A SPECIFICATION OF A MULTI-VERSION STORAGE ENGINE USING TLA+

By

Goodwill T. Khoa

Student Number: 219080534

Supervisor:

Prof. J. Quenum

Department of Computing and Informatics

A thesis submitted in partial fulfillment of the requirements for the degree of BACHELOR OF COMPUTER SCIENCE Honours (Software Development)

at the

NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY (NUST) ,

Windhoek, Namibia.

(07 October 2022)

Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis entitled "**A SPECIFICATION OF A MULTI-VERSION STORAGE ENGINE USING TLA+**" written by **Goodwill T. Khoa**, (**Student Number: 219080534**), of NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY (NUST) , has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of Honours degree.

Signature: _____

Name of Supervisor: **Prof. J. Quenum**

Date: _____

Signature (HoD): _____

Date: _____

Signature (Dean): _____

Date: _____

Approval

It is certified that the contents and form of the thesis entitled "**A SPECIFICATION OF A MULTI-VERSION STORAGE ENGINE USING TLA+**" submitted by **Goodwill T. Khoa** have been found satisfactory for the requirement of the degree.

Supervisor: **Prof. J. Quenum**

Signature: _____

Date: _____

Committee Member 1: _

Signature: _____

Date: _____

Committee Member 2: _

Signature: _____

Date: _____

Committee Member 3: _

Signature: _____

Date: _____

Dedication

This thesis is dedicated to all the deserving children who do not have access to quality education especially young girls.

DECLARATION

I Goodwill T. Khoa, hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at Department of Computing and Informatics at the NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY (NUST) or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at the NAMIBIA UNIVERSITY OF SCIENCE AND TECHNOLOGY (NUST) or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Goodwill T. Khoa**

Signature: _____

Date: _____

Acknowledgments

To God Almighty for the strength to never give up and pursue all things towards His Kingdom and Glory. To my family who through all sleepless nights, never complained about my present-absence. My wife Lutopu and son Michael and daughter Lillian as well as newly adopted children.To my Work colleagues for patience and support, To Prof J. Quenum for patience and faith in me. Thank you all.

Goodwill T. Khoa

Contents

List of Figures

List of Tables

Abstract

It is well known that; within complex systems and concurrency, fault deduction and system failures is rated as the greatest risk factor amongst software developers. Even with various efforts to eliminate some of the expected errors and or bugs. Distributed Systems tend not to be for faint-hearted software developers. As a result this working document will focus on the ability to check the correctness of our system (a distributed storage engine) beyond testing. One such solution is to use the Temporal Logic of Actions (TLA) in the designing and developmental processes before actual coding takes place. Temporal Logic of Actions Plus (TLA+) is the specification language used throughout this research process. The specification is focused on the key operations of the storage engine: inserting, updating and deleting and reading data from either a distributed physical storage medium or a local physical storage medium.

Chapter 1

Introduction and Motivation

1.1 Introduction

Currently cloud computing and outsourcing is a general norm amongst developers. African, and Namibian software developers to be precise, finds themselves with a hot bill to swallow with regards to outsourcing storage, either for personal projects or cliental. It is with this in mind that a small and compact team from the Namibia University of Science and Technology embarked on the development of a storage engine, that aims to solve some of the basic but crucial architectural aspects of the development and deployment process within our country and continent as a whole. According to the Wikipedia contributors (2022), "A database engine (or storage engine) is the underlying software component that a database management system (DBMS) uses to create, read, update and delete (CRUD) data from a database."

Storage engines can be classified as either as transactional or non-transactional, and categorized as local or distributed. The storage engine under the microscope, is a transactional and distributed storage engine which aims to have high availability with high scalability as well as strong consistency and can be used for data analysis. A brief highlight of the level of complexity within the distributed storage engine includes; Conversions, Version Control, Synchronizations, Concurrent Read-Write operations, Transaction Support, Backups and Quality Assurance, High availability and reduced time complexity in terms of overall performance. A similar dilemma was faced by Senior Developer at Amazon Web Service (AWS). It is with this in mind that the researcher intends to utilize TLA+ a formal specification method to specify the key operations

of the multi-versioned storage engine. According to Lamport (2003). "The language of PlusCal expressions is TLA+, a high-level specification language based on set theory and first order logic. The researcher assumes that the reader has a background in Mathematics with specific reference to the Set theory and Set Notation as the main translation of the derived algorithm will be denoted in Set Theory Notation. Below are small snippets of what to expect through out the document.

- Actions: Å, M, M1, M2
- Predicates: P, Init
- Variables: Path, VersionNumber, Pointer
- Primed variables: rootPath‘ PreVersion‘, PPoniter‘
- States: s, s‘
- State function: f
- Behavior: <s0, s1,..., sn>
- Values: Data items, e.g. Integers, constants, String, Boolean
- Semantics: [f],[Pointer], K[VersionNumber], rootPath[f]
- Formulas: F, G,
- Operators: $\Box, \diamond, \neq, \vee, \wedge Quantifiers : \forall, \exists$

- Symbols: $\triangleq, <, >, \backsim, [,], (,), =, \equiv, \cdots$

1.2 Problem Statement and Contribution

(Alvarez, 2020) "As can be inferred, the selection, implementation, operation and maintenance of a System for Automatic Diagnosis of Failures is not a simple task, requiring at each stage, care so that the result provided by the system, after its implementation, is within the one initially specified.” Is it therefore possible to detect system faults within a storage engine and rectify such faults to increase the quality and effectiveness of the storage engine, by focusing on the reliability and correctness of the storage engine, given its complexity? Through experience, the researcher has come to the understanding that

all software design, algorithms and coding processes contains one or more errors, some which are easy to detect while others not.

If there was a method to avoid errors in the core system processes and development stages, what would that look like? And how effective would it be? This dilemma lead Professor Leslie Lamport to seek and develop an alternative approach to solving complex systems through formal methods. "When I developed TLA, I realized that, for the first time, I had a formalism that really was completely formal, so formal that mechanically checking TLA proofs should be straightforward." (Urban Engberg, Leslie Lamport, Peter Grønning, 1992).

The purpose of this research is therefore to actively use TLA+ as a formal method in the developing process of the storage engine to specify and evaluate the error reduction rate, by focusing on the safety and correctness of the system of the storage engine. Through the use of TLA+ the researcher aims to formally specify the critical components