

## **HADOOP CONFIGURATION TUNING FOR PERFORMANCE OPTIMIZATION**

By

Christian  
11302024



SWISS GERMAN UNIVERSITY  
The Prominence Tower  
Jalan Jalur Sutera Barat no. 15, Alam Sutera  
Tangerang, Banten 15143 - Indonesia

August 2017

**Revision after the Thesis Defense on 19 July 2017**

## STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in the thesis.

Christian

Student

Approved by:

Dipl.-Inf. Kho I Eng

Date

Thesis Advisor

Date

Ir. Heru Purnomo Ipung, M.Eng

Thesis Co-Advisor

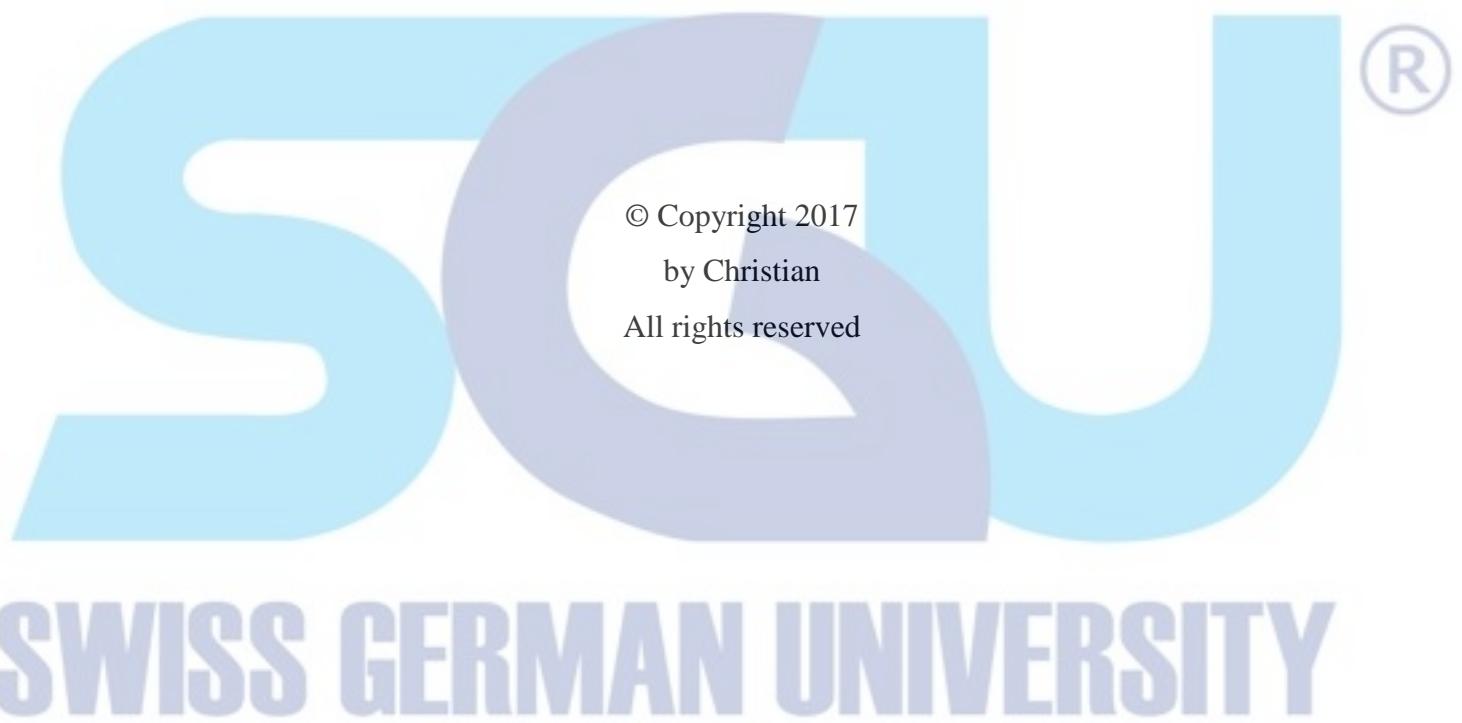
Date

Dr. Ir. Gembong Baskoro, M.Sc

Dean

Date

Christian



## DEDICATION

I dedicate this work to myself and Swiss German University.



## ACKNOWLEDGEMENTS

First of all, I would like to thank to my advisors Kho I Eng and Heru Purnomo Ipung for giving me the opportunity to work on this research and advising me each step of the way. Furthermore, I would like to thank to all my lecturers for any recommendations and guidance regarding my research. Finally, I give special thanks to my parents for any assistance and support to complete this research.



## ABSTRACT

### HADOOP CONFIGURATION TUNING FOR PERFORMANCE OPTIMIZATION

By

Christian  
Dipl.-Inf. Kho I Eng, Advisor  
Ir. Heru Purnomo Ipung, M.Eng, Co-Advisor

SWISS GERMAN UNIVERSITY



Configuration parameter tuning is an essential part of the implementation of Hadoop clusters. Each parameter in a configuration plays a role that impacts the overall performance of the cluster. However, we need to learn the characteristics of said parameter and understand the impact in hardware utilization in order to achieve optimal configuration.

