



# **Comparative Study and measurement of performance of Serial, Parallel and Concurrent Mark Sweep Collectors algorithm on JVM**

## **Dissertation**

### **Submitted To**

Central Department of Computer Science & Information Technology

**Tribhuvan University**

Kirtipur, Kathmandu

Nepal

In Partial Fulfillment of the Requirements for the Degree of Master of Science  
in Computer Science & Information Technology

Submitted By:-

**Saroj Bhatta**

August, 2014

Supervisor

**Mr. Arjun Singh Saud**



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

**Student's Declaration**

I hereby declare that I am the only author of this work and that no sources other than the listed here have been used in this work.

.....

**Saroj Bhatta**

**Date: August, 2014**



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

**Supervisor's Recommendation**

I hereby recommend that the dissertation prepared under my supervision by **Mr. Saroj Bhatta** entitled “**Comparative Study and Measurement of Performance of Serial, Parallel, and Concurrent Mark Sweep Collectors on JVM** ” be accepted as in fulfilling partial requirement for the completion of Masters Degree of Science in Computer Science & Information Technology.

-----  
**Mr. Arjun Singh Saud**

Lecturer,

Central Department of Computer Science and Information Technology,

Institute of Science and Technology,

Kirtipur, Kathmandu, Nepal

**Date: August, 2014**



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

**LETTER OF APPROVAL**

We certify that we have read this dissertation work and in our opinion it is appreciable for the scope and quality as a dissertation in the partial fulfillment of the requirements of Masters Degree of Science in Computer Science & Information Technology.

**Evaluation Committee**

---

**Asst. Prof. Nawaraj Paudel**  
**Head of Department**  
Central Department of Computer Science  
& Information Technology  
Tribhuvan University  
Kirtipur

---

**Mr. Arjun Singh Saud**  
**Lecturer**  
Central Department of Computer Science  
and Information Technology  
(Supervisor)

---

**(External Examiner)**

---

**(Internal Examiner)**

**Date: August, 2014**

## ABSTRACT

Java Virtual Machine (JVM) works as a software module that executes java application bytecode and translates the byte code into hardware and operating system-specific instructions. By doing so, the JVM enables java program to be executed in different environments. It also performs the function of memory allocation as objects are created and freeing when they are no longer needed. In java programming language, garbage collector automatically manage the objects generated by the keyword new inside Java Virtual machine. But in other programming languages like C/C++ objects created are managed by free or delete.

Garbage collection is the process of automatic storage reclamation in which those objects which are no longer referenced from any live objects or from program are collected. One of the advantages of garbage collection is that the garbage collection ensures program integrity. It is an important part of java security strategy. Garbage collectors are becoming the essential part of compilers. Most of the high level languages like java and C# have incorporated garbage collectors for automatic memory management. This study compares three garbage collectors (Serial, Parallel and Concurrent Mark Sweep). After performing different tests, this dissertation work showed that Serial GC is better choice if we have to use single threaded programs and parallel GC is better choice in case of multithreaded.

**Keywords:** *Serial garbage Collector, Parallel Garbage Collector, Concurrent mark Sweep Garbage Collector, Java Virtual Machine, Java Heap Memory.*

## Acknowledgement

I would like to express my gratitude to all the people who supported and accompanied me during the preparation of this dissertation “**Comparative Study and measurement of performance of Serial, Parallel, and Concurrent Mark Sweep Collectors algorithm on JVM**”. This research work has been performed under Central Department of Computer Science and Information Technology (*Tribhuvan University*), Kirtipur. I am very grateful to my department for giving me an enthusiastic support.

Firstly, I would like to deeply extend my heartfelt acknowledgement to my respected teacher and dissertation supervisor **Mr. Arjun Singh Saud**, for his wholehearted cooperation, encouragement and strong guidelines throughout the preparation of this study. He is the one who listened to all my problems I faced during this thesis and showed me the way to overcome them.

Most importantly I would like to thank to respected Head of Department of Central Department of Computer Science and Information Technology, Asst. Prof. Nawaraj Paudel for his kind support, help and constructive suggestions. I am very much grateful and thankful to all the respected teachers Prof. Dr. Shashidharam Joshi, Prof. Dr. Subarna Sakya, Prof. Sudarsan Karanjit, Mr. Min Bahadur Khatai, Mr. Bishnu Gautam, Mr. Jagdish Bhatta, Mr. Dheeraj Kedar Pandey, Mr. Sarbin Sayami, Mrs. Lalita Sthapit, Mr. Yoga Raj Joshi and Mr. Bikash Balami of CDCSIT, TU, for providing me such a broad knowledge and inspirations.

All my class fellows are worthy of my gratefulness for their direct or indirect support in completion of my dissertation. Finally, I would like to thank my friends Mr. Dabal Singh Mahara and Mr. Niranjan Kathayat for their kind co-operation during my work.

