

**ENHANCED AUTOMATED HETEROGENEOUS
DATA DUPLICATION MODEL USING
PARALLEL DATA COMPRESSION AND
SORTING TECHNIQUE**

NUR AQILAH PASKHAL ROSTAM

UNIVERSITI SAINS MALAYSIA

2019

**ENHANCED AUTOMATED HETEROGENEOUS
DATA DUPLICATION MODEL USING
PARALLEL DATA COMPRESSION AND
SORTING TECHNIQUE**

by

NUR AQILAH PASKHAL ROSTAM

**Thesis submitted in fulfilment of the requirements
for the degree of
Master of Science**

May 2019

ACKNOWLEDGEMENT

Alhamdulillah, all praise to Allah, I would like to grab this opportunity to express my gratitude to my supervisor, Prof Dr. Rosni Abdullah for being so thoughtful and giving me the best guidance, valuable feedback, full support and encouragement throughout my research work. I gained a lot of new knowledge after working with her. I believe it is impossible to complete this research at this stage without her guidance and immense expertise.

My sincere acknowledgement should also go to Mr Daniel and Puan Suzi on behalf of Sophic Automation Sdn Bhd, for their suggestion and financial support, keeping me on track with my study. Not to forget to my examiners and panels, Dr. Adib, Prof. Dr. Zulaikha, Associates Prof. Putra Sumari, Associates Prof. Zurinahni Zainol and Dr. Manmeet for correcting my mistakes and providing valuable suggestions in improving my thesis. Next, I would like to give my sincere gratitude to my parents Mariyati and Rostam for always relentlessly showering me with advice, motivation and unstoppable love, God knows and I cannot write it all here. My special thanks also dedicated to my lovable sisters and my happy pills, Talib, Nana and Ika and to my sweet, supportive and caring friends, Khairun, Farahin, Shahirah, Haziqah, Izzati, Amirah, Kak Jiha, Kak Sarah and Jaziem who were always be with me along my master's journey that the completion of this research would be impossible without all of them.

Lastly, I am glad that I joined School of Computer Sciences as a place to pursue my study of a Master's degree in Universiti Sains Malaysia and this work was supported by Crest with the collaboration of Sophic Automation Sdn Bhd. Without

them, I would never have industrial opportunity. I have obtained precious memories and embarked a wonderful odyssey of knowledge throughout my study life in this university. Lastly, to the most important person, my outmost motivation and my blessing in disguise,

Mama, this is for you!

TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	ii
TABLE OF CONTENTS.....	iv
LIST OF TABLES.....	vii
LIST OF FIGURES.....	ix
LIST OF ABBREVIATIONS.....	xi
ABSTRAK.....	xiv
ABSTRACT.....	xvi
CHAPTER 1 INTRODUCTION	1
1.1 Research Background.....	1
1.2 Problem Statement	7
1.3 Research Objectives	10
1.4 Scope of the Study.....	10
1.5 Limitation of the Study	11
1.6 Significance of the Study	11
1.7 Outline of Thesis	12
CHAPTER 2 BACKGROUND & LITERATURE REVIEW	14
2.1 Introduction	14
2.2 Basic Architecture and Characteristics of Flash Memory.....	15
2.3 Basic Concept of Digital Data, Data Storage and Data Duplication.....	18
2.3.1 Database Replication	22
2.3.2 Hard Disk Duplication	24

2.3.3	Flash Duplication	26
2.4	Related Work on Data Compression Technique	33
2.5	The Use of Sorting Techniques in Data Storage and Compression	37
2.6	The Use of Parallel Technique in Data Transfer or Duplication Process	40
2.7	The Use of Parallel Method in Data Compression Process	43
2.8	Research Gap Discussion	45
2.9	Summary	47
CHAPTER 3 RESEARCH METHODOLOGY		48
3.1	Introduction	48
3.2	Research Framework	48
3.2.1	Stage 1 : Problem Formulation	50
3.2.2	Stage 2 : Proposed Model and Solutions	54
3.2.3	Stage 3 : Experiments and Evaluation	57
3.3	Summary	66
CHAPTER 4 PROPOSED ENHANCED HETEROGENOUS DATA DUPLICATION MODEL USING PARALLEL DATA COMPRESSION AND SORTING TECHNIQUE		68
4.1	Introduction	68
4.2	Proposed Enhanced Data Duplication Model	68
4.3	Design of Enhanced Data Duplication Model.....	70
4.4	Results and Analysis	78
4.4.1	Experiment 1 Test Result and Analysis	79
4.4.2	Experiment 2 Test Result and Analysis	82
4.4.3	Experiment 3 Test and Results Analysis	85

4.4.3(a)	Bit Reduction and Compression Utilities.....	85
4.4.3(b)	Chosen Compression Level on Data Duplication Performance.....	87
4.4.4	Experiment 4 Test and Results Analysis	88
4.4.4(a)	Data Duplication by using Multithreading.....	89
4.4.4(b)	Data Duplication by using Parallel.foreach API.....	91
4.5	Discussion	95
4.6	Summary	97
CHAPTER 5 CONCLUSION AND FUTURE WORK.....		99
5.1	Conclusion.....	99
5.2	Future Works.....	100
REFERENCES.....		101
APPENDIX		

LIST OF TABLES

	Page
Table 2.1	A Summary of the Data Duplication Techniques 22
Table 2.2	A Summary of Hard Disk Duplication Related Work 26
Table 2.3	A Summary of Flash Duplication Method 26
Table 2.4	A Summary of Duplication Method with Dataset & Performance 29
Table 2.5	A Summary of Compression Technique 36
Table 2.6	A Summary of Sorting and Compression in Data Transfer 37
Table 2.7	A Summary Comparison of Parallel Method Performance 42
Table 2.8	A Summary of All Techniques in Data Transfer 43
Table 3.1	eMMC Card Product 50
Table 3.2	Duplicators Specification 51
Table 3.3	Comparison of Duplicator Results 52
Table 3.4	Dataset Division for the First Round 54
Table 3.5	Dataset Division for the Second Round 54
Table 3.6	eMMC Socket Specification 56
Table 3.7	Operating System Comparison 57
Table 4.1	Recorded Duration for Duplication Process in Experiment 1 80
Table 4.2	Duplication Process Duration for Experiment 2 82
Table 4.3	Comparison between Experiment 1 and Experiment 2 84
Table 4.4	Data Size Reduction for Individual Dataset 85
Table 4.5	Data Size Reduction for Individual Dataset Using “Fastest” 85
Table 4.6	Data Size Reduction for Individual Dataset Using “Optimal” 86
Table 4.7	Duplication Process Duration after Compression in Experiment 3 ... 87