Several configuration changes includes mapper count, reduces count, HDFS block size, and MapReduce compression codec selection. The experiment also includes the rebuilding Hadoop source to produce the 64bit version over the 32bit version release from Apache.

To prove any performance gain, we performed benchmark test every experiment we conducted. The benchmark includes TeraGen, TeraSort, and TeraValidate. We used 1GB, 10GB, 50GB of data size that we generated initially using TeraGen which will be used throughout all benchmarks. TeraSort is the program that runs the benchmark, we measure the time needed to complete the sort of the set of data and the CPU utilization during the benchmark. TeraValidate only validates the output of TeraSort to ensure that the output is correct.

From the experiments that we conducted, we have observed significant performance improvements. However, the results may vary between different cluster configuration.

*Keywords:* apache hadoop, computer cluster, configuration tuning, terasort benchmark

---

Christian

## TABLE OF CONTENTS

STATEMENT BY THE AUTHOR .....	2
DEDICATION .....	4
ACKNOWLEDGEMENTS .....	5
ABSTRACT .....	6
TABLE OF CONTENTS .....	7
LIST OF FIGURES .....	10
LIST OF TABLES .....	11
CHAPTER 1 - INTRODUCTION .....	12
1.1 Background .....	12
1.2 Research Purpose .....	13
1.3 Problem Identification .....	13
1.4 Research Limitation .....	13
1.5 Research Scope .....	14
1.6 Research Questions .....	14
1.7 Hypothesis.....	14
1.8   Significance of Study .....	15
1.9 Document Structure .....	15
CHAPTER 2 - LITERATURE REVIEW .....	16
2.1 Cluster Computing .....	16
2.2 High Performance Computing .....	16
2.3 Apache Hadoop Framework .....	17
2.3.1 Apache Hadoop .....	17
2.3.2 Hadoop MapReduce.....	17
2.3.3 Hadoop YARN.....	17
2.4 Hadoop Distributed File System .....	18
2.4.1 Namenode .....	18
2.4.2 Datanode .....	18
2.5 Compression .....	19
2.5.1 GZip .....	19
2.5.2 BZip2 .....	19
2.5.3 Snappy.....	19
2.6 Related Works.....	20

SWINBURNE UNIVERSITY



2.6.1 A Framework for Performance Analysis and Tuning in Hadoop Based Clusters .....	20
2.6.2 Hadoop Performance Tuning Guide .....	20
2.6.3 Towards A Scalable HDFS Architecture .....	21
2.6.4 Workload Analysis, Implications, and Optimization on a Production Hadoop Cluster .....	21
CHAPTER 3 - RESEARCH METHOD .....	22
3.1 Methodology .....	22
3.2 Experiment Setup .....	22
3.2.1 Cluster Design .....	22
3.2.2 Hadoop Installation .....	25
3.3 Monitoring and Analysis Tools .....	26
3.3.1 Time (Linux utility command) .....	26
3.3.2 Zabbix .....	26
3.4 Benchmarking Tools .....	26
3.4.1 TeraGen .....	26
3.4.2 TeraSort .....	27
3.4.3 TeraValidate .....	27
3.5 Experiments .....	28
3.5.1 Initial benchmark .....	28
3.5.2 Mappers and Reduces .....	28
3.5.3 Native Hadoop Library .....	29
3.5.4 HDFS Block size .....	30
3.5.5 Compression .....	30
CHAPTER 4 - RESULTS AND DISCUSSION .....	31
4.1 Initial Benchmark Result .....	31
4.1.1 Scaling up .....	31
4.1.2 Benchmark baseline .....	32
4.2 Mappers and Reduces balancing .....	33
4.3 Native Hadoop Library .....	37
4.4 HDFS Blocksize tuning .....	38
4.5 Compression .....	39
CHAPTER 5 - CONCLUSIONS .....	40
5.1 Conclusions .....	40
5.2 Future works .....	42

5.2.1 Further experiments .....	42
5.2.2 Hadoop 3 .....	42
5.2.3 SSD Storage .....	42
REFERENCES .....	43
APPENDIX .....	45
Appendix 1: Minimal Hadoop cluster configuration files .....	45
hadoop-env.sh .....	45
core-site.xml .....	46
hdfs-site.xml .....	46
yarn-site.xml .....	47
mapred-site.xml .....	47
slaves .....	47
Appendix 2: Mappers and Reduces configuration files .....	48
mapred-site.xml .....	48
Appendix 3: HDFS Block size configuration files .....	49
hdfs-site.xml .....	49
Appendix 4: Compression configuration files .....	50
mapred-site.xml .....	50
CURRICULUM VITAE .....	51

SWISS GERMAN UNIVERSITY