be completed within time t, where $h(t)$ is the probability density function.

$$P(T \leq t) = \int_0^t h(t)dt \tag{2.6}$$

The mean time to repair (MTTR) is typically used to quantify maintainability. However, similar to the MTTF calculation, the statistical distribution must also be considered (EBELING, 2004). Equation 2.7 represents the calculation used to obtain this value, where $H(t)$ is the cumulative distribution function.

$$MTTR = \int_0^\infty th(t)dt = \int_0^\infty (1 - H(t))dt \tag{2.7}$$

Analytical models have been widely adopted for dependability assessments (BERNARDI; MERSEGUER; PETRIU, 2012). A model is an abstraction of a system whose purpose is to enable understanding before it is designed. A dependability model considers the abstractions required to represent system failures and their consequences. Modeling can then be defined

according to the interaction or structure of the components of a system.

For more complex interactions and representing dependencies between systems, state-space models can provide a more accurate representation by analyzing the dynamic behavior when events occur (MACIEL et al., 2011). Combinatorial models relate to the structural relationships between the elements of a system; however, they assume that the failure or recovery of one component is not affected by the behavior of another element.

Reliability block diagrams (RBD), fault trees, and reliability graphs are representative combinatorial models, whereas Markov chains and stochastic Petri nets (SPN) are the most widely used state space-based models (MACIEL et al., 2011).

2.5 CONTINUOUS MARKOV CHAINS

The Markov chain is a mathematical formalism based on state spaces, proposed in (MARKOV, 1906) for modeling systems in several areas, both for descriptive purposes and for analysis. A Markov model can be described as a discrete state space diagram associated with a Markov process, that is, a case of stochastic processes (BOLCH et al., 2006).

A stochastic process is a collection of random variables ($X(t)$) indexed by a parameter t belonging to a set T. Often, T is taken to be a set of non-negative integers (although other sets are perfectly possible), and $X(t)$ represents a measurable characteristic of interest at time t (BOLCH et al., 2006). The set of all possible values of $X(t)$ (for each $t \in T$) is called the state space S. If set T is discrete, the process is classified as discrete-time; otherwise, it is considered continuous. Similarly, the state space S can be discrete or continuous; consequently, the stochastic processes can also be discrete or continuous. In this thesis, discrete times are not adopted; therefore, the continuity section focuses on continuous-time stochastic processes ($T = \{t : 0 \leq t < \infty\}$).

A stochastic process is said to be Markovian if (HAVERKORT, 2000),

$$P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k, X(t_{k-1}) = x_{k-1}, \ldots X(t_1) = x_1, X(t_0) = x_0\} = P\{X(t_{k+1}) \leq x_{k+1} | X(t_k) = x_k\}, \tag{2.8}$$

for all $t_0 \leq t_1 \leq \ldots t_k \leq t_{k+1}$. This means that a stochastic process is said to be a Markovian process only if the future state depends exclusively on the present state ($X(t_k) = x_k$) and not on the previous states. Therefore, this particular case of stochastic processes is also called the

memoryless process (HAVERKORT, 2000).

A Markov chain can be represented as a state diagram, where the vertices represent the states and the arcs represent the transitions between states. Transitions between states represent the occurrence of events (MACIEL et al., 2011). The weights of the transitions are assigned according to the type of random variable adopted to represent the duration of the events in the system. For example, for *Discrete-Time Markov Chain*s (DTMCs), one can assign a value $0 < p < 1$ to the weight of an arc, where p represents the probability of transition from state s_i to s_j. For a *Continuous-Time Markov Chain* (CTMC), the value assigned to the arc weight between two transitions represents the rate at which the change in state occurs (BOLCH et al., 2006). Owing to the Markov (memoryless) property, the time between activities must follow a memoryless distribution. Therefore, in CTMCs, an exponential distribution is adopted (BOLCH et al., 2006).

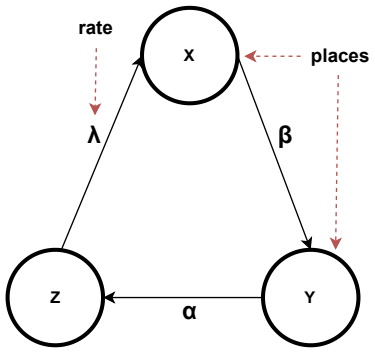


Figure 2.6: CTMC example (own work (2023)).

Figure 2.6 shows an example of a CTMC with three states, which is adopted to explain the following analysis method. Markov chains can be represented in a matrix form (transition rate or generator matrix). The generating matrix Q is composed of components q_{ii} and q_{ji}, where q is the transition rate from state i to j, and $\sum q_{ij} = -q_{ii}$. Then, for this hypothetical CTMC with the state space $S = \{X,Y,Z\} = \{0,1,2\}$, it can be stated that the resulting generating matrix Q is

$$Q = \begin{pmatrix} q_{00} & q_{01} & q_{02} \\ q_{10} & q_{11} & q_{12} \\ q_{20} & q_{21} & q_{22} \end{pmatrix} = \begin{pmatrix} -\beta & \beta & 0 \\ 0 & -\alpha & \alpha \\ \lambda & 0 & -\lambda \end{pmatrix} \quad (2.9)$$

The stationary analysis of a Markov chain consists of determining the probability that the system will reach a specific state over a long runtime. These probabilities are independent of the initial state of the system. They are represented by the vector $\pi = \{\pi_1, \pi_2, \pi_3, ..., \pi_n\}$, where

π_i is the stationary probability of state i. Equation 2.10 estimates the state probability vector.

$$\pi \times Q = 0, \quad \sum_{i \in S} \pi_i = 1 \tag{2.10}$$

Although Markov chains are a fairly representative mathematical formalism, this thesis adopts generalized stochastic Petri nets (GSPNs) for the proposed models. However, as presented in the next section, generating an equivalent CTMC is one step in performing a stationary analysis of GSPNs (BOLCH et al., 2006; KLEINROCK, 1975). From the CTMC state space, the metrics assumed in a given GSPNS model can be estimated through numerical analysis.

2.6 STOCHASTIC PETRI NETS

This section explains the mathematical formalism adopted in this thesis to represent the data storage systems. This formalism has been adopted to create analytical models for evaluating the performance, energy consumption, and dependability of such systems. First, the concept of Petri nets is introduced to facilitate the understanding of its fundamentals. Then, its extension, generalized stochastic Petri nets, is presented.

2.6.1 Petri nets

PNs are a family of formalisms suitable for modeling various systems because of their features such as concurrency, synchronization, asynchronism, distribution, and non-determinism are well represented (MURATA, 1989). Introduced by Carl Adam Petri in 1962 (PETRI, 1962; BOLCH et al., 2006), Petri nets were not originally developed for the purpose of performance evaluations, despite their ability to represent complex systems (FRANCÊS, 2003). As a graphical tool, Petri nets can be used as a visual communication aid, similar to flow charts and block diagrams (MURATA, 1989), thus, enabling a description of the existing relationships between conditions and events (O'CONNOR; O'CONNOR; KLEYNER, 2012). The elements constituting a Petri net are shown in Figure 2.7, which is explained below.

In general, a Petri net is a directed bipartite graph consisting of two types of *nodes*: *place* and *transition*. Graphically, places are represented by circles or ellipses (REISIG, 2013) and are associated with a passive component intended to portray a condition or store an object (BAUSE; KRITZINGER, 2002). The changing conditions of a system, which can also be seen as a change in values, are represented by transitions, which are symbolized by a rectangle and are

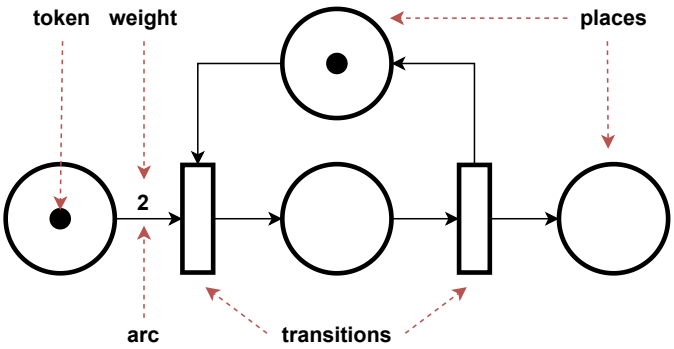


Figure 2.7: Petri net elements (own work(2023)).

characterized as the active components of Petri nets (REISIG, 2013).

Because it is a bipartite graph, the connection between elements must be made by considering both types of nodes; that is, a place can only connect to a transition, and vice versa (BAUSE; KRITZINGER, 2002). Places and transitions are directly connected by *arcs*. Graphically, an arc is represented by an arrow and does not constitute a system component but only an abstract relationship, such as a logical connection (REISIG, 2013). Two types of arcs exist: *input arcs* and *output arcs*. Input arcs represent a connection from an *input place* to a transition, whereas the output arcs connect a transition to an *output place* (BOLCH et al., 2006).

For a transition to be executed (*fired*), it must be enabled. A transition is enabled if all its entry places have at least one mark (*token*). A token, graphically represented by a black dot, portrays the system's state at a given moment (O'CONNOR; O'CONNOR; KLEYNER, 2012). The *firing* of an enabled transition represents the execution of an action that causes the absorption and generation of tokens at the input and output locations, respectively (Figure 2.8), thereby taking the model to a new state of marking (O'CONNOR; O'CONNOR; KLEYNER, 2012).

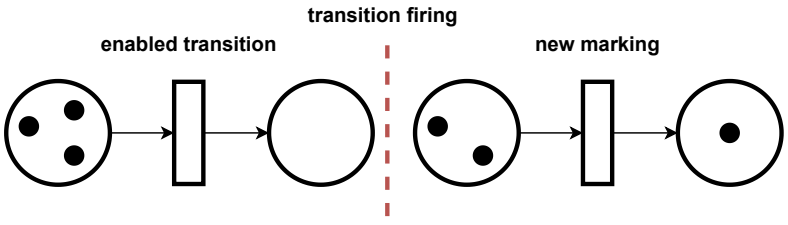


Figure 2.8: Firing of a transition (own work (2023)).

The formal definition of a Petri net is as follows (MURATA, 1989):

Definition 2.1. *A Petri net is a 5-tuple,* $PN = (P, T, F, M, M_0)$*, where:*

- $P = \{p_1, p_2, ..., p_m\}$ *is the finite set of places*;

- $T = \{t_1, t_2, ..., t_n\}$ *is the finite set of transitions*;
- $F \subseteq (P \times T) \cup (T \times P)$ *is the set of arcs*;
- $M : F \to \aleph$ *is the function that assigns weight to the arcs*;
- $M_0 : P \to \aleph$ *is the initial marking, where* $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

All reachable markings and steps of a Petri net can all be compiled into a *reachability graph* (RG). In this directed graph, the nodes represent reachable markings, whereas the edges reflect the transitions between them. In principle, the reachability graph is an appropriate starting point for net analysis and verification of some (behavioral and structural) properties, as long as it presents a finite number of reachable markings (REISIG, 2013). Figure 2.9 illustrates a hypothetical Petri net and its corresponding reachability graph.

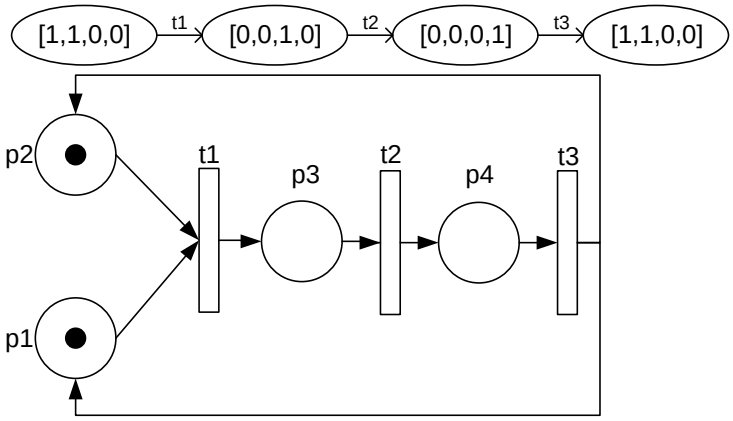


Figure 2.9: Petri net and corresponding reachability graph (own work (2023)).

2.6.1.1 Petri nets properties

The study of Petri net properties allows for the analysis of several characteristics of a modeled system. The properties of Petri nets can be divided into two major groups: behavioral and structural (MACIEL; LINS; CUNHA, 1996; MURATA, 1989).

The behavioral properties depend on the markings of the Petri net model. The definitions of the behavioral properties addressed in this thesis are as follows:

- **Reachability**: indicates the possibility of reaching a given marker by firing a finite number of transitions from an initial marker. Given a marked Petri net $RM = (R, M_0)$,

the firing transition t_0 changes the marking of the Petri net. A marking $M^{'}$ is accessible from M_0 if there exists a sequence of transitions that, after firing, leads to marking $M^{'}$. That is, if marking M_0 enables transition t_0, firing this transition achieves the marking M_1. Marking M_1 enables t_1, which, if fired, achieves M_2, and so on, until marking $M^{'}$ is obtained;

- **Boundedness**: a Petri net is *bounded* if, for any reachable marking$M^{'}$ from the initial marking M_0, the number of tokens never exceeds the value k. Thus, this net is said to be k-bounded;

- **Liveness**: this property is defined as a function of the possibility of firing transitions. A Petri net is considered *live* if, regardless of the markings that are reachable from M_0, it is always possible to fire a transition in the Petri net through a sequence of firing transitions. This property allows us to analyze whether events that will never be fired have been included in the Petri net model. If a model is live, this means that it is deadlock-free;

The structural properties of Petri nets reflect their marking-independent characteristics. These properties depend solely on the structures of the Petri nets. The structural properties discussed in this thesis are defined as follows:

- **Boundedness:** a Petri net $R = (P, T, F, W, M_0)$ is classified as structurally bounded if it is bounded (it maintains the number of tokens) for any initial marking;

- **Conservativeness:** this property allows for the verification of non-destruction or creation of resources through the conservation of token marks in a given Petri net. A net is considered conservative if the weighted sum of the marks in the net does not changed for any possible firing sequence;

- **Consistency:** a Petri net is considered consistent if firing a sequence of transitions from an initial label M_0 returns to the same initial label M_0; but, all transitions are fired at least once. A net can be considered *partially consistent* if $M_0[s > M_0$ and some transitions t_i ($t_i \in T$) fire at least once in a sequence of transitions s.

2.6.2 Generalized stochastic Petri nets

Petri nets have proven to be suitable for modeling computer systems because of their ability to represent concurrency and synchronization. Petri nets have demonstrated to be a widely used resource for evaluating system properties by enabling the representation of the relationships between events and conditions. However, performance and dependability analyses are unfeasible because time is not considered in its definition. This demand has led several authors to propose modifications to the basic definition to obtain a modeling tool more applicable to the representation of real systems (AJMONE MARSAN; CONTE; BALBO, 1984a).

In particular, including time associated with a transition specifies the delay between enabling and firing it. In this context, Zuberek et al. (ZUBEREK, 1980) established a fixed time to model the performance of a computational system at a specific level. Moreover, Merlin et al. (MERLIN; FARBER, 1976) introduced the timed Petri nets including a maximum and minimum time to fire each transition. In contrast, Molloy et al. (MOLLOY, 1982) proposed a *Stochastic Petri Net* (SPN) in which the firing time of the transitions is randomly and exponentially distributed.

Using exponential distribution to define the temporal characteristics makes this extension valuable, mainly because of its memoryless property. This property makes it unnecessary to distinguish between the distributions of the current delay and those that are yet to occur (BALBO, 2001), thus, making the reachability graph of a bounded SPN isomorphic to a continuous-time Markov chain (MURATA, 1989; TRIVEDI, 2008). For example, this characteristic allows the computation of steady-state probabilities of a marking.

The introduction of SPNs makes it possible to combine the concepts of graphical and probabilistic models, thus, becoming a useful tool for estimating the performance of a system, as well as an alternative to the generation of Markov chains through the adoption of simulation techniques (TRIVEDI, 2008). Its limitations are associated with the size of the represented system, which can increase the complexity of its graphical representation and the number of states of the associated Markov chain (in the case of stationary analyses). Therefore, SPNs are appropriate for modeling systems with a limited state space (AJMONE MARSAN; CONTE; BALBO, 1984a).

Next, a formal definition of stochastic Petri nets is presented (BAUSE; KRITZINGER, 2002):

Definition 2.2. *Formally, an SPN is a 2-tuple, SPN* $= (PN, \Lambda)$*, where:*

- $PN = (P, T, F, M, M_0)$ *is formed by the Petri net discussed in Definition 2.1*;
- $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_n\}$ *is the set of rates, where the rate* λ_i *is associated with the transition* t_i.

Depending on the complexity of the system, not associating random times for the representation of an action may become desirable. Moreover, representing short activities only logically can be particularly convenient, especially if the number of states in the generated Markov chain is reduced. In this regard, Balbo et al. (AJMONE MARSAN; CONTE; BALBO, 1984a) introduced generalized stochastic Petri nets, which have two types of transitions: timed, represented by a white rectangle, and *immediate*, denoted by a black rectangle. By definition, immediate transitions have no delay, whereas timed transitions are associated with an exponential distribution time, as mentioned in the definition of SPNs. Note that immediate transitions, when enabled, take precedence over timed transitions (MARSAN et al., 1994).

Other extensions have been proposed for GSPNs, the most relevant of which are explained as follows (BOLCH et al., 2006):

- **Inhibit arc:** an inhibit arc connects a place to a transition and is graphically represented by a line with a white circle at the end opposite the place. When the number of marks in a place is equal to or greater than the multiplicity constrained by the arc, the transition is disabled;
- **Priorities:** although inhibit arcs can be used to specify priority relationships, such assignments are best defined when they are explicitly introduced into a paradigm. Priorities are specified by the integers associated with these transitions. Thus, a transition t_i can be enabled if its priority is higher than that of the other transitions in the net, that is, $t_i > t_n$;
- **Weight:** if weights w_i and w_j are associated with the respective immediate transitions, t_i and t_j, and only both are enabled, the firing probability of t_i is given by $w_i/(w_i + w_j)$ (BAUSE; KRITZINGER, 2002);
- **Server semantics:** this thesis addresses *single-server* and *infinite-server* semantics. Enabling an immediate single-server transition allows the absorption of tokens in an individual manner; that is, the occurrence of a new firing is conditional on the

completion of the previous delay. As for infinite-server semantics, the absorption of tokens is performed in parallel, so as the respective transition is enabled, it can be fired (MARSAN et al., 1994).

Considering the presented GSPN extensions, their formalism can be defined as follows:

Definition 2.3. *Formally, a GSPN is an 8-tuple,* $GSPN = (P, T, I, O, H, \Pi, W, M_0)$*, where:*

- $P = \{p_1, p_2, ..., p_m\}$ *is the finite set of places*;
- $T = \{t_1, t_2, ..., t_n\}$ *is the finite set of immediate* (T_{im}) *and timed transitions* (T_{timed})*,* $P \cap T = \emptyset$ *e* $T = T_{im} \cup T_{timed}$;
- $\Pi : T \to \aleph$ *is the priority function, where* $\Pi(t) \geq 1$*, if* $t \in T_{im}$*, or* $\Pi(t) = 0$*, if* $t \in T_{timed}$;
- $I, O, H : T \to Bag(P)$ *are the input, output, and inhibit functions, respectively, where* $Bag(P)$ *is the multiset of* $P(Bag(P) : P \to \aleph)$;
- $W : T \to \Re$ *is the weight or rate assignment function, where* $W(t) = w_t$*, if* $t \in T_{im}$*, or* $W(t) = \lambda_t$*, if* $t \in T_{timed}$;
- $M_0 : P \to \aleph$ *is the initial marking, where* $P \cap T = \emptyset$ *e* $P \cup T \neq \emptyset$.

Adding the concept of immediate transitions to GSPNs has increased their ability to model real systems; however, their analysis is more complex than that of SPNs. The isomorphism between SPNs and CTMCs does not occur in the same way for GSPNs because of the existence of two types of markings: *vanishing* and *tangible* (MARSAN et al., 1994). Volatile markings represent states upon enabling at least one immediate transition, whereas tangible markings are associated with timed transitions.

For a given GSPN, an *Extended Reachability Graph* (ERG) is generated, containing information from both types of markings. However, to avoid stochastic discontinuity, volatile markings must be eliminated to obtain a reachability graph isomorphic to a corresponding Markov chain (MARSAN et al., 1994). Eliminating vanishing markings is an essential step in generating an equivalent CTMC. Two techniques can be employed: *on-the-fly* and *post-elimination*.

Figure 2.10 shows the steps for eliminating volatile markings using an on-the-fly technique. Initially, the respective ERG is generated, which allows the identification of volatile markings to be disregarded. Volatile markings are then avoided by redirecting the arcs to the

markings that should constitute the reachability graph (in this example, $m0$ connects $m3$ and $m4$, avoiding $m1$) (BOLCH et al., 2006). Finally, the corresponding CTMC is generated, allowing for stochastic analysis. Notably, this splitting considers the probabilities of the immediate transitions involved in the process and associates them with the rates of the resulting arcs.

Generating a CTMC from a GSPN makes it possible to estimate the performance and dependability metrics of the represented system. The equivalent CTMC can then be computed using numerical analysis (stationary analysis). Although it provides accurate results, the stationary analysis of a GSPN is a solution method that requires exponential statistical distributions associated with timed transitions (BOLCH et al., 2006). An alternative to this restriction is the use of the *moment matching* technique, with which it is possible to approximate nonexponential delays using *phase-type distributions* (Section 2.6.3). Simulation techniques can also be used to obtain performance and dependability metrics (BOLCH et al., 2006). However, it is important to note that simulation methods provide results based on a particular significance level, whereas numerical analyses of a GSPN result in a single-point value (TUFFIN et al., 2007).

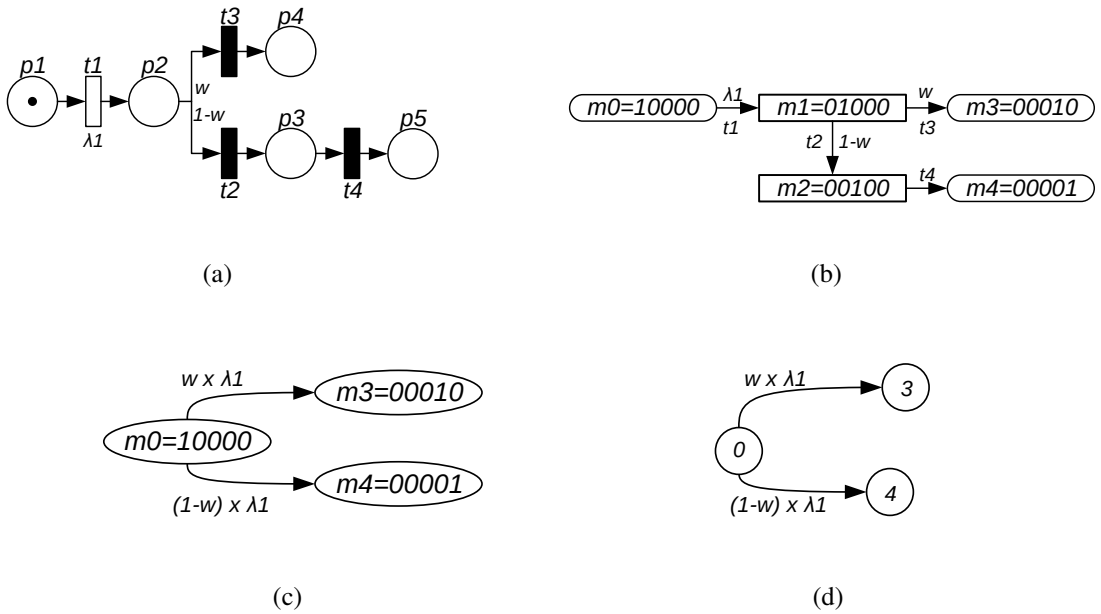


Figure 2.10: Eliminating vanishing markings demonstrated by (a) a given GSPN, (b) equivalent ERG, (c) resulting RG, and (d) corresponding CTMC (adapted from BOLCH et al. (2006)).

2.6.3 Phase-type distributions

In GSPNs, timed transitions are associated to exponential distributions; however, a non-exponential delay can be approximated using phase-type distributions (DESROCHERS; AL-JAAR, 1995), more specifically, Erlang, hyperexponential and hypoexponential. A trapezoidal transition, namely the s-transition (Figure 2.11), is adopted to denote a subnet, which models a delay using a phase-type distribution.

This thesis utilizes the technique described in (DESROCHERS; AL-JAAR, 1995), in which an algorithm adopts the inverse of the coefficient of variation (CV): $1/CV = \mu_d/\sigma_d$. μ_d is the mean delay, and σ_d is the standard deviation. The algorithm is as follows:

- If $\mu_d = \sigma_d$, only a single timed transition is adopted;
- Assuming $\mu_d/\sigma_d \in \mathbb{N}$ and $\mu_d/\sigma_d \neq 1$, the phase approximation considers an Erlang subnet (Figure 2.12), such that $\gamma = \left(\frac{\mu_d}{\sigma_d}\right)^2$ and $\lambda = \gamma/\mu_d$;
- Considering that $\mu_d > \sigma_d$, a hypoexponential subnet is adopted (Figure 2.13) and

$$\left(\frac{\mu_d}{\sigma_d}\right)^2 - 1 \leq \gamma < \left(\frac{\mu_d}{\sigma_d}\right)^2, \tag{2.11}$$

$$\lambda_1 = \frac{1}{\mu_1} \quad and \quad \lambda_2 = \frac{1}{\mu_2}, \tag{2.12}$$

$$\mu_1 = \frac{\mu_d \pm \sqrt{\gamma(\gamma+1)\,\sigma_d^2 - \gamma\mu_d^2}}{\gamma+1}, \tag{2.13}$$

$$\mu_2 = \frac{\gamma\mu_d \mp \sqrt{\gamma(\gamma+1)\,\sigma_d^2 - \gamma\mu_d^2}}{\gamma(\gamma+1)}. \tag{2.14}$$

- If $\mu_d < \sigma_d$, the approximation assumes a hyperexponential subnet (Figure 2.14), in which

$$\omega_1 = \frac{2\mu_d^2}{\left(\mu_d^2 + \sigma_d^2\right)}, \tag{2.15}$$

$$\omega_2 = 1 - \omega_1, \tag{2.16}$$

$$\lambda_h = \frac{2\mu_d}{(\mu_d^2 + \sigma_d^2)}. \tag{2.17}$$

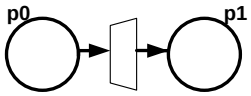


Figure 2.11: *s-transition* (BORBA; TAVARES; MACIEL, 2022).

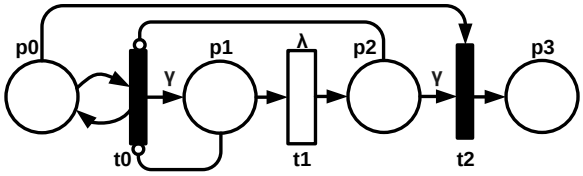


Figure 2.12: Erlang distribution (BORBA; TAVARES; MACIEL, 2022).

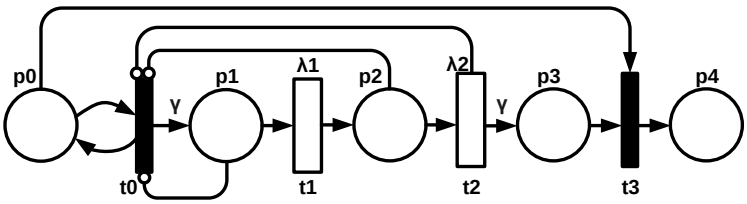


Figure 2.13: Hipoexponential distribution (BORBA; TAVARES; MACIEL, 2022).

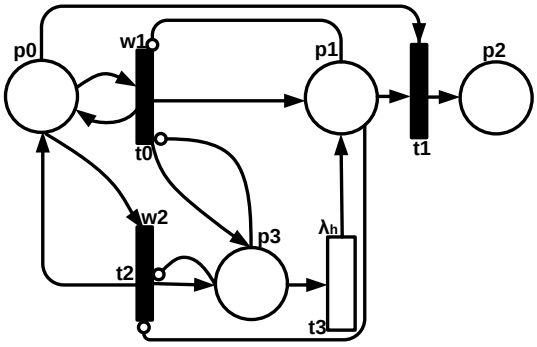


Figure 2.14: Hiperexponential distribution (BORBA; TAVARES; MACIEL, 2022).

2.7 RELIABILITY BLOCK DIAGRAMS

A reliability block diagram (RBD) is a graphical representation of the combination of successes or failures of components in a system. It represents the logical relationship among a system, subsystems, and components, considering their individual reliability values (KAPUR; PECHT, 2014; RAUSAND; ARNLJOT et al., 2004; KUO; ZUO, 2003). Furthermore, although it was initially proposed for reliability calculations, RBDs can also be used to estimate other dependability metrics, such as availability and maintainability (MACIEL et al., 2011).

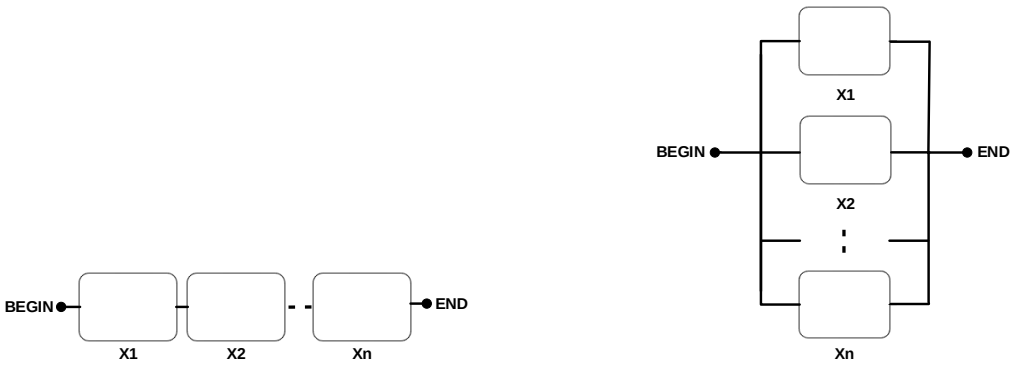
RBDs consist of components and their logical relationships, which are graphically represented by rectangles and arcs, respectively. In addition to connecting the components, the arcs are also linked to the start and end vertices. The start vertex is typically positioned on the left side of the model, whereas the end vertex is arranged on the opposite side (MACIEL et al., 2011).

Usually, the arrangement of the components in a model corresponds to the physical arrangement of the items in a system. However, some cases do not follow this rule, such as when the failure of one of the two resistors physically arranged in parallel causes system failure. In this case, the appropriate model would be two blocks arranged in series.

RBDs are often used to model the effects of system failures (MODARRES; KAMINSKIY; KRIVTSOV, 2009). A serial RBD arrangement (Figure 2.15(a)) should be considered when the failure of one block results in the failure of the entire system. This arrangement implies that all the subsystems must be operational to allow the system to operate (KAPUR; PECHT, 2014). A system that maintains its operability as long as at least one of its n components is functional is represented by parallel arrangements (RAUSAND; ARNLJOT et al., 2004), as shown in Figure 2.15(b).

In the event of component failure, when redundancy exists, redundant behavior can occur in the following ways (KAPUR; PECHT, 2014):

- ***Hot standby*:** redundant equipment governed by this concept is characterized by being active, even when not in use. Therefore, it has the same failure rate as the component being used. Parallel arrangements typically consider this type of redundancy;
- ***Cold standby*:** in this specific case, the standby component does not fail until it is requested, because it is inactive until the main component fails;



(a) Series (b) Parallel

Figure 2.15: RBD arrangement (BORBA; TAVARES, 2017).

- ***Warm standby*:** the redundant component has a low failure rate compared with the main component because it is only required at regular intervals (e.g., system backup).

Assuming n components (blocks) in series, the availability (A_s) or reliability (R_s) of a system can be estimated as

$$D_s = \prod_{i=1}^{n} d_i \tag{2.18}$$

where d_i represents the availability (A_i) or reliability (R_i) of the i component (EBELING, 2004).