Table 4.8	Comparison of Results between Experiment 2 and Experiment 3	88
Table 4.9	Result of Multithreading for Experiment 4.....	89
Table 4.10	Comparison Result between <i>Parallel.Foreach</i> and Multithreading ..	91
Table 4.11	Results of Naïve Algorithm in the First Experiment	93
Table 4.12	Comparison of Duplication Performance from Three Experiments ..	93

LIST OF FIGURES

	Page
Figure 1.1	Mobile Handset Booting Architecture (Tsai, 2011).....4
Figure 1.2	eMMC Shipments by Application (Yang, 2014).....4
Figure 1.3	Ball Grid Array (BGA) of eMMC5
Figure 1.4	Thesis Structure 13
Figure 2.1	Literature Review Outline..... 14
Figure 2.2	NAND Cell Structure..... 15
Figure 2.3	Overall Architecture of Flash Memory System (Liu et al., 2009) 16
Figure 2.4	Data Representation in Digital Data Format..... 19
Figure 2.5	Conversion of Analog Signal to Digital Data 19
Figure 2.6	Abstraction of Data Storage Concept in Real Life 20
Figure 2.7	Basic Data Transfer Process (Barr & Rewini, 2005)..... 21
Figure 2.8	Database Replication Architecture (Wiesmann et al., 2000)..... 23
Figure 2.9	Disk Cloning Process 24
Figure 3.1	Research Methodology 49
Figure 3.2	eMMC Socket 55
Figure 3.3	Experimental Design..... 61
Figure 3.4	The Summary of Four Experiments..... 66
Figure 4.1	The Proposed Enhanced Data Duplication Model..... 70
Figure 4.2	Proposed Data Duplication Model (High Level View) 71
Figure 4.3	Difference between Sequential and Data Parallelism 76
Figure 4.4	Summary of all Experiments with Specified Sub- Experiments..... 79
Figure 4.5	Duplication of 800MB Image Dataset 81

Figure 4.6	Duplication of of 800MB Mix Dataset (Overall).....	82
Figure 4.7	Duplication of Overall of 800MB Dataset After Sorting.....	83
Figure 4.8	Sequential Data Duplication (One Thread to Duplicate Each Data) .	90
Figure 4.9	Parallel Data Duplication (Multiple Thread Data Duplication).....	90
Figure 4.10	Local Duplication Result in Experiment 1 until 3	95
Figure 4.11	Across Devices Duplication Result in Experiment 1 until 3	96

LIST OF ABBREVIATIONS

RAM	Random Access Memory
ROM	Read Only Memory
PROM	Programmable Read-Only Memory
EPROM	Erasable Programmable Read Only
EEPROM	Electrically Erasable Programmable Read Only Memory
CD-ROM	Compact Disc Read-Only Memory
eMMC	Embedded Multi-media Card
MMC	Multimedia Card
USB	Universal System Bus
FTL	Flash Translation Layer
I/O	Input Output
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MTD	Memory Technology Device
MicroSD	Micro Secure Digital
SD	Secure Digital
SSD	Solid State Drive

SATA	Serial Advanced Technology Attachment
FPGA	Field-Programmable Gate Array
DSP	Digital Signal Processor
ECC	Error Correction Code
CF	Compact Flash
CR	Compression Ratio
CPU	Central Processing Unit
HDD	Hard Disk Drive
HS-MMC	High-Speed MMC
BGA	Ball Grid Array
PC	Personal Computer
JEDEC	Joint Electron Device Engineering Council
XIP	eXecute In Place
KB	Kilobyte
MB	Megabyte
GB	Gigabyte
MB/s	Megabyte Per Second
HID	Human Interface Device

MCU	Multipoint Control Unit
UART	Universal Asynchronous Receiver-Transmitter
LAN	Local Area Network
WAN	Wide Area Network
DMA	Direct Memory Access
EDT	Efficient Data Transfer
GPU	Graphic Processing Unit
SIMD	Single Instruction Multiple Data
API	Application Program Interface
RnD	Research and Development

**MODEL DUPLIKASI DATA HETEROGEN TERTINGKAT
MENGUNAKAN TEKNIK SELARI, PEMAMPATAN DAN PENGISIHAN
DATA**

ABSTRAK

Mesin duplikator bertujuan untuk memperbaiki masa yang diambil untuk proses duplikasi atau pemindahan data. Proses duplikasi dilakukan dengan menyalin setiap bit data dari sumber (*master*) peranti ke peranti yang lain (*slaves*) termasuk ruang storan ingatan yang tidak digunakan. Walau bagaimanapun, untuk menduplikasi 64GB Kad Memori Multimedia Terbenam (eMMC) biasanya memakan masa yang panjang antara 2 jam hingga 7 jam. Di samping itu, spesifikasi kelajuan produk yang dijanjikan oleh vendor berbeza daripada apa yang mereka telah dakwa apabila diuji dalam proses duplikasi yang sebenar. Tambahan pula, data yang lebih besar mencipta masalah semasa penghantaran data dan menyebabkan kelewatan semasa duplikasi data. Hal ini akan mengurangkan prestasi proses duplikasi dari segi tempoh duplikasi data. Oleh itu, kajian ini mencadangkan satu kaedah untuk mempertingkatkan tempoh teknik duplikasi. Ini dapat dicapai dengan menggunakan konsep penyimpanan dan penghantaran data melalui teknik penyusunan dan pemampatan. Teknik selari juga digunakan untuk mempertingkatkan penghantaran data ke beberapa peranti hamba yang lain. Pemerhatian terhadap kesan jenis dan struktur data kepada prestasi proses duplikasi juga dilakukan. Empat eksperimen dijalankan dengan menggunakan saiz data digital heterogen yang sama (iaitu dokumen, gambar, audio dan filem). Secara keseluruhannya, hasil eksperimen menunjukkan proses duplikasi yang menggunakan jenis data yang berbeza telah memberikan tempoh yang berlainan. Teknik yang

dicadangkan telah mengurangi penggunaan masa atau tempoh duplikasi data sebanyak 20% hingga 50% bergantung pada teknik duplikasi yang dilakukan secara setempat atau merentas peranti.