I have done my best to complete this research work. Suggestions from the readers are always welcomed, which will improve this work.

Saroj Bhatta

August, 2014

## ABBREVIATIONS

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

# TABLE OF CONTENTS

	<b>Pages</b>
ABSTRACT.....	i
ACKNOWLEDGEMENT.....	ii
ABBREVIATION.....	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES.....	vii
LIST OF TABLES.....	viii
CHAPTER ONE.....	1
<b>BACKGROUND AND INTRODUCTION.....</b>	<b>1</b>
1.1 Background.....	1
1.1.1 Java Virtual Machine.....	1
1.1.2 Explicit vs. Automatic Memory Management.....	3
1.1.3 Memory Fragmentation and Compaction.....	4
1.1.4 Garbage Collection.....	6
1.1.4.1 Desirable Garbage Collector Characteristics.....	7
1.1.5 Java Heap Memory.....	8
1.2 Introduction.....	10
1.2.1 Problem Definition.....	10
1.2.2 Objectives.....	11
1.3 Performance Metrics.....	11
1.4 Motivation.....	11
1.5 Structure of Thesis.....	12
<b>CHAPTER TWO.....</b>	<b>13</b>
Research Methodology and Literature Review.....	13
2.1 Literature Review.....	13
2.2 Research Methodology.....	16
<b>CHAPTER THREE.....</b>	<b>17</b>
GARBAGE COLLECTION ALGORITHMS.....	17
3.1 Serial Garbage Collector.....	17



3.2 Parallel Garbage Collector.....	19
3.3 Concurrent Mark Sweep Garbage Collector.....	20
<b>CHAPTER FOUR.....</b>	<b>22</b>
<b>DESIGN AND IMPLEMENTATION.....</b>	<b>22</b>
4.1 Programming Language.....	22
4.2 Tools Used.....	22
4.2.1 VisualVM.....	22
4.2.2 Javac.....	22
4.2.3Java.....	23
4.3 Data Structures.....	24
4.3.1 Integer Object.....	24
4.3.2 Vector.....	24
4.3.3 Strings.....	24
4.4 Programming Language Features.....	24
4.4.1 Multithreading.....	24
4.4.2 Generics.....	25
4.5 Experimental Setup.....	26
4.6 Test Case Design.....	26
4.6.1 Test Case1.....	27
4.6.2 Test Case2.....	28
4.6.3 Test Case3.....	29
4.6.4 Test Case4.....	31
<b>CHAPTER FIVE.....</b>	<b>33</b>
<b>DATA COLLECTION AND ANALYSIS.....</b>	<b>33</b>
5.1 CPU and Memory Usage for Test Case1.....	33
5.2 CPU and Memory Usage for Test Case2.....	35
5.3 CPU and Memory Usage for Test Case3.....	37
5.4 CPU and Memory Usage for Test Case4.....	39
5.5 Summarizing GC Performance.....	40

<b>CHAPTER SIX</b> .....	43
CONCLUSION AND RECOMMENDATION.....	43
6.1 Conclusion.....	43
6.2 Further Recommendation.....	44
<b>REFERENCES</b> .....	45

## LIST OF FIGURES

Figure 1.1: Compiling Java Program.....	1
Figure 1.2: JVM Architecture.....	2
Figure 1.3: Memory Fragmentation.....	5
Figure 1.4: Memory Compaction.....	5
Figure 1.5: Java Heap Memory.....	9
Figure 3.1: Serial Garbage Collector.....	17
Figure 3.2: Serial Young Generation Collection.....	18
Figure 3.3: Parallel Garbage Collector.....	19
Figure 3.4: Concurrent Mark Sweep Collector.....	21
Figure 5.1: CPU and Memory Usage with Serial GC.....	33
Figure 5.2: CPU and Memory Usage with parallel GC.....	33
Figure 5.3: CPU and Memory Usage with CMS GC.....	34
Figure 5.4: CPU and Memory Usage with Serial GC.....	35
Figure 5.5: CPU and Memory Usage with Parallel GC.....	35
Figure 5.6: CPU and memory Usage with CMS GC.....	36
Figure 5.7: CPU and memory Usage with Serial GC.....	37
Figure 5.8: CPU and memory Usage with Parallel GC.....	37
Figure 5.9: CPU and memory Usage with CMS GC.....	38
Figure 5.10: CPU and memory Usage with Serial GC.....	39
Figure 5.11: CPU and memory Usage with Parallel GC.....	39
Figure 5.1 2: CPU and memory Usage with CMS GC.....	40
Figure 5.14: Graph for Table 5.13.....	41
Figure 5.16: Graph for Table 5.15.....	42

## LIST OF TABLES

Table 5.13: Summary of maximum CPU and Memory Usage.....	41
Table 5.15: Average CPU and Memory Usage.....	42