In a system with n parallel components, availability or reliability is calculated as

$$D_s = 1 - \prod_{i=1}^{n} [1 - d_i] \tag{2.19}$$

where d_i corresponds to the availability (A_i) or reliability (R_i) of the i component (EBELING, 2004).

2.8 SUMMARY

This chapter explored some of the fundamental concepts necessary to understand this thesis. First, the storage devices studied in this thesis were presented in detail to provide an understanding of their functionalities and components. In this way, the influence of these elements on the performance, reliability, and power consumption of HDDs and SSDs was discussed. Next, the concept of hybrid systems was approached, and for didactic purposes, two commonly suggested storage policies for hybrid storage devices were demonstrated for didactic

purposes. Subsequently, this chapter addressed the performance and dependability concepts fundamental to the evaluations performed in this thesis. The mathematical formalisms of the CTMC, GSPN, and RBD were introduced to better understand the analytical conceived in this work. Finally, the optimal utilization of storage devices through data placement strategies was discussed.

3

RELATED WORKS

This chapter presents existing studies that propose solutions to address issues concerning the performance, dependability, and energy consumption of data storage systems. The related works detailed in this chapter are categorized into sections according to the main aspects suggested for improving such storage devices. The following aspects are considered: analytical models, architectures, dependability evaluation, energy consumption, data management, and flash memory management. Finally, this chapter concludes by comparing the existing solutions and the contributions of this thesis.

3.1 OVERVIEW

Hybrid storage devices have demonstrated their viability in data storage systems owing to their significant contribution to the performance of computer systems. Several architectures have been proposed to provide more efficient configurations for the growing number of requests, which is a consequence of the considerable increase in the demand for cloud services. In this context, data management in data storage devices is a prominent field of research that has motivated several studies. For example, several researchers have proposed approaches for a more appropriate data allocation to decrease the response time and energy consumption of storage devices. However, these approaches do not consider using analytical models to concomitantly investigate issues such as the energy consumption, dependability, and performance of data storage systems.

3.2 MODELS

Driven by the massive amount of data generated over the years, the demand for higher storage capacity and throughput has grown steadily. In this sense, HDDs play an essential role in data storage systems because of their high volumes and relatively low prices. However, HDDs may become a performance bottleneck when subjected to random operations, as such an access pattern demands intense movements from their mechanical components. Consequently, the service time may substantially increase because of data seeking, disk rotation, and transmission delays. Therefore, a balance between performance and price is essential for magnetic disks. Xie et al. (XIE; XIA; XU, 2020) propose a numerical approach to estimate HDD performance. A multiple-state-dependent approach is used to model HDDs, resulting in an M/G$^{[b]}$/1/K queuing model that can be solved to estimate throughput and response times under various workload types. The authors present a method for computing steady-state probability using embedded Markov chains to solve the proposed model. A comprehensive study of disk performance is conducted, and experimental results show the benefits of their solution for workloads composed of random write operations and small request sizes. In addition, the authors claim that the proposed model is feasible and can be used in future HDD optimization studies concerning, for example, the impact of different cache policies.

In flash-based storages, updating data means writing new information in an empty block rather than overwriting old data and setting the block containing the former information as erase-available. When the garbage collection mechanism is triggered, a cleaning algorithm claims space and erases these marked blocks, which may still store information derived from a different application (i.e., unrelated but valid data). This information is kept and copied to another block, which therefore causes extra write operations; as such requests, in principle, should not be required. This amplification is expressed as the ratio of the total pages written to the original number of requests. Verschoren et al. (VERSCHOREN; VAN HOUDT, 2020) investigate the performance of SSDs by specifically focusing on the influence of various garbage collection algorithms. Of particular interest is the d-choice garbage collection algorithm, designed to mitigate the adverse effects of write amplification (i.e., the volume of data a solid-state drive controller writes relative to the data written by the host flash controller). However, these algorithms can introduce variability in the number of program/erase cycles experienced by flash memory cells, consequently affecting the lifespan of the SSD. To address these concerns, the

authors propose a new HCWF (swap) mode, which is a mechanism through which it is possible to balance wear and minimize write amplification. For this, a "hot" and "cold" data separation is adopted, aiming to identify the most used memory cells and, consequently, avoid their use during the execution of the proposed approach. A Markov chain model is developed to represent the SDDs and proposed mode. Simulations demonstrate relative errors below 0.1%; therefore, the authors consider that the model presents high accuracy in estimating SSD performance when implementing the proposed solution.

Boukhelef et al. (BOUKHELEF et al., 2017) propose a cost model for storing database objects in cloud infrastructure. The approach comprises costs from storage utilization (occupation per GB), consumed energy, endurance, SLA violation, and object movement between storage units. Constant values are assumed for the power cost model because the authors do not consider the workload characteristics (e.g., access patterns) for this metric. The migration cost model provides insights into the transfer of objects. However, different applications and their effects are not considered in this approach. Additionally, the storage lifetime is estimated by considering the number of operations a given storage has been subjected to and its respective manufactory-report endurance. The authors built a hybrid storage system (HDD and SSD) and kernel module that captures I/O requests from different devices. Factory specifications (e.g., average response time, idle power, and endurance) from different storage technologies (HDD and SSD) were adopted for the experiments using the cost models. The results indicate that energy consumption is responsible for 5–28% of the overall cost of databases using HDDs. SSD failures were found to be responsible for the highest costs and can account for up to 90% (for write-intensive workloads) of the overall cost. These models do not consider network-related costs, and the authors intend to approach their impacts in the future.

3.3 ARCHITECTURES

In (NAKASHIMA et al., 2017), the authors investigate the performance of hybrid storage systems composed of M.2 SSDs (solid-state devices that use the PCI-express interface), HDDs, and *Serial Advanced Technology Attachment* (SATA) SSDs. The experiments focus on sequential requests, initially best processed by magnetic disk devices. Consequently, the hybrid composition of HDD + M.2 SSDs (acting as a system cache) obtained higher I/O values than the homogeneous devices (HDD, SATA SSD, and M.2 SSD). Nakashima et al. (NAKASHIMA; KON; YAMAGUCHI, 2018) evaluate the SSD architecture as a cache and propose a method to

increase the throughput by focusing on Big Data applications. This method considers the wear level of SSDs when many writes are requested. In addition, considering the request behavior, SSDs are adopted only for storing anticipated data (cache information), whereas massive writes are directed to HDDs. Thus, read operations become predominant in SSDs. Experimental results indicate a 78.2% improvement in the response time of the solid-state devices.

Li et al. (LI et al., 2019) claim that architectures with SSDs as cache are ineffective because of the significant performance differences between solid-state and magnetic disk devices. The authors justify their assertion by considering that a cache miss on the order of 1% can degrade the I/O rate by a factor of 10 (considering that the adopted SSDs have a throughput rate three times higher than that of HDDs). Therefore, the authors propose a scheme in which primary replicas are stored on SSDs, whereas backups are saved on HDDs. The proposed architecture includes a mechanism to transform small random writes into sequential requests to compensate for the performance differences between the technologies. This mechanism is adopted only to back up replicas (i.e., requests originally intended for HDDs). This proposal aims to provide efficient virtual disks for virtual machines in cloud computing environments. The experimental results demonstrate that the proposed approach can achieve better performance than commercial block storage services such as *Amazon Web Service*s (AWSs).

3.4 DEPENDABILITY EVALUATION

Han et al. (HAN et al., 2021) study the correlated failures in nearly one million SSDs of 11 drive models based on a dataset of *Self-Monitoring Analysis and Reporting Technology* (SMART) logs (COMMITTEE, 1995), trouble tickets, physical locations, and applications. The authors conduct an exploratory analysis in order to guide the design of highly reliable storage systems. The SMART attributes and storage drive characteristics (drive models, lithography, and capacity) are approached to investigate their correlation with failures, and Spearman's rank technique (JAIN, 1990) is adopted. The results indicate the significant effects of write-dominant applications and multi-level cell (MLC) technologies on SSD failures.

In (XU et al., 2021), the authors propose a technique to select the SMART log attributes as learning features in an automated and robust manner. This technique combines different feature ranking results and automatically generates the final feature selection based on change point detection of wear-out degrees. The authors claim that the proposed method can be used for large-scale SSD failure prediction of different drive models and suppliers, as experimental results

indicate accuracy improvements of 10-14% compared to existing feature selection approaches. Similarly, Zhang et al. (ZHANG et al., 2019b) suggest a machine learning approach to predict HDD failures. This work address the lack of data about minority disks, that is, storage devices deployed to augment system capacity or replace recently failed drives. A transfer learning approach is adopted, and an iterative algorithm is developed to improve predictive accuracy. Consequently, a prediction model might be leveraged for a different model by transferring sufficient SMART training data (health state information) from an other disk model.

In (CHAMAZCOTI et al., 2017), the authors propose a solution to increase the reliability of *Redundant Array of Independent Disks* (RAID)-based SSDs. Specifically, a new parity bits distribution policy is proposed based on existing policies for RAIDs (e.g., the uniform or unequal distribution of parity bits). A uniform distribution of bits among SSDs may increase the probability of concurrent failures when, for example, the number of programming/deletion cycles reaches the endurance limit of flash memory. Conversely, an uneven distribution accelerates the wear of some solid-state devices, resulting in a high probability of failure. The authors then suggest a structure called Hybrid RAID, which uses both of the mentioned parity policies and adopts the number of programming/deletion cycles in each flash memory as the factor determining the behavior of the designed control algorithm. Furthermore, a quantitative model is adopted to estimate the reliability of SSD-based RAIDs, and the experimental results prove the approach's feasibility.

Yin et al. (YIN et al., 2018a) state that existing approaches for decreasing the power consumption of storage systems do not consider the possibility of negative effects on the reliability. For example, several techniques suggest dynamic power management by increasing or decreasing the rotation of magnetic disks according to the frequency of the requests. Inevitably, the mechanical components involved in this process will suffer from increased wear and, consequently, a reduction in their lifetime. As an alternative, the authors propose a hybrid storage system and suggest an approach to balance issues such as reliability, energy efficiency, and performance. A middleware is designed to allocate frequent write requests to HDDs (because of the limited number of write operations on SSDs). Simultaneously, less intense workloads are directed to solid-state devices (to decrease the frequency at which the HDD spin-down is requested). The approach is validated through experimental results, in which the proposed system achieved a 40% decrease in power consumption, a 50% increase in throughput (IOPS), and a 15% decrease in reliability (a value considered acceptable by the authors) when compared to systems composed

of SSDs only.

3.5 ENERGY CONSUMPTION

Energy consumption in data centers has been continuously increasing, and storage systems may represent up to 40% of the overall demand. Existing energy-saving techniques commonly do not consider their impacts on storage endurance, which becomes a challenging issue, especially when it involves storage-as-a-service contracts, where data unavailability penalties are considerably impactful. Previous solutions approach the rotation rate at which magnetic disk devices should operate to reduce the power demand. The idea is to determine an optimal trade-off between the HDD's power states (active and sleeping), that is, to define when the platters should rotate at high or low speeds. These approaches may cause HDDs to malfunctioning, leading to permanent data loss. As for SSDs, intense write workloads may shorten their lifetime owing to the limited number of erasure cycles of flash memory.

In this context, Yin et al. (YIN et al., 2018b) propose a middleware called DuoFS, which improves the energy efficiency of storage systems. By integrating HDDs and SSDs, the proposed system distributes data according to the number of requests and workload behavior. This approach exploits the low power consumption and high performance of SSDs and the large storage capacity and lifetime of HDDs to save energy in data storage systems. Moreover, DuoFS is scalable and allows storage arrangements to expand as resource demand increases. Regarding I/O analyses, access patterns are investigated to identify data popularity and categorize such requests as hot (heavily accessed) or cold (lightly accessed). Specifically, two separate file systems (file system-hot and file system-cold) are adopted for different types of storage nodes. Hot file systems comprises SSDs (better performance and low power consumption), whereas HDDs are used for cold file systems (larger capacity and longer lifetime). Frequently read data demand better performance and, therefore, are cached into SSDs. The remaining of the stored data are led to disks that are pushed to idle mode to save energy. The FIO benchmark tool (AXBOE, 2021) is used for the experiments, and the results indicate that SSD-only architectures have better performance values than the proposed DuoFS. However, the authors claim energy savings of up to 60% when the number of concurrent processes is less than 32. In addition, a 5% reduction in the expected storage lifetime was obtained because the solution was used to save energy. The authors intend to extend this study to cover more I/O scenarios and consider fault tolerance.

Similarly, Yin et al. (YIN et al., 2016) propose a storage layer called RESS to improve the

energy efficiency of storage systems without impairing their reliability. The authors developed a middleware on which HDD-based storage nodes seamlessly integrate with SSDs. RESS uses HDDs as the primary storage method because of their high capacity, and SSDs are employed to handle recently accessed data. The proposed approach relies on disabling SSDs under many requests because of the limitations of erasure cycles and shifting to active mode if the rate of data access is low. In the active mode, the SSDs are used as a cache for HDDs transitioning into low-power mode to save energy. More specifically, a workload monitor tracks the I/O patterns to determine the number of active nodes, stores the data inside HDDs, and creates replicas to be loaded into SDDs. In other words, energy savings are achieved by replicating data strips in the SSD nodes and reducing the number of simultaneously operational disk nodes. Experiments were conducted on a five-node cluster (four HDDs and one SSD) by using the MPI-IO benchmark tool (GRIDER; NUNEZ; BENT, 2008) to generate synthetic workloads. Experimental results show performance gains as the number of processes or concurrent I/Os increases. Additionally, the authors claim substantial energy savings to compensate for the high prices of solid-state devices. The authors intend to improve the measurement of the power values because the adopted approach does not support continuous monitoring and may lead to inaccurate results.

3.6 DATA MANAGEMENT

Cloud service providers, such as Amazon EC2, and private cloud platforms, such as OpenStack, use virtualization techniques to manage resources and efficiently provide dynamic scalability efficiently. Although existing virtualization techniques provide efficient resource management, a significant amount of overhead can occur because of the additional abstraction layer without a proper data management policy. For instance, in SSDs, sequential write workloads are more likely to generate a small number of invalid blocks, whereas random write workloads may generate numerous blocks with a small number of invalid pages. Consequently, random writes increase the garbage collection overhead, which induces more internal page copy operations and negatively impacting SSD performance.

In this regard, Kim et al. (KIM; EOM; SON, 2019) present an address-mapping technique for SSDs to improve the spatial locality and performance of random write operations. This technique transforms random write requests into sequential requests by changing the virtualization layer. To achieve this, the authors create a metadata checker in the *Virtual Machine* (VM) filesystem and a sequentializer in a *Kernel-Based Virtual Machine* (KVM). While the checker

categorizes the data and metadata for a file type-oriented approach, the sequentializer remaps random write requests from the VM into sequential write requests. The experimental results indicate a performance improvement of 97% compared with the other systems.

In (BOUKHELEF et al., 2019), the authors investigate cost-based object placement strategies for hybrid storage systems in a cloud infrastructure. This study aims to optimize the overall storage cost while considering customer requirements and constraints such as capacity and performance. The authors propose a heuristic-based solution and genetic algorithms to optimize the object placement problem. The idea involves computing a data placement solution for a given problem and improving it by considering the I/O characteristics and SLA penalties. Experiments were conducted using synthetic workloads and two storage devices (one SSD and one HDD). The results show an improvement of 40% in storage costs when the number of objects is low. In addition, the authors claim an average resource over-provisioning of 8% to comply with SLAs.

Wu et al. (WU; HUANG; CHANG, 2019) propose a data management method for hybrid storage systems based on object priority. A migration mechanism moves high-priority data (i.e., the most accessed objects) to SSDs, whereas low-priority objects are kept in HDDs. Experimental results indicate improvements in I/O performance. Nevertheless, the approach does not consider prominent issues, such as access patterns and object sizes, for defining object priorities.

3.7 FLASH MEMORY MANAGEMENT

Flash memory requires a regular garbage collection mechanism to optimize space and improve the efficiency of the entire storage device. However, this process involves identifying valid data, copying them to an empty block, updating the address table, and deleting invalid data. Although this is an essential element for the correct operation of SSDs, the aforementioned mechanism may prevent the use of storage during the execution period. Moreover, because it does not have a defined execution period (i.e., is nondeterministic), the system's reliability (regarding response time) for real-time applications can be compromised. In light of this, MCEWAN et al. (MCEWAN; KOMSUL, 2018) present a solution for real-time garbage collection with a deterministic runtime. Moreover, the proposed method considers the wear level of the memory cells as a threshold to dynamically define the most appropriate execution time dynamically. Simulations performed using the Disksim tool suggest that the proposed approach can increase

the lifetime and performance of SSDs.

One disadvantage of SSDs is the limited amount of programs/deletions handled by each memory block. The absence of a control mechanism can cause extreme numbers of programming and deletions in certain blocks, leading to early deterioration. Therefore, SSDs have a wear leveling mechanism that equalizes the usage of their memory blocks as much as possible. Typically, wear-leveling techniques assume a uniform wear limit, provided by the manufacturer, for all blocks. However, owing to manufacturing variations, different blocks in the same chip may exhibit distinct P/E. In (SHI et al., 2018), a new wear leveling scheme is proposed to estimate the wear of memory blocks in SSDs more accurately. In contrast to existing schemes, the authors suggest a new technique that performs a dynamic and individual evaluation of memory blocks concerning P/E and the number of performed deletions. The authors developed a simulator using the C programming language to evaluate the proposed approach for four real workloads. The results indicate a 17% reduction in the wear of the memory blocks; however, a performance loss of 5% is also observed.

3.8 COMPARISON

This section compares this thesis and the aforementioned existing solutions in the literature, which have been explained in this chapter. For didactic purposes, related works have been classified according to the main aspects analyzed by the authors to solve issues related to the energy consumption, performance, and dependability of storage systems.

Table 3.1 lists the studies mentioned in this chapter. For comparison purposes, the works are classified according to the following approaches and contributions: measurement (meas.), performance model (perf.), energy consumption (e.c.), dependability (dep.), performability (perfor.), data management (d.m.), and cost (c.).

In contrast to previous studies, this thesis proposes models based on GSPNs and RBDs mathematical formalisms to evaluate the performance, dependability, and energy consumption of homogeneous and hybrid storage systems. The proposed models allow the design of data storage systems that may be adopted, for instance, in data centers. In addition, the proposed approach can be utilized to analyze distinct workloads, data management mechanisms, and storage arrangements. This study also considers real-world workloads to demonstrate the practical suitability of the proposed models.

Furthermore, issues related to the performability of data storage systems can be assessed

Table 3.1: Comparison between this thesis and related work.

	meas.	perf.	e.c.	dep.	perfor.	d.m.	c.
This thesis	✔	✔	✔	✔	✔	✔	✔
(XIE; XIA; XU, 2020)	✔	✔				✔	
(VERSCHOREN; VAN HOUDT, 2020)	✔	✔		✔		✔	
(BOUKHELEF et al., 2017)	✔	✔	✔	✔			✔
(NAKASHIMA et al., 2017)		✔				✔	
(NAKASHIMA; KON; YAMAGUCHI, 2018)	✔	✔		✔			
(LI et al., 2019)		✔		✔		✔	
(HAN et al., 2021)	✔			✔			
(XU et al., 2021)	✔			✔			
(ZHANG et al., 2019b)	✔			✔			
(CHAMAZCOTI et al., 2017)		✔		✔		✔	
(YIN et al., 2018a)	✔	✔	✔	✔		✔	
(YIN et al., 2018b)		✔	✔	✔		✔	
(YIN et al., 2016)		✔	✔	✔		✔	
(KIM; EOM; SON, 2019)		✔		✔		✔	
(BOUKHELEF et al., 2019)	✔	✔	✔				✔
(WU; HUANG; CHANG, 2019)	✔	✔				✔	
(MCEWAN; KOMSUL, 2018)		✔		✔		✔	
(SHI et al., 2018)		✔		✔		✔	

using the proposed hierarchical modeling approach. For example, aspects such as SLA compliance, disaster prevention, and job scheduling can be addressed using the solution presented in this thesis because the effects of storage failures on the performance of the entire system can be estimated. It is also important to note that the operation (read and write) delays adopted for the validation, and experimentation of the proposed models, were collected from real devices under industry-based workloads (HDDs, SSDs, and Hybrid). In addition, to demonstrate the feasibility of the proposed approach, this thesis presents experimental results using the designed models following established industry standards. The conceived models abstract away issues, such as bad blocks, data addressing, file systems, and data migration between storage devices.

3.9 SUMMARY

This chapter summarized the representative works related to this thesis. It was demonstrated that although many solutions involve analytical models, the ability to concomitantly evaluate the performance, availability, and energy consumption of storage systems is not found in the literature. Although some studies have investigated and proposed various approaches for managing data storage systems, none of the previous studies have considered a solution that allows the representation and evaluation of the impacts of different applications may cause. For

example, in the context of new architectures, in addition to the cited solutions being restricted to performance analysis, the suggested storage arrangements and policies are static and do not consider the effects of different workloads on device endurance. Finally, a comparison of previous works and this thesis has been presented to demonstrate the differences between the proposed contributions and existing solutions.

4

METHODOLOGY AND TOOLS

This chapter describes the methodology utilized to model and assess the performance, dependability, and energy consumption of data storage systems. The proposed models' design, validation, and experimentation are thoroughly explained, including the steps, methods, techniques, and tools used. Preliminaries necessary to comprehend the notation adopted for presenting the methodology are introduced in Section 4.1. Section 4.2 outlines the steps involved in modeling and evaluating the performance and energy consumption of storage systems. Section 4.3 covers these aspects specifically for dependability. Finally, in Section 4.4, the tools and environment employed to collect the performance and power consumption data for the considered data storage devices are demonstrated. In addition, storage failure datasets from representative companies are detailed.

4.1 PRELIMINARIES

Figure 4.1 presents the elements used for the high-level representation of the methodology proposed in this thesis through a process flow diagram based on *Unified Modeling Language* (UML) notation (UML, 2023). Figure 4.1(a) shows the elements representing the activity to perform. Figure 4.1(b) depicts the decision to be made that dictates the next step to be executed. The element denoted in Figure 4.1(c) represents a feasible use case when adopting the solution proposed in this thesis. Activities that can occur in parallel are shown in Figure 4.1(e), whereas the joining of different flows is illustrated in Figure 4.1(d). Figures 4.1(f) and 4.1(g) show the elements that represent the beginning and end of the flowchart, respectively.

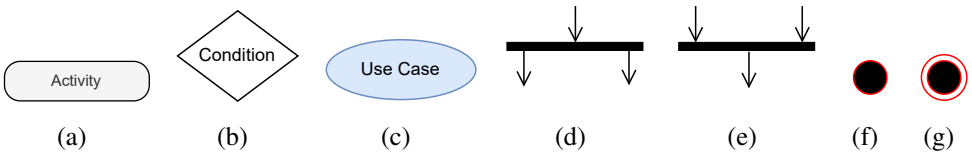


Figure 4.1: UML elements adopted to illustrate the methodology proposed in this thesis (own work (2023)).

4.2 MODELING STORAGE SYSTEMS FOR PERFORMANCE AND ENERGY CONSUMPTION EVALUATION

This section provides a feasible methodology for estimating data storage systems' performance and energy consumption. Figure 4.2 illustrates the steps of the methodology proposed in this study. Initially, the proposed modeling methodology consists of defining the problem and gathering the requirements for the *Conception* of the abstract models. In the *Investigation of the real system* step, data related to storage behavior are collected and analyzed using statistical techniques to represent the devices in question. Subsequently, the *Validate models* activity determines the need for adjustments to the designed models. In addition, in the *computation* step, the models are fitted and computed (*Refine models for experiments* and *Solve models*) to provide sufficient information for the analysis phase of the following experiments. In the *Workload-driven* step, the performance and energy consumption of the homogeneous and hybrid storage systems are evaluated for different workload characteristics. In the *Optimizaton* step, a case study is performed to identify the optimal storage arrangements according to the given constraint. *Scalability* step investigates the behavior of the conceived model as components are added.

4.2.1 Conception of performance and energy consumption models

Regarding the *Conception* step, the *Specify system characteristics and requirements* activity is concerned with observing and gathering the requirements required to represent homogeneous and hybrid data storage systems in data centers. In addition, in this step, the activity *Define metrics of interest* defines the metrics to be considered in the conception of the analytical models as well as for validation and experimentation. The average response time, throughput, and energy consumption were the metrics adopted in this study. The choice of these metrics considers the essential requirements regularly encountered in SLAs with cloud

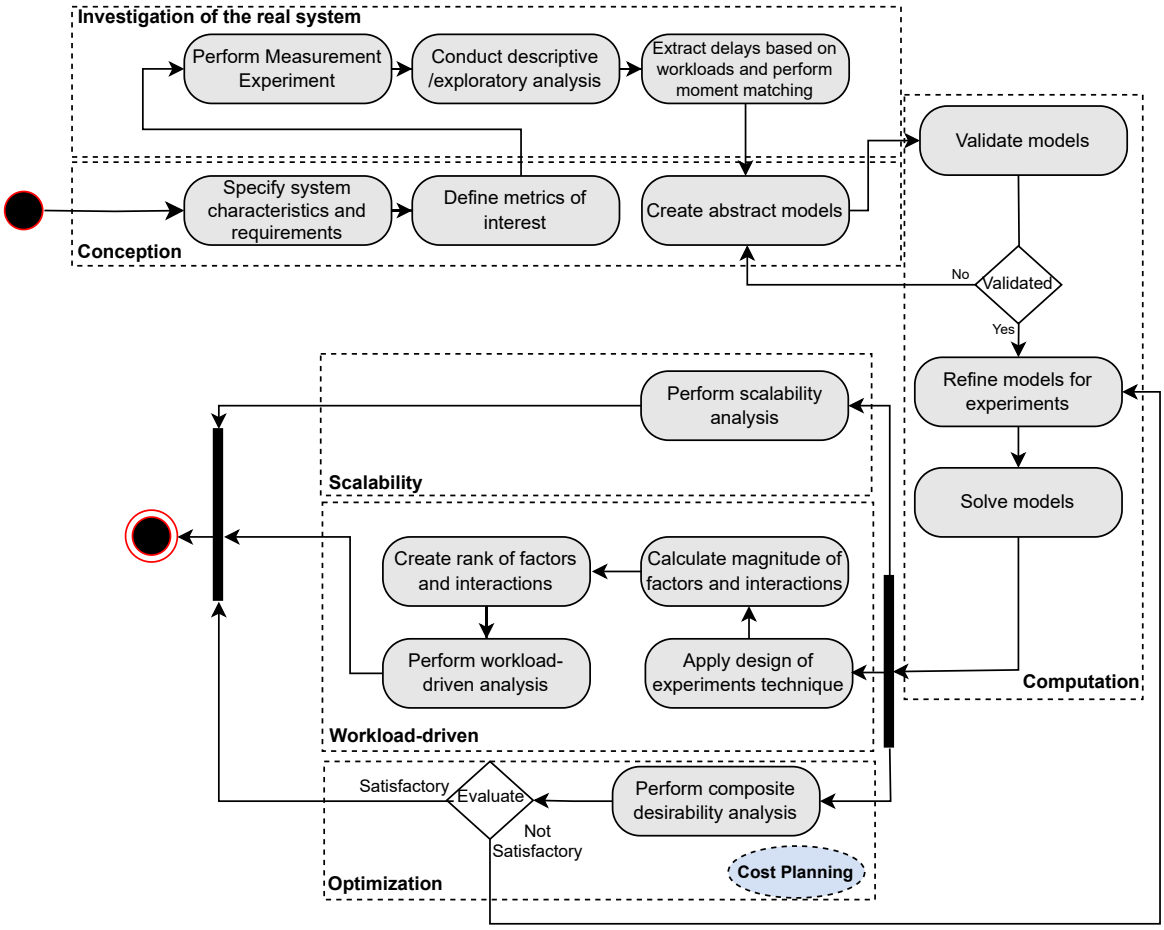


Figure 4.2: Supporting methodology for performance and energy consumption modeling of data storage systems (own work (2023)).

computing service providers (ARSHAD et al., 2022; MUSTAFA et al., 2019). In addition, energy consumption is crucial for storage system designers in estimating the cost required to provide a service. Activity *Create abstract models* represents the modeling stage, in which the requirements and metrics defined previously are considered. In this thesis, the mathematical formalism GSPN has been adopted to design the models. GSPN is a suitable formalism for storage system design, as, differently from queueing network models (for instance), synchronization, resource sharing, and conflicts are naturally represented. Also, phase approximation technique may be applied for modeling non-exponential activities, and events with zero delays (e.g., workload selection) may adopt immediate transitions (concepts detailed in Sections 2.6.3 and 2.6.2, respectively). The conceived models represent requests from one or more clients, characterization of the workload (e.g., access pattern, object size, and operation type), and execution of read or write operations. Also, the proposed GSPN models can estimate the average response time, throughput, and energy consumption of homogeneous and hybrid data-storage systems.

4.2.2 Measurement and investigation of storage performance and energy consumption

In *Investigation of the real system*, the *Perform Measurement experiment* activity is about collecting performance and power values, which will be used in the following steps (*Moment matching* activity, which adopts the technique explained in Section 2.6.3). The tools and environment adopted are demonstrated in Section 4.4. *Descriptive/exploratory analysis* provides an assessment of the performance and energy consumption of storage devices and hybrid mechanisms in order to provide insights about the benefits of each technology. For that, this work evaluates storage systems utilizing an approach based on *Design of Experiment* (DoE) (MONTGOMERY; RUNGER, 2014). More specifically, a factorial design is adopted ($\prod_{i=1}^{k} l_i$) with 20 replications (to obtain mean values with an approximate normal distribution). In this technique, factors (k) refer to the variables that can be controlled during the experiment, while levels (l) represent the values each factor can take. A treatment (i) corresponds to a unique combination of factor levels. Three experiments are carried out and the metrics of interest are response time, IOPS (input/output per second) and energy consumption. The experiments are explained below:

- The first experiment adopts a screening approach for identifying the suitable technology to compose a hybrid storage system. Specifically, this is an experiment with the purpose of identifying those factors (in this experiment, storage technologies) that have the best values on the metrics of interest, considering all possible treatments.

- The second experiment assesses the performance and energy consumption of a hybrid storage device and its components (HDD and SSD) individually. This experiment considers main factors and second-order interactions, since higher order interactions are usually negligible (MONTGOMERY; RUNGER, 2014). An analysis of variance (ANOVA) (JAIN, 1990) is performed to calculate the impacts of such factors and interactions. For this work, the hybrid storage system redirects the request only to an idle device (which is not performing any request). Additional assessments are also carried out for factor interactions and some factor levels are fixed to represent real-world workloads.

- The third experiment adopts a composite desirability (CD) approach (MONTGOMERY;

RUNGER, 2014) to estimate the best combination for all *technology* levels. CD aims to optimize a set of metrics and the value ranges from 0 to 1. CD tends to 0 for the worst configuration and 1 represents the best system.

4.2.3 Model validation, refinement, and solving

Computation step evaluates the abstract models created so that, if satisfactory, they can be refined and computed according to the experiments to be performed. The first activity, *Validation*, analyzes whether the estimated results of the performance metrics (e.g., average response time and throughput) and energy consumption through stationary analysis of analytical models are consistent with the values of the actual data storage devices obtained from the measurement experiment. For this analysis, the *delays* associated with the execution of write and read operations in the designed GSPNS models are derived from the results of applying the moment matching technique. For validation, parameters such as technology (SSD, HDD, or *Hybrid*), operation type (write or read), access pattern (sequential or random), and object size (small or large) are considered. Validation is confirmed, for each metric, when under the same conditions (i.e., subjected to the same workload); the value obtained through the stationary analysis from the GSPN models is contained within the confidence interval estimated from the results of the measurements performed on the real data storage systems. The non-conformity between the results leads to performance and energy consumption model adjustments. Further details regarding the validation conducted in this study can be found in Section 6.2. *Refine models for experiments* and *Solve models* involve tuning and computing the models designed to correctly represent and obtain data related to the planned performance and energy consumption experiments. These activities assume that the designed formal state-space-based models are validated. Similar to the validation step, this process considers, for all experiments, the *delays* computed through the moment matching technique, which are assigned to the execution of write and read operations in the designed GSPNs models.