**ENHANCED AUTOMATED HETEROGENOUS DATA DUPLICATION
MODEL USING PARALLEL DATA COMPRESSION AND SORTING
TECHNIQUE**

ABSTRACT

A duplicator machine aims to improve the time taken for duplication or data transfer. The process of duplication is done by copying each data bit from the source (master) device to the slaves including the unused memory region. However, to duplicate a 64GB Embedded Multimedia Card (eMMC) memory is usually very time consuming which takes between 2 hours to 7 hours. In addition, the product speed specification promised by the vendor is different from what they claimed to be when it is tested in real life. Moreover, bigger data creates a transmission problem, causing delay during data duplication. Consequently, this will reduce duplication performance in terms of duration. Therefore, this study was proposed to enhance the duplication technique duration. This was achieved by adopting data storage and transmission concepts through sorting and compression techniques. Parallel technique was adopted to enhance data duplication process using multiple slaves. The impact of data type and data structure to the duplication performance was also studied. Four experiments were conducted by using the same size of heterogeneous digital data (i.e. document, picture, audio and movie). Overall, the results showed that data duplication process using different data type render a different duration. The proposed technique has reduced time consumption by 20% to 50% during data duplication depending on the technique and the environment of local and across devices.

CHAPTER 1

INTRODUCTION

1.1 Research Background

Along with digital revolution is the emergence of digital computers leading the creation of new information relevant to human life. The creation of digital content has been consistently facilitated with the digital revolution as it becomes progressively more accessible and available. Going digital provides a trouble-free process of data transportation by reducing reliance on hard documents like books or papers.

Film industry today has benefited the digital cinema concept where the cinema distributes and projects a movie in a form of digital media with supported technology to the theaters. In the past, movies were sent as film rolls in a container to theatres where they were projected and returned later by couriers such as FedEx and other related courier companies. Unfortunately, film rolls were easily exposed to misfortunes such as damage, stolen and lost. Moreover, it is extremely costly to produce and deliver thousands of films rolls.

Today, digital cinema drastically reduces the cost of distributing movies and ensures a greater degree of security against theft or damage. Digital movies are usually sent in hard disks, Blu Ray discs or secondary memory such as SD card which functions as the storage medium. To rapidly reproduce the films, the industry uses duplicator machine to speed up the process of duplication by cloning the digital media. The digital media such as movie is stored in eMMC card using eMMC duplicator.

eMMC card is usually used due to its cost-effective and mobility advantages to ease the film production and distribution process.

Despite the convenience of data storage and digital data, the trend also has made digital data backup as a crucial process for many industries. Other industries to benefit may include digital forensic, banking, marketing, and especially the industry that are responsible for handling important data. The relevance of this process is usually to retain data for a long period of time by storing the data in secondary storage due to its non-volatile characteristics.

Storage can be divided into two types which are volatile and non-volatile memories. Volatile memory requires power to maintain stored information but eventually will lose its data when power is interrupted. This may include a Random Access Memory (RAM) with static and dynamic RAM. Therefore, the need to store files over a long period of time and maintain users data when the computer is switched off has led to the non-volatile memory which is able to retain information during power outage. Read Only Memory (ROM), Programmable Read-Only Memory (PROM), Erasable Programmable Read Only (EPROM) or Electrically Erasable Programmable Read Only Memory (EEPROM) are examples of non-volatile memory and Flash memory is an example of EEPROM (Bez et al., 2003).

Duplicator machine is one type of flash application that is utilized to speed up the transfer and duplication process by creating multiple copies of flash storage content from the master (sources) to the slaves (target sources). As it can copy all the content from flash applications such as USB storage drives, eMMC, SD/MicroSD Cards and other flash application, flash duplicators make a great accessory to any

offices or workshops since the flash content can only be reproduced through duplication. Therefore, eMMC duplicator is designed to meet eMMC card content duplication & verification manufacturing needs in compliance with its specification (Kingston, 2013). eMMC is usually used in many industries due to its cost-effective value and belongs to flash memory family that is fast in reading and writing makes it very suitable for Research and Development (R&D).

Flash memory has recently been adopted in embedded applications with several features, including its non-volatility, fast access speed, shock resistance, and, low power consumption (Chung et al., 2009). In addition, eMMC as external storage which is not connected directly to CPU or stand-alone is suitable for high-performance applications and can be used as a replacement for traditional storage media, such as hard disk drive (HDDs). Moreover, there are various end-use applications of eMMC such as smartphones, digital cameras, portable devices, and tablets.

eMMC is designed as the embedded memory standard specification defined by the MMC (MultiMediaCard Association). Besides, Baker (2010) explained that eMMC typically includes flash memory component, multimedia card interface, flash memory controller, and high-speed MMC (HS-MMC) driver. Tsai (2011) provided a trend chart of the eMMC market that shows the proliferating eMMC usage driven by rapid growth in smartphone sales (see Figure 1.1). On the other hand, the study of Yang (2014) showed that the history of eMMC shipment by applications and the forecast of the growth (see Figure 1.2).

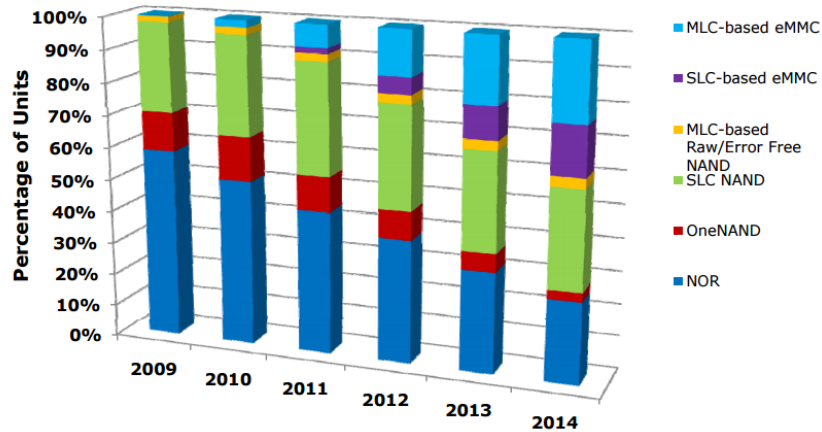


Figure 1.1 Mobile Handset Booting Architecture (Tsai, 2011)

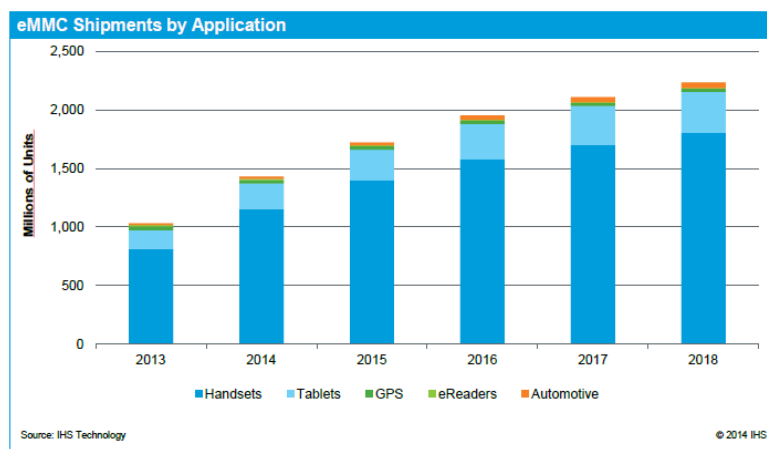


Figure 1.2 eMMC Shipments by Application (Yang, 2014)

Figure 1.2 presents the high shipment by application of eMMC pertaining to its advantages. Moreover, the increase of various secondary memory today has created convenience to retain information during a power outage and can satisfy the need to store abundant of digital data. eMMC consists of one embedded storage where inside the MMC Interface, NAND Flash, and Controller, they are packaged into one small chip called Ball Grid Array (BGA) as shown in Figure 1.3.

