

A LIGHTWEIGHT CACHING FRAMEWORK

Sameera Dinusha Nelson

148232M

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2018

A LIGHTWEIGHT CACHING FRAMEWORK

Sameera Dinusha Nelson

148232M

Dissertation submitted in partial fulfillment of the requirements for the degree Master
of Science in Computer Science and Engineering

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2018

DECLARATION

I declare that this is my own work and this MSc Research Project does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.

.....

S.D. Nelson

.....

Date

I certify that the declaration above by the candidate is true to the best of my knowledge and that this project report is acceptable for evaluation for the CS6997 MSc Research Project.

.....

Dr. Indika Perera

.....

Date

ACKNOWLEDGEMENTS

I would like to convey my heartfelt gratitude to Dr. Indika Perera, my supervisor, for the excellent supervision, support and advice given throughout to make this research a success. I would also like to express my deep gratitude to Mr. Aruna Dissanayake, Head of Engineering, Persistent Software Lanka Pvt Ltd for helping to plant this project concept. Moreover, I would like to convey my sincere appreciation to the staff of the Computer Science and Engineering, University of Moratuwa for their generous support towards the completion of this task. I would also like to convey my appreciation to my family for their continuous support and motivation given to make this research project a success.

ABSTRACT

Node management techniques especially designed for the modern systems are becoming more and more advanced and complex with the research involvement from the industry and institutions today. Almost all type industries with the computing systems are trending towards the distributed environments now. In this context node management plays a very significant and important role which helps to fulfill the main aspect of the distributed-ness. Clustering is the process of managing, maintaining and binding different nodes together sharing common set of configurations to work together for achieving a common goal.

In this project, we design and implement an approach towards a lightweight caching framework for a distributed environment. Here, we evaluate several protocols and choose WebSockets protocol for implementing the node management of the cluster. In our approach, each node maintains node server and node client set in a multithreaded environment to cater the node management. We have detailed out the design and implementation aspects on chapter 4. Then we demonstrate the performance achievements of the caching framework by using it on an application, having small to medium scale cluster. Then through a simulated use-case of the caching cluster framework, the system stability is monitored while up scaling the system step by step. Finally, the above-mentioned use case is extended to a simulated experimental evaluation making a comparison with a reference framework. For that experiment, a reference framework which based on TCP protocol is implemented with the help of Netty framework. We have concluded by highlighting the significance of our caching cluster framework by the outstanding behaviors and features such as efficiency, lightweight-ness, and scalability, stability on scaling and very low system overhead.

TABLE OF CONTENTS

DECLARATION.....	iii
ACKNOWLEDGEMENTS.....	iv
ABSTRACT	v
CHAPTER 1	1
1 INTRODUCTION.....	1
1.1 Overview.....	2
1.2 Scope	3
1.3 Issues.....	3
1.4 Cluster Management for Caching.....	4
1.4.1 Consistency.....	4
1.4.2 Static Mode	5
1.4.3 Dynamic Mode.....	5
1.5 Configuration Management	5
1.6 Fault Tolerance	6
1.7 Distributed Caching	6
1.8 Research Problem.....	6
1.9 Motivation	7
1.10 Objectives	7
CHAPTER 2	8
2 LITERATURE REVIEW	8
2.1 Overview.....	9
2.2 History and Background.....	9
2.3 Related Work	9
2.4 Group Communication Protocols for Node Management.....	10

2.4.1	Multicasting Protocols.....	10
2.4.2	Gossip Protocols.....	12
2.5	Client- Server Protocols for Node Management.....	15
2.5.1	HTTP Protocol.....	15
2.5.2	WebSockets Protocol.....	16
2.6	Caching, Distribution and Replication	18
	CHAPTER 3	19
	3 METHODOLOGY	19
3.1	Overview.....	20
3.2	Cluster Membership.....	22
3.3	Caching.....	22
3.4	Cluster Setup Steps.....	23
	CHAPTER 4	25
	4 DESIGN AND IMPLEMENTATION	25
4.1	Tools Used for the implementation work	26
4.1.1	Netty Framework.....	26
4.1.2	WebSockets for Cluster Management	28
4.1.3	Framework Thread Model	30
4.1.4	Message Model.....	32
4.1.5	Security measurements in the Framework.....	35
4.2	Node Implementation	36
4.3	Server Implementation	37
4.4	Client Implementation	37
4.5	Node Registry	38
4.6	Managing Cluster Memberships.....	39
4.6.1	Phase 1: Cluster Setup	39

4.6.2	Phase 2: Cluster Sync up.....	41
4.6.3	Detecting Node Failures	42
4.6.4	Node Recovery	42
4.7	Caching	42
	CHAPTER 5	44
	5 EVALUATION AND RESULTS	44
5.1	Tests and Results: Using Cache Cluster Framework.....	45
5.1.1	Node Server Startup.....	45
5.1.2	Node Client Startup.....	46
5.1.3	Client Failure Sync up.....	46
5.1.4	Messaging	47
5.2	Assessment of the Advantages of Cache Framework.....	48
5.3	Feature Evaluation	49
5.3.1	Ease of usage	49
5.3.2	Lightweight-ness of the Framework	51
5.3.3	Scalability & Stability	54
5.3.4	Fault Tolerance.....	56
5.4	Performance Evaluation and Analysis	57
5.4.1	Application Use case.....	57
5.4.2	Experimental Evaluation	62
	CHAPTER 6	65
	6 CONCLUSION & FUTURE WORK.....	65
	REFERENCES	67