4.2.4 Experiments utilizing the models

The following steps refer to experiments performed using the models proposed in this thesis. Similar to the measurement experiment, a factorial design is adopted but, in this case, five experiments are carried out using GSPN models to contemplate the workload characteristics highlighted as significant for storage evaluation by the *Storage Performance Council* (SPC) (COUN-

CIL, 2019; HURSON, 2013). (*Workload-driven* step). This work also presents a case study to illustrate the feasibility of the proposed models for assessing distinct storage arrangements (*Optimization* step). Lastly, *Scalability* step demonstrate the size of state space (i.e., CTMC size) and evaluation time of the conceived models considering the increase of storage components and workers (clients). Further details regarding experiments are outlined below:

- The first experiment adopts a screening approach for identifying the magnitude of each factor and interactions. For the sake of validation and comparison in the experiments, the energy consumption is assumed for one second, aligning with the sampling interval of the oscilloscopes used for voltage value collection. Subsequently, a rank is created using the calculated impacts of the factors and interactions in question. This rank is used as a reference for decision-making in the following experiments.

  Four additional experiments are adopted, which utilize the results from the screening approach and the guidelines for benchmarks developed by the SPC. Such a council is composed of companies that define methodologies to evaluate storage devices and systems. The experiments also utilize two supplementary metrics: (i) IOPS/energy consumption, and (ii) price/IOPS. The former represents energy efficiency and a higher value is better. The latter is the relation between storage system price and performance, and lower values are preferable. Storage system price is calculated as storage capacity $\times$ cost per GB. In this work, US$0.075 and US$1.0 (YIN et al., 2018a) per gigabyte for HDD and SSD, respectively, have been considered. The same factors as those used in the screening approach were considered, and the levels representing specific workloads were fixed according to each respective experiment. The additional experiments are explained as follows.

- The second experiment evaluates the performance of storage systems, in which the application's access pattern is predominately random (e.g., database systems). In this case, the objects are stored on device blocks without a specific order (SAXENA; KUMAR, 2014). Besides, write operations contemplate 70% of the workload.

- The third experiment (*sequential access*) assesses the behavior of storage systems for applications that require large-scale sequential data access, such as in financial processing applications, where tasks involve analyzing historical market data, for

instance. Sequential access assumes the objects are stored on contiguous blocks in the storage devices (PARK et al., 2011). The workload also assumes an equal proportion (50%) of write and read operations.

- The fourth experiment (*read operations*) evaluates storage systems for queries in very large databases (e.g., data mining). In this case, the workload is predominantly composed of sequential (100%) and read (99%) requests.

- The fifth experiment, namely, *mixed*, represents raw data workloads, which are usually composed of small random requests (80%) and commonly have mixed operations (50% write) from simultaneous clients (e.g., 4 workers) (COUNCIL, 2019; MONTAZERI et al., 2018). The workload also assumes 20% of sequential requests with large object sizes (1MB).

- A case study demonstrates an evaluation of distinct storage arrangements in a cloud computing environment. This study adopts composite desirability (CD) (MONTGOMERY; RUNGER, 2014) to evaluate the adoption of storage devices concerning different workloads. CD aims to optimize a set of metrics (IOPS and response time), and the value ranges from 0 to 1. CD tends to 0 for the worst configuration, and 1 represents the best system. The case study takes into account four distinct systems (to assess all possible device type configurations, including HDD-only, SSD-only, and hybrid configurations with varying HDD and SSD proportions), which may only adopt two device types with the following costs (CLOUD, 2020): $1TBHDD$ - US$40.96 and $120GBSSD$ - US$20.4. All designs consider the maximum amount of devices, in which the total price is less than US$105.0, a value from which it is possible to evaluate hybrid systems with more HDDs than SSDs and vice versa. For workloads, the case study takes into account three applications previously mentioned: database systems, data mining, and raw data.

4.3 MODELING STORAGE SYSTEMS FOR DEPENDABILITY EVALUATION

This section aims to provide a methodology to estimate availability and performability of data storage systems. Figure 4.3 illustrates the proposed approach. The activity flowchart denotes

the main steps for the *Conception* of the models as well as the proposed *Investigation of the real system* for a proper representation of storage reliability behavior. In the *Computation* step, the abstract models are refined and the resulting hierarchical model is solved. Lastly, statistical techniques can be employed to provide *Optimization*-based insights.

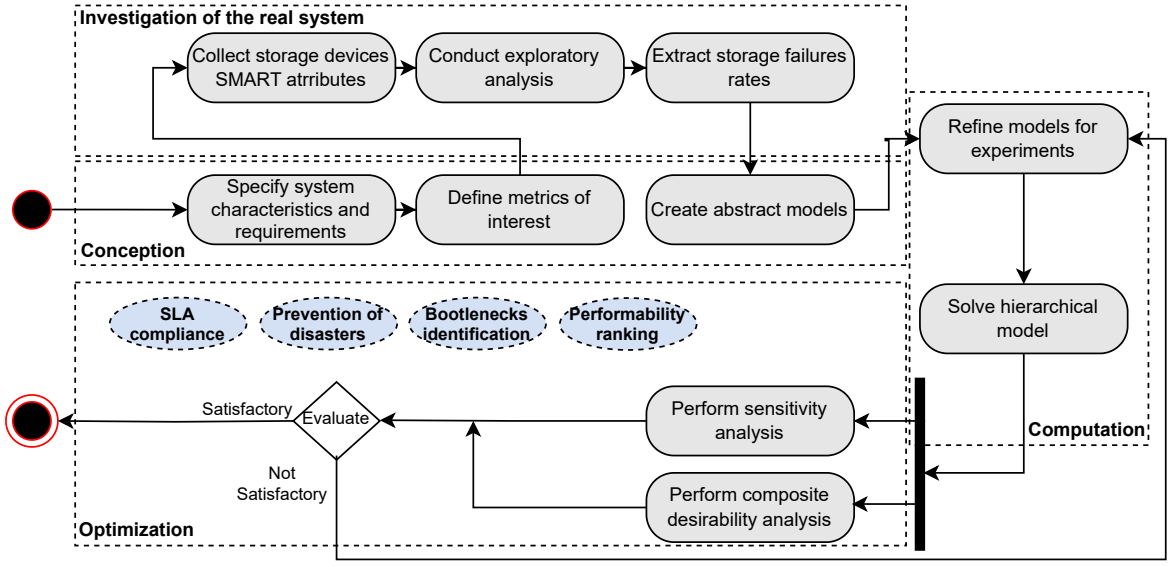


Figure 4.3: Supporting methodology for dependability modeling of data storage systems (own work (2023)).

Regarding the *Conception* step, the system characteristics and requirements (in this study, the storage nodes arrangement, workloads, and metrics of interest) are essential to conceive the abstract models. For instance, *Metrics of interest* activity may reflect the desired information (e.g., response time and availability) for diagnosing the represented data storage system. In addition, this description may represent the dependability relationship between the storage devices and the technologies adopted. RBDs and GSPNs are the mathematical formalisms employed in this stage.

Investigation step comprises gathering failure-related data from SSDs and HDDs (in this work, the datasets from Alibaba and Backblaze, both detailed further in Section 4.4.1). In this activity (the first in this branch), only attributes related to storage reliability are considered. Next, exploratory analysis is performed to identify the applications' effects on SSDs and HDDs failures. Subsequently, the results of such extraction are considered to estimate failure rates regarding the related application.

The *Computation* step involves refining the abstract models and solving the hierarchical model, considering the previous steps. *Refine models* activity consists of representing the

previously conceived abstract models following the constraints from the planned experiments and considering the outcomes derived from the *Investigation* step (e.g., storage system architecture and parameters like MTTF/MTTR and request processing delays). *Solve hierarchical model* activity involves first the computation of the refined availability model so that its results can be employed as an attribute in the performability model (please, see Section 5.2.2.1).

The *Optimization* step involves using the results of computing the models to provide solutions that meet the desired efficiency criteria, employing statistical techniques. If the results are unsatisfactory (i.e., if the criteria that represent the required level of performance that must be reached even in the presence of failures for a given modeled storage system are not met), the models must be further refined and computed. *Perform sensitivity analysis* and *Perform composite desirability* (MONTGOMERY; RUNGER, 2014) activities are examples of methods that can be employed to conduct analyses to optimize the modeled storage system. SLA compliance, disaster prevention, bottleneck identification, and performability ranking are studies that can be performed using the proposed modeling approach. SLA compliance involves ensuring that the storage system satisfies predefined service-level agreements. This guarantees that the system provides the required performance, availability, and reliability to satisfy the expectations of users. Disaster prevention concerns assessing and implementing measures to prevent or mitigate potential disasters or data loss within storage systems. It aims to enhance the system's resilience and minimize the impact of unforeseen events. Bootleneck identification within a storage system is essential for optimizing performance. This allows for identifying points in the system where resource constraints limit the overall performance and addressing these bottlenecks to improve system efficiency. Performability ranking evaluates and ranks system performance under failure events, providing valuable insights for decision-making and facilitating system improvements.

4.4 TOOLS AND ENVIRONMENT SETTING

This work adopts the tool Iometer (LEVINE, 1998; NAKASHIMA; KON; YAMAGUCHI, 2018; LI et al., 2015) to characterize storage devices for read and write operations. The results are utilized on the conceived GSPN models for validation and experiments.

Figure 4.4 depicts the adopted system, whose components are detailed in Table 4.1. Using Iometer, the server executes the workload on each drive (or simultaneously for the *hybrid* approach). An oscilloscope collects instantaneous voltage (using shunt resistors), the power is estimated and, then, energy consumption is obtained using numerical integration . For each

treatment, the system collects 20 samples, a sample size considered suitable for estimating mean delays associated with read/write operations and the metrics of interest when assessing a storage, as referenced in (LEVINE, 1998). These samples are then used to estimate IOPS, mean response time, and energy consumption.

Figure 4.5 depicts the electrical circuit to collect voltage values from HDDs and SSDs. V1, V2, and V3 represent the voltage the respective oscilloscope measures at a given time. The HDD and SSD electric power (EP_{HDD} and EP_{SSD}) are then estimated using Equations 4.1 and 4.2, where R represents the resistance of a shunt resistor.

$$EP_{HDD} = \frac{(12 - V_1)}{R} \times V_1 + \frac{(5 - V_2)}{R} \times V_2 \tag{4.1}$$

$$EP_{SSD} = \frac{(5 - V_3)}{R} \times V_3 \tag{4.2}$$

Fio tool (AXBOE, 2021) has also been adopted to generate additional real-world workloads (KISHANI; AHMADIAN; ASADI, 2019; LEE et al., 2015; MOTI et al., 2021): OLTP (online transaction processing) and Varmail. OLTP represents financial system transactions (WU et al., 2016; PARK et al., 2011), which is mainly composed of small ($4KB$) random (100%) writes (99%). Requests are performed by 211 threads (*workers*), which aligns with OLTP workload characteristics. Regarding Varmail, the workload contemplates both small ($4KB$) random (50%) and large ($1MB$) sequential (50%) requests (YANG; ZHU, 2015; KISHANI; AHMADIAN; ASADI, 2019). 16 *workers* are adopted to perform the requests, in accordance with Varmail workload characteristics.

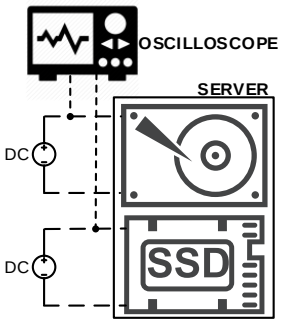


Figure 4.4: Environment setting (BORBA; TAVARES; MACIEL, 2022).

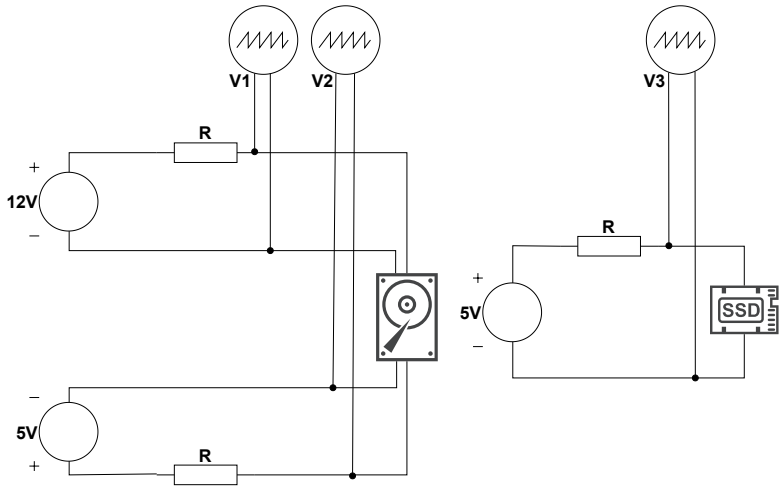


Figure 4.5: Electrical circuit for measurement of HDD and SSD voltage values (own work (2023)).

Table 4.1: Experiment components.

component	description
Main HDD	HDD 500GB
80*GBHDD*	HDD 80GB
500*GBHDD*	HDD 500GB
1*TBHDD*	HDD 1TB
1*TBWDHDD*	HDD 1TBWD
120*GBSSD*	SSD 120GB
Server	quad-core 3.10GHz 8GB RAM

This work adopts Mercury (SILVA et al., 2013; OLIVEIRA et al., 2017) and TimeNET (ZIMMERMANN et al., 2006) tools for evaluating GSPN models. The validation has been carried out on a computer with Intel core 2 Duo 2.4GHz, 8GB RAM, Windows 10.

4.4.1 HDDs and SSDs failure logs

This section outlines the monitoring software and datasets employed in this study to acquire health and performance information from solid-state and hard disk drives.

4.4.1.1 SMART logs

SMART (COMMITTEE, 1995) is a commonly utilized software designed for monitoring distinct characteristics of storage devices. To accomplish this, performance and reliability metrics are gathered and compared to pre-established thresholds in order to report the current behavior of such devices.

In this study, a subset of seven out of approximately 255 distinct attributes from SMART

Table 4.2: SMART attributes adopted for HDD and SSD analysis (✳ means an attribute is included in the respective technology).

attribute	description	SSD	HDD
5	Reallocated sectors count	✳	✳
173	Wear leveling count	✳	
187	Uncorrectable errors		✳
188	Command timeout		✳
198	Uncorrectable sector count		✳
241	Number of blocks written	✳	✳

logs has been selected, prioritizing those widely recognized as strongly correlated with device failure (HAN et al., 2021; ZHANG et al., 2019b, 2020), to conduct the exploratory analysis and failure behavior collecting. However, as SMART software is technology- and vendor-specific, discrepancies between storage models may occur. Therefore, this selection process aims to contemplate attributes in common among the storage devices featured in the utilized datasets. Table 4.2 presents the chosen attributes for SSDs and HDDs and their respective descriptions.

4.4.1.2 Datasets

Two public datasets have been adopted to investigate the behavior and failure rates of SSDs and HDDs. The first dataset contains 18387 failed SSD' tickets from Alibaba's data centers containing 965495 SSDs from three distinct manufacturers and 11 models. Such SSDs were monitored over two years. In addition, the dataset allows to identify which application was used for a given SSD. The following applications have been identified: *Data Analytics Engine* (DAE), *Database* (DB), *Network Attached Storage* (NAS), *Resource Management* (RM), *SQL Services* (SS), *Web Proxy Services* (WPS), *Web Services* (WS), *Web Service Management* (WSM), none (not identified).

The second dataset contains information regarding HDDs from a Backblaze data center, including 231309 HDDs from four manufacturers and 29 models. Backblaze has been monitoring such devices for eight years, during which 2963 failures have occurred. Unfortunately, this dataset does not explicitly specify which applications have been adopted during the monitoring period, which initially may reduce the accuracy of further failure predictions based on statistics extracted from these logs.

4.5 SUMMARY

This chapter presented the proposed methodology for modeling and evaluating homogeneous and hybrid storage systems. The methods explained herein consist of two approaches. While one has dealt with modeling and evaluating models for performance and energy consumption, the other has addressed a similar question for dependability. The model design process and investigating data from real devices have been shown in both directions. The technique adopted for planning the experiments has also been explained, along with the method and statistical techniques for evaluating storage systems' performance, dependability, and energy consumption. It has then presented the tools and measurement environment adopted for validating the GSPN models and obtaining essential data to conduct the experiments. This chapter also detailed the datasets from data collection regarding SMART attributes of HDDs and SSDs, which have been performed in the data centers of Alibaba and Backblaze companies.

5

MODELS

This chapter presents the performance and dependability models conceived to represent data storage systems. Section 5.1 briefly introduces the proposed performance models and their limitations, assumptions, and metrics of interest. In addition, the mathematical notation used in this thesis to calculate the performance and energy consumption metrics is introduced. Subsequently, the two proposed performance models and metrics of interest are presented in detail in Sections 5.1.1 and 5.1.2. These sections show how workload characteristics can be modeled using the adopted GSPN formalism elements and how delays are approximated through moment matching to non-exponential distributions. Section 5.1.1.1 describes how this technique can be adopted for the latter. Section 5.2 explains the dependability modeling approach proposed in this thesis. Then, the conceived availability and performability models are presented in Sections 5.2.1 and 5.2.2, respectively. Section 5.2.2.1 shows an example of the adopted hierarchical modeling technique.

5.1 PERFORMANCE MODELING

The conceived performance models represent read and write operations under different workloads, access patterns, and object sizes. Besides, the modeling approach has been conceived for stationary analysis (BALBO, 2001), in which (without loss of generality) the analysis assumes a system's long run.

Two models are proposed, and they are based on GSPN formalism: (i) single storage model; and (ii) multiple storage model. The single storage model represents client requests to a system with a single storage device (e.g., SSD) or a hybrid system as a black box (i.e., without

distinguishing its components). The multiple storage model is adopted for assessing the impact of workloads on different arrangements of storages (e.g., hybrid storage systems). Unlike the single model, this approach allows system designers to explicitly evaluate the components of hybrid systems.

The metrics of interest are throughput, mean response time, and energy consumption. Throughput represents IOPS (MEISTER; BRINKMANN, 2010), which estimates the amount of processed requests (write or read) in one second. Mean response time is the average time for a single operation to complete.

The proposed modeling approach also allows the analysis/verification of behavioral and structural properties (MURATA, 1989). As an example, a given model can be bounded and live. The former indicates the state space size is finite, and, thus, no data overflow may occur in buffers (e.g., multiple unprocessed write/read operations). The latter means the absence of deadlock states (MURATA, 1989).

For the sake of explanation, the multiple storage model is presented with only two different devices (HDD and SSD). However, this is not a limitation of the model, which is capable of representing storage systems with additional components (e.g., 4 HDDs; 2 SDDs and 4 HDDs). Additional storages may lead to state space size explosion (VALMARI, 1998), but simulation techniques may also be taken into account, as an alternative to CTMC generation (MELO et al., 2015).

Specific features, such as metadata manipulation, are not explicitly represented on the conceived models, as, in the context of storage devices, there is no distinction of the data type being accessed or stored. Also, this study has assumed that data management mechanisms (garbage collection and wear leveling) eventually occur, and, thus, the time for their execution is considered on mean delays for the write/read operations. Similarly, the proposed approach does not deal with interferences in flash memory cells, since they are not the focus of this work. However, as such occurrences may affect response time, they may also be considered on mean delays. This abstraction level allows the assessment of different systems in a more concise manner, without dealing with a detailed model that may not be feasibly evaluated.

Regarding workload, this work assumes two access patterns: random and sequential. These two types of access represent widely relevant and comprehensive usage scenarios for most storage applications and workloads. They represent real demands, applicability in various use cases, and the ability to evaluate the maximum performance (such as maximum bandwidth)

and minimum performance (minimum latency). Given the importance of assessing storage systems when processing various object sizes, objects were chosen to be either small or large to better evaluate their impact on HDDs and SSDs. The reason for not considering medium-sized objects is that, concerning object sizes, minor variations usually do not significantly affect the performance and energy consumption of the storage systems (WU et al., 2018; MEI et al., 2019; HSU; SMITH, 2004).

Concerning energy consumption, the proposed approach has focused on assessing storage devices during the active energy state (i.e, when processing read and write operations) and, thus, other energy states are not explicitly represented. However, as idle and standby states may eventually occur between requests, the respective effects on mean delays for power values have been considered.

The following notation is adopted: $E\{\#p\}$ represents the mean value of the inner expression, in which $\#p$ denotes the number of tokens in place p; and $W(T)$ represents the firing rate associated with transition T.

Additionally, function $\eta : T_{imm} \rightarrow [0,1]$ maps each immediate transition ($t \in T_{imm}$) to a normalized weight. More specifically, the weights represent the transition firing probability in a conflict set (BALBO, 2001), and, for the adopted models, each immediate transition can only be in one conflict set due to the characteristics of a request, which can be categorized as either random or sequential, small or large, and read or write. Next sections present the models using building blocks (i.e., submodels).

5.1.1 Single storage model

Figure 5.1 depicts the GSPN model for representing systems with a single storage. Additionally, Table 5.1 shows the existing transitions in the model and their respective attributes (type, server semantics, weight, and priority).

workload generator block is responsible for representing user requests. The marking of place *pRequests* (N) denotes the amount of concurrent requests from simultaneous clients (workers), and transition *tRequesting* indicates the arrival of a request within a storage. This transition adopts *infinite server semantics* (BALBO, 2001) to represent concurrent arrivals. Tokens in place *pForward* represent the request prepared for writing ($tWrite$) or reading ($tRead$).

A block *workload classifier*$_{op}$ is adopted for each operation. Transitions $tWrite$ and *tRead* denote the amount of requests for the respective activity, and they have weights indicating

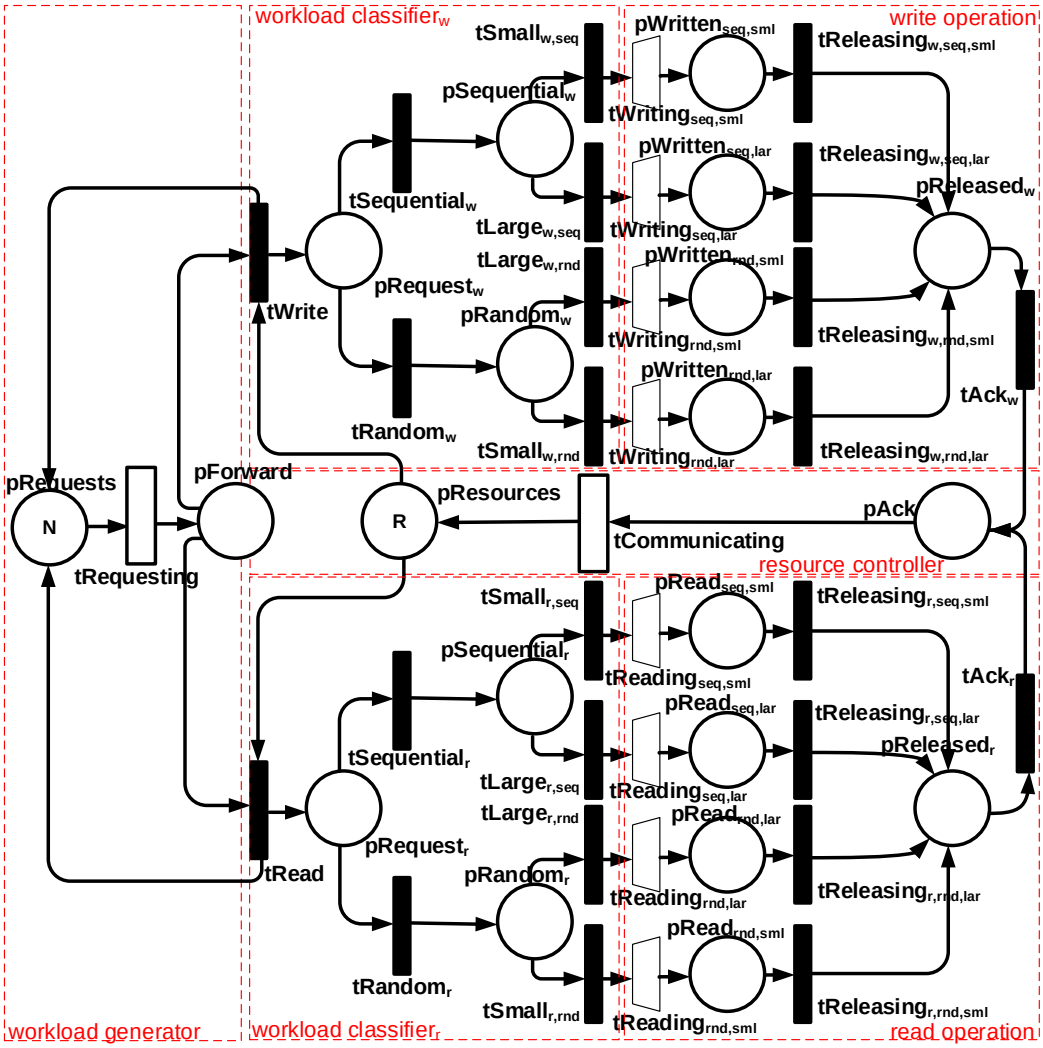


Figure 5.1: Single storage model (BORBA; TAVARES; MACIEL, 2022).

the probability of each operation. For instance, in *mixed* operations, read and write may have the same probability (0.5). Tokens in places $pRequest_{op}$ indicate read or write requests are queued. Immediate transitions $tSequential_{op}$ and $tRandom_{op}$ define the access pattern for a workload, and, similarly, their weights indicate the amount of requests associated with each pattern. Transition $tSmall_{op,pt}$ and $tLarge_{op,pt}$ represent the object size.

write and *read operation* blocks model the operation execution, and the delay is denoted by s-transition $tWriting_{pt,os}$ and $tReading_{pt,os}$ (Section 2.6.3). Tokens in places $pWritten_{pt,os}$ and $pRead_{pt,os}$ represent the conclusion of an activity. $tReleasing_{op,pt,os}$ and $tAck_{op}$ indicate the notification of resource release to the storage controller.

resource controller block denotes the storage readiness to execute read or write operations. A token in place *pAck* indicates a resource is ready to be released, in which the communication

with the controller is depicted by transition *tCommunicating*. Besides, the marking of place *pResource* (R) indicates the storage is ready for executing one or more operations. Also, the marking of place *pResouce* (R) may denote the adopted technology. For instance, for traditional SSDs (SATA interface), the marking of place *pResource* is 1, as only one operation at a time is carried out (i.e., writing/reading and releasing of resource) (KIM; KIM; KIM, 2020). Concerning SSDs-NVME, the marking in place *pResources* may be assumed as the number of threads for concurrently processing I/O requests (I/O completion thread) (KIM; KIM, 2017). In general, eight simultaneous threads are suitable to represent SSDs-NVMe (BAHN; CHO, 2020).

Table 5.1: Transition attributes - single storage model.

transition	type	server semantics	weight	priority
$tRequesting$	timed	infinite server	-	-
$tWrite$	immediate		κ	1
$tRead$	immediate		$1-\kappa$	1
$tSequential_{op}$	immediate		$1-\alpha$	1
$tRandom_{op}$	immediate		α	1
$tSmall_{op,pt}$	immediate		β	1
$tLarge_{op,pt}$	immediate		$1-\beta$	1
$tWriting_{pt,os}$	timed	single server	-	-
$tReading_{pt,os}$	timed	single server	-	-
$tReleasing_{op,pt,os}$	immediate		1	1
$tAck_{op}$	immediate		1	1
$tCommunicating$	timed	infinite server	-	-

For the proposed model, the mean response time is estimated using Little's law (TRIVEDI, 2008), expressed in Equation 5.1, where R represents the mean response time, L is the average number of requistions, and λ represents the arrival rate of requests. For this model, Equations 5.2 and 5.3 show the calculations used to obtain L and λ, respectively. The system throughput (i.e., IOPS) is estimated using Equation 5.4.

$$R = L/\lambda \tag{5.1}$$

$$L = N - E\{\#pRequests\} \tag{5.2}$$

$$\lambda = E\{\#pRequests\} \times W(tRequesting) \tag{5.3}$$

$$TH = E\{\#pAck\} \times W(tCommunicating) \tag{5.4}$$

For energy consumption, the workload features (e.g., access pattern) must be taken into account, as they influence the system power consumption. This work takes into account the proportion of each factor, which is represented as weights in immediate transitions ($\eta(t)$). For the single device model, the following weights are taken into account: $\eta(tWrite) = \kappa$; $\eta(tRead) = 1-\kappa$; $\eta(tRandom) = \alpha$; $\eta(tSequential) = 1-\alpha$; $\eta(tSmall) = \beta$; and $\eta(tLarge) = 1-\beta$. System energy consumption (EC) is then estimated as follows:

$$\begin{aligned} EP_w = &\kappa(EP_{w_1} * \alpha * \beta + EP_{w_2} * (1-\alpha) * \beta + EP_{w_3} \\ &* \alpha * (1-\beta) + EP_{w_4} * (1-\alpha) * (1-\beta)), \end{aligned} \tag{5.5}$$

$$\begin{aligned} EP_r = &(1-\kappa) * (EP_{r_5} * \alpha * \beta + EP_{r_6} * (1-\alpha) * \beta + \\ &EP_{r_7} * \alpha * (1-\beta) + EP_{r_8} * (1-\alpha) * (1-\beta)), \end{aligned} \tag{5.6}$$

$$EC = (EP_w + EP_r) * TH * time. \tag{5.7}$$

EP_{op} is the mean power consumption for an operation (read - *r* or write - *w*), which is estimated using the mean power of each workload feature. For instance, EP_{w1} denotes the power of a write operation (w) using random access (α) and a small object (β). *time* is the time of interest.

The equations above allow the evaluation of different storage technologies' performance and energy consumption when subjected to workloads with distinct characteristics. It is essential to state that the modeled data storage systems and workloads must consider the devices' write and read operation delays (represented through phase-type distributions) and the composition of the operations (described as weights of the immediate transitions), respectively. Table 5.2 presents the metrics adopted for the solution proposed in this thesis.

Notably, concerning behavioral properties, the single-storage model is reachable, bounded, and free of deadlocks. For structural properties, the GSPN model is conservative and consistent.

Table 5.2: GSPN metrics - single storage model.

equation	metric	syntax
5.1	average response time	$R = (N - E\{\#pRequests\})/(E\{\#pRequests\} \times W(tRequesting))$
5.4	throughput	$TH = E\{\#pAck\} \times W(tCommunicating)$
5.7	energy consumption	$EC = (EP_w + EP_r) * TH * time$

5.1.1.1 Phase-type distribution example

This section presents an example for the adoption of moment matching technique to represent non-exponential distributions (Section 2.6.3). This example refers to the single storage model, but the same technique may be adopted for the multiple storage model described in next section.

Figure 5.2 contains a particular s-transition ($tWriting_{seq,sml}$), which represents the execution of a writing operation for a sequential workload (seq) and small object (sml). Let us consider a measurement activity was carried out, in which the mean value (μ_d) is greater than the standard deviation (σ_d) for such a write operation ($\mu_d > \sigma_d$). This delay may be approximated using a hypoexponential distribution.

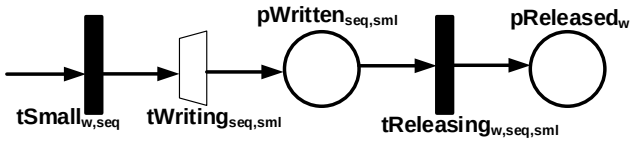


Figure 5.2: s-transition example (BORBA; TAVARES; MACIEL, 2022).

Figure 5.3 depicts the modeled delay using a hypoexponential subnet. The parameters γ (number of phases), μ_1, and μ_2 are obtained using Equations 2.11, 2.13, and 2.14, respectively. The average delay assigned to exponential transition $tWriting1_{seq,sml}$ is μ_1 ($\lambda_1 = 1/\mu_1$), and the delay associated with transition $tWriting2_{seq,sml}$ is μ_2 ($\lambda_2 = 1/\mu_2$). γ is an integer value, which is the arc weight from immediate transition t_0 to place p_1 and from $tWriting2_{seq,sml}$ to place $pWritten_{seq,sml}$.