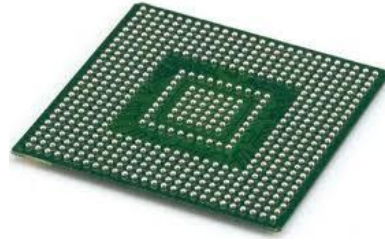


Figure 1.3 Ball Grid Array (BGA) of eMMC

Even though eMMC duplicator is known for its speed but the machine would still consume a lot of time when the volume of data increases. This is due to bit by bit methodology during the duplication process that compromises the time. Moreover, the specification provided by the eMMC duplicator vendor is usually not the same. The speed of the duplicator is usually slower than what it claims to be during the actual testing. This has led to the main objective of this research which is to improve the existing duration of the duplication process.

Weaknesses in the duplication method in term of duration or speed can be seen through flash hardware technology that is usually driven by a desire to increase the capacity and performance. The trend requires a new method and architecture system for a new type of devices. However, these techniques are only described in patents, not in technical articles (Gal & Toledo, 2005). The disadvantages of patents are difficult to read, not detail, usually perform without quantitative comparisons to alternative techniques, and no theoretical analysis as compared to technical articles, journals, and conference proceedings article where insight of the method is given (Gal & Toledo, 2005).

These reasons have made the research process more complicated as the duplicator has never been exposed with duplication method and recent research on flash domain is mainly focused on the flash characteristics and not on data transfer in

term of duplication duration. This research looks into duplication method of flash application. Some examples of duplication work that have been conducted previously were (Labský, 2018; Aldaej et al., 2017; Shah et al., 2014; Sawant & Shah, 2013; Rashid & Khan, 2012; Tiwari et al., 2011; Panichprecha et al., 2011; Kim, 2011; Agus, 2010). Most of these research were standalone and only focused on data transfer without any improvements taken with other methods or hybrid algorithm. Moreover, the overall past research were limited to only certain type of data or homogenous data. A thorough research would be impossible with the limitation of testing data as each data type has a different data structure.

Lastly, to further improve the performance of the algorithms, one of the crucial steps to be taken into consideration is by doing a hybrid. Numerous methods have been suggested to improve the duplication performance as suggested by Kammer (2014). The two most common measures are speed and memory usage. The other measures could include transmission speed, temporary disk usage, long-term disk usage, power consumption, the total cost of ownership, and response time to external stimuli.

Many of these measures depend on the size of the input to the algorithm, for example, the amount of data to be processed. In addition, by adopting data storage concept for hybrid, the design goal of data compression helps to represent data with as few bits as possible by reducing the data size to save storage and transmission channel capacity (Longley et al., 2005). Logically, in data storage, by reducing the data size, less bit of data is presented and duplicated. Hence, this helps to reduce the time taken for a duplication process to complete. Data compression also helps to improve operating time, lower communication latencies, reduce costs, and make more

effective use of available bandwidth and storage (Armen, 2016). Data compression on duplication or data transfer work has been done in the past. On the other hand, parallel techniques can be adapted to increase the performance of sequential data duplication as it has been widely used in data transfer and compression in the past.

Regarding the homogenous data of past research, Armen (2016) mentioned the need for a network, data file size, and data type as important requisites to likely affect data transfer performance. This research, therefore, investigates the use of heterogeneous data that consists of basic digital data such as documents, pictures, videos, and audios.

1.2 Problem Statement

General duplication process is done by copying each data bit from the master device to the targeted device including the unused memory region (Mazonka, 2009). The efficiency of duplicating the memory content for a standard product has become a challenge as it determines the overall throughput success of the product. The problem is, to duplicate a 64GB eMMC memory typically takes up between two to seven hours. Due to the sequential bit-by-bit methodology used, it takes a long time to duplicate a large volume of data. This has increased the time taken or duration during the duplication process.

Moreover, the specifications provided by vendors Phiyo (2009) and Dediprogram (2014) claimed to have the highest average data duplication speed designed that is up to 200MB/second. In fact, some of the duplicators can only perform at the speed of

3MB to 6MB/second in real experience during product analysis and benchmarking process.

Unfortunately, recent algorithm or method related to PC-duplicator is very hard to find and the available duplicator method only describes in patents, not in technical articles (Gal & Toledo, 2005). The disadvantages of patents are difficult to read, not detail, and no theoretical analysis and quantitative comparisons to alternative techniques as compared to technical articles or journals. These reasons have made the research process complicated to gain insight into the method as the duplication method has never been exposed and studied (Gal & Toledo, 2005).

However, the recent research only discussed on the architecture and limitation of flash such as Flash Translation Layer (FTL) especially on the; (i) mapping scheme (Chiao & Chang, 2011; Wang et al., 2016) (ii) or concern with the use of NAND flash chip on how to reduce energy or power consumption during erase operation (Shim et al., 2012; Tseng et al., 2013) (iii) and how to lengthen the flash storage duration or lifetime (Kim, 2015; Jimenez et al., 2013; Lee & Kang, 2016). Moreover, FTL has limitation due to “erase before write” characteristic as prescribed in patents. Nevertheless, by erasing the whole flash unit in effect worsening the flash performance (Gal & Toledo, 2005; Parthey, 2007). Other common issues such as quality of the data being stored and delay perceived by the user are due to big amount of data which have created transmission problem (Longley et al., 2005) are rarely discussed.

To summarize, due to lack of duplication enhancement as the duplicator speed is not as high as the vendor claim and research in flash domain has focused a lot on

NAND, FTL architecture and the limitation of flash characteristic “erase before write”, has led many research to overcome the limitation of NAND but not to improve the performance of data duplication. In addition, the use of homogenous data as the dataset as in the past could not guarantee a thorough research study as Armen (2016) mentioned that the researchers to not neglect the other device characteristics such as processor speed, memory, network interfaces, data file size, and data type. This is because, different types of data have different data structure. There are possibilities of changes in the performance depending on the data type used.

