Optimistic Concurrency Control in a Distributed NameNode Architecture for Hadoop Distributed File System

Qi Qi

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Doctor Luís Manuel Antunes Veiga

Examination Committee

Chairperson:	Doctor Luís Eduardo Teixeira Rodrigues
Supervisor:	Doctor Luís Manuel Antunes Veiga
Member of the Committee:	Doctor Nuno Preguiça

September 2014

Acknowledgments

The work presented is delivered as final thesis report at Instituto Superior Técnico - IST (Lisbon, Portugal). It is in partial fulfillment of the European Master in Distributed Computing - EMDC program 2012-2014. Royal Institute of Technology - KTH (Stockholm, Sweden) is the coordinator for this Erasmus Mundus master program. The study track has been composed of a first two semesters at IST, 3rd semester at KTH, and for this work and 4th semester, a degree project in Computer Systems Laboratory at Swedish Institute of Computer Science - SICS (Stockholm, Sweden).

Special thanks to my advisor Dr. Jim Dowling for his support throughout the project. With more than ten years' professional industry experience, Jim is always patient to help. He's the cool guy who gives answers faster than Google and StackOverFlow.

Thanks to Salman Niazi and Mahmoud Ismail for all the practical help. Without them I might have to spend quite a long time studying the code base of the precedent work.

I'm also grateful to my supervisor Prof. Luís Antunes Veiga for his continuous support and encouragement. When I was in IST, I liked staying in the classroom after his class and chatted with him for a while. Veiga was like a big brother there taking care of us.

I would like to thank the good friends I met in Portugal and Sweden, who leveled me up during these two years. Without you guys, this journey wouldn't have been such a legendary in my life.

I am truly thankful to my family for nursing me with all their affections and love.

Last, special appreciation to this young man, Qi Qi, who always has the guts to go on any adventure in his life.

September 5, 2014, Stockholm

Qi Qi

Dedication

To my father, a man of integrity, who supports all my adventurous decisions so that I can live outside of the box.

Resumo

[To be added] Portuguese Abstract

Abstract

The *Hadoop Distributed File System* (HDFS) is the storage layer for Apache Hadoop ecosystem, persisting large data sets across multiple machines. However, the overall storage capacity is limited since the metadata is stored in-memory on a single server, called the *NameNode*. The heap size of the NameNode restricts the number of data files and addressable blocks persisted in the file system. (Shvachko 2010)

The *Hadoop Open Platform-as-a-service* (Hop) is an open platform-as-a-Service (PaaS) support of the Hadoop ecosystem on existing cloud platforms including Amazon Web Service and OpenStack. The storage layer of Hop, called the Hop-HDFS, is a highly available implementation of HDFS, based on storing the metadata in a distributed, in-memory, replicated database, called the *MySQL Cluster*. It aims to overcome the NameNode's limitation while maintaining the strong consistency semantics of HDFS so that applications written for HDFS can also run on Hop-HDFS without modifications.

Precedent thesis works have contributed for a transaction model for Hop-HDFS. From system-level coarse grained locking to row-level fine grained locking, the strong consistency semantics has been ensured, but the overall performance is restricted compared to the original HDFS.

In this thesis, we first analyze the limitation of HDFS NameNode implementation and provide an overview of Hop-HDFS illustrating how we overcome those problems. Then we give a systematic assessment on precedent works for Hop-HDFS comparing to HDFS, and we analyze the restriction when using pessimistic locking mechanism to ensure strong consistency semantics. Finally, as a proof of concept, we demonstrate and evaluate that how to improve the performance by designing a new model based on optimistic concurrency control with snapshot isolation. The correctness of this new model has been validated by ensuring that 300+ unit tests pass for Apache HDFS.

Palavras Chave
Keywords

Palavras Chave [To be corrected by native Portuguese speaker]

HDFS

MySQL Cluster

Controle de Concorrência

Snapshot Isolation

Transação

Vazão

Keywords

HDFS

MySQL Cluster

Concurrency Control

Snapshot Isolation

Transaction

Throughput

Index

List of Figures

List of Tables

I

Introduction and Background

1 Introduction

1.1 A

AAA

1.2 B

BBB

1.3 C

CCC

1.4 D

DDD

2 Background

2.1 A

AAA

2.2 B

BBB

2.3 C

CCC

2.4 D

DDD

II

Assessment in Hop-HDFS

3 Strong Consistency Semantics in Hop-HDFS

3.1 A

AAA

3.2 B

BBB

3.2.1 B1

BBB1

3.2.2 B2

BBB2

3.3 C

CCC

3.4 D

DDD

4 Systematic Assessment of Operation Performance in Hop-HDFS

Neque porro quisquam est qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit...

– Cerico

4.1 A

AAA

4.2 B

BBB

4.2.1 B1

BBB1

4.2.2 B2

BBB2

4.3 C

CCC

4.4 D

DDD

III
Solution

5 Design

5.1 A

AAA

5.2 B

BBB

5.2.1 B1

BBB1

5.2.2 B2

BBB2

5.3 C

CCC

5.4 D

DDD

6 Implementation

6.1 A

AAA

6.2 B

BBB

6.2.1 B1

BBB1

6.2.2 B2

BBB2

6.3 C

CCC

6.4 D

DDD

IV
Evaluation and Conclusion

7 Evaluation

7.1 A

AAA

7.2 B

BBB

7.2.1 B1

BBB1

7.2.2 B2

BBB2

7.3 C

CCC

7.4 D

DDD

8 Conclusion

8.1 A

AAA

8.2 B

BBB

8.2.1 B1

BBB1

8.2.2 B2

BBB2

8.3 C

CCC

8.4 D

DDD

Bibliography

Shvachko, K. V. (2010). Hdfs scalability: The limits to growth. *login 35*(2), 6–16.

V
Appendices

Apache Unit Testing