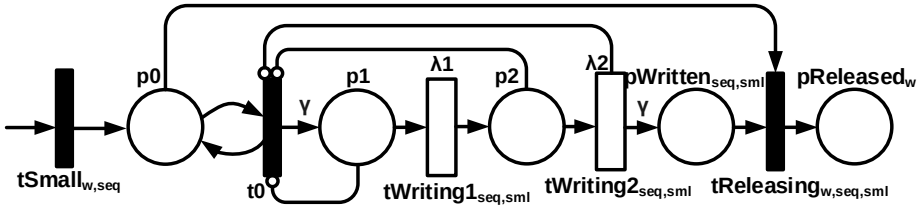


Figure 5.3: Hipoexponential subnet example (BORBA; TAVARES; MACIEL, 2022).

However, the measurement activity results may have a different mean and standard deviation relation. In this case, the delay of the given operation can be approximated to other distributions as follows:

- If $\mu_d = \sigma_d$, then the exponential distribution can be considered to represent write operations in data storage systems. Figure 5.4 illustrates a subnet that represents an exponential distribution with rate λ, which is associated with the $tWriting_{seq,sml}$ transition;

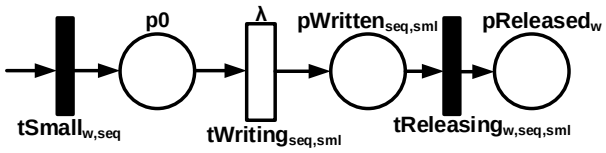


Figure 5.4: Exponential example (own work (2023)).

- Assuming $\mu_d/\sigma_d \in \mathbb{N}$ e $\mu_d/\sigma_d \neq 1$, the delay is approximated to an Erlang distribution, which is modeled as shown in (Figure 5.5). In this case, the rate associated with the transition $tWriting_{seq,sml}$ is λ, whereas the number of phases is represented by γ.

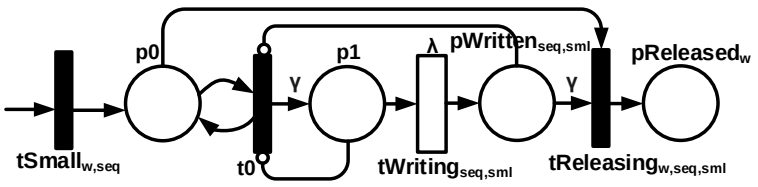


Figure 5.5: Erlang subnet example (own work (2023)).

- If $\mu_D < \sigma_D$, the approximation is modeled as a hyperexponential subnet (Figure 5.6). The parameters of this distribution are λ_h (rate), w_1 and w_2 (weights). λ_h is associated with the $tWriting_{seq,sml}$ transition, whereas w_1 and w_2 are assigned to immediate t_0 and t_2 transitions, respectively.

5.1.2 Multiple storage model

Figure 5.7 depicts the GSPN model for representing systems with multiple storage devices. For a better understanding, this section presents the model using a hybrid storage system (1 SSD and 1 HDD). However, it is essential to note that homogeneous device arrangements (i.e., only HDDs or SSDs) can be represented. The attributes of the transitions in the multiple model are listed in Table 5.3, and the same terminology as in the previous section is adopted.

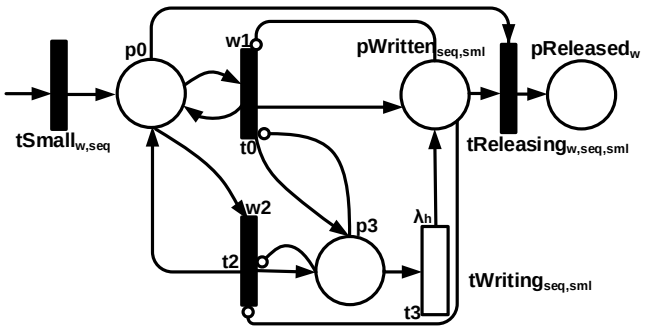


Figure 5.6: Hiperexponential subnet example (own work (2023)).

Table 5.3: Transition attributes - multiple storage model.

transition	type	server semantics	weight	priority
$tRequesting$	timed	infinite server	-	-
$tForward_d$	immediate		$\eta(tForward_d)$	1
$tController_d$	immediate		1	1
$tWrite_d$	immediate		κ	1
$tRead_d$	immediate		$1-\kappa$	1
$tSequential_{d,op}$	immediate		$1-\alpha$	1
$tRandom_{d,op}$	immediate		α	1
$tSmall_{d,op,pt}$	immediate		β	1
$tLarge_{d,op,pt}$	immediate		$1-\beta$	1
$tWriting_{d,pt,os}$	timed	single server	-	-
$tReading_{d,pt,os}$	timed	single server	-	-
$tReleasing_{d,op,pt,os}$	immediate		1	1
$tAck_{d,op}$	immediate		1	1
$tCommunicating$	timed	infinite server	-	-

Similar to the previous model, *workload generator* block represents the creation of user requests, in which the marking N in place $pRequests$ indicates the number of concurrent requests. Timed transition $tRequesting$ adopts *infinite server semantic* to represent concurrent arrivals. Immediate transitions $tForward_d$ denote a request is redirected to a storage d. Tokens in places $pHDD$ and $pSSD$ ($pStorage$) indicate read or write requests are queued in a storage device.

Similar to the single storage model, *read operation*$_d$ and *write operation*$_d$ blocks represent, respectively, the reading and writing activities. For each storage device in the system, both blocks are adopted.

resource controller$_d$ block models the available resources for performing an operation in a request. The number of tokens (e.g., R_1) in places $pResource_d$ denotes the number of operations are concurrently carried out. Transition $tController_d$ represents the device is informing the controller about the conclusion of an operation.

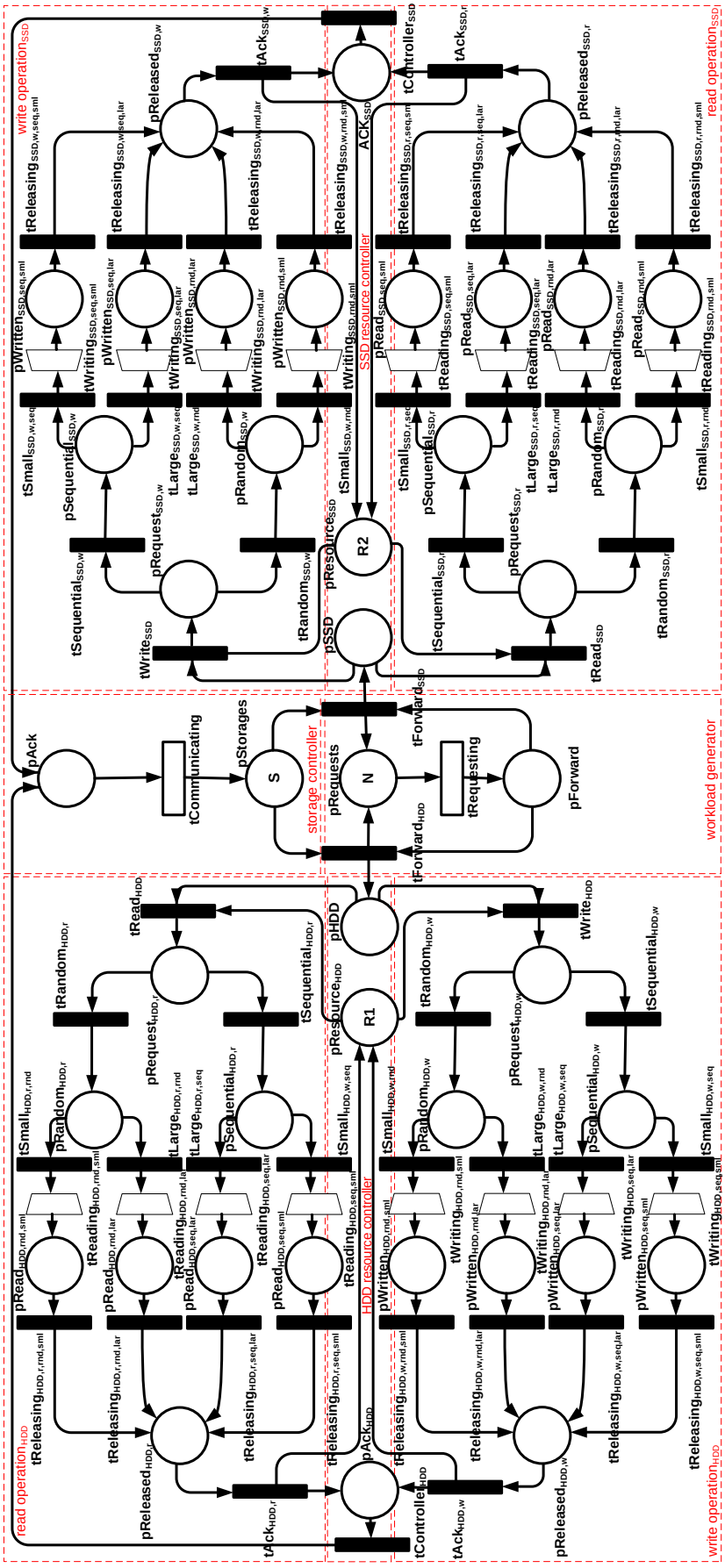


Figure 5.7: Multiple storage model (BORBA; TAVARES; MACIEL, 2022).

In *storage controller* block, a token in place *pAck* represents a storage concluded the operation, and transition *tCommunicating* denotes the controller delay for receiving the acknowledgment. This work assumes the storage controller can simultaneously receive acknowledgments from all devices (i.e., *infinite server semantics*). The marking in place *pStorages* (S) denotes the number of devices in the system.

Mean response time and throughput are estimated using Equations 5.8 and 5.9, respectively. Energy consumption (EC_h) is obtained from the power consumption of the workload features ($EP_{d,op,i,j}$) in all storage devices (n):

$$R_h = (N - E\{\#pRequests\})/(E\{\#pRequests\} \times W(tRequesting)), \tag{5.8}$$

$$TH_h = E\{\#pAck\} \times 1/W(tCommunicating), \tag{5.9}$$

$$EP_d = \sum_{op}\sum_{i}\sum_{j} \eta(op) * \eta(i) * \eta(j) * EP_{d,op,i,j}, \tag{5.10}$$

$$EC_h = \left(\sum_{d=0}^{n} \eta(tForward_d) * EP_d \right) * TH_h * time, \tag{5.11}$$

where $op \in (tWrite_d, tRead_d)$, $i \in (tSequential_{d,op}, tRandom_{d,op})$ and $j \in (tSmall_{d,op,i}, tLarge_{d,op,i})$.

The model has been presented considering two distinct devices for a hybrid system. However, additional devices can be included by considering additional *read*, *write* and *resource controller* blocks. For a better understanding, the equations utilized to calculate the metrics of interest using the designed model have been gathered in Table 5.4.

Table 5.4: GSPN metrics - multiple storage model.

equation	metric	syntax
5.8	average response time	$R_h = (N - E\{\#pRequests\})/(E\{\#pRequests\} \times W(tRequesting))$
5.9	throughput	$TH_h = E\{\#pAck\} \times 1/W(tCommunicating)$
5.11	energy consumption	$EC_h = \left(\sum_{d=0}^{n} \eta(tForward_d) * EP_d\right) * TH_h * time$

Notably, concerning behavioral properties, the multiple storage model is reachable, bounded, and free of deadlocks. For structural properties, the designed GSPN model is conservative and consistent.

5.2 DEPENDABILITY MODELING

This section presents the dependability modeling approach proposed in this thesis, which includes an *availability model* for assessing system availability and a *performability model* to evaluate its impact on performance.

Regarding the availability model, arrangements in series or parallel (in this thesis, hot standby redundancy) of different technologies can be described to represent several compositions of a storage node. This model also allows the representation of spare components, which is a common approach adopted by data centers and cloud computing environments. It is important to emphasize that only storage devices have been considered for representing such composition, as the refereed component is the focus of this work.

The performability model allows the representation of data requests (read or write) and their processing, and can estimate the performance of storage nodes when subject to failures. Specifically, this model assumes a composite measure to describe the degradation in the performance of storage nodes as a result of a failure. In this work, a hierarchical modeling approach has been adopted for combining results from the proposed availability model into the conceived performability model. It is important to note that the performability model has been conceived for stationary analysis (BALBO, 2001), in which (without loss of generality) the analysis assumes a system's long run. However, simulation techniques may also be adopted for estimating performance and dependability metrics.

The metrics of interest are availability, throughput and response time. Mean response time is the average time for a single operation to complete and is estimated using Little's law $R = L/\lambda$ (TRIVEDI, 2008), in which R represents the mean response time, L is the average number of requisitions and λ represents the arrival rate of requests. Throughput (IOPS) quantifies the number of requests (write or read) processed within a single second. Availability is the probability of a system to be in an operational state. The availability of the system may be obtained by the mean time to failure (MTTF) and mean time to repair (MTTR) of a given system. Thus, availability of a component j is estimated as $A_j = MTTF/(MTTF + MTTR)$ in which $MTTF$ is the mean time to failure and $MTTR$ is the mean time to repair (MACIEL et al., 2011) for such a component.

As a limitation of the conceived models, it is worth to state that filesystems, metadata manipulation, cache memories, and energy states are not explicitly represented, since they are

not the focus of this work. On the other hand, this abstraction level allows the assessment of different systems without dealing with a detailed model that may not be feasibly evaluated.

For the sake of explanation, the availability and performability models are presented with only three and two different nodes, respectively. However, this is not a limitation of the model, which is capable of representing storage systems with additional nodes.

Additional storages may lead to state space size explosion (VALMARI, 1998), but simulation techniques may also be taken into account, as an alternative to CTMC generation (MELO et al., 2015). Next sections present the models using building blocks (i.e., submodels).

5.2.1 Availability model

Figure 5.8 depicts the conceived reliability block diagram for modeling the availability of storage systems. Three distinct configurations are assumed, but other arrangements can be adopted to represent more complex redundancy policies (e.g., RAID). For this specific abstract model, the system's operational state is given by its functional components; therefore, at least one device (HDD or SSD) in any of the storage nodes must be operational.

A block $Node_k$ models the arrangement in parallel (at least one of them must be operational) of distinct technologies (HDD and SSD). $HDD_{x,k}$ denotes the HDD of number x and $SSD_{w,k}$ indicates the SSD of number w, both deployed at the same storage node k. It is worth to note that, in this type of storage node, the number of hard-disk drives and solid-state drives are not necessarily equal.

$Node_i$ and $Node_j$ blocks denote homogeneous technologies in parallel. x indicates the amount of redundant HDDs enforced at the storage node i. As for j, it represents a storage node containing only SSDs, and, for this case, w represents the number of flash-based devices.

To represent a particular system, a refined model can be conceived and, in this case, the respective MTTFs and MTTRs must be specified by the system designers. Afterwards, a reduction technique based on series and/or parallel arrangement can be utilized to obtain the single block (KUO; ZUO, 2003).

5.2.2 Performability model

Figure 5.9 depicts the GSPN model for representing data processing in storage systems with multiple storage nodes. For a better understanding, this section presents the model using two storage nodes.

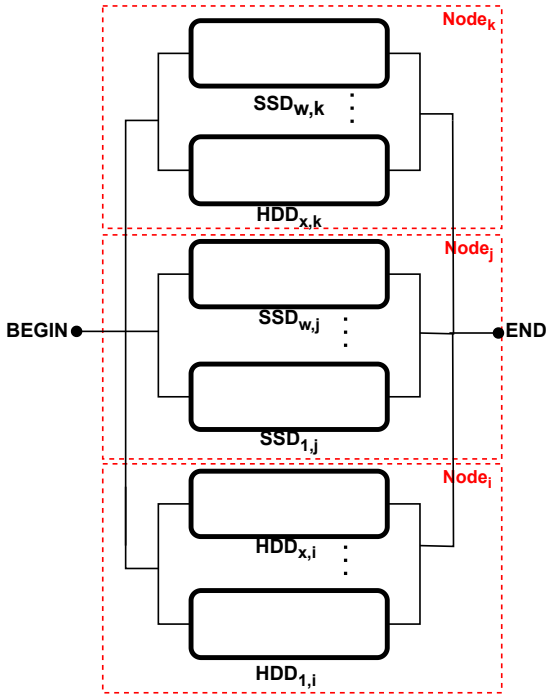


Figure 5.8: RBD model for three storage node configurations (own work (2023)).

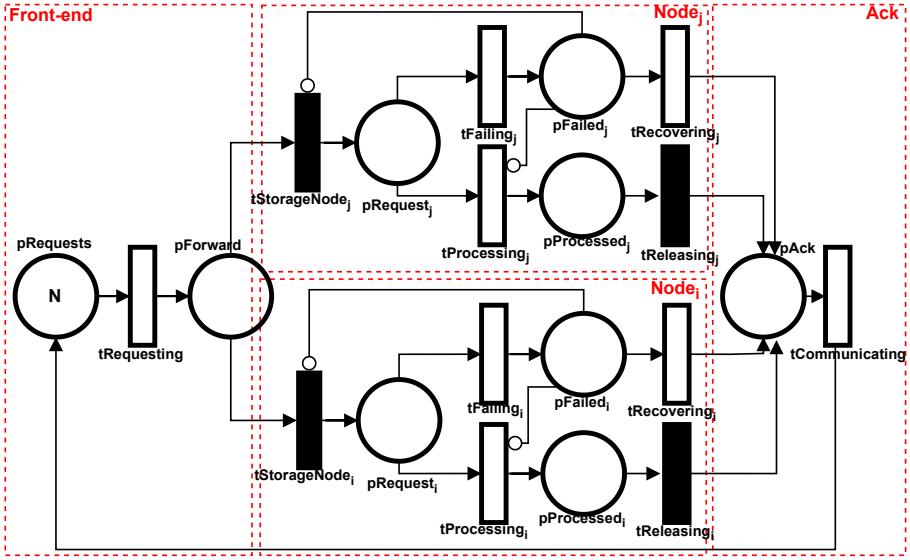


Figure 5.9: Performability model (own work (2023)).

front-end block is responsible for representing the creation of data requests. The marking of place *pRequests* (N) denotes the initial state, which indicates the number of concurrent requests in the system. Transition *tRequesting* adopts infinite server semantics (BALBO, 2001) in order to represent concurrent arrivals of requests. Tokens in place *pForward* represents a request prepared to be redirected to the respective storage node.

$Node_i$ and $Node_j$ blocks represent the processing of requests using two storage nodes

(i and j). Firing of the immediate transitions $tStorageNode_i$ or $tStorageNode_j$ indicates a request is redirected to one of the storage nodes. The probability of redirection to a storage node is modeled as weight in the respective immediate transition (e.g., weight 0.5 for each transition) (AJMONE MARSAN; CONTE; BALBO, 1984b)). However, since a storage node may not be operational, an inhibitory arc guarantees this transition as active only when the respective node is available (denoted by the absence of tokens at $pFailed$ locations). Tokens in places $pRequest$ indicate read or write requests are queued and ready for processing. Transitions $tFailing$ firing represents a failure of a storage node. The loss of the requisition (due to a failed storage node) is indicated by the presence of tokens in places $pFailed$. The maintenance of a failed node is represented by transitions $tRecovering$. Transitions $tProcessing$ denotes the actual processing of the request, and its conclusion is depicted by tokens in places $pProcessed$. If the respective storage node is not operational, an inhibitory arc prevents this transition from being active. The immediate transition $tReleasing$ and the timed transition $tRecovering$, respectively, indicate the notification of resource release and the repair process for the corresponding storage node. All timed transitions in this block adopt infinite server semantics, as each node may contain one or more storage devices and, therefore, process multiple requests.

In Ack block, a token in place $pAck$ represents a storage node concluded the operation, and transition $tCommunicating$ denotes the delay for transmitting the acknowledgment. This work assumes the storage controller can simultaneously receive acknowledgments from all storage nodes (i.e., it adopts infinite server semantics).

Additionally, this work considers the processing of requests based on exponential distribution, an approach similar to that adopted in previous studies (VARKI et al., 2004; KHAZAEI; MISIC; MISIC, 2012).

For the proposed model, mean response time is then estimated as follows:

$$R = \frac{N - E\{\#pRequests\}}{E\{\#pRequests\} \times W\left(tRequesting\right)}. \tag{5.12}$$

System throughput (i.e., IOPS) is estimated as:

$$TH = E\{\#pAck\} \times W\left(tCommunicating\right). \tag{5.13}$$

It is important to note that the model has been presented considering two distinct storage nodes. However, additional nodes can be included by considering additional $Node_i$ and $Node_j$

blocks. $Node_k$ blocks may also be integrated to represent storage nodes with hybrid storage technologies (HDDs and SSDs).

5.2.2.1 Hierarchical modeling example

This section presents an example of adopting a hierarchical modeling approach to investigate the effects of data storage device failures on their performance. This example refers to a storage system containing two nodes, but the same technique can be adopted to design systems with more nodes and different internal arrangements of storage devices. In addition, an example of the proposed performability model's execution can be found in Appendix C.

Figure 5.10 depicts the modeled storage system using the proposed performability and availability models. In the GSPN model, the storage node i contains the transitions $tFailing_i$ and $tRecovering_i$, which represent the time between failures and repairs, respectively. The MTTF and MTTR delays assigned to these respective transitions are obtained by computing the blue-highlighted RBD. In other words, the parallel arrangement between $HDD_{1,i}$ and $HDD_{x,i}$ is reduced to one single block (KUO; ZUO, 2003) and, next, the metrics of interest are estimated.

Similarly, the MTTF and MTTR delays assigned to transitions $tFailing_j$ and $tRecovering_j$ also derives from the reduction of a parallel arrangement ($SSD_{1,j}$, $SSD_{2,j}$ and $SSD_{w,j}$). However, in this case, the red-highlighted RBD one (Figure 5.10) represents the storage node of interest.

Finally, the computation of the resulting hierarchical modeled storage system can provide performance results for the modeled storage nodes (Equations 5.12 and 5.13); however, in this case, taking into account the possible effects that failures may infringe on performance.

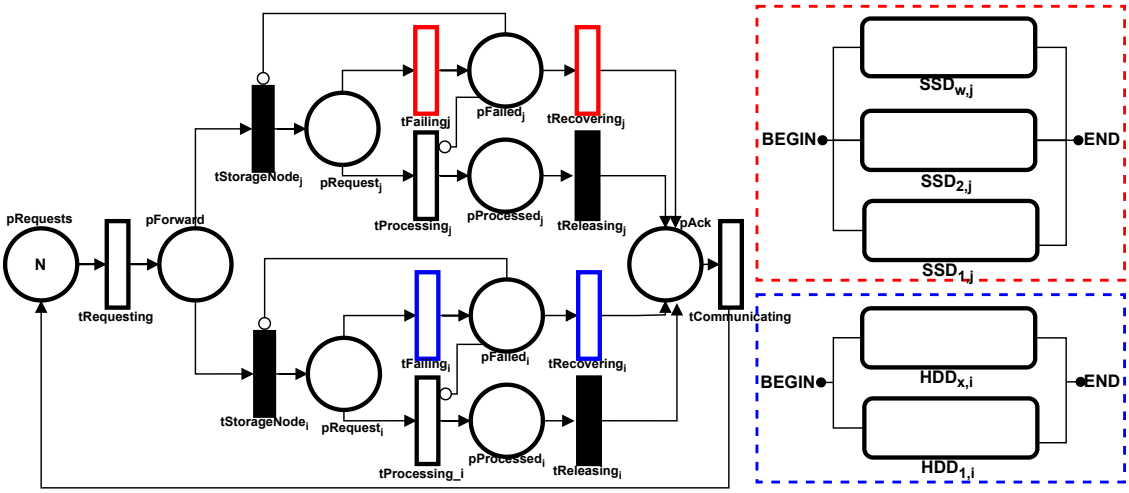


Figure 5.10: Hierarchical modeling example (own work (2023)).

5.3 SUMMARY

This chapter presented the GSPN and RBD models designed to represent data storage systems and estimate performance, energy consumption, and availability. Initially, performance and dependability models have been briefly introduced in their respective sections to present the purpose of the proposed solution. Subsequently, the considerations and limitations of the adopted approach have been addressed (e.g., specific features, such as metadata, are not represented). The mathematical notation for understanding the metrics of interest is introduced, and then the conceived models are presented. Examples have been provided to demonstrate the adopted moment matching and hierarchical modeling technique. This chapter also detailed the metrics of interest (average response time, throughput, availability, and energy consumption) and their respective equations.

The models presented in this chapter make it possible to represent homogeneous and hybrid storage systems under different workloads, which is of great importance for system designers. However, it is important to emphasize that in order to adopt these models, it is necessary to have a previous grounding regarding the formalism adopted for their conception.

6

EXPERIMENTS

This chapter presents experimental results to demonstrate the practical feasibility of the modeling approach proposed in this thesis. The conducted experiments follow the methodology outlined in Chapter 4, which also contains the details and characteristics of the adopted tools, techniques, and datasets. Section 6.1 shows an exploratory analysis of the performance and energy consumption data collected by measuring an SSD and HDDs under different workloads. Section 6.2 introduces the workloads, considerations, and storage characteristics assumed to validate the proposed GSPN models. It then demonstrates the moment matching technique for approximating delays to non-exponential distributions, in addition to the results of the conducted validations. Section 6.3 details the experiments conducted using the proposed models. Specifically, a screening and four workload-driven experiments demonstrate the feasibility of the designed models for evaluating homogeneous and hybrid data storage systems. Furthermore, a case study demonstrates the utilization of models for cost planning in data centers, followed by a study on model scalability. Section 6.4 concludes the chapter with the results of an exploratory analysis of industry-representative datasets containing information on storage failures.

6.1 MEASUREMENT EXPERIMENT - EXPLORATORY ANALYSIS

This section presents an evaluation of the performance and energy consumption of storage devices and hybrid mechanisms in order to provide insights into the benefits of each technology. This analysis employed performance and energy consumption values obtained from HDDs and an SSD via a measurement experiment. These measurements are also used for the validation and experiments with the proposed models, which are shown in the following sections. Lastly,

a workload evaluation is presented to assist in the conception of an optimized data-placement policy for hybrid storage systems.

6.1.1 Experiment I: screening

This experiment adopts an approach based on DoE for identifying the suitable technology to compose a hybrid storage system. A factorial design is adopted, and five factors are taken into account by which it is possible to represent the workloads most found in cloud providers and, consequently, conduct a more credible analysis of storage systems (COUNCIL, 2019): storage technology (*technology*), object size (*object_size*), operation type (*operation*), access pattern (*pattern*), and number of threads (*workers*). These factors contemplate the following levels (Table 6.1): (i) *technology* - 80*GBHDD*, 500*GBHDD*, 1*TBHDD*, 1*TBWDHDD*, and 120*GBSSD*; (ii) *object_size* - 4*KB*, 128*KB*, 512*KB*, 1*MB*; (iii) *operation* - *write*, *read*, and *mix* (50% *read* + 50% *write*); (iv) *pattern* - *rnd* (random), *seq* (sequential), and 80%*rnd* (since in real workloads it is common for more than 80% of the write and read requisitions to be random (MONTAZERI et al., 2018); and (v) *workers* - 1, 2, and 4.

Table 6.1: Factors and levels.

Factor	Levels
technology	80GBHDD, 500GBHDD, 1TBHDD, 1TBWDHDD, 120GBSSD
object_size	4KB, 128KB, 512KB, 1MB
operation	write, read, mix
workers	1, 2, 4
pattern	rnd, seq, 80%rnd

Table 6.2 shows the mean values for each *technology* level, standard deviation (*StDev*), and the respective 95% confidence interval (95% *C.I.*). Results indicate 120*GBSSD* and 1*TBWDHDD* have the best values concerning response time (18.0905ms and 14.4540ms, respectively) and IOPS (128.6322 and 157.4852), respectively. Regarding energy consumption, 120*GBSSD* saves on average 7.48J compared to 80*GBHDD* (the worst *technology* level).

Taking into account only HDD technology, 1*TBWDHDD* has the best results concerning response time and IOPS. Although 500*GBHDD* seems to have a lower energy consumption, a t-test for equal means indicates the difference between 500*GBHDD* and 1*TBWDHDD* is not statistically significant.

Based on the aforementioned results, 1*TBWDHDD* and 120*GBSSD* are utilized to compose the hybrid storage system adopted in the next experiments.

Table 6.2: Experiment I - mean values.

technology	Energy consumption (J)			Response time (ms)			IOPS		
	Mean	StDev.	95% C.I.	Mean	StDev.	95% C.I.	Mean	StDev.	95% C.I.
80GBHDD	9.9047	2.1814	(9.7327; 10.0769)	36.1313	33.1560	(35.0180; 37.2440)	63.6399	95.1474	(62.1388; 65.2188)
500GBHDD	5.9244	1.0406	(5.7524; 6.0966)	28.5852	24.8860	(27.4720; 29.6980)	80.5769	131.6655	(78.1799; 83.1186)
1TBHDD	6.5848	0.6020	(6.4127; 6.7569)	28.9358	29.0490	(27.8230; 30.0490)	74.6976	89.7827	(72.6374; 76.8816)
1TBWDHDD	6.1126	7.4970	(5.9410; 6.2850)	14.4540	12.6020	(13.3410; 15.5670)	157.4852	252.5890	(148.5884; 167.5041)
120GBSSD	2.4209	4.5651	(2.2489; 2.5930)	18.0905	27.5640	(16.9780; 19.2030)	128.6322	101.0101	(122.6391; 135.2447)

6.1.2 Experiment II: hybrid storage evaluation

This section presents a comparative assessment of the performance and energy consumption by examining a hybrid storage system and its constituent components, including HDD and SSD.

Similar to the previous experiment, the same factors and levels are considered, but *technology* factor contemplates 120*GBSSD*, 1*TBWDHDD* and *Hybrid* (120*GBSSD*+1*TBWD HDD*). Some factor levels are fixed to represent real-world workloads. Real workloads may be composed of small (4KB) random (80%) requisitions, which commonly have mixed operations (read - 50.4%, write - 49.6%) from simultaneous clients (e.g., *workers*=4) (MONTAZERI et al., 2018).

Table 6.3 shows the analysis of variance (ANOVA) with significance level $\alpha = 0.05$. Column *Factor/Interaction* describes the most significant factors and second-order interactions. Other factors and interactions do not considerably impact the adopted metrics and, thus, they are not shown for readability purposes. *Error* represents noise in the measurements. The influence of each factor (or interaction) is represented by *Var.%* and *df* denotes the degree of freedom. *F-stat.* represents the F statistic with the respective *p-value*.

As follows, results are described using Tukey's procedure (a post-hoc test) (MONTGOMERY; RUNGER, 2014).

Table 6.3: Experiment II - ANOVA two-way analysis.

Factor/Interaction	Energy consumption				Response time				IOPS			
	Var.%	df	F-stat.	p-value	Var.%	df	F-stat.	p-value	Var.%	df	F-stat.	p-value
operation	0.11	2	11.89	≤ 0.001	4.84	2	640.68	≤ 0.001	5.29	2	628.85	≤ 0.001
technology	58.81	2	6580.78	≤ 0.001	1.47	2	194.94	≤ 0.001	10.37	2	1231.87	≤ 0.001
object_size	6.24	3	465.18	≤ 0.001	21.42	3	1889.89	≤ 0.001	23.49	3	1859.89	≤ 0.001
pattern	0.06	2	6.96	≤ 0.001	6.79	2	898.90	≤ 0.001	8.35	2	992.03	≤ 0.001
workers	0.00	2	0.33	0.721	15.88	2	2101.82	≤ 0.001	0.03	2	3.12	0.044
operation*technology	0.19	4	10.58	≤ 0.001	7.79	4	515.70	≤ 0.001	12.53	4	744.29	≤ 0.001
operation*workers	0.09	4	5.15	≤ 0.001	1.60	4	106.00	≤ 0.001	0.07	4	4.28	0.002
technology*object_size	4.69	6	174.85	≤ 0.001	2.24	6	98.95	≤ 0.001	5.67	6	224.58	≤ 0.001
Error	28.69	6401			24.19	6401			26.94	6401		

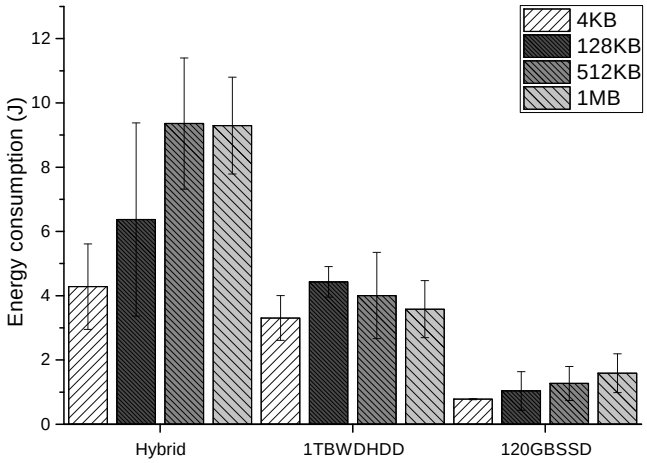


Figure 6.1: Experiment II - energy consumption (BORBA et al., 2020).

