Tribhuvan University Institute of Science and Technology

Quantitative Evaluation of Garbage Collector Algorithms in JVM

A Dissertation

Submitted to The Central Department of Computer Science and Information Technology, Institute of Science and Technology Tribhuvan University

> In Partial Fulfillment of the Requirements for the degree of Master of Science in Computer Science and Information Technology

> > By Ambika Kumari December 2012

Under the Supervision of Lecturer Mr. Arjun Singh Saud

Tribhuvan University Institute of Science and Technology Central Department of Computer Science and Information Technology

Date.....

Recommendation

I hereby recommend that the dissertation prepared under my supervision by **Mrs. Ambika Kumari** entitled **"Quantitative Evaluation of Garbage Collector Algorithms in JVM**" be accepted as fulfilling in part requirements for the degree of Masters of Science. In my best knowledge this is an original work in computer science.

Lecturer

Mr. Arjun Singh Saud Central Department of Computer Science And Information Technology

Tribhuvan University

Institute of Science and Technology

Central Department of Computer Science and Information Technology

We certify that we have read this dissertation work and in our opinion it is satisfactory in the scope and quality as a dissertation in the partial fulfillment for the requirement of Master of Science in Computer Science and Information Technology.

Evaluation Committee

Mr. Nawaraj Poudel Asst. Professor Head, Central Department of Computer Science and Information Technology (Acting) Tribhuvan University

Mr. Arjun Singh Saud Lecturer Computer Science and Information Technology (Supervisor)

(External Examiner)

(Internal Examiner)

Date: -----

ABSTRACT

JVM performs the function of allocating memory as objects are created and freeing when they are no longer needed because different operating systems and hardware platform vary in the way they manage the memory. In java programming language, garbage collector automatically manage the objects generated by the keyword new inside Java Virtual Machine unlike in C and C++ programming language where the objects created are managed by the keywords free or delete. Garbage Collectors are the programs that reclaim the memory occupied by the objects that are no longer in use by the main program. In java platform, memory is automatically managed by the garbage collector that reduces the unnecessary burden of explicitly managing the memory for developers. The dissertation shows how the different garbage collectors perform the garbage collector in java. The dissertation also compares the three different types of garbage collectors algorithms namely, Mark Sweep algorithm, Concurrent Mark Sweep collector and Garbage First Garbage collector (G1GC) on the basis of throughput, pause time and footprint. After performing different tests, G1GC is a good option, if a computer (generally server) has powerful CPU and RAM but for a computer with less powerful CPU and powerful RAM, then CMS holds the edge over G1GC.

ACKNOWLEDGEMENT

The Dissertation has been prepared under Central Department of Computer Science and Information Technology (Tribhuvan University), Kirtipur and the study was done for the "Quantitative Evaluation of Garbage Collector algorithms in JVM". I would like to express heartfelt regards to my respected teacher and thesis supervisor, Mr. Arjun Singh Saud, lecturer of Central Department of Computer Science and Information Technology for his full cooperation, support, inspiration and guidelines throughout the preparation of this thesis work.

I would like to extend my deep gratitude towards respected Head of Department of Computer Science and Information Technology; Mr. Nawaraj Poudel for his encouragement, full support and kind help.

I am grateful and thankful to all the respected teachers of CDCSIT, TU, for granting me broad knowledge and inspirations within the time period of two years.

My special thanks go to my parents and family members for their blessings, encouraging words and always being with me in every step throughout preparation of my thesis work. I am thankful to my colleagues and also to those who were directly or indirectly involved in making this thesis a success.

There may be some errors in my dissertation, so any suggestion regarding the mistake of this dissertation will be always welcomed.

ABBREVIATIONS

CPU	Central Processing Unit
CMS	Concurrent Mark Sweep
DFS	Depth First Search
GB	Giga Byte
GC	Garbage Collector
GC	Garbage Collection
G1GC	Garbage First Garbage Collector
GHz	Giga Hertz
JDK	Java Development Kit
JVM	Java Virtual Machine
KB	Kilo Byte
MHz	Mega Hertz
RS	Remembered Set

TABLE OF CONTENTS

ABSTRACTi		
ACKNOWLEDGEMENTii		
ABBREVIATIONiii		
TABLE OF CONTENTS iv		
LIST OF FIGURESvii		
LIST OF TABLESviii		
CHAPTER ONE1		
BACKGROUND AND INTRODUCTION1		
1.1. Background1		
1.1.1 Java Virtual Machine JVM1		
1.1.2 Memory Management		
1.1.3 Memory Fragmentation and Compaction4		
1.1.4 Garbage Collector concept5		
1.1.4.1. Garbage Collection and the java platform memory model5		
1.1.4.2. Characteristics of Garbage Collector		
1.1.4.3. Design Choices7		
1.1.5. Java Heap Description		
1.2. Introduction		
1.2.1. Problem Definition		
1.2.2. Objectives of the study10		
1.3. Performance Matrix10		
1.4. Motivation10		
1.5. Organization of the study11		
CHAPTER TWO12		
RESEARCH METHODOLOGY AND LITERATURE REVIEW12		

2.1.	Literature Review	12
2.2.	Research Methodology	15
CHAF	PTER THREE	17
DESC	CRIPTION OF GARBAGE COLLECTION ALGORITHMS	17
3.1.	Mark and Sweep Algorithm	17
3.2.	Concurrent Mark Sweep garbage Collector	18
3.3.	Garbage First Garbage Collector	20
CHAF	PTER FOUR	23
DESIG	GN AND IMPLEMENTATION	23
4.1.	Programming Language	23
4.2.	Tools Used	23
	4.2.1. PrintGCDetails	23
	4.2.2. PrintGCTimeStamps	23
	4.2.3. Xloggc	23
	4.2.4. VisualVM	24
4.3.	Experimental Setup	24
4.4.	Test Case Design	25
	4.4.1. Test Case1	25
	4.4.2. Test Case2	26
	4.4.3. Test Case3	27
CHAF	PTER FIVE	29
DATA	A COLLECTION AND ANALYSIS	29
5.1.	Memory and CPU Usage for Test Case1	29
5.2.	CPU and Memory Usage for Test Case2	31
5.3.	CPU and Memory Usage for Test Case 3	33

5.4.	Summarizing GC Performance	.35
CHAP	TER SIX	.37
CONC	LUSION AND RECOMMENDATION	37
6.1.	Conclusion	.37
6.2.	Further Recommendation	38
REFE	RENCES	.39
APPEN	NDICES	.42