Therefore, rather than focusing on the FTL or NAND chip, some improvements can be done by focusing more in terms of data transfer speed so that the process is able to reduce time consumption during the duplication process and to utilize the use of heterogenous data in the implementation. With that, two research questions are explored:

1. How to reduce the duration or time consumption during the duplication process with and without flash device such as eMMC by neglecting the FTL technique and focuses on the duplication or data transfer?

2. What is the impact of using different data type, with the same data size, and data arrangement to a duplication performance?

1.3 Research Objectives

The main aim of this research is to determine a suitable method to improve data duplication process while increasing time efficiency during the duplication process.

The research objectives are:

1. To propose a method to enhance the eMMC data duplication performance in terms of duration by using compression, sorting, and parallel method.
2. To study the effect of data duplication performance in terms of duration when using heterogeneous data.

1.4 Scope of the Study

This study is proposed to enhance data duplication in flash domain that usually involves both software and hardware. The execution of the enhanced duplication method is PC-based software with eMMC version 5.0 as the flash memory storage. Owing to the limitation and lack of references on the eMMC duplicator, the research focuses on the duplication algorithms that involve USB data structures of USB eMMC card reader adapter version 2.0 which is used to ease the accessibility of data during duplication process. Besides, benchmarking process of the standalone duplicator will be made among the existing duplicator market products such as Phiyo and Dediprogram. For implementation part, basic duplication method is adopted from previous researcher (Sawant & Shah, 2013) and the enhanced proposed solutions are based on the past work on flash domain and focuses on data storage concept which encompasses sorting, compression, and parallel method.

1.5 Limitation of the Study

1. The maximum duplication speed is hindered by the communication interface between PC and eMMC device as JEDEC.
2. The number of slaves in the configuration of one master device to multiple slaves implementation is limited depending on the number of ports and the budget of customizing the required hardware.

1.6 Significance of the Study

‘Big data’ and mass of data are used for analysis and transfer via a network to be stored. This process crucially needs data backup for disaster recovery purpose in many industries. Digital data is normally transmitted across the network with large amount of data that may create problem regarding storage space and transmission time (Longley, 2001). An enhanced data duplication method is proposed to improve data duplication performance that explores data transfer efficiency.

Therefore, it is an interesting research to look for a potential method in enhancing the duplication speed from many aspects especially from the data structure, data flow, and dataset used that mostly has less interest from the current research trend in flash memory domain. This method eventually may help the industries to implement a cost-effective and time-effective process using a PC from spending more on expensive hardware. This is a good way to save research and development cost and protecting customers’ data. In addition, this may address the global trend on storage and throughput imbalance between the data production on such systems and their I/O subsystems.

1.7 Outline of Thesis

A brief overview of each chapter are as follows:

Chapter 2 reviewed in detail the related work of domain problem and at the same time emphasis on the selected domain which consist of the following: (i) some background study of the duplicator, (ii) eMMC memory card, (iii) flash memory in general and related work of previous research on data duplication, (iv) sorting, (v) data compression technique, (vi) and parallelism technique during data transfer or data duplication to help in understanding the overall context of the thesis. Chapter 3 described the research methodology in this thesis which consisted of research proposed model, data sources, problem description, performance measure and experimentation in fulfilling the objectives of this research. Chapter 4 discussed the detail on each performance of the selected method. The experiments were divided into four where one experiment acts as a control experiment, which was the performance of naive duplication without any improvement and three experiments were conducted using three different methods which were sorting, compression and parallel method and are evaluated using respective performance measure. Finally, Chapter 5 discussed the conclusion of the thesis along with the future works. The thesis structure is presented in Figure 1.4.

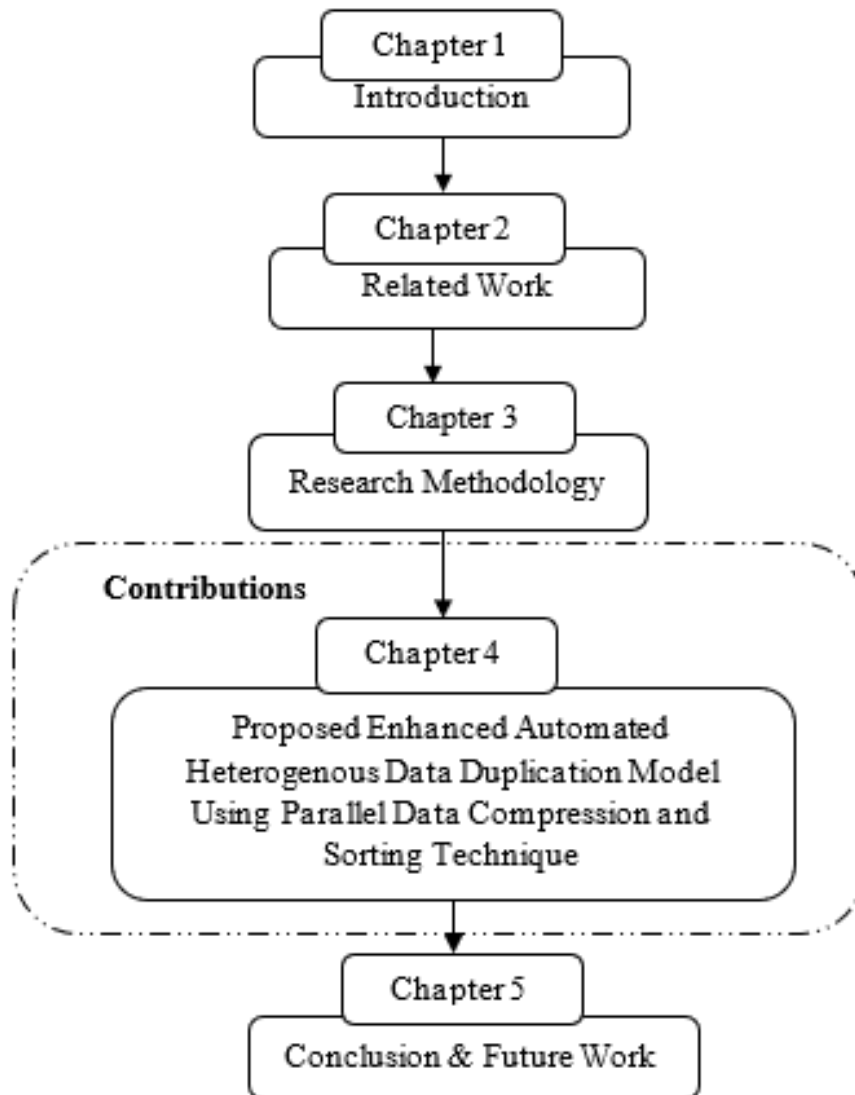


Figure 1.4 Thesis Structure

CHAPTER 2 BACKGROUND & LITERATURE REVIEW

2.1 Introduction

This chapter provides the reviews on related works in the area of flash domain duplication process and also highlights on a suitable method for the domain problem as well as the methodology for the research study. Methods for improvement will also be discussed namely compression, sorting, and parallel method. Additionally, the research gap discusses the potential direction derived from the review and within the scope of this research study. The organization of this chapter is illustrated in Figure 2.1.