6.1.2.1 Evaluation of the impact of factors on energy consumption

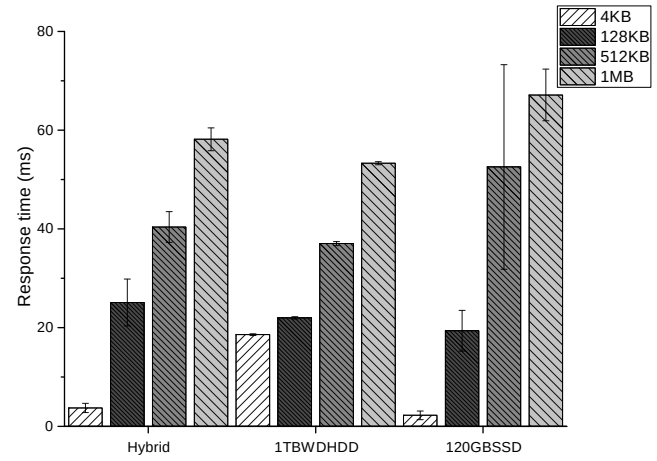
Table 6.3 indicates *technology* as the factor with the greatest impact on energy consumption (58.81%), followed by *object_size* (6.24%). Other factors have a small effect, even with a *p-value*$\leq$ 0.001. Concerning interactions, *technology* $*$ *object_size* is the only one with a substantial impact (4.69%). *Error* is also a significant source of variation (28.69%).

Figure 6.1 depicts an analysis about *technology* $*$ *object_size* interaction using 95% confidence interval. For this evaluation, *pattern* (80%*rnd*), *workers* (4), and *operation* (*mix*) have been fixed. 120*GBSSD* provides the best saving for all object sizes. Compared to 1*TBWDHDD*, the hybrid system increases the mean energy consumption for all object sizes: 29.53% (4*KB*), 43.85% (128*KB*), 133.57% (512*KB*), and 159.51% (1*MB*).

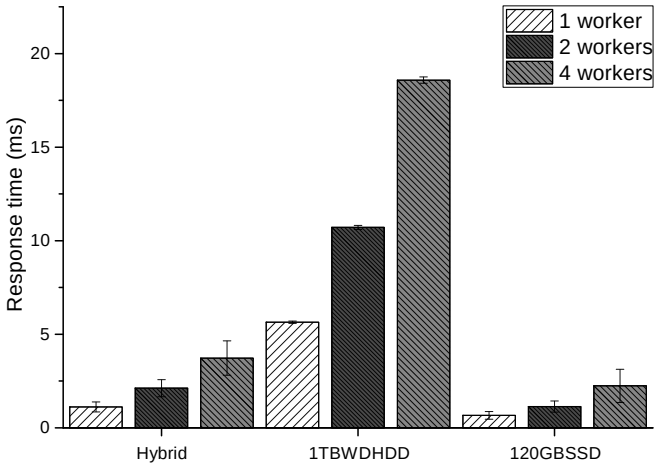
6.1.2.2 Evaluation of the impact of factors on response time

object_size accounts for most of the impact on response time (21.42%), as depicted in Table 6.3. Also, *workers* has a great influence (15.88%). Even with a *p-value* $\leq$ 0.001, *operation* (4.84%), *technology* (1.47%), and *pattern* (6.79%) have smaller contributions. Regarding interactions, *operation* $*$ *technology* (7.79%) and *object_size* $*$ *workers* (6.33%) represent the highest variation on response time.

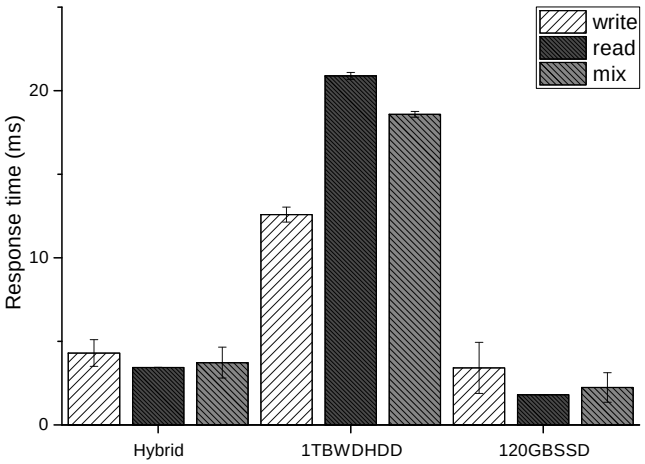
Figure 6.2 shows an analysis about the most significant factors and their interactions also using 95% confidence intervals. In Figure 6.2(a) (*object_size*), *pattern* (80%*rnd*), *workers* (4) and *operation* (*mix*) are also fixed. 120*GBSSD* is the best level for small data (4*KB* and 128*KB*), whereas 1*TBWDHDD* presents the best results for 512*KB* and 1*MB* object sizes. The



(a)



(b)



(c)

Figure 6.2: Experiment II - response time: (a) *object_size*; (b) *workers*; and (c) *operation* $*$ *technology* (BORBA et al., 2020).

hybrid system decreases response time by 79.92% for small objects (4*KB*) in comparison to 1*TBWDHDD*. Besides, hybrid system presents a response time 13.38% lower than 120*GBSSD* for larger objects (1*MB*).

Since *workers* has a substancial relevance on response time (Table 6.3), Figure 6.2(b) depicts its influence in *technology* factor. *operation* (*mix*), *object_size* (4*KB*), and *pattern* (80%*rnd*) are kept fixed. 1*TBWDHDD* presents the worst results for all *workers* levels. Compared to 1*TBWDHDD*, *Hybrid* decreases the mean response time in 80.21%, 80.19% and 79.92% for 1, 2 and four *workers*, respectively.

To evaluate the interaction of *operation* and *technology* the following factors have been fixed: *object_size* (4*KB*), *pattern* (80%*rnd*), and *workers* (4). Figure 6.2(c) highlights the improvement obtained by *Hybrid* level. Indeed, for write operations, considering the confidence intervals, *Hybrid* has a response time similar to 120*GBSSD*.

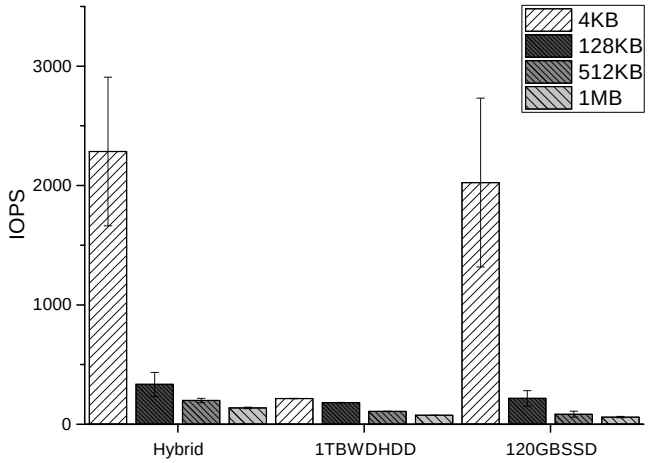
6.1.2.3 Evaluation of the impact of factors on IOPS

Regarding IOPS, *object_size* presents the major variation (23.49%) (Table 6.3). *technology* and *pattern* contemplate, respectively, 10.37% and 8.35% of variation for this metric. As *workers* presents a *p-value* $= 0.044$, there is no statistical evidence that it influences IOPS. $operation * technology$ interaction provides a significant contribution (12.53%), but other interactions have a minimal impact or do not statistically affects IOPS.

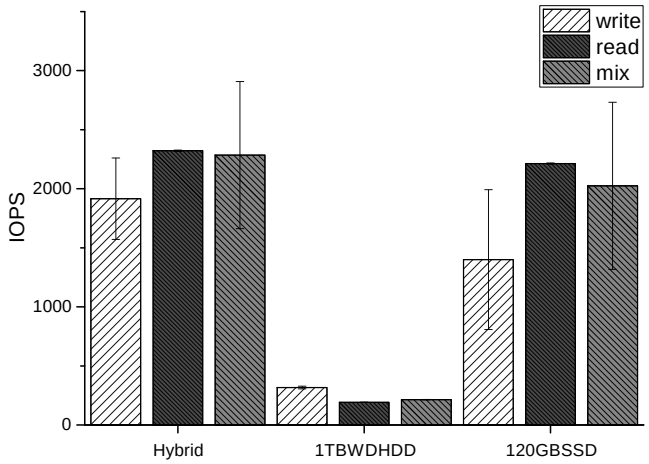
Figure 6.3 depicts an analysis regarding *object_size*, *pattern* and *technology*, using 95% confidence intervals. In Figure 6.3(a) (*object_size*), factors *operation*, *pattern*, and *workers* have fixed values: *mix*, 80%*rnd*, and 4, respectively. *Hybrid* approach provides better results than 120*GBSSD* and 1*TBWDHDD*. For instance, *Hybrid* increases the IOPS over 12.87% (4*KB*) and 53.62% (128*KB*) compared to 120*GBSSD*. Also, *Hybrid* presents a IOPS higher than 1*TBWDHDD*, 84.51% and 83.59% for 512*KB* 1*MB* levels, respectively.

Evaluation of $technology * operation$ is depicted in Figure 6.3(b). The levels for *object_size*, *pattern* and *workers* are 4*KB*, 80%*rnd* and 4, respectively. *Hybrid* provides better results than 1*TBWDHDD* and 120*GBSSD*. For instance, *Hybrid* increases IOPS about 36.83% and 4.9% for *write* and *read* operations in comparison to 120*GBSSD*. For *mix* level, there is no statistical difference between *Hybrid* and 120*GBSSD* levels.

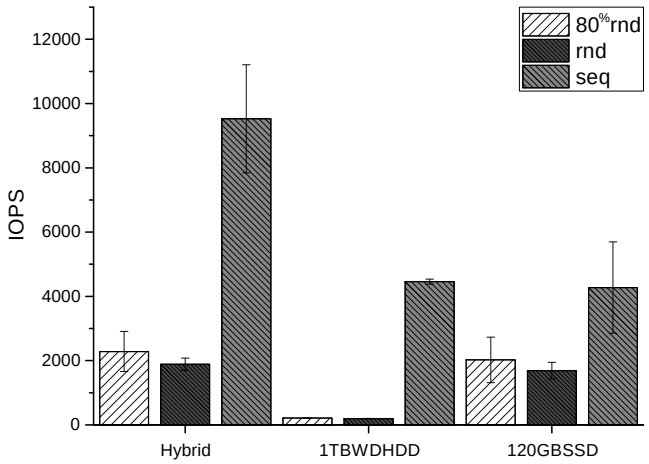
Figure 6.3(c) depicts the influence of *pattern* in *technology*. For this investigation, *operation*, *object_size*, and *workers* are *mix*, 4*KB*, and 4, respectively. However, for *seq* level,



(a)



(b)



(c)

Figure 6.3: Experiment II - IOPS: (a) $object\_size$; (b) $operation * technology$; and (c) $pattern$ (BORBA et al., 2020).

Hybrid significantly increases IOPS in comparison to 120*GBSSD* (122.94%) and 1*TBWDHDD* (113.69%).

6.1.3 Experiment III: composite desirability

To determine the optimal configuration for factor levels using the adopted metrics, this work adopts composite desirability (CD). Table 6.4 details the estimated values. For the sake of explanation, better configurations related to the hybrid approach (*Hybrid*) are highlighted.

For write operations, *Hybrid* has the highest CD with 80%*rnd* (4*KB* - CD=0.971670) and *rnd* (4*KB* - CD=0.973242; 512*KB* - CD=0.874362). Assuming read *operation*, for 512*KB* objects, *Hybrid* has better composite desirability than 120*GBSSD* and 1*TBWDHDD*: 0.933250 (80%*rnd*) and 0.933862 (*rnd*). Similarly, *Hybrid* presents the highest desirability factors for mixed operations: 0.891833 (80%*rnd*) and 0.893063 (*rnd*).

Additionally, the results also indicate important configurations for HDD (1*TBWDHDD*) and SSD (120*GBSSD*) individually, which are also helpful in conceiving new hybrid storage systems.

Table 6.4: Experiment III - Composite desirability.

		Write				Read				Mix			
		4KB	128KB	512KB	1MB	4KB	128KB	512KB	1MB	4KB	128KB	512KB	1MB
Hybrid													
	80%rnd	**0,971670**	0.930369	0.876671	0.856091	0.978956	0.955543	**0.933250**	0.913947	0.966710	0.942020	**0.891833**	0.860040
	rnd	**0.973242**	0.921885	**0.874362**	0.856213	0.982986	0.949870	**0.933862**	0.916804	0.970997	0.926738	**0.893063**	0.863578
	seq	0.984091	0.945875	0.916111	0.891693	0.97732	0.969090	0.957613	0.950324	0.972220	0.939454	0.922348	0.886884
1TBWDHDD													
	80%rnd	0.969087	0.945811	0.879324	0.875919	0.948292	0.944900	0.909992	0.907181	0.957243	0.931762	0.889341	0.873772
	rnd	0.967672	0.934639	0.874346	0.873613	0.949542	0.936502	0.907860	0.907818	0.959230	0.934315	0.887870	0.875080
	seq	0.976532	0.972171	0.938088	0.932333	1.00000	0.956782	0.956655	0.937133	0.984040	0.969712	0.939495	0.921194
120GBSSD													
	80%rnd	0.949314	0,905724	0.818134	0.807204	0.990064	0.979037	0.926342	0.916437	0.977825	0.943714	0.872314	0.849240
	rnd	0.947593	0.895778	0.813897	0.806624	0.994225	0.972445	0.925801	0.918702	0.981559	0.937647	0.872200	0.852268
	seq	0.989643	0.950569	0.887680	0.872018	0.97131	0.992339	0.979253	0.964988	0.998072	0.977784	0.929965	0.902654

6.2 PERFORMANCE MODEL VALIDATION

This section presents the validation for the conceived GSPN models. Experiments were performed with real systems (Section 6.1) and compared to the values obtained with GSPN models (using stationary analysis). The single model is validated for HDD and SSD storages, and the multiple storage model is assessed utilizing a hybrid system composed of 1 HDD and 1 SSD, as the objective is to validate a baseline before expanding to more complex configurations. Small (4*KB*) objects (*os*) have been considered for both single and multiple storage models, and large

(1*MB*) objects only for the single storage model. Two access patterns (*pt*) are assumed: random (*rnd*) and sequential (*seq*). The validation also takes into account read and write operations (*op*).

The models utilize a delay of 1μs (following an exponential distribution) for transition *tRequesting*. For all GSPN models, the marking of place *pRequests* is 1, which denotes only one *worker*. The marking *Rx* (place $pResources_d$) is also 1, as storages (1*TBWDHDD* and 120*GBSSD*) with the serial advanced technology attachment (SATA) interface have been adopted. In that technology, a device does not carry out simultaneous operations. For the multiple storage model, marking *S* (*pStorages*) is 2, since the system controller manages 2 devices.

The delays for write and read operations have been approximated using phase-type distributions (Section 2.6.3). Table 6.5 details the results for the moment matching, considering data collected on the real system using Iometer. *mean* is the mean delay and *st.dev.* is the standard deviation. *distribution* denotes the probability distribution, and hypoexponential (*hypo*) has been adopted due to the algorithm described in Section 2.6.3. *phases* represents the number of phases. A limit of 10 phases has been adopted, since additional phases do not influence the results (BOLCH et al., 2006). Table 6.6 shows the mean power of each drive (1*TBWDHDD* and 120*GBSSD*) for distinct workload features.

Table 6.5: Moment matching - HDD and SSD.

			1TBWDHDD				120GBSSD			
op	**os**	**pt**	**mean (ms)**	**st.dev. (ms)**	**phases**	**distribution**	**mean (ms)**	**st.dev. (ms)**	**phases**	**distribution**
write	4KB	rnd	3.510000	0.950510	10	hypo.	0.968840	0.778870	1	hypo.
		seq	0.072336	0.024602	8	hypo.	0.223670	0.142030	2	hypo.
	1MB	rnd	9.920000	3.820000	6	hypo.	29.950000	17.410000	2	hypo.
		seq	5.690000	0.131970	10	hypo.	9.930000	4.540000	4	hypo.
read	4KB	rnd	8.000000	0.839160	10	hypo.	0.612730	0.056047	10	hypo.
		seq	0.047958	0.019129	6	hypo.	0.210620	0.040588	10	hypo.
	1MB	rnd	14.190000	0.276980	10	hypo.	4.470000	0.234440	10	hypo.
		seq	5.580000	0.107310	10	hypo.	4.010000	0.022829	10	hypo.

Table 6.6: Mean power values.

			power (W)	
op	**os**	**pt**	**1TBHDD**	**120GBSSD**
write	4KB	rnd	0.0102094	0.0008643
		seq	0.0003109	0.0001321
	1MB	rnd	0.0411532	0.0328646
		seq	0.0229215	0.0075362
read	4KB	rnd	0.0252030	0.0004737
		seq	0.0001598	0.0001919
	1MB	rnd	0.0455828	0.0086180
		seq	0.0183449	0.0081872

A hypothesis testing has been adopted to assess whether the proposed model may

adequately represent a storage system. In this case, the null hypothesis states that the difference between the model estimate and the system mean value is not statistically significant. A 95% confidence interval (for each considered workload feature) has been computed to determine whether the null hypothesis may be rejected or not (MONTGOMERY; RUNGER, 2014).

Table 6.7 depicts the values for real systems and the estimates using the single storage model. The metrics are energy consumption, response time and IOPS^{-1}. For all metrics, the model values are contained in the 95% confidence intervals (95% *c.i.*) obtained from the systems and, thus, the hypothesis of equivalence cannot be refuted. Similarly, Table 6.8 provides results for hybrid systems and multiple storage models. As the confidence intervals do contain the model estimates, the hypothesis of equivalence cannot be discarded.

Table 6.7: Validation results - single storage model.

				energy consumption (J)		**response time (ms)**		**IOPS^{-1}**	
device	**op**	**os**	**pt**	**95% c.i.**	**GSPN**	**95% c.i.**	**GSPN**	**95% c.i.**	**GSPN**
1TBHDD	write	4KB	rnd	(2.8450; 2.9597)	2.8909	(3.4862; 3.5218)	3.5190	(0.003400; 0.003523)	0.003520
			seq	(3.5052; 4.3209)	3.9166	(0.0808; 0.0823)	0.0813	(0.000081; 0.000083)	0.000082
		1MB	rnd	(3.5870; 4.6392)	4.0975	(9.8138; 9.9631)	9.9290	(0.009818; 0.009966)	0.009930
			seq	(3.6820; 4.3836)	4.0297	(5.6867; 5.7031)	5.6989	(0.005689; 0.005705)	0.005699
	read	4KB	rnd	(2.9513; 3.1926)	3.0812	(8.0046; 8.0584)	8.0090	(0.008007; 0.008061)	0.008010
			seq	(2.8750; 2.9931)	2.9007	(0.0561; 0.0574)	0.0569	(0.000056; 0.000057)	0.000057
		1MB	rnd	(3.0870; 3.3231)	3.2061	(14.1660; 14.2321)	14.2000	(0.014171; 0.014238)	0.014200
			seq	(3.2260; 3.3567)	3.2888	(5.5831; 5.5922)	5.5900	(0.005585; 0.005594)	0.005590
120GBSSD	write	4KB	rnd	(0.8625; 0.9753)	0.8830	(0.7839; 1.0923)	0.9778	(0.000785; 0.001095)	0.000978
			seq	(0.7909; 0.8507)	0.8195	(0.1456; 0.1754)	0.1602	(0.000146; 0.000176)	0.000161
		1MB	rnd	(1.1791; 1.6549)	1.4096	(22.5770; 23.7726)	23.3125	(0.022593; 0.023791)	0.023313
			seq	(0.9168; 1.1199)	0.9811	(5.8073; 8.9683)	7.6800	(0.005813; 0.008986)	0.007681
	read	4KB	rnd	(0.7598; 0.7708)	0.7647	(0.6145; 0.6217)	0.6217	(0.000615; 0.000622)	0.000622
			seq	(0.8585; 0.8875)	0.8701	(0.2183; 0.2200)	0.2196	(0.000219; 0.000220)	0.000220
		1MB	rnd	(1.3416; 2.5265)	1.9236	(4.4292; 4.4795)	4.4790	(0.004430; 0.004480)	0.004480
			seq	(1.4217; 2.6731)	2.0366	(3.9933; 4.0016)	4.0009	(0.003994; 0.004002)	0.004020

Table 6.8: Validation results - multiple storage model.

		energy consumption (J)		**response time (ms)**		**IOPS^{-1}**	
op	**pt**	**95% c.i.**	**GSPN**	**95% c.i.**	**GSPN**	**95% c.i.**	**GSPN**
write	rnd	(3.6922; 6.4811)	5.4950	(1.1334; 1.3907)	1.3634	(0.0005670; 0.0006960)	0.0006827
	seq	(3.9363; 6.1721)	4.6159	(0.0959; 0.1872)	0.0978	(0.0000487; 0.0000520)	0.0000492
read	rnd	(3.5179; 4.6106)	4.0630	(1.1573; 1.1647)	1.1623	(0.0005791; 0.0005828)	0.0005811
	seq	(3.6575; 4.3266)	3.9854	(0.0871; 0.0907)	0.0874	(0.0000433; 0.0000435)	0.0000433

A model validation has also been conducted considering real-world workloads, more specifically, OLTP (*oltp*) and Varmail (*varmail*) (Section 4.2.3). However, since processing such workloads typically involves the use of multiple storages in data centers, only the multiple storage model (1HDD + 1SSD) is assessed in this context, as it is specifically designed for representing two or more storages. Particularly, simulation is adopted to avoid the state space size explosion.

Table 6.9 depicts the results for the real system (*r.s.*) and the multiple storage model (*GSPN*). The metrics are response time and IOPS. Similar to previous validation, a hypothesis testing is conducted to determine a significant difference between the model estimate and the real system results. Confidence intervals (95% *c.i.*) contain the model estimates, and, therefore, the hypothesis of equivalence cannot be discarded.

Table 6.9: Validation using Fio tool - multiple storage model.

workload	**system/model**	**response time (ms)**			**IOPS**		
		mean	**st.dev.**	**95% c.i.**	**mean**	**st.dev.**	**95% c.i.**
oltp	r.s.	0.007344	0.0027	(0.006131; 0.008556)	25781.0	7658.54	(22425.48; 29138.37)
	GSPN	0.007633	1.22E-5	(0.007631; 0.007633)	27154.0	34.27	(27151.16; 27157.25)
varmail	r.s.	0.025723	0.0040	(0.023957; 0.027489)	630.0	111.42	(581.43; 679.09)
	GSPN	0.027470	6.89E-5	(0.027447; 0.027484)	592.5	0.994	(592.11; 592.93)

6.3 EXPERIMENTAL RESULTS

This section presents experimental results to demonstrate the feasibility of the proposed modeling approach. Specifically, the performance and energy consumption of homogeneous and hybrid storages are analyzed utilizing the conceived models. First, a screening experiment investigates the factors and interactions . Subsequently, four experiments evaluate the behavior of storage systems under industry-based benchmarks. Next, a case study demonstrates cost planning for different storage configurations. Finally, the scalability of the proposed models is discussed.

6.3.1 Experiment I: screening

This experiment evaluates the effects of each factor and their interactions, taking into account a DoE based on factorial design. Effect is the change in response due to a change in the factor level. Five factors ($k = 5$) are taken into account: (i) storage technology ($technology$); (ii) object size ($object\_size$); (iii) operation type ($operation$); (iv) access pattern ($pattern$); and (v) number of threads ($workers$). Table 6.10 depicts the levels (l_i) for each factor, and the metrics of interest are response time, IOPS and energy consumption.

Table 6.11 shows a rank for main and second-order interactions. The rank is ordered in descending order taking into account the absolute values of all effects. This work considers only main effects and second-order interactions, since high-order interactions do not considerably impact the adopted metrics (MONTGOMERY; RUNGER, 2014). Besides, the nine most

Table 6.10: Screening - factors and levels.

factor	levels
technology	1TBWDHDD, 120GBSSD, Hybrid
object_size	4KB, 1MB
operation	write, read
workers	1, 4
pattern	random (*rnd*), sequential (*seq*)

significant effects are illustrated, as other effects do not remarkably affect the metrics. For *technology* factor, the adopted levels for estimating an effect are indicated in parenthesis (e.g., $technology(120GBSSD - Hybrid)$).

Considering energy consumption, $technology$, $object\_size$, $operation$ and respective interactions (e.g., $operation * technology(120GBSSD - Hybrid)$) are the most significant effects. Nevertheless, the adoption of a hybrid system (i.e., $technolog\ y(120GBSSD - Hybrid)$ and $technology(1TBWDHDD - Hybrid)$) considerably contribute to energy consumption (change of $5.6919J$ and $3.6281J$).

The main effects account for most of the impact on response time, and interactions do not significantly influence this metric. $object\_size$ and *workers* are the factors with considerable variation: $20.1874ms$ and $16.2618ms$, respectively. *Hybrid* is the best level for $technology$, since it reduces response time in 3.4529ms and 2.8260ms compared to $120GBSSD$ and $1TBWDHDD$.

Regarding IOPS, $object\_size$ has the greatest influence followed by $technology$. *Hybrid* level considerably improves IOPS, as it may increase throughput by more than 280 operations per second. $pattern$ also influences system throughput: *rnd* - 158.3573 and *seq* - 376.9530. Besides, some interactions also have an effect on IOPS, for instance, $object\_size * technology(120GBSSD - Hybrid)$ (270.4418), and $operation * technology(120GBSSD - Hybrid)$ (188.1111).

Table 6.11: Rank of main and interaction effects.

energy consumption (J)		response time (ms)		IOPS	
factor/interaction	**effect**	**factor/interaction**	**effect**	**factor/interaction**	**effect**
technology(120GBSSD-Hybrid)	5.6919	*object_size*	20.1874	*object_size*	721.7838
technology(1TBHDD-Hybrid)	3.6281	*workers*	16.2618	*technology(1TBHDD-Hybrid)*	285.0174
*operation*technology(120GBSSD-Hybrid)*	2.1027	*pattern*	9.7670	*technology(120GBSSD-Hybrid)*	281.6786
*object_size*technology(1TBHDD-Hybrid)*	2.0719	*operation*	7.2928	*object_size*technology(120GBSSD-Hybrid)*	270.4418
technology(1TBHDD-120GBSSD)	2.0637	*technology(120GBSSD-Hybrid)*	3.4529	*pattern*	218.5956
object_size	2.0182	*technology(1TBHDD-Hybrid)*	2.8260	*operation*technology(120GBSSD-Hybrid)*	188.1111
*object_size*technology(120GBSSD-Hybrid)*	1.8911	*technology(1TBHDD-120GBSSD)*	0.6269	*operation*technology(1TBHDD-120GBSSD)*	146.8808
*object_size*operation*	1.2833	*object_size*technology(1TBHDD-120GBSSD)*	0.0060	*pattern*technology(1TBHDD-Hybrid)*	144.0635
operation	1.0491	*operation*technology(1TBHDD-Hybrid)*	0.0040	*object_size*operation*	136.4913

Results show the factors do not similarly influence all metrics (i.e., have the same rank position). Thus, for the next experiments, some factors levels have been fixed and mixed to better

assess their effects on storage systems. This approach also aims to represent workloads found in data centers. Furthermore, real-workloads are usually composed of concurrent requests (MONTAZERI et al., 2018); henceforth, *workers* is fixed on four to represent simultaneous clients. This value is a prominent balance for assessing concurrent requests without affecting model evaluation time (MEI et al., 2018; LIN et al., 2017).

6.3.2 Experiment II: random accesses

This section presents results for storages considering a workload mainly composed of random requests (Table 6.12). This experiment contemplates the following factors and levels: (i) $technology$ - $1TBHDD$, $120GBSSD$ and $Hybrid$; (ii) $object\_size$ - $4KB$; (iii) $operation$ - $70\%\_w$; (iv) $pattern$ - rnd; and (v) $workers$ - 4.

Table 6.12: Experimental results.

experiment	*technology*	energy consumption (J)	response time (ms)	IOPS
random accesses	SSD	0.850	3.162	1264.690
	HDD	2.937	17.248	231.897
	Hybrid	6.179	8.636	926
sequential access	SSD	2.055	36.901	108.396
	HDD	3.580	21.299	187.793
	Hybrid	10.734	14.173	564.599
read operations	SSD	1.719	18.494	216.281
	HDD	2.887	18.565	215.450
	Hybrid	8.464	17.927	446.447
mixed	SSD	1.571	10.862	368.226
	HDD	3.089	19.711	202.922
	Hybrid	6.639	11.746	681.211

Results indicate $120GBSSD$ as the best technology regarding all metrics due to the absence of mechanical components. The performance of magnetic disks is jeopardized because of excessive disk rotations. Compared to $1TBWDHDD$, $Hybrid$ has better values for response time and IOPS, but hybrid system consumes more energy. Considering the ratios IOPS/energy consumption and price/IOPS (Figures 6.4 and 6.5), SSD has better results followed by hybrid system.

Usually, SSDs are known for their remarkable performance for read operations (WAN et al., 2017). Additionally, this experiment corroborates the ability of SSDs to handle random requests, even under a workload consisting mostly of write requests (70%) (MEI et al., 2018).

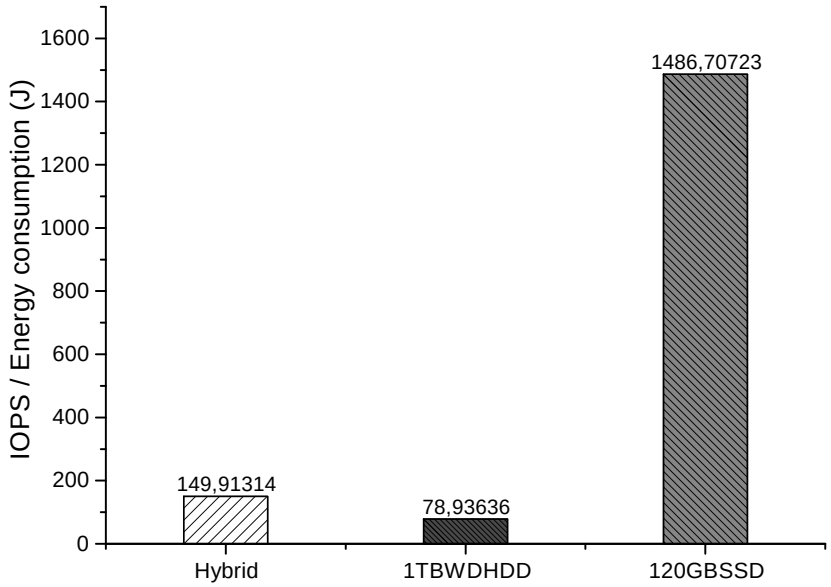


Figure 6.4: *Random accesses* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022).

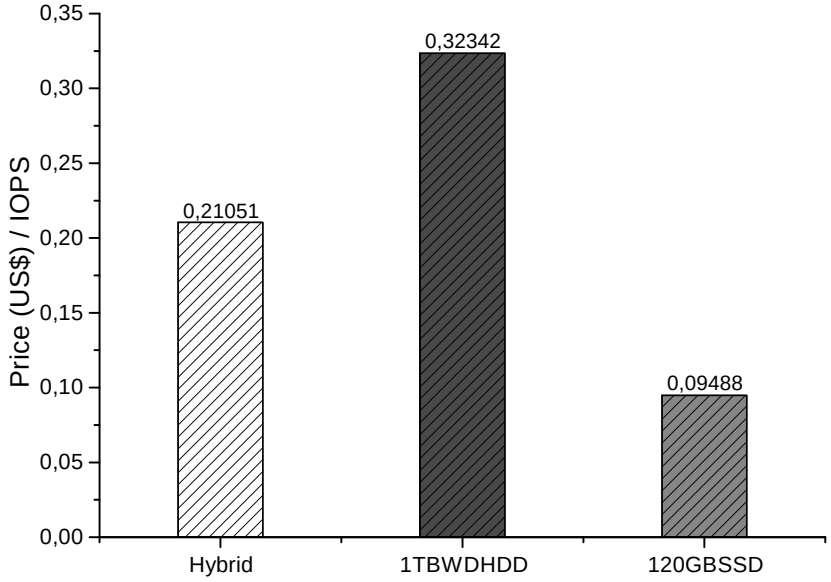


Figure 6.5: *Random access* - price/IOPS (BORBA; TAVARES; MACIEL, 2022).

6.3.3 Experiment III: sequential accesses

This section takes into account a workload represented by sequential requests. The experiment considers the following levels: (i) *technology* - $1TBHDD$, $120GBSSD$ and *Hybrid*; (ii) $object\_size$ - $1MB$; (iii) *operation* - $50\%\_w$; (iv) *pattern* - *seq*; and (v) *workers* - 4.

Similar to the previous experiment, the hybrid system has the worst values for energy consumption (Table 6.12). However, this system is capable of reducing response time (33.45%) and increasing IOPS (200.64%), compared to $1TBWDHDD$ (commonly considered the most

suitable technology for sequential workloads (LIN et al., 2017)). Results highlight the improvement obtained with *Hybrid* for large objects. Except for energy consumption, 120*GBSSD* has not presented significant results.

Figure 6.6 depicts 120*GBSSD* does not have a prominent IOPS/energy ratio, compared to other technologies. Regarding the ratio price/IO, Figure 6.7 indicates SSD has the highest cost.

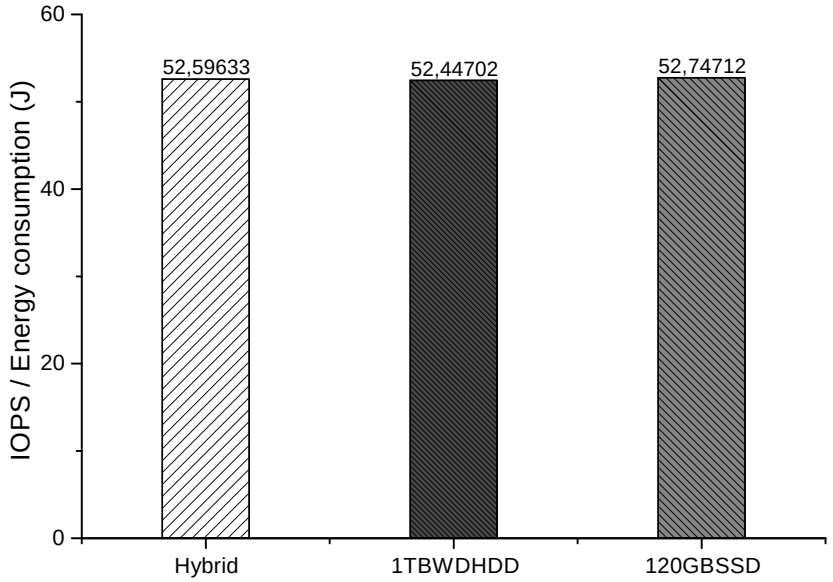


Figure 6.6: *Sequential accesses* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022).

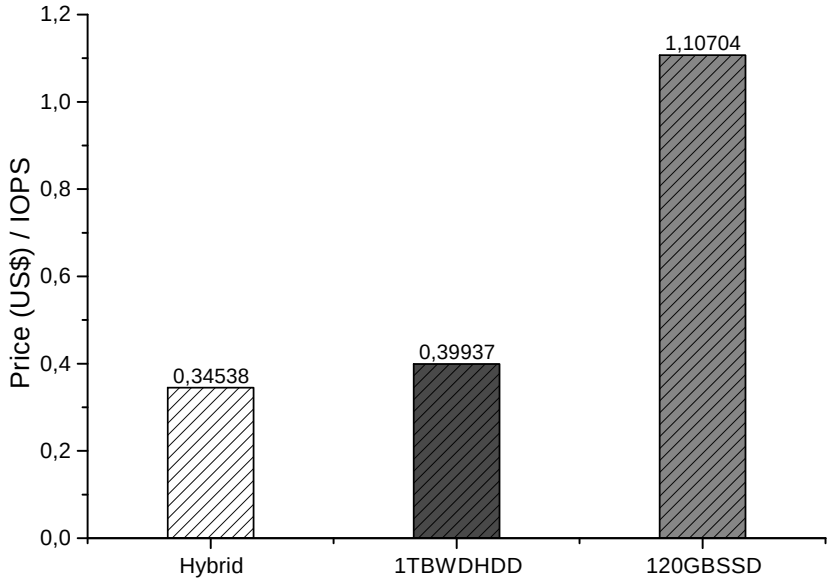


Figure 6.7: *Sequential access* - price/IOPS (BORBA; TAVARES; MACIEL, 2022).

6.3.4 Experiment IV: read operations

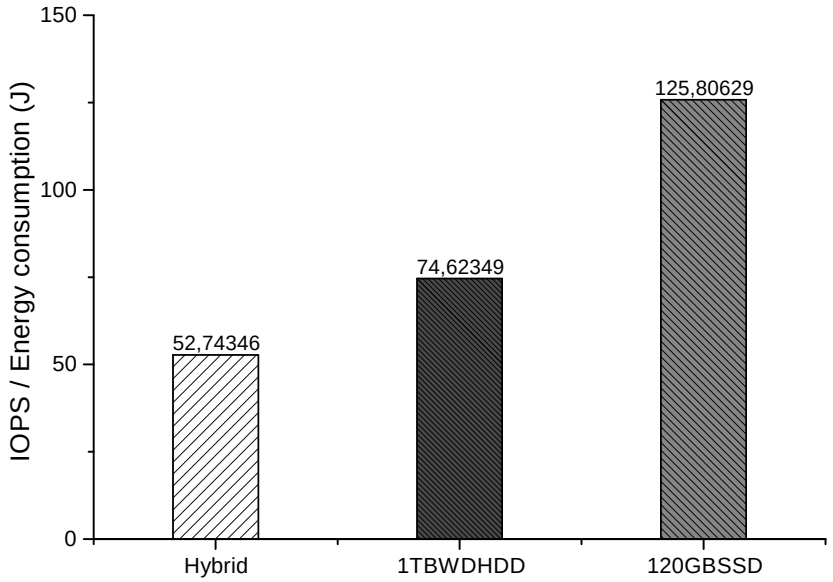


Figure 6.8: *Read operations* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022).

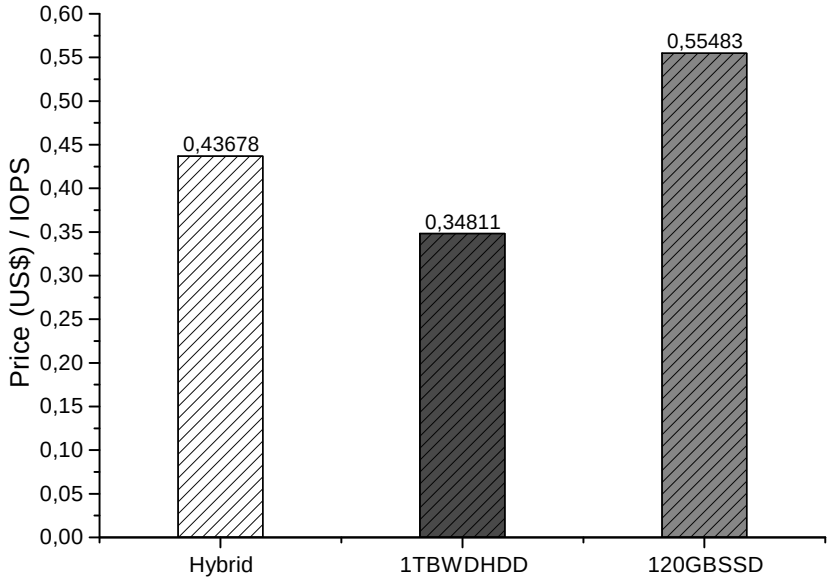


Figure 6.9: *Read operations* - price/IOPS (BORBA; TAVARES; MACIEL, 2022).

Table 6.12 depicts the results for a workload mainly composed of read operations. The following levels are assumed for this experiment: (i) *technology* - $1TBHDD$, $120GBSSD$ and *Hybrid*; (ii) $object\_size$ - $1MB$; (iii) $operation$ - $99\%\_r$; (iv) *pattern* - *seq*; and (v) *workers* - 4.

$120GBSSD$ and $1TBWDHDD$ have closer results concerning performance, but energy in magnetic disks is much higher (67.94%) than in solid-state memories. *Hybrid* has the best values for response time and IOPS, but energy consumption is still an issue for this system. As a

consequence, IOPS/energy is the lowest (Figure 6.8), and price/IOPS ratio is slightly better than a system based only on SSD (Figure 6.9).

For this workload, results indicate *Hybrid* may not have a prominent balance regarding cost and energy consumption, even having the best values for IOPS and response time.

6.3.5 Experiment V: mixed

Table 6.12 depicts the results for a workload composed of mixed operations, access patterns and distinct object sizes. This experiment takes into account the following levels: (i) *technology* - $1TBHDD$, $120GBSSD$ and *Hybrid*; (ii) $object\_size$ - $20\%\_los$; (iii) *operation* - $50\%\_w$; (iv) *pattern* - $80\%\_rnd$; and (v) *workers* - 4.

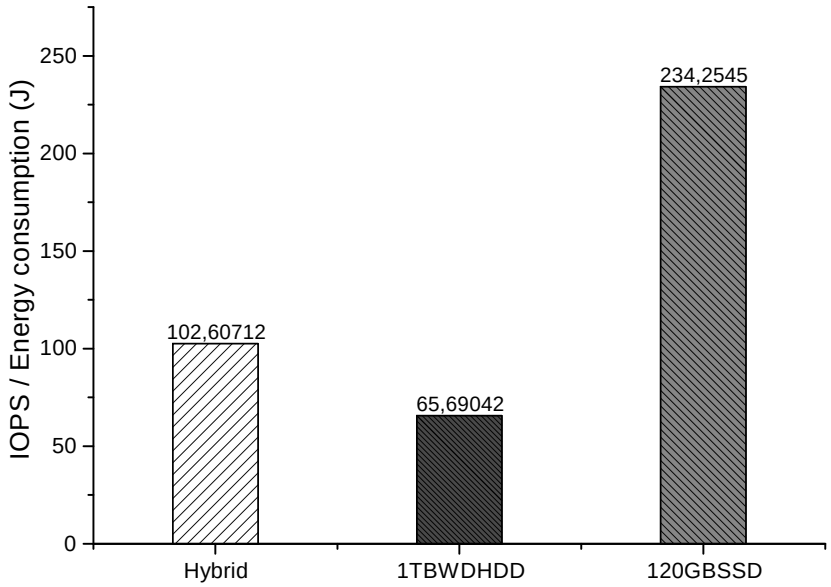


Figure 6.10: *Mixed* - IOPS/energy consumption (BORBA; TAVARES; MACIEL, 2022).

$120GBSSD$ has the smallest value for response time ($10.86ms$), influenced by small random requests ($4KB$ and *rnd*). *Hybrid* has the highest IOPS (681.211) and $1TBWDHDD$ has the worst performance, except for energy consumption.

Figure 6.10 indicates $120GBSSD$ has the best energy efficiency, about 128.30% higher than *Hybrid*. However, Figure 6.11 shows *Hybrid* has the best price-performance. Indeed, for the hybrid system, the high values for energy consumption and price are strongly compensated by system throughput.

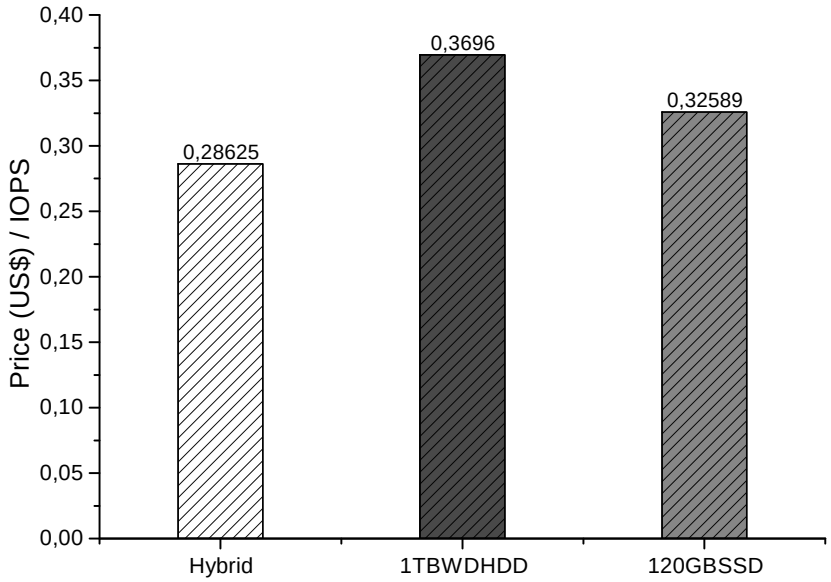


Figure 6.11: *Mixed* - price/IOPS (BORBA; TAVARES; MACIEL, 2022).

6.3.6 Case study

This section presents a case study to illustrate the practical feasibility of the conceived models for assessing storage systems in cloud computing environments. This work adopts a composite desirability technique to obtain the best configuration for storage systems using the adopted metrics (response time and IOPS).

Table 6.13: Case study - chosen configurations.

technology	**configuration**	**capacity (GB)**	**price (US$)**
SSD_{hom}	5 SSDs	600	102.00
HDD_{hom}	2 HDDs	2048	81.92
$Hybrid_1$	2 HDDs + 1 SSD	2168	102.32
$Hybrid_2$	1 HDD + 3 SSDs	1384	102.16

Table 6.13 depicts 4 storage systems that meet the price constraint (US$105.0), which consider homogeneous (HDD_{hom} and SSD_{hom}) and hybrid storage systems ($Hybrid_1$ and $Hybrid_2$). Table 6.14 details results for each system. For better visualization, *technology* is ordered in descending order taking into account the composite desirability values (CD).

For *database applications* (mostly composed of random operations), SSD_{hom} and $Hybrid_2$ have closer results concerning CD (1.0 and 0.946, respectively), but IOPS in the former (5952.38) is much higher than in the latter (3676.47). These results confirm the ability of SSDs to better handle random requests.

Concerning *data mining* application, both hybrid systems achieve significant results for

IOPS and response time. However, $Hybrid_1$ is the best configuration, as it has a composite desirability value equal to 0.5, whereas $Hybrid_2$ is equal to 0.495. For $Hybrid_1$, IOPS and response time are 929.281 and 12.931ms, respectively. The results corroborate the observations in Section 6.3.4, in which the hybrid storage system (*Hybrid*) is indicated as the most suitable technology.

Table 6.14: Case study results summary - composite desirability.

application	*technology*	IOPS	response time (ms)	CD
database systems	SSD_{hom}	5952.380	3.355	1.000
	$Hybrid_2$	3676.470	4.364	0.946
	$Hybrid_1$	1044.932	11.489	0.574
	HDD_{hom}	760.282	10.521	0.529
data mining	$Hybrid_1$	929.281	12.913	0.500
	$Hybrid_2$	1149.425	13.918	0.495
	SSD_{hom}	1082.251	18.483	0.223
	HDD_{hom}	431.127	18.555	0.066
raw data	$Hybrid_2$	1980.198	8.070	0.834
	SSD_{hom}	2044.989	9.773	0.796
	$Hybrid_1$	902.934	13.290	0.612
	HDD_{hom}	406.058	19.701	0.235

Assuming raw data workloads, $Hybrid_2$ presents the highest desirability value (0.834). Considering this system, 1980.198 and 8.070ms are the values for IOPS and response time, respectively. However, $Hybrid_2$ is slightly better than a system based only on SSD. SSD_{hom} achieves significant results for performance (IOPS - 2044.989; response time - 9.773ms) and, thus, may also be considered as a prominent solution.

The results indicate important insights, since all adopted configurations meet the price constraints. Capacity has not been considered for computing the composition desirability, as the focus has been on performance for this case study. However, a designer may consider other metrics and attributes for jointly assessing storage systems using the proposed models (e.g., IOPS/byte and ms/byte).

6.3.7 Scalability

This section presents a scalability experiment concerning the conceived models. The size of state space (i.e., CTMC size) and evaluation time are demonstrated considering the increase of storage components and workers.

The experiment adopts the multiple storage model and raw data workload utilized in Section 6.3.5. For evaluation, Mercury tool is utilized due to the scripting language that facilitates

Table 6.15: Scalability evaluation - storages.

storages	size	time (min)
2	9816	1.7
4	58463	44.0
8	210300	720.0
16	-	0.2
32	-	0.21

Table 6.16: Scalability evaluation - workers.

workers	size	time (min)
4	9816	1.7
8	17532	2.4
16	32964	9.0
32	63828	24.0
64	125556	61.0
128	249012	158.0
1000	-	0.28

model creation. Tables 6.15 and 6.16 depict the results, in which: *storages* is the number of storage devices; *workers* is the amount of concurrent clients; *size* is the CTMC size; and *time* is the time spent for numerical evaluation or simulation.

Considering additional storage components and workers, the state space considerably increases and, for some models, the tool was not able to generate the respective CTMC in a feasible period. In these cases, a simulation has been utilized. Results indicate the addition of storage components has a greater impact on state space than increasing the number of workers.

It is important to explain that the state space of the proposed models is finite (and so the respective CTMC), as the models are structurally bounded (MURATA, 1989). Additionally, the results may change (e.g., CTMC generation) using other evaluation tools.

6.4 HDDs AND SSDs FAILURES ANALYSIS

This section presents an explanatory analysis of storage failures using two representative industry datasets. This investigation aims to understand the impact that workloads may have on SSDs and HDDs and promote insights regarding the utilization of such devices. Such information is fundamental, as extracted statistics can be utilized for assessing storage systems using the proposed dependability models.

For instance, the annual failure rate (AFR) calculated in this analysis is determined using Equation 6.1, where F represents the number of failed storages, T represents the total number

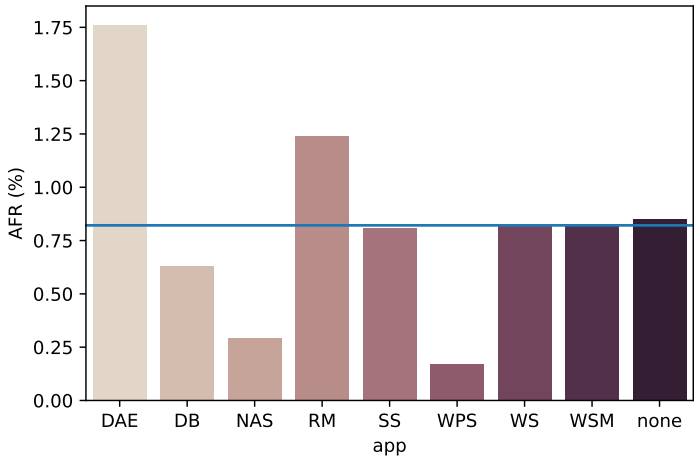


Figure 6.12: SSD annual failure rate per application (own work (2023)).

of storages, and M indicates the operational duration in months. Based on this statistic, $MTTF$ (mean time to failure) can be expressed as a function of the number of hours in a year (H_{year}) and AFR, as shown in Equation 6.2. The resulting $MTTF$ can then be utilized as an attribute in the availability model of the respective modeled storage device, allowing for estimation of the overall availability and mean time to failure of the storage system.

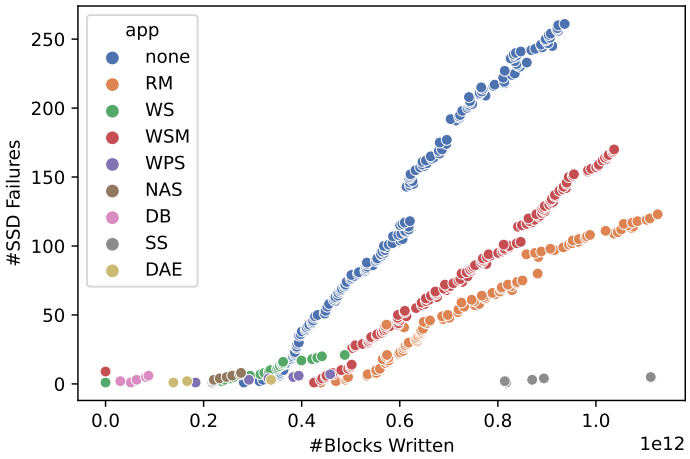
$$AFR = \frac{F}{T} \times \frac{12}{M} \tag{6.1}$$

$$MTTF = \frac{H_{year}}{AFR} \tag{6.2}$$

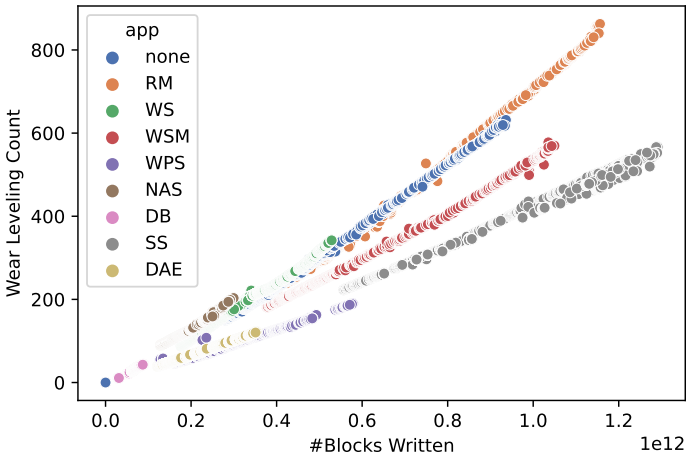
6.4.1 SSD analysis

Figure 6.12 depicts the annual failure rate of solid-state drives (SSDs) concerning different applications (a detailed application overview can be found in Appendix A). The horizontal blue line indicates an annual failure rate of nearly 0.8%, which is the average calculated from all SSD failures for all applications. Two applications, "*DAE*" and "*RM*," had the highest annual failure rate values, which suggests that these applications have a considerable effect on SSD wear. In contrast, the remaining applications have values approximately equal to or below the average and, thus, are not the most harmful to SSDs.

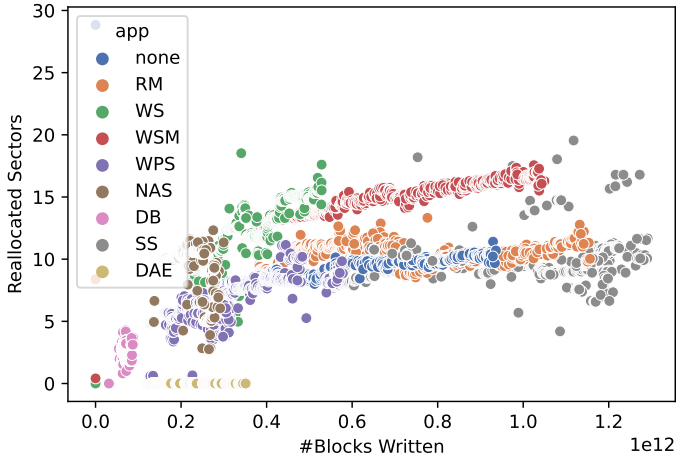
Figure 6.13 shows a graphical analysis of solid-state drives' failure rate and wear level over the number of blocks written considering different applications. More specifically, Fig-



(a) Failed SSDs x #Blocks written.



(b) Wear leveling x #Blocks written.



(c) Reallocated sectors x #Blocks written.

Figure 6.13: Attributes and SSD failures over the number of written blocks (own work (2023)).

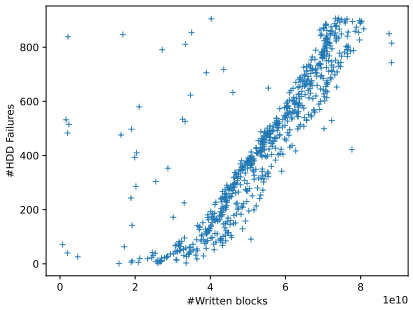
ure 6.13(a) depicts the failure rate of SSDs, while the other two subfigures (Figures 6.13(b) and 6.13(c)) show the device wear level through two SMART parameters (wear leveling and reallocated sector counts). As can be seen, the impact of applications with the same number of blocks written on the SSDs is considerably different among the three analyses. Moreover, this investigation reveals a similar steady increase in the SSD failure rate with a higher number of written blocks, mainly for three specific applications: *WSM*, *RM*, and *none*. By comparing these results with those illustrated in Figure 6.12, it is possible to visually identify the same applications that substantially influence the SSD annual failure rate, which strengthens the hypothesis that distinct workloads are responsible for different wearing levels. As for *DAE*, despite this application demonstrating a high annual failure rate (also observed in Figure 6.12), the same was not observed when analyzing the evolution of the number of SSD failures according to the number of blocks written. During the investigation, it was found that most of the devices submitted to the *DAE* application lack a record of the number of blocks written; therefore, any statistics involving this specific application may be inconsistent and, thus, unreliable.

This analysis provides essential information regarding SSDs subjected to different applications. This suggests that specific applications might be more prone to induce SSD failures than others and that this risk should be considered when selecting and adopting such devices. In addition, these findings are essential to estimate failure statics considering storage utilization, which can be adopted for the conceived dependability models and to study optimized data placement solutions.

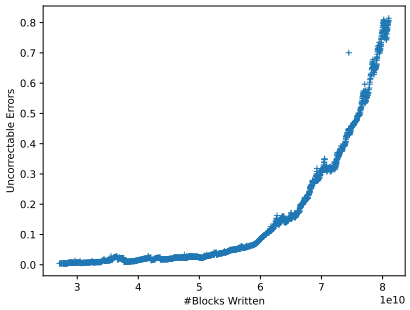
6.4.2 HDD analysis

Figure 6.14 depicts the results of an exploratory analysis of HDD failures and the evolution of their SMART attributes concerning the number of daily written blocks. The analysis specifically focuses on HDDs sourced from Backblaze data centers. Additional information can be found in Appendix B. The HDD model chosen for the exploratory analysis is highlighted in Table 4.2.

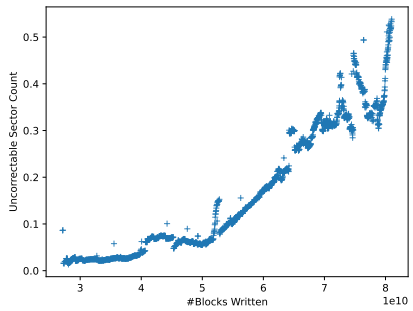
Figures 6.14(b) – 6.14(e) show that the evolution of HDDs SMART attributes representing storage wear over the number of blocks written follows a similar pattern as depicted in Figure 6.14(a). This suggests that the HDD failure rate may also be modeled as a function of the number of written blocks, considering the workload type (similar to the insights obtained from the previous SSD analysis). Furthermore, as illustrated in Figures 6.14(g) and 6.14(h),



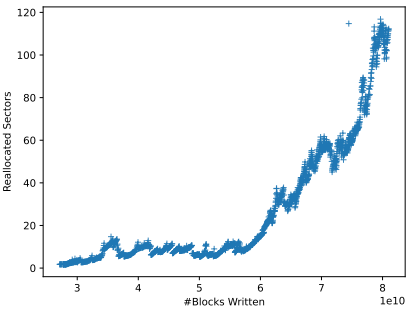
(a) Failed HDDs x #Blocks written.



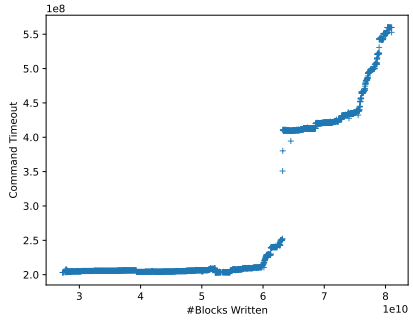
(b) Uncorrectable errors x #Blocks written.



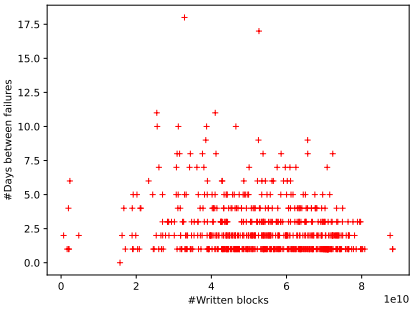
(c) Uncorrectable sectors errors x #Blocks written.



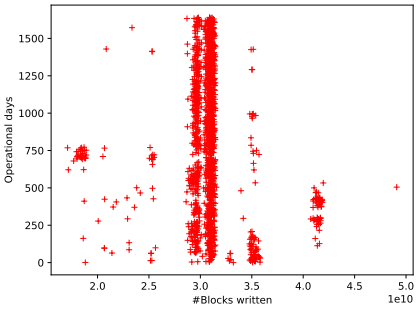
(d) Reallocated sectors x #Blocks written.



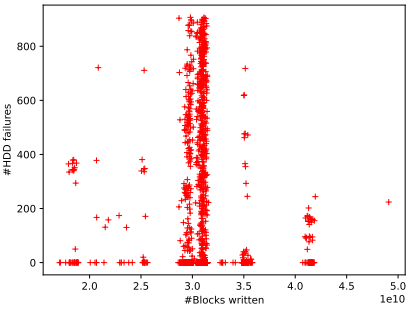
(e) Command timeout x #Blocks written.



(f) Days between failures x #Blocks written.



(g) Operational days x #Blocks written.



(h) #HDD failures per day x #Blocks written.

Figure 6.14: Attributes and HDD failures over number of blocks written (own work (2023)).

the number of blocks written per day when drive failures occur is nearly the same, indicating a daily backup, which means that the information extracted from this dataset is mainly related to a specific application. Therefore, although the dataset does not distinguish or mention which applications are being used, it is still possible to use statistics extracted from it in the context of workload-driven failures because the dataset includes at least one application. In addition, Figure 6.14(f) depicts the relationship between the number of days to HDD failure as blocks are written. As can be seen, the time between failures steadily decreases (resulting in more failures) as the number of written blocks increases. This outcome strengthens the evidence that statistics related to time to failure over the number of blocks written is a reasonable method to predict storage failures arising from the application put into practice.

These results provide valuable insights into the aspects contributing to HDD failures and highlight the importance of investigating multiple factors when modeling failure rates in storage systems. Furthermore, the computed failure rates may be adopted as attributes for the proposed dependability models.

6.5 SUMMARY

This chapter presented experimental results that demonstrated the feasibility of the proposed analytical models. Initially, a descriptive analysis based on the results of the measurement experiment for storage performance and energy consumption was presented. Subsequently, the validation of the designed GSPN models was explained, and for this purpose, the composition of the adopted workloads, storage devices, and initial definitions for the analytical models was detailed. Then, the chapter presented the results of the application of the moment matching technique, as well as the validation of the GSPN models. For experiments using these models, a DoE technique was adopted, and a factorial design was assumed. A rank was then created based on the results of the screening experiment to assess the most impactful factors and levels for storage devices. Next, experiments were conducted considering distinct workloads, and the results allowed a comparison of the performance and energy consumption between SSDs, HDDs, and Hybrid technologies. A case study was presented to demonstrate the usability of the models for applications, such as complaints of SLAs (in this case, cost planning). Next, a scalability experiment assessed the size of the space state and evaluation time of such models when increasing the number of storage components and workers. Finally, an exploratory analysis of the data extracted from the datasets of Alibaba and Backblaze companies was shown.