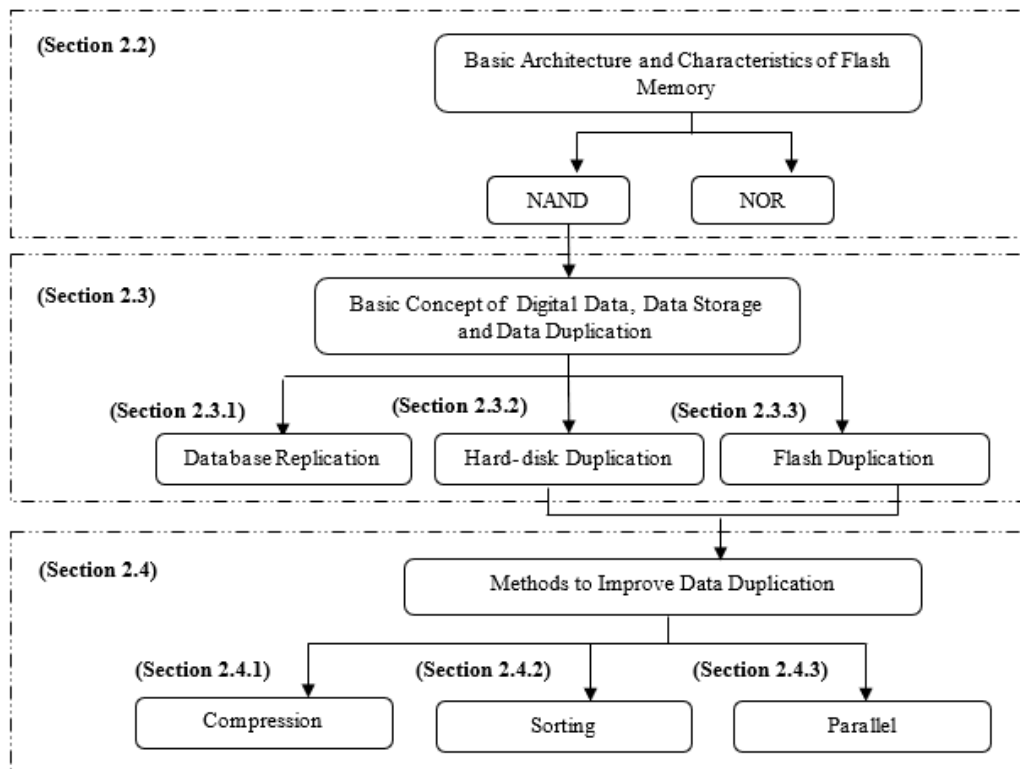


Figure 2.1 Literature Review Outline

2.2 Basic Architecture and Characteristics of Flash Memory

Two major types of flash memory in the current market are NAND and NOR flash. NOR is the best used for code storage and execution, usually in small capacities, offers eXecute In Place (XIP) capabilities and high read performance (Iniewski, 2010). It is cost-effective in low capacities but suffers from extremely low write and erase performance. On the other hand, NAND architecture offers extremely high cell densities and high capacity combined with fast write and erase rates (Iniewski, 2010). NAND obviously offers many advantages in terms of duplication performance due to its fast write and erase rates. NAND flash chip consists of die, plane, block, and page (see Figure 2.2).

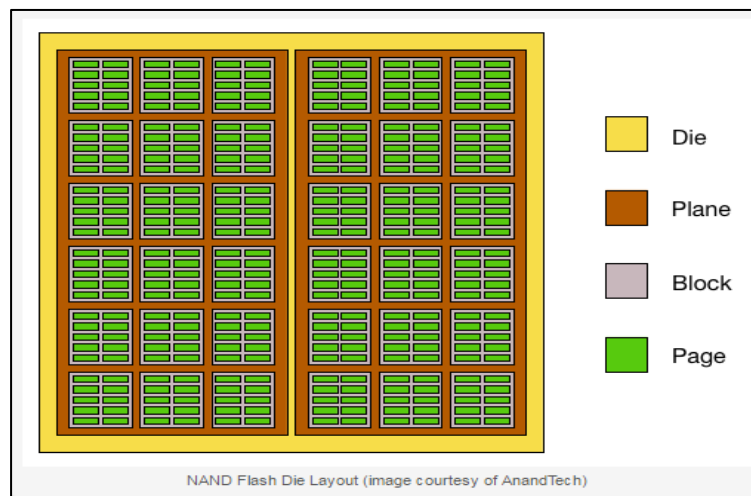


Figure 2.2 NAND Cell Structure

Each plane contains a number of blocks which are the smallest unit that can be erased. Each block contains a number of pages which are the smallest unit that can be programmed or written to. The erase operation happens at block level while read and write are at the page level. Writing operation takes place to a page which might be 8KB to 16KB in size but erase operation takes place to a block which might be 4MB

to 8MB in size that makes all of the pages in a block need to be candidates for erasure since a block needs to be erased before it can be written (Gal & Toledo, 2005). Bytes are organized into pages, that consist of 512 bytes of data area and 16 bytes of the spare area in every single page. Data area stores the actual data while the spare area is used for memory management purposes (Rahiman & Sumari, 2011). Figure 2.3 shows the Flash Translation Layer (FTL) which is a middleware deployed between memory technology device (MTD) and file system. The MTD provides primitive functions such as read, write, and erase that directly operate on a flash memory system.