7

CONCLUSION

The infrastructure for cloud computing services demands high-throughput and low response-time access to meet the requirements defined in SLAs (LI et al., 2019). Nowadays, real-time data processing and access are required for many transactions. These transactions typically consist of small random files and require high reliability. Therefore, several studies have been conducted to improve the availability and performance of storage devices, which are often bottlenecks in computing systems that operate services with intense requests (YIN et al., 2018a).

Several techniques, such as data management across multiple disks, caching, write buffering, prefetching, request scheduling, and parallel I/O, have been employed in traditional storage devices to reduce these adversities (PARK; LEE; KIM, 2017). However, despite these improvements, hard disk devices do not perform sufficiently well to fully meet the current demands of cloud computing platforms, and they account for 78% of equipment replacements in data centers (HUANG et al., 2019). Replacing HDDs with SSDs is one possible alternative, as SSDs exhibit better performance and lower power consumption (ELYASI et al., 2018; YIN et al., 2018a). However, the high cost per gigabyte and low durability (compared to HDDs) still make it unfeasible to use SSDs exclusively in data centers. Consequently, data centers oversize their architecture (which leads to additional costs) to meet the availability of contracted services stipulated in service level agreements (NARAYANAN et al., 2016).

As an alternative, research on hybrid storage systems has attracted increasing attention from industry and academia (WU et al., 2018; BOUKHELEF et al., 2019). Several architectures have been proposed to create a storage system capable of taking advantage of both storage technologies (e.g., the higher reliability of HDDs, and lower energy consumption of SSDs).

Many studies have failed to concurrently and comprehensively evaluate multiple aspects. For instance, it is crucial to consider the influence of storage reliability on both the performance and energy consumption of storage systems. However, most related studies did not evaluate several aspects concomitantly; for example, they did not consider the impact of storage reliability has on the performance and energy consumption of storage systems.

This thesis proposed a stochastic model-based approach for analyzing the performance, availability, and energy consumption of homogeneous and hybrid data storage systems. The performance models are based on stochastic Petri nets, which enable the assessment of different storage architectures under varying workloads. In addition, the proposed RBD and GSPN dependability models estimate the availability of the data storage systems and their impact on performance. To validate the proposed approach, the performance models were evaluated using benchmark tool (Iometer and Fiotool) results and voltage measurements obtained using an oscilloscope in the selected environment. The experiments involved analyzing both traditional technologies (HDDs or SSDs only) and a hybrid storage system consisting of one HDD and one SSD. The evaluation considered the most significant factors for assessing storage devices and followed industry standards as benchmarks.

The experiments confirmed the practical feasibility of the modeling approach and provided meaningful evaluations for storage system designers. For instance, hybrid systems typically consume more energy than conventional systems. However, whenever performance requirements prevail over energy savings, hybrid storage is a prominent alternative, primarily for sequential accesses and raw workloads. SSDs may exhibit performance issues with sequential accesses, however, they are suitable for services with small random read operations. Concerning HDDs, results confirm the issues associated with the processing of small objects. Nevertheless, HDDs are still a feasible option for some systems represented by sequential accesses, owing to the favorable IOPS/energy and price/IOPS ratios. Although the experiments confirm the practical feasibility of the proposed approach, the results may not be generalizable to all types of data storage systems. The following sections describe the principal contributions of the proposed method and outline future works.

7.1 CONTRIBUTIONS

This thesis introduced a novel stochastic model-based approach for evaluating the performance, availability, and energy consumption of data storage systems. This approach can

facilitate decision-making regarding the selection and configuration of storage systems for various workloads and enable the comparison of different architectures. The specific contributions os this study are as follows:

- **Measurement.** An experiment was conducted using the Iometer tool and oscilloscopes to obtain performance and power values from real storage devices. Statistical techniques were used to conduct an exploratory analysis of the collected values. The results of the performance measurements and the estimated power values can be helpful for future research on magnetic and solid-state storage devices;

- **Performance and energy consumption models.** Analytical models based on a state space mathematical formalism (generalized stochastic Petri nets) that can be use to represent read and write operations. Furthermore, workloads are characterized according to their pattern (random or sequential) and object size (small or large). These models allow the estimation of throughput, average response time, and energy consumption of homogeneous and hybrid storage systems;

- **Factor and interaction impacts.** A screening characterization study to estimate the effects of the main factors and interactions in homogeneous and hybrid storage systems on performance and energy consumption. It is important to note that there have been no reports of similar analyses in past work. Based on this study, it is possible to identify and eliminate the least significant factors and interactions to allow for a more accurate, noise-free analysis in experiments that evaluate the throughput, average response time, and energy consumption of write and read requests;

- **Experimental results.** In this study, experiments have been performed using the conceived GSPN models for assessing different storage technologies. In addition, workload parameters have been defined using a characterization study (screening) and industry-standard benchmarks. The results obtained from these experiments provide a novel analysis for comparing hybrid systems with traditional devices. Specifically, neither the computed performance and energy results nor analysis investigating ratios IOPS/energy consumption and IOPS/price ratios are found in previous literature;

- **Storage failure analysis.** An exploratory analysis was performed on industry datasets from Alibaba and Backblaze to investigate workloads' impact on SSDs

and HDDs failures. Such investigations provide a means of understanding the effects of distinct workloads on these storage technologies;

- **Availability and performability models.** Two analytical models have been proposed based on the mathematical formalisms RBD and GSPN to evaluate the availability and performance of the storage devices. A hierarchical modeling approach is adopted for combining results from the proposed availability model into the conceived performability model. These models allow the simulation of real-world scenarios and the assessment of the impact of workloads on storage device failures and, consequently, performance;

- **Methodology to evaluate performance and energy consumption of data storage systems.** the methodology proposed in this thesis assists in making decisions regarding the storage systems in data centers. The adoption of the designed GSPN models, and the planning of experiments, allow the evaluation of different architectures and storage policies for homogeneous and hybrid systems, while still in the design phase. It is important to emphasize that using the designed models may demand a prior grounding regarding the formalism adopted;

- **Methodology to evaluate the dependability of data storage systems.** the suggested approach involves investigating in advance the effects on storage that a specific application can cause on the storage system to be designed. Thus, a more accurate dependability study can be conducted using the designed GSPN and RBD models. The methodology also includes an optimization step, in which the models must be refined for use cases such as SLA compliance, prevention of disasters, performability-sensitive data management, and bottleneck identification. It is essential to state that a GSPN and RBD practitioner may be required to adopt such a technique.

7.2 LIMITATIONS

The proposed stochastic models can be utilized to evaluate the performance and energy consumption evaluation of homogeneous and hybrid storage systems. However, an abstraction level has been adopted at which specific features are not directly represented. Therefore, this section summarizes some assumptions of the proposed technique.

The models do not distinguish between data types; thus, metadata manipulation is not explicit in the conceived models. Interference due to data management mechanisms (garbage collection and wear leveling) are not the focus of this work. However, their respective impacts on the mean delays for write and read operations have been considered.

Regarding the workload, this study investigates only two access patterns (random and sequential) and two object sizes (small and large). Concerning energy consumption, the assessment of storage devices has focused on the active energy state. Nonetheless, the influence of idle and standby states on the mean delays for power values has also been considered.

As presented in Section 6.3.7, although the state space of the proposed models is finite, the explicit representation of many storage components considerably increases the size of the state space (CTMC). However, the proposed approach allows for the abstraction of several components in a storage submodel several components. Besides, simulation may be utilized as an alternative to CTMC generation.

7.3 PUBLICATIONS

This section shows the articles written and published during this study. All the articles presented here are related to the research conducted in this thesis and are listed below:

7.3.1 Journals

- **Borba, Eric**; TAVARES, EDUARDO. Stochastic modeling for performance and availability evaluation of hybrid storage systems. JOURNAL OF SYSTEMS AND SOFTWARE, 2017;

- **Borba, Eric**; TAVARES, EDUARDO; Maciel, Paulo. A modeling approach for estimating performance and energy consumption of storage systems. JOURNAL OF COMPUTER AND SYSTEM SCIENCES, 2022.

7.3.2 Conferences

- **Borba, Eric**; PONTES, JONAS; TAVARES, EDUARDO. Performance and availability modeling of hybrid storage systems. In: 2017 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2017, Banff;

- **Borba, Eric**; TAVARES, EDUARDO; Maciel, Paulo; LIRA, VICTOR; ARAUJO, CARLOS GOMES. Performance and Energy Consumption Evaluation of Hybrid Storage Systems. In: 2020 IEEE International Systems Conference (SysCon), 2020, Montreal;

- **Borba, Eric**; Tavares, Eduardo; Maciel, Paulo; Gomes, Carlos. Analytical models for performance and energy consumption evaluation of storage devices. In: 36th International Conference on Massive Storage Systems and Technology (MSST 2020), 2020.

7.4 FUTURE WORKS

This thesis has investigated various aspects of data storage systems, such as their performance, dependability, and energy consumption. Despite the comprehensive analysis provided, ample scope remains for further improvements and extensions to the present work. The following points outline some potential avenues for such enhancements:

- **Failure sources:** the study conducted in this thesis has investigated storage failures by considering the impacts different applications may have on performance. However, other failures can be considered and extended to the designed models. Examples include human error, silent errors, and silent data corruption (XU et al., 2019). The latter two refer to errors that occur when a device sends corrupted data to the host without any signaling. Fault correction mechanisms are also a potential extensions of these models. Thus, different error correction code (ECC) techniques can be proposed and evaluated;

- **Data management solutions:** the models proposed in this thesis allow for the evaluation of data management solutions (i.e., defining which storage device will process a given request) for a given data storage system architecture. Therefore, optimization techniques other than those proposed in this paper can be used to identify novel storage architectures and policies that have not yet been investigated. For example, the greedy randomized adaptive search procedure (GRASP), is a metaheuristic characterized by providing a good quality solution within a finite set of elements, according to a defined maximum number of interactions (FEO; RESENDE, 1995);

- **Flash management modeling:** Flash memories have a garbage collection mechanism that can impact the data storage device in two ways: increased response time (due to blocking for request execution) and wear on the flash memory chips (due to excessive deletions) (MCEWAN; KOMSUL, 2018). Wear leveling is another essential mechanism in the operation of solid-state devices. However, to equalize the number of programs/deletions along the memory blocks, wear leveling can cause premature wear on flash memory (CHIKHAOUI; BOUKHALFA; BOUKHOBZA, 2018). An extension of the models proposed in this thesis could provide a means for system designers to estimate the impacts of different solutions on such mechanisms.

References

AGRAWAL, N.; PRABHAKARAN, V.; WOBBER, T.; DAVIS, J. D.; MANASSE, M. S.; PANIGRAHY, R. Design Tradeoffs for SSD Performance. In: **USENIX Annual Technical Conference**. [S.l.: s.n.], 2008. p.57–70.

AJMONE MARSAN, M.; CONTE, G.; BALBO, G. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. **ACM Transactions on Computer Systems (TOCS)**, [S.l.], v.2, n.2, p.93–122, 1984.

AJMONE MARSAN, M.; CONTE, G.; BALBO, G. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. **ACM Transactions on Computer Systems (TOCS)**, [S.l.], v.2, n.2, p.93–122, 1984.

AL MAMUN, A.; GUO, G.; BI, C. **Hard disk drive**: mechatronics and control. [S.l.]: CRC press, 2006. v.23.

APPUSWAMY, R.; MOOLENBROEK, D. C. van; TANENBAUM, A. S. Integrating flash-based SSDs into the storage stack. In: **Mass Storage Systems and Technologies (MSST), 2012 IEEE 28th Symposium on**. [S.l.: s.n.], 2012. p.1–12.

ARSHAD, U.; ALEEM, M.; SRIVASTAVA, G.; LIN, J. C.-W. Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers. **Renewable and Sustainable Energy Reviews**, [S.l.], v.167, p.112782, 2022.

AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B. Fundamental concepts of dependability. **Department of Computing Science Technical Report Series**, [S.l.], 2001.

AVIZIENIS, A.; LAPRIE, J.-C.; RANDELL, B. et al. **Fundamental concepts of dependability**. [S.l.]: University of Newcastle upon Tyne, Computing Science, 2001.

AXBOE, J. **Flexible I/O Tester Synthetic Benchmark**. Acessed: 2021/08/15, Available at: `https://github.com/axboe/fio`.

BAHN, H.; CHO, K. Implications of NVM Based Storage on Memory Subsystem Management. **Appl. Sci.**, [S.l.], v.10, n.3, p.999, 2020.

BALBO, G. Introduction to stochastic Petri nets. In: **Lectures on Formal Methods and PerformanceAnalysis**. [S.l.]: Springer, 2001. p.84–155.

BAUSE, F.; KRITZINGER, P. S. **Stochastic Petri Nets**. [S.l.]: Vieweg Wiesbaden, 2002. v.1.

BERISHA, B.; MËZIU, E.; SHABANI, I. Big data analytics in Cloud computing: an overview. **Journal of Cloud Computing**, [S.l.], v.11, n.1, p.24, 2022.

BERNARDI, S.; MERSEGUER, J.; PETRIU, D. C. Dependability modeling and analysis of software systems specified with UML. **ACM Computing Surveys (CSUR)**, [S.l.], v.45, n.1, p.2, 2012.

BHARANY, S.; SHARMA, S.; KHALAF, O. I.; ABDULSAHIB, G. M.; AL HUMAIMEEDY, A. S.; ALDHYANI, T. H.; MAASHI, M.; ALKAHTANI, H. A systematic survey on energy-efficient techniques in sustainable cloud computing. **Sustainability**, [S.l.], v.14, n.10, p.6256, 2022.

BOLCH, G.; GREINER, S.; MEER, H. de; TRIVEDI, K. S. **Queueing networks and Markov chains**: modeling and performance evaluation with computer science applications. [S.l.]: John Wiley & Sons, 2006.

BORBA, E.; TAVARES, E. Stochastic modeling for performance and availability evaluation of hybrid storage systems. **Journal of Systems and Software**, [S.l.], v.134, p.1–11, 2017.

BORBA, E.; TAVARES, E.; MACIEL, P. A modeling approach for estimating performance and energy consumption of storage systems. **Journal of Computer and System Sciences**, [S.l.], v.128, p.86–106, 2022.

BORBA, E.; TAVARES, E.; MACIEL, P.; LIRA, V.; ARAÚJO, C. G. Performance´ and energy consumption evaluation of hybrid storage systems. In: IEEE INTERNATIONAL SYSTEMS CONFERENCE (SYSCON), 2020. **Anais...** [S.l.: s.n.], 2020. p.1–6.

BOUKHELEF, D.; BOUKHALFA, K.; BOUKHOBZA, J.; OUARNOUGHI, H.; LEMARCHAND, L. COPS: cost based object placement strategies on hybrid storage system for dbaas cloud. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON CLUSTER, CLOUD AND GRID COMPUTING (CCGRID), 2017. **Anais...** [S.l.: s.n.], 2017. p.659–664.

BOUKHELEF, D.; BOUKHOBZA, J.; BOUKHALFA, K.; OUARNOUGHI, H.; LEMARCHAND, L. Optimizing the cost of DBaaS object placement in hybrid storage systems. **Future Generation Computer Systems**, [S.l.], v.93, p.176–187, 2019.

BREWER, J.; GILL, M. **Nonvolatile Memory Technologies with Emphasis on Flash**: a comprehensive guide to understanding and using flash memory devices. [S.l.]: John Wiley & Sons, 2011. v.8.

BU, K.; WANG, M.; NIE, H.; HUANG, W.; LI, B. The optimization of the hierarchical storage system based on the hybrid ssd technology. In: **Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on**. [S.l.: s.n.], 2012. p.1323–1326.

CARNS, P.; HARMS, K.; ALLCOCK, W.; BACON, C.; LANG, S.; LATHAM, R.; ROSS, R. Understanding and improving computational science storage access through continuous characterization. **ACM Transactions on Storage (TOS)**, [S.l.], v.7, n.3, p.8, 2011.

CHAMAZCOTI et al. Hybrid RAID: a solution for enhancing the reliability of ssd-based raids. **IEEE Transactions on Multi-Scale Computing Systems**, [S.l.], v.3, n.3, p.181–192, 2017.

CHEN, F.; DING, X.; JIANG, S. Exploiting disk layout and block access history for i/o prefetch. **Advanced Operating Systems and Kernel Applications: Techniques and Technologies: Techniques and Technologies**, [S.l.], p.201, 2009.

CHEN, F.; KOUFATY, D. A.; ZHANG, X. Hystor: making the best use of solid state drives in high performance storage systems. In: **Proceedings of the international conference on Supercomputing**. [S.l.: s.n.], 2011. p.22–32.

CHIKHAOUI, A.; BOUKHALFA, K.; BOUKHOBZA, J. A Cost Model for Hybrid Storage Systems in a Cloud Federations. In: FEDERATED CONFERENCE ON COMPUTER SCIENCE AND INFORMATION SYSTEMS (FEDCSIS), 2018. **Anais...** [S.l.: s.n.], 2018. p.1025–1034.

CLOUD, G. **Official Site**. Acessed: 2020/10/04, Available at: `https://cloud.google.com/compute/disks-image-pricing#disk`.

COMMITTEE, S. F. F. **Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T.)**. Acessed: 2023/07/20, Available at: `https://www.linux-mips.org/pub/linux/mips/people/macro/S.M.A.R.T./SFF-8035i.pdf`.

COUNCIL, S. P. **Official Site**. Acessed: 2019/09/27, Available at: `http://www.storageperformance.org`.

DESROCHERS, A. A.; AL-JAAR, R. Y. **Applications of Petri nets in manufacturing systems**: modeling, control, and performance analysis. [S.l.]: IEEE, 1995.

EBELING, C. E. **An introduction to reliability and maintainability engineering**. [S.l.]: Tata McGraw-Hill Education, 2004.

EL MAGHRAOUI, K.; KANDIRAJU, G.; JANN, J.; PATTNAIK, P. Modeling and simulating flash based solid-state disks for operating systems. In: **Proceedings of the first joint WOSP/SIPEW international conference on Performance engineering**. [S.l.: s.n.], 2010. p.15–26.

ELYASI, N.; ARJOMAND, M.; SIVASUBRAMANIAM, A.; KANDEMIR, M. T.; DAS, C. R. Content popularity-based selective replication for read redirection in ssds. In: IEEE 26TH INTERNATIONAL SYMPOSIUM ON MODELING, ANALYSIS, AND SIMULATION OF COMPUTER AND TELECOMMUNICATION SYSTEMS (MASCOTS), 2018. **Anais...** [S.l.: s.n.], 2018. p.1–15.

FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. **Journal of global optimization**, [S.l.], v.6, n.2, p.109–133, 1995.

FRANCÊS, C. R. L. Introdução às redes de petri. **Laboratório de Computação Aplicada, Universidade Federal do Pará**, [S.l.], 2003.

FRANK, A.; YANG, D.; BRINKMANN, A.; SCHULZ, M.; SÜSS, T. Reducing false node failure predictions in HPC. In: IEEE 26TH INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING, DATA, AND ANALYTICS (HIPC), 2019. **Anais...** [S.l.: s.n.], 2019. p.323–332.

GRIDER, G.; NUNEZ, J.; BENT, J. **LANL MPI-IO Test**. 2008.

HAN, L.; SHEN, Z.; SHAO, Z.; LI, T. Optimizing RAID/SSD controllers with lifetime extension for flash-based SSD array. **ACM SIGPLAN Notices**, [S.l.], v.53, n.6, p.44–54, 2018.

HAN, S.; LEE, P. P.; XU, F.; LIU, Y.; HE, C.; LIU, J. An In-Depth Study of Correlated Failures in Production SSD-Based Data Centers. In: USENIX CONFERENCE ON FILE AND STORAGE TECHNOLOGIES (FAST 21), 19. **Anais...** [S.l.: s.n.], 2021. p.417–429.

HAVERKORT, B. R. Markovian models for performance and dependability evaluation. In: SCHOOL ORGANIZED BY THE EUROPEAN EDUCATIONAL FORUM. **Anais...** [S.l.: s.n.], 2000. p.38–83.

HSU, W. W.; SMITH, A. J. The performance impact of I/O optimizations and disk improvements. **IBM Journal of Research and Development**, [S.l.], v.48, n.2, p.255–289, 2004.

HUANG, S.; LIANG, S.; FU, S.; SHI, W.; TIWARI, D.; CHEN, H.-b. Characterizing disk health degradation and proactively protecting against disk failures for reliable storage systems. In: IEEE INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING (ICAC), 2019. **Anais...** [S.l.: s.n.], 2019. p.157–166.

HURSON, A. R. **ADVANCES IN COMPUTERS Green and Sustainable Computing**: part ii preface. [S.l.]: ELSEVIER, 2013.

JAIN, R. **The art of computer systems performance analysis**: techniques for experimental design, measurement, simulation, and modeling. [S.l.]: John Wiley & Sons, 1990.

JOO, Y.; RYU, J.; PARK, S.; SHIN, H.; SHIN, K. G. Rapid prototyping and evaluation of intelligence functions of active storage devices. **IEEE Transactions on Computers**, [S.l.], v.63, n.9, p.2356–2368, 2014.

KANOUN, K.; SPAINHOWER, L. **Dependability benchmarking for computer systems**. [S.l.]: John Wiley & Sons, 2008. v.72.

KAPUR, K. C.; PECHT, M. **Reliability engineering**. [S.l.]: John Wiley & Sons, 2014.

KATAL, A.; DAHIYA, S.; CHOUDHURY, T. Energy efficiency in cloud computing data centers: a survey on software technologies. **Cluster Computing**, [S.l.], v.26, n.3, p.1845–1875, 2023.

KHAZAEI, H.; MISIC, J.; MISIC, V. B. Performance analysis of cloud computing centers using m/g/m/m+ r queuing systems. **IEEE Transactions on parallel and distributed systems**, [S.l.], v.23, n.5, p.936–943, 2012.

KIM, H.-J.; KIM, J.-S. A user-space storage I/O framework for NVMe SSDs in mobile smart devices. **IEEE Trans. Consum. Electroni.**, [S.l.], v.63, n.1, p.28–35, 2017.

KIM, K.; KIM, S.; KIM, T. FAST I/O: qos supports for urgent i/os in nvme ssds. In: INTERNATIONAL CONFERENCE ON INTELLIGENT INFORMATION TECHNOLOGY, 2020. **Proceedings...** [S.l.: s.n.], 2020. p.146–151.

KIM, S.; EOM, H.; SON, Y. Improving spatial locality in virtual machine for flash storage. **IEEE Access**, [S.l.], v.7, p.1668–1676, 2019.

KISHANI, M.; AHMADIAN, S.; ASADI, H. A Modeling Framework for Reliability of Erasure Codes in SSD Arrays. **IEEE Trans. Comput.**, [S.l.], v.69, n.5, p.649–665, 2019.

KLEINROCK, L. Queueing systems. Volume I: theory. , [S.l.], 1975.

KUO, W.; ZUO, M. J. **Optimal reliability modeling**: principles and applications. [S.l.]: John Wiley & Sons, 2003.

LEE, C.; SIM, D.; HWANG, J.; CHO, S. F2FS: a new file system for flash storage. In: USENIX} CONFERENCE ON FILE AND STORAGE TECHNOLOGIES ({FAST} 15), 13. **Anais...** [S.l.: s.n.], 2015. p.273–286.

LEE, D.; MIN, C.; EOM, Y. I. Effective flash-based SSD caching for high performance home cloud server. **IEEE Transactions on Consumer Electronics**, [S.l.], v.61, n.2, p.215–221, 2015.

LEE, S.-W.; MOON, B.; PARK, C. Advances in flash memory SSD technology for enterprise database applications. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2009. **Proceedings...** [S.l.: s.n.], 2009. p.863–870.

LEVINE, D. D. Iometer user's guide. **Intel Server Architecture Lab**, [S.l.], v.40, 1998.

LI, C.; FENG, D.; HUA, Y.; WANG, F. A high-performance and endurable SSD cache for parity-based RAID. **Frontiers of Computer Science**, [S.l.], v.13, n.1, p.16–34, 2019.

LI, H.; ZHANG, Y.; LI, D.; ZHANG, Z.; LIU, S.; HUANG, P.; QIN, Z.; CHEN, K.; XIONG, Y. Ursa: hybrid block storage for cloud-scale virtual disks. In: FOURTEENTH EUROSYS CONFERENCE 2019. **Proceedings...** [S.l.: s.n.], 2019. p.1–17.

LI, Z.; CHEN, M.; MUKKER, A.; ZADOK, E. On the trade-offs among performance, energy, and endurance in a versatile hybrid drive. **ACM Transactions on Storage (TOS)**, [S.l.], v.11, n.3, p.1–27, 2015.

LILJA, D. J. **Measuring computer performance**: a practitioner's guide. [S.l.]: Cambridge university press, 2005.

LIN, M.; CHEN, R.; XIONG, J.; LI, X.; YAO, Z. Efficient sequential data migration scheme considering dying data for HDD/SSD hybrid storage systems. **IEEE Access**, [S.l.], v.5, p.23366–23373, 2017.

MACIEL, P.; LINS, R.; CUNHA, P. **Introdução às redes de Petri e aplicações**. [S.l.]: UNICAMP-Instituto de Computacao, 1996.

MACIEL, P.; TRIVEDI, K.; MATIAS, R.; KIM, D. Performance and dependability in service computing: concepts, techniques and research directions, ser. **Premier Reference Source. Igi Global**, [S.l.], 2011.

MANOGARAN, G.; THOTA, C.; LOPEZ, D. Human-computer interaction with big data analytics. In: **Research Anthology on Big Data Analytics, Architectures, and Applications**. [S.l.]: IGI global, 2022. p.1578–1596.

MAO, B.; JIANG, H.; WU, S.; TIAN, L.; FENG, D.; CHEN, J.; ZENG, L. HPDA: a hybrid parity-based disk array for enhanced performance and reliability. **ACM Transactions on Storage (TOS)**, [S.l.], v.8, n.1, p.4, 2012.

MAO, B.; WU, S.; JIANG, H. Improving storage availability in cloud-of-clouds with hybrid redundant data distribution. In: **Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International**. [S.l.: s.n.], 2015. p.633–642.

MARKOV, A. A. Extension of the law of large numbers to dependent quantities. **Izv. Fiz.-Matem. Obsch. Kazan Univ.(2nd Ser)**, [S.l.], v.15, p.135–156, 1906.

MARSAN, M. A.; BALBO, G.; CONTE, G.; DONATELLI, S.; FRANCESCHINIS, G. **Modelling with generalized stochastic Petri nets**. [S.l.]: John Wiley & Sons, Inc., 1994.

MCEWAN, A. A.; KOMSUL, M. Z. Age aware pre-emptive garbage collection for SSD RAID. **Microprocessors and Microsystems**, [S.l.], v.56, p.13–21, 2018.

MEI, L.; FENG, D.; ZENG, L.; CHEN, J.; LIU, J. A High-Performance and High-Reliability RAIS5 Storage Architecture with Adaptive Stripe. In: INTERNATIONAL CONFERENCE ON ALGORITHMS AND ARCHITECTURES FOR PARALLEL PROCESSING. **Anais...** [S.l.: s.n.], 2018. p.562–577.

MEI, L.; FENG, D.; ZENG, L.; CHEN, J.; LIU, J. Exploiting flash memory characteristics to improve performance of RAIS storage systems. **Frontiers of Computer Science**, [S.l.], v.13, n.5, p.913–928, 2019.

MEISTER, D.; BRINKMANN, A. dedupv1: improving deduplication throughput using solid state drives (ssd). In: **Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on**. [S.l.: s.n.], 2010. p.1–6.

MELO, A.; TAVARES, E. A. G.; SOUSA, E.; NOGUEIRA, B. C. e. S.; MARINHO, M. Dependability approach for evaluating software development risks. **IET Software**, [S.l.], v.9, n.1, p.17–27, 2015.

MERLIN, P.; FARBER, D. Recoverability of communication protocols–implications of a theoretical study. **IEEE transactions on Communications**, [S.l.], v.24, n.9, p.1036–1043, 1976.

MIAO, R.; ZHU, L.; MA, S.; QIAN, K.; ZHUANG, S.; LI, B.; CHENG, S.; GAO, J.; ZHUANG, Y.; ZHANG, P. et al. From luna to solar: the evolutions of the compute-to-storage networks in alibaba cloud. In: ACM SIGCOMM 2022 CONFERENCE. **Proceedings...** [S.l.: s.n.], 2022. p.753–766.

MICHELONI, R.; MARELLI, A.; ESHGHI, K. **Inside solid state drives (SSDs)**. [S.l.]: Springer Science & Business Media, 2012. v.37.

MODARRES, M.; KAMINSKIY, M. P.; KRIVTSOV, V. **Reliability engineering and risk analysis**: a practical guide. [S.l.]: CRC press, 2009.

MOLLOY, M. K. Performance analysis using stochastic Petri nets. **IEEE Transactions on computers**, [S.l.], v.31, n.9, p.913–917, 1982.

MONTAZERI, B.; LI, Y.; ALIZADEH, M.; OUSTERHOUT, J. Homa: a receiver-driven low-latency transport protocol using network priorities. In: CONFERENCE OF THE ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION, 2018. **Proceedings...** [S.l.: s.n.], 2018. p.221–235.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied statistics and probability for engineers**. [S.l.]: John Wiley & Sons, 2014.

MOTI, N.; SCHIMMELPFENNIG, F.; SALKHORDEH, R.; KLOPP, D.; CORTES, T.; RÜCKERT, U.; BRINKMANN, A. Simurgh: a fully decentralized and secure nvmm user space file system. In: INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS. **Proceedings...** [S.l.: s.n.], 2021. p.1–14.

MURATA, T. Petri nets: properties, analysis and applications. **Proceedings of the IEEE**, [S.l.], v.77, n.4, p.541–580, 1989.

MUSTAFA, S.; SATTAR, K.; SHUJA, J.; SARWAR, S.; MAQSOOD, T.; MADANI, S. A.; GUIZANI, S. Sla-aware best fit decreasing techniques for workload consolidation in clouds. **IEEE Access**, [S.l.], v.7, p.135256–135267, 2019.

NAKASHIMA, K.; KON, J.; YAMAGUCHI, S. I/o performance improvement of secure big data analyses with application support on ssd cache. In: INTERNATIONAL CONFERENCE ON UBIQUITOUS INFORMATION MANAGEMENT AND COMMUNICATION, 12. **Proceedings...** [S.l.: s.n.], 2018. p.1–7.

NAKASHIMA, K.; KON, J.; YAMAGUCHI, S.; LEE, G. J.; FORTES, J. 1A study on big data I/O performance with modern storage systems. In: IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA), 2017. **Anais...** [S.l.: s.n.], 2017. p.4798–4799.

NARAYANAN, I.; WANG, D.; JEON, M.; SHARMA, B.; CAULFIELD, L.; SIVASUBRAMANIAM, A.; CUTLER, B.; LIU, J.; KHESSIB, B.; VAID, K. SSD failures in datacenters: what? when? and why? In: ACM INTERNATIONAL ON SYSTEMS AND STORAGE CONFERENCE, 9. **Proceedings...** [S.l.: s.n.], 2016. p.1–11.

NIJIM, M.; NIJIM, Y.; SKER, R.; REDDY, V.; RAJU, R. N. DM-pas: a data mining prefetching algorithm for storage system. In: **High Performance Computing and Communications (HPCC), 2011 IEEE 13th International Conference on**. [S.l.: s.n.], 2011. p.500–505.

O'CONNOR, P. D.; O'CONNOR, P.; KLEYNER, A. **Practical reliability engineering**. [S.l.]: John Wiley & Sons, 2012.

OLIVEIRA, D.; BRINKMANN, A.; ROSA, N.; MACIEL, P. Performability evaluation and optimization of workflow applications in cloud environments. **Journal of Grid Computing**, [S.l.], v.17, p.749–770, 2019.

OLIVEIRA, D.; MATOS, R.; DANTAS, J.; FERREIRA, J.; SILVA, B.; CALLOU, G.; MACIEL, P.; BRINKMANN, A. Advanced stochastic petri net modeling with the mercury scripting language. In: EAI INTERNATIONAL CONFERENCE ON PERFORMANCE EVALUATION METHODOLOGIES AND TOOLS, 11. **Proceedings...** [S.l.: s.n.], 2017. p.192–197.