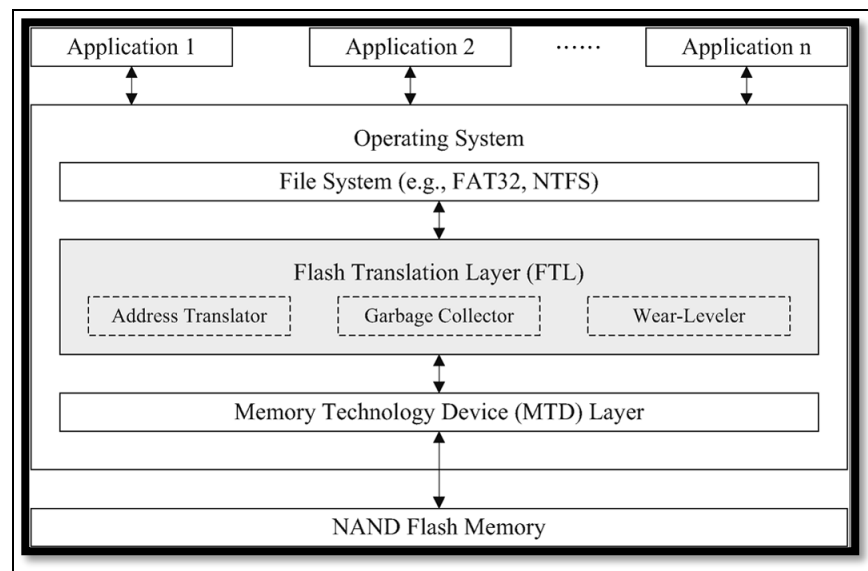


Figure 2.3 Overall Architecture of Flash Memory System (Liu et al., 2009)

Liu et al. (2009) described the FTL as a block device interface to the user by emulating the operations of hard disk drives using flash memory and the key role of the FTL is to translate the logical sector address from the operating systems into the physical flash address used by flash memory. The address translation can be done by some mapping algorithms, page, block, and hybrid mapping. In-depth information of flash memory data structure is studied by Gal and Toledo (2005). This research only

highlighted the drawbacks and limitation of the scheme in general and proposed a better method to improve duplication and data transfer technique.

Page level Mapping is a flexible scheme that writes data in units of a page. Any logical pages can be mapped to any physical pages in flash memory. In a simple word, the row size of the logical-to-physical mapping table corresponds to the logical pages recognized by the file system. Kale and Jahagirdar (2012) highlighted that page level FTL algorithm shows good overall performance for both read and write operations. However, it requires a large amount of Random Access Memory (RAM) for operation. Thus, it is hardly feasible for the small embedded system.

To overcome the large RAM requirement of page mapping scheme, block-level FTL algorithms are proposed (Ban, 1999). In block-level mapping, the row size of the logical-to-physical mapping table is the same as the total number of blocks in flash memory. As a result, the size of the mapping table is much smaller, unfortunately, it causes performance bottleneck (Suh et al., 2012) and increases garbage collection overhead (Lee et al., 2008).

Hybrid mapping schemes have been introduced due to the shortcomings of both page and block level mapping schemes. Generally, hybrid mapping combines the merit of both page and block mapping concept by using the block mapping technique to obtain the corresponding physical block and then, the page mapping technique to locate an available free page within the physical block. However, hybrid mapping requires more mapping information than the block mapping scheme (Chung et al., 2009).

From the background study of flash domain, the trend shows that most of the research focus on the limitation of flash characteristic, “erase before write”, therefore,

various studies have been made to cater the consequences of the limitation such as to conserve the energy or power during erasure operation (Shim et al., 2012; Tseng et al., 2013). Besides, some research (Kim, 2015; Jimenez et al., 2013; Lee & Kang, 2016) focused on lengthening the lifetime of NAND chip as a consequence of excess erasure operation. Studies in the past also centered upon improving various mapping method from page-mapping until hybrid mapping owing to similarities of FTL and flash performance (Chiao & Chang, 2011; Wang et al., 2016).

Unfortunately, the drawbacks of FTL method transparently maps blocks of a block device to physical erase units on the flash device. It takes care of wear leveling and therefore, it allows the transparent use of traditional block-oriented file systems on raw flash memories which do not provide any flash specific wear leveling mechanisms. This block device approach is highly inefficient because no matter how little data is written, any write access to the device requires erasing a whole flash erase unit. If the data units that have been written are smaller than one erase unit of the flash device, and the file system is not aware of the underlying hardware, it uses the hardware in a very inefficient way (Parthey, 2007).

Therefore, rather than focusing on the FTL, some improvements can be done by focusing more on the data storage aspect which is the data itself and how data is arranged and transferred during data duplication process across flash devices.

2.3 Basic Concept of Digital Data, Data Storage and Data Duplication

The importance of storage or memory capacity to computers has led to the development of many advance computers with the availability of much larger and

much less expensive data storage (Fenwick, 2018). Computers capable of handling data and receive them as 0s and 1s. A single bit can represent the numerical values 0 and 1 bits and are used to represent all type of digital data such as numbers, letters, sound or even represents colors (see Figure 2.4). Most modern computers today address consist 8-bit of 1 byte and a process known as digitization which is used to convert any information into the discrete unit of data as shown in Figure 2.5.

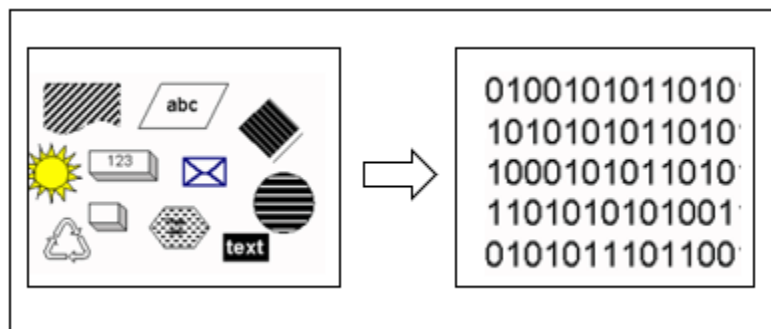


Figure 2.4 Data Representation in Digital Data Format

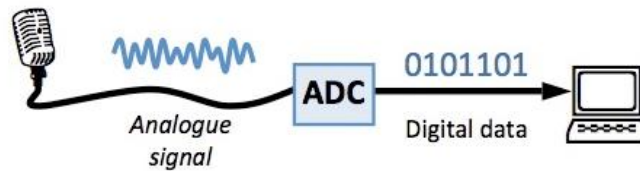


Figure 2.5 Conversion of Analog Signal to Digital Data

Longley et al. (2005) visualized data storage in real life as a car bonnet that full of passengers' items. Meanwhile, the car represents the data storage or database, and the passengers' items represent the data. Same as in real life, we tend to reduce the item size or the number of items so that all of the items can be stored in the bonnet. The time taken to get to the destination can also be reduced as lesser time is needed to store and arrange the items. The same abstraction occurs in data storage. Data will not be stored once the data size exceeds the storage size. In addition, bigger data size will cause a delay in transferring the data from the source to the target destination.

Indirectly, the relation between bits, bytes reflects the size of data. A bigger data size consists more bits compared to smaller data size and this affects the duration of data transfer and data duplication.



Figure 2.6 Abstraction of Data Storage Concept in Real Life

Inspired by data storage concept, this research understands that the duplication process will face the same problem which is the delay during data transfer. To explore more in details, previous work is studied in order to understand the state of the art of data transfer and data storage before the implementation of an enhanced automated data duplication model.

Generally, duplication is a process that is done by copying each data bit or bytes present from the source (master device) to the destination (targeted device) including the unused memory region or can simply define by a data transfer process from one location to another location to produce multiple copies from one source (Mazonka, 2009). In layman term, each data bit and empty region of the data in data storage is copied. Duplication process can be represented by basic data transfer cycle works by first defining the address located in the memory or I/O space. An address is defined in two operations; (i) when a transfer is to occur, (ii) after the process of defining equal length to transfer the data. Each half of the operation (defining the address and data

transfer) takes 100ns, or 200ns to complete. This process is repeated until the entire block of data is transferred (see Figure 2.7).

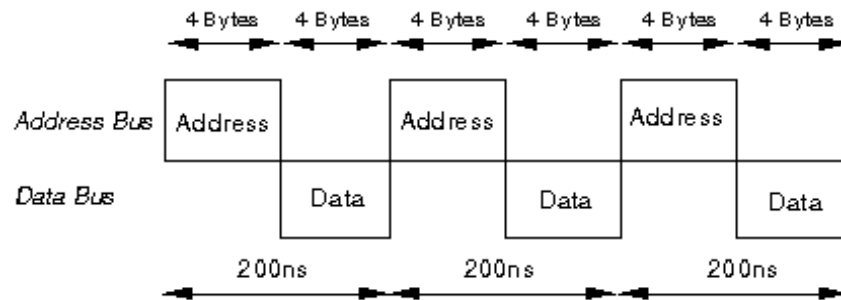


Figure 2.7 Basic Data Transfer Process (Barr & Rewini, 2005).

There are many examples of duplication works that have been done in the past (Labský, 2018; Aldaej et al., 2017; Shah et al., 2014; Sawant & Shah, 2013; Rashid & Khan, 2012; Tiwari et al., 2011; Panichprecha et al., 2011; Kim, 2011; Agus, 2010). Most of the works may come with different keywords such as data transfer, replication or duplication but the goal to create multiple copies for recovery and data backup is still the same. The review will focus on past work duplication strategies and is organized based on database replication, hard disk duplication, and flash duplication. Lastly, suggestions on several other ways to improve data transfer are also discussed. The summary of duplication past work is shown in Table 2.1.

Table 2.1 A Summary of the Data Duplication Techniques

Author(s)	Database Replication	Hard- disk duplication	Flash Duplication	Other Method
Tiwari et al. (2011)	/			
Mazilu (2010)	/			
Panichprecha et al. (2011)		/		
Aldaej et al. (2017)		/		
Shah et al. (2014)		/	/	
Labsky (2018)			/	
Agus (2010)			/	
Rashid & Khan (2016)			/	
Sawant & Shah (2013)			/	
Kim (2011)			/	
Nagwanshi et al. (2015)				/

2.3.1 Database Replication

Replication represents the process of sharing information to ensure consistency between redundant resources, such as software or hardware components, to improve reliability, fault-tolerance, or accessibility (Mazilu, 2010) which is classified into active and passive replication. Active replication happens when the same request is processed at every replicated instance and passive replication is when each request is processed on a single replica and then, its state is transferred to the other replicas (Mazilu, 2010).

Database replication is the process of creating and maintaining multiple instances of the same database and the process of sharing data or database design changes between databases in different locations without having to copy the entire database (Tiwari et al., 2011). In most implementations of database replication, one database server maintains the master copy of the database and the additional database servers maintain slave copies of the database. The two or more copies of a single database remain synchronized. The original database is called a design master and

each copy of the database is called a replica. Database replication system architecture is illustrated in Figure 2.8.

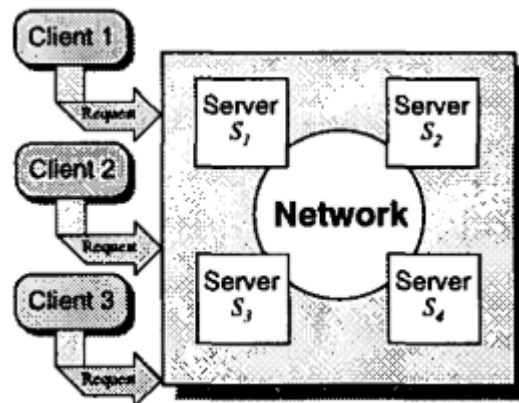


Figure 2.8 Database Replication Architecture (Wiesmann et al., 2000)

Database writes are sent to the master database server and are then replicated by the replicas. Despite the convenience of data replication still, drawbacks exist like replicas being frequently updated and may lose any historical state where backup saves a copy of data unchanged over time (Tiwari et al., 2011). Overall, the general data transfer of the duplication process by creating multiple copies approaches are still the same where the design master is the master and the replica represents the slaves.

Database replication can be performed in at least three different ways: snapshot, merging, and transactional replication. Snapshot replication happens when data on one database server is plainly copied to another database server, or to another database on the same server. Merging replication is when data from two or more databases are combined into a single database. Lastly, transactional replication is when users obtain complete initial copies of the database and then, obtain periodic updates as data changes (Mazilu, 2010). Even though the solution given is in a different environment where database replication has been extensively studied in the context of

distributed database systems, the duplication concept still can be studied and adopted in this research study.

2.3.2 Hard Disk Duplication

Hard disk duplication is a process of making copies of the data stored in a hard disk (Panichprecha et al., 2011). Such process is commonly used in data backup to save the original copy. Data duplication setting up multiple computers that have the same configuration and making copies of digital data for digital forensics. Just like any flash application, the common approach of hard disk duplication is creating duplicated data in a bit-by-bit manner where the entire disk even empty spaces are copied. Hard disk duplication or known as disc cloning process for creating an exact copy is shown in Figure 2.9.

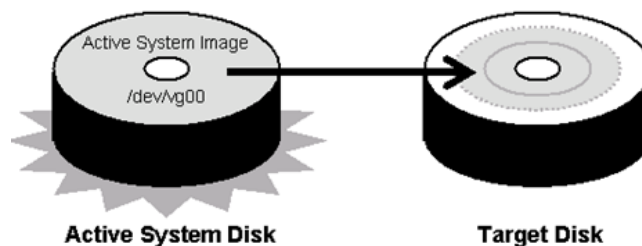


Figure 2.9 Disk Cloning Process

Due to the sequential bit-by-bit methodology used, it takes a long time to duplicate a large disk (Panichprecha et al., 2011). Panichprecha et al. (2011) performed hard disk duplication of 1GB of document data with duration approximately 25.88s. The method used in the research does not provide detailed explanations pertaining to the implementation of the hard disk duplication.