PALACIOS CHAVARRO, S.; NESPOLI, P.; DÍAZ-LÓPEZ, D.; NIÑO ROA, Y. On the Way to Automatic Exploitation of Vulnerabilities and Validation of Systems Security through Security Chaos Engineering. **Big Data and Cognitive Computing**, [S.l.], v.7, n.1, p.1, 2022.

PARK, J.; LEE, S.; KIM, J. DAC: dedup-assisted compression scheme for improving lifetime of nand storage systems. In: DESIGN, AUTOMATION & TEST IN EUROPE CONFERENCE & EXHIBITION (DATE), 2017. **Anais...** [S.l.: s.n.], 2017. p.1249–1252.

PARK, S.; KIM, Y.; URGAONKAR, B.; LEE, J.; SEO, E. A comprehensive study of energy efficiency and performance of flash-based SSD. **Journal of Systems Architecture**, [S.l.], v.57, n.4, p.354–365, 2011.

PETRI, C. A. Kommunikation mit automaten. , [S.l.], 1962.

RAUSAND, M.; ARNLJOT, H. et al. **System reliability theory**: models, statistical methods, and applications. [S.l.]: John Wiley & Sons, 2004. v.396.

REISIG, W. **Understanding petri nets**. [S.l.]: Springer, 2013.

RICHTER, D. **Flash memories**: economic principles of performance, cost and reliability optimization. [S.l.]: Springer Science & Business Media, 2013.

SALKHORDEH, R.; BRINKMANN, A. Online Management of Hybrid DRAM-NVMM Memory for HPC. In: IEEE 26TH INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE COMPUTING, DATA, AND ANALYTICS (HIPC), 2019. **Anais...** [S.l.: s.n.], 2019. p.277–289.

SALKHORDEH, R.; KREMER, K.; NAGEL, L.; MAISENBACHER, D.; HOLMBERG, H.; BJØRLING, M.; BRINKMANN, A. Constant time garbage collection in ssds. In: IEEE INTERNATIONAL CONFERENCE ON NETWORKING, ARCHITECTURE AND STORAGE (NAS), 2021. **Anais...** [S.l.: s.n.], 2021. p.1–9.

SAXENA, P.; KUMAR, P. Performance evaluation of HDD and SSD on 10GigE, IPoIB & RDMA-IB with Hadoop cluster performance benchmarking system. In: INTERNATIONAL CONFERENCE-CONFLUENCE THE NEXT GENERATION INFORMATION TECHNOLOGY SUMMIT (CONFLUENCE), 2014. **Anais...** [S.l.: s.n.], 2014. p.30–35.

SHI, X.; WU, F.; WANG, S.; XIE, C.; LU, Z. Program error rate-based wear leveling for NAND flash memory. In: DESIGN, AUTOMATION & TEST IN EUROPE CONFERENCE & EXHIBITION (DATE), 2018. **Anais...** [S.l.: s.n.], 2018. p.1241–1246.

SIAPOUSH, M. S.; JAMALI, S.; BADIRZADEH, A. Software-defined networking enabled big data tasks scheduling: a tabu search approach. **Journal of Communications and Networks**, [S.l.], v.25, n.1, p.111–120, 2023.

SILVA, B.; CALLOU, G.; TAVARES, E.; MACIEL, P.; FIGUEIREDO, J.; SOUSA, E.; ARAUJO, C.; MAGNANI, F.; NEVES, F. Astro: an integrated environment for dependability and sustainability evaluation. **Sustainable computing: informatics and systems**, [S.l.], v.3, n.1, p.1–17, 2013.

SINGHAL, S.; SHARMA, P.; AGGARWAL, R. K.; PASSRICHA, V. A global survey on data deduplication. **International Journal of Grid and High Performance Computing (IJGHPC)**, [S.l.], v.10, n.4, p.43–66, 2018.

STRUNK, J. D. Hybrid Aggregates: combining ssds and hdds in a single storage pool. **ACM SIGOPS Operating Systems Review**, [S.l.], v.46, n.3, p.50–56, 2012.

TRIVEDI, K. S. **Probability & statistics with reliability, queuing and computer science applications**. [S.l.]: John Wiley & Sons, 2008.

TUFFIN, B.; CHOUDHARY, P.; HIREL, C.; TRIVEDI, K. Simulation versus analytic-numeric methods: a petri net example. In: VALUETOOLS CONFERENCE, 2. **Proceedings...** [S.l.: s.n.], 2007.

TULI, S.; GILL, S. S.; XU, M.; GARRAGHAN, P.; BAHSOON, R.; DUSTDAR, S.; SAKELLARIOU, R.; RANA, O.; BUYYA, R.; CASALE, G. et al. HUNTER: ai based holistic resource management for sustainable cloud computing. **Journal of Systems and Software**, [S.l.], v.184, p.111124, 2022.

UML. **Official Site**. Accessed: 04/05/2023, Available at: `http://www.uml.org`.

VALMARI, A. The state explosion problem. **Lectures on Petri nets I: Basic models**, [S.l.], p.429–528, 1998.

VARKI, E.; MERCHANT, A.; XU, J.; QIU, X. Issues and challenges in the performance analysis of real disk arrays. **IEEE Transactions on Parallel and Distributed Systems**, [S.l.], v.15, n.6, p.559–574, 2004.

VEF, M.-A.; STEINER, R.; SALKHORDEH, R.; STEINKAMP, J.; VENNETIER, F.; SMIGIELSKI, J.-F.; BRINKMANN, A. DelveFS-An Event-Driven Semantic File System for Object Stores. In: IEEE INTERNATIONAL CONFERENCE ON CLUSTER COMPUTING (CLUSTER), 2020. **Anais...** [S.l.: s.n.], 2020. p.35–46.

VERSCHOREN, R.; VAN HOUDT, B. How to improve the performance of the d-choices garbage collection algorithm in flash-based SSDs. In: EAI INTERNATIONAL CONFERENCE ON PERFORMANCE EVALUATION METHODOLOGIES AND TOOLS, 13. **Proceedings...** [S.l.: s.n.], 2020. p.180–187.

WAN, J.; WU, W.; ZHAN, L.; YANG, Q.; QU, X.; XIE, C. DEFT-Cache: a cost-effective and highly reliable ssd cache for raid storage. In: IEEE INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS), 2017. **Anais...** [S.l.: s.n.], 2017. p.102–111.

WANG, S.; LU, Z.; CAO, Q.; JIANG, H.; YAO, J.; DONG, Y.; YANG, P.; XIE, C. Exploration and Exploitation for Buffer-Controlled HDD-Writes for SSD-HDD Hybrid Storage Server. **ACM Transactions on Storage (TOS)**, [S.l.], v.18, n.1, p.1–29, 2022.

WANG, S. X.; TARATORIN, A. M. **Magnetic Information Storage Technology**: a volume in the electromagnetism series. [S.l.]: Academic press, 1999.

WANG, Y.; JIANG, S.; HE, L.; PENG, Y.; CHOW, T. W. Hard Disk Drives Failure Detection Using A Dynamic Tracking Method. In: IEEE 17TH INTERNATIONAL CONFERENCE ON INDUSTRIAL INFORMATICS (INDIN), 2019. **Anais...** [S.l.: s.n.], 2019. v.1, p.1473–1477.

WOO, Y.-J.; KIM, J.-S. Diversifying wear index for MLC NAND flash memory to extend the lifetime of SSDs. In: **Embedded Software (EMSOFT), 2013 Proceedings of the International Conference on**. [S.l.: s.n.], 2013. p.1–10.

WU, C.-H.; HUANG, C.-W.; CHANG, C.-Y. A data management method for databases using hybrid storage systems. **ACM SIGAPP Appl. Comput. Rev.**, [S.l.], v.19, n.1, p.34–47, 2019.

WU, J.; WANG, Y.; WANG, J.; WANG, H.; LIN, T. How does solid-state drives cluster perform for distributed file systems: an empirical study. **Concurrency and Computation: Practice and Experience**, [S.l.], p.e7709, 2023.

WU, S.; CHEN, G.; CHEN, K.; LI, F.; SHOU, L. HM: a column-oriented mapreduce system on hybrid storage. **IEEE Transactions on Knowledge and Data Engineering**, [S.l.], v.27, n.12, p.3304–3317, 2015.

WU, S.; MAO, B.; CHEN, X.; JIANG, H. LDM: log disk mirroring with improved performance and reliability for ssd-based disk arrays. **ACM Trans. Storage**, [S.l.], v.12, n.4, p.1–21, 2016.

WU, W.; LIN, W.; HSU, C.-H.; HE, L. Energy-efficient hadoop for big data analytics and computing: a systematic review and research insights. **Future Gener. Comput. Syst.**, [S.l.], v.86, p.1351–1367, 2018.

WU, W.; XIA, W.; YU, Z.; LIU, Q. Exploring the potential of coupled array of SSD and HDD for multi-tenant. In: IEEE 3RD INTERNATIONAL CONFERENCE ON CLOUD COMPUTING AND BIG DATA ANALYSIS (ICCCBDA), 2018. **Anais...** [S.l.: s.n.], 2018. p.653–657.

WU, Z.; LI, Y.; LEE, P. P.; XU, Y. Modeling SSD RAID reliability under general settings. In: ACM INTERNATIONAL CONFERENCE ON COMPUTING FRONTIERS, 15. **Proceedings...** [S.l.: s.n.], 2018. p.155–164.

XIE, M.; XIA, L.; XU, J. On M/G [b]/1/K queue with multiple state-dependent vacations: a real problem from media-based cache in hard disk drives. **Performance Evaluation**, [S.l.], v.139, p.102085, 2020.

XIE, X.; YANG, T.; LI, Q.; WEI, D.; XIAO, L. Duchy: achieving both ssd durability and controllable smr cleaning overhead in hybrid storage systems. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, 47. **Proceedings...** [S.l.: s.n.], 2018. p.1–9.

XU, E.; ZHENG, M.; QIN, F.; XU, Y.; WU, J. Lessons and Actions: what we learned from 10k ssd-related storage system failures. In: USENIX ANNUAL TECHNICAL CONFERENCE (USENIX ATC 19), 2019. **Anais...** [S.l.: s.n.], 2019. p.961–976.

XU, F.; HAN, S.; LEE, P. P.; LIU, Y.; HE, C.; LIU, J. General feature selection for failure prediction in large-scale SSD deployment. In: ANNUAL IEEE/IFIP INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS (DSN), 2021. **Anais...** [S.l.: s.n.], 2021. p.263–270.

YANG, J.; REN, Z.; WANG, J.; LI, L. Determining the Sampling Size with Maintaining the Probability Distribution. In: THEORETICAL COMPUTER SCIENCE: 40TH NATIONAL CONFERENCE, NCTCS 2022, CHANGCHUN, CHINA, JULY 29–31, 2022, REVISED SELECTED PAPERS. **Anais...** [S.l.: s.n.], 2022. p.61–74.

YANG, Y.; CAO, Q.; YAO, J.; JIANG, H.; YANG, L. Batch-file Operations to Optimize Massive Files Accessing: analysis, design, and application. **ACM Transactions on Storage (TOS)**, [S.l.], v.16, n.3, p.1–25, 2020.

YANG, Y.; ZHU, J. Algebraic modeling of write amplification in hotness-aware SSD. In: ACM INTERNATIONAL SYSTEMS AND STORAGE CONFERENCE, 8. **Proceedings...** [S.l.: s.n.], 2015. p.1–11.

YIN, S.; JIAO, B.; ZHU, X.; RUAN, X.; CHEN, S.; TANG, Z. DuoFS: a hybrid storage system balancing energy-efficiency, reliability, and performance. In: EUROMICRO INTERNATIONAL CONFERENCE ON PARALLEL, DISTRIBUTED AND NETWORK-BASED PROCESSING (PDP), 2018. **Anais...** [S.l.: s.n.], 2018. p.478–485.

YIN, S.; JIAO, B.; ZHU, X.; RUAN, X.; CHEN, S.; TANG, Z. DuoFS: a hybrid storage system balancing energy-efficiency, reliability, and performance. In: EUROMICRO INTERNATIONAL CONFERENCE ON PARALLEL, DISTRIBUTED AND NETWORK-BASED PROCESSING (PDP), 2018. **Anais...** [S.l.: s.n.], 2018. p.478–485.

YIN, S.; XIAO, Z.; LI, K.; HUANG, J.; RUAN, X.; ZHU, X.; QIN, X. RESS: a reliable energy-efficient storage system. In: IEEE 22ND INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED SYSTEMS (ICPADS), 2016. **Anais...** [S.l.: s.n.], 2016. p.1193–1198.

YU, X.; ZHANG, C.; LIANG, C.; KHALED, A.; ZHENG, J.; ZHANG, Q. et al. A high-performance hierarchical snapshot scheme for hybrid storage systems. **Chinese Journal of Electronics**, [S.l.], v.27, n.1, p.76–85, 2018.

ZENG, J.; DING, D.; KANG, K.; XIE, H.; YIN, Q. Adaptive DRL-based virtual machine consolidation in energy-efficient cloud data center. **IEEE Transactions on Parallel and Distributed Systems**, [S.l.], v.33, n.11, p.2991–3002, 2022.

ZHANG, J.; FENG, D.; LIU, J.; FANG, C.; LIU, C.; ZHANG, Z. WB-RAIS: white-box redundant array of independent ssds. **Wireless Personal Communications**, [S.l.], v.102, n.4, p.2807–2821, 2018.

ZHANG, J.; ZHOU, K.; HUANG, P.; HE, X.; XIAO, Z.; CHENG, B.; JI, Y.; WANG, Y. Transfer Learning based Failure Prediction for Minority Disks in Large Data Centers of Heterogeneous Disk Systems. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, 48. **Proceedings...** [S.l.: s.n.], 2019. p.1–10.

ZHANG, J.; ZHOU, K.; HUANG, P.; HE, X.; XIAO, Z.; CHENG, B.; JI, Y.; WANG, Y. Transfer learning based failure prediction for minority disks in large data centers of heterogeneous disk systems. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING, 48. **Proceedings...** [S.l.: s.n.], 2019. p.1–10.

ZHANG, J.; ZHOU, K.; HUANG, P.; HE, X.; XIE, M.; CHENG, B.; JI, Y.; WANG, Y. Minority disk failure prediction based on transfer learning in large data centers of heterogeneous disk systems. **IEEE Transactions on Parallel and Distributed Systems**, [S.l.], v.31, n.9, p.2155–2169, 2020.

ZHAO, D.; ZHOU, J. An energy and carbon-aware algorithm for renewable energy usage maximization in distributed cloud data centers. **Journal of Parallel and Distributed Computing**, [S.l.], v.165, p.156–166, 2022.

ZHOU, G.; ZHOU, J.; HUAI, X.; ZHOU, F.; JIANG, Y. A two-phase liquid immersion cooling strategy utilizing vapor chamber heat spreader for data center servers. **Applied Thermal Engineering**, [S.l.], v.210, p.118289, 2022.

ZIMMERMANN, A.; KNOKE, M.; HUCK, A.; HOMMEL, G. Towards version 4.0 of TimeNET. In: **Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2006 13th GI/ITG Conference**. [S.l.: s.n.], 2006. p.1–4.

ZUBEREK, W. M. Timed Petri nets and preliminary performance evaluation. In: **Proceedings of the 7th annual symposium on Computer Architecture**. [S.l.: s.n.], 1980. p.88–96.

Appendix

A

SSD dataset - applications overview

Table A.1: SSD dataset - applications overview.

app	#total	#failed	%write	%read
DAE	16000	1214	81.80	19.20
DB	26781	203	26.71	73.29
NAS	14454	541	62.13	37.87
RM	183981	3016	76.11	23.89
SS	32936	184	42.09	57.91
WPS	44676	529	30.69	69.31
WS	17740	232	97.09	2.91
WSM	380170	8916	79.05	20.95
none	248757	3552	69.20	30.80

B

HDD dataset - SMART attributes overview

Table B.1: HDD failures and attributes overview (✳ means that a specific attribute has been found on logs from the respective HDD model).

model	op_time (days)	#total	#failed	r_sectors	u_errors	b_written	command_timeout	pending_sector	u_sector	AFR (%)
HMS5C4040BLE640	1641	15514	217	✳	-	-	-	✳	✳	0.30
HUH721212ALN604	1449	10973	164	✳	-	-	-	✳	✳	0.37
HUH728080ALE600	1641	1197	30	✳	-	-	-	✳	✳	0.54
ST12000NM0007	1641	38822	1981	✳	✳	✳	✳	✳	✳	1.11
ST12000NM0008	993	20607	557	✳	✳	✳	✳	✳	✳	0.97
ST14000NM001G	654	10880	151	✳	✳	✳	✳	✳	✳	0.76
ST14000NM0138	576	1689	116	✳	✳	✳	✳	✳	✳	4.29
ST16000NM001G	993	18181	118	✳	✳	✳	✳	✳	✳	0.23
ST4000DM000	1641	32186	1864	✳	✳	✳	✳	✳	✳	1.27
ST8000NM0055	1641	15290	907	✳	✳	✳	✳	✳	✳	**1.30**
MD04ABA400V	1641	146	3	✳	-	-	-	✳	✳	0.45
MG07ACA14TA	1484	38831	551	✳	-	-	-	✳	✳	0.34
WUH721414ALE6L4	631	8446	37	✳	-	-	-	✳	✳	0.24

C

Example of the performability model execution

The following example shows the important steps in the execution of the performability model proposed in this thesis. Initially, the model's behavior is presented when representing the creation and forwarding of requests to the respective storage node. Next, the request processing is illustrated. The failure of a storage node and its impact on request processing is also approached. Finally, the repair of the storage node in question is presented. For a better understanding, the transitions that are enabled are highlighted in yellow. Inhibitory arcs that prevent the activation of a transition are highlighted in blue.

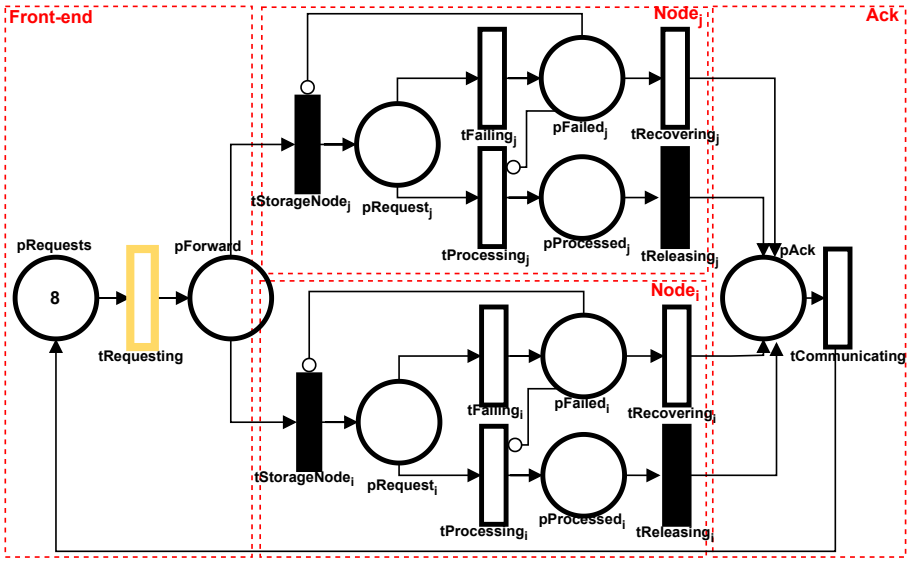


Figure C.1: Performability model execution - initial marking (own work (2023)).

Figure C.1 shows the initial marking of the model. In this example, the number of concurrent requests that can be generated is represented by eight tokens in place *pRequests*. As shown, transition *trequesting* is enabled to fire according to the delay assigned to it.

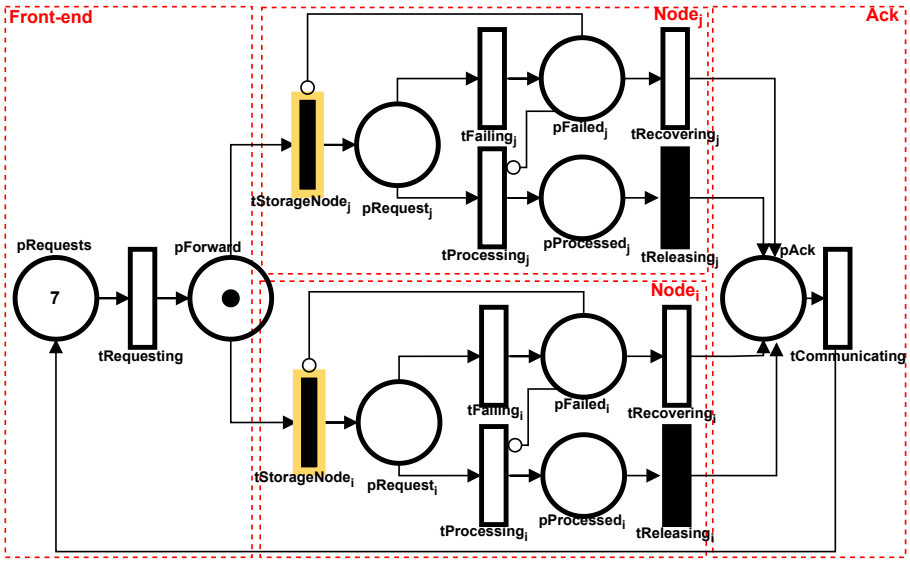


Figure C.2: Performability model execution - all storage nodes available (own work (2023)).

Figure C.2 depicts a generated request and the system's readiness to forward it for processing by storage nodes. Specifically, the firing of transition *tRequesting* results in the absorption of a token from place *pRequests* and the generation of a token in place *pForward*. Transitions $tStorageNode_i$ and $tStorageNode_j$ are enabled, and their firing as immediate transitions depends on their assigned priorities.

Figure C.3 illustrates a token generated at place $pRequest_j$, indicating the firing of transition $tStorageNode_j$ and the absorption of the token from place *pForward*. At the current stage of execution, transitions *tRequesting*, $tFailing_j$ and $tProcessing_j$ are enabled, with their firing dependent on the respective assigned delays. The firing of transition $tFailing_j$ represents the failure of the storage node in question, whereas triggering transition $tProcessing_j$ represents the processing of a request.

Figure C.4 illustrates a processed request, denoted by the tokens in place $pProcessed_j$. This processing is represented by the firing of transition $tProcessing_j$, resulting in the absorption of the token from place $pRequest_j$. The enabled transition $tReleasing_j$ indicates the storage node can transmit information regarding the concluded processing of requests. Once this is fired, a token is absorbed from place $pProcessed_j$ and generated in place *pAck*.

In Figure C.5, *tCommunicating* is depicted as enabled, and its firing absorbs a token

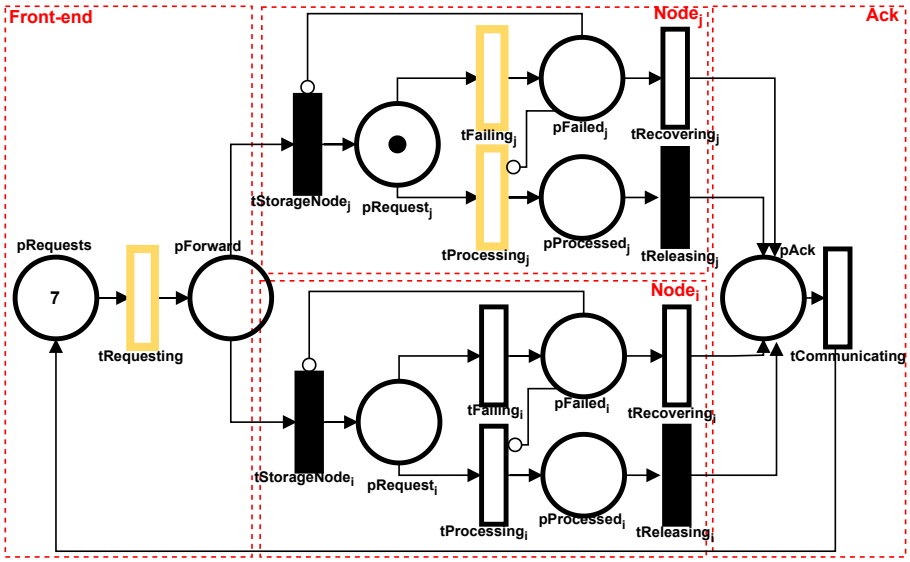


Figure C.3: Performability model execution - ready for processing (own work (2023)).

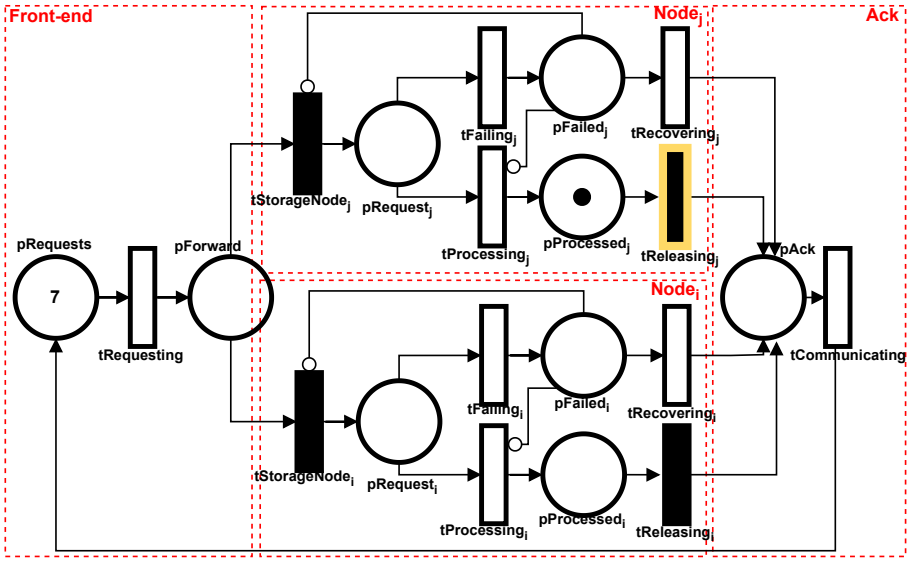


Figure C.4: Performability model execution - processing request (own work (2023)).

from place *Ack* and generates a token in place *pRequests*. This firing represents the transmission of an acknowledgment regarding a processed request or storage node repair.

Figure C.6 shows the storage node j as unavailable, represented by tokens in place $pFailed_j$. Inhibiting arcs prevents transitions $tStorageNode_j$ and $tProcessing_j$ from being enabled. This means that the respective storage node cannot receive new requests or processes that might have been waiting at the time of failure. Therefore, new requests are directed to storage node i, represented by tokens in place *pForward* and the enabling of transition $tStorageNode_i$.

The repair of storage node j is shown in Figure C.7. This is represented by the absorption

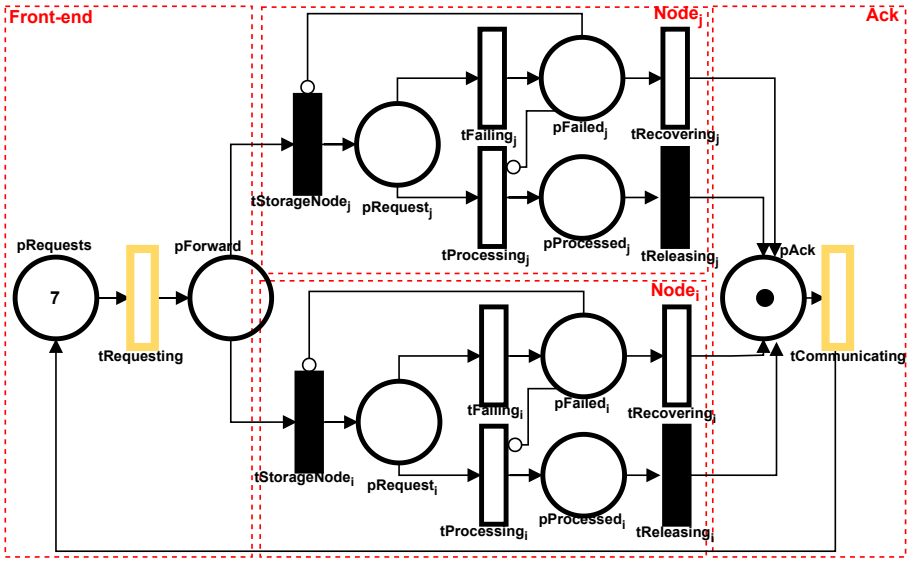


Figure C.5: Performability model execution - ready for communicating (own work (2023)).

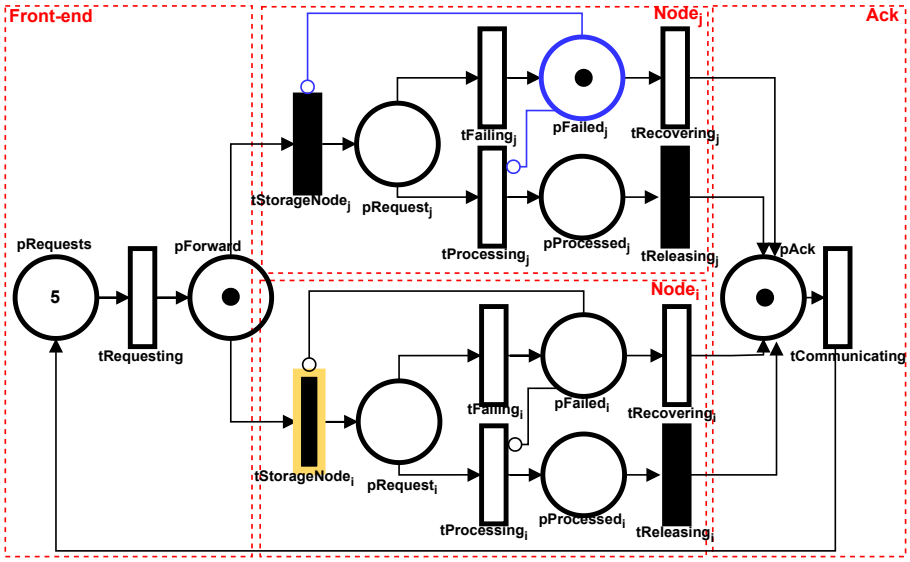


Figure C.6: Performability model execution - storage node unavailable (own work (2023)).

of tokens from place $pFailed_j$ and their generation in place $pAck$. As can be seen, new requests can once again be directed to both storage nodes considered in this model. In this manner, the tokens in place $pForward$ enable transitions $tStorageNode_i$ and $tStorageNode_j$.

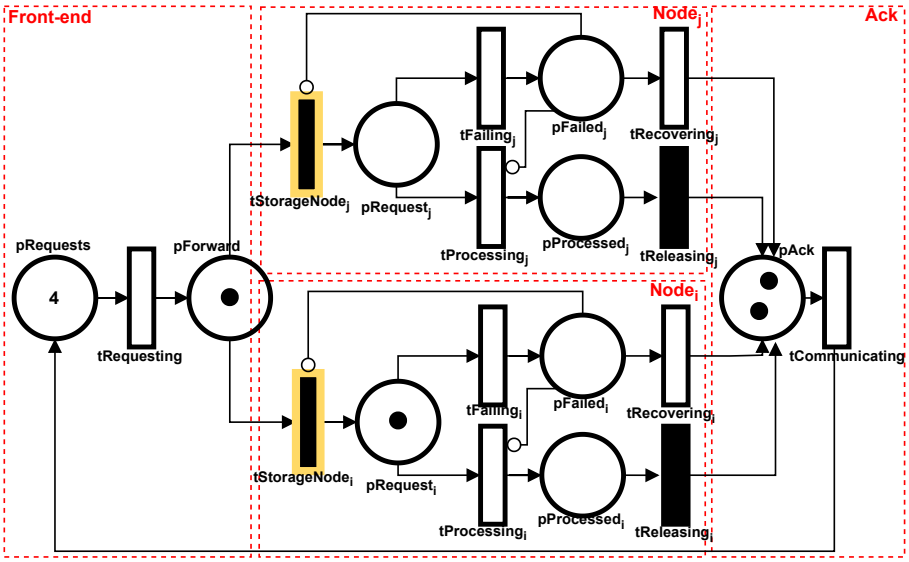


Figure C.7: Performability model execution - repairing storage node (own work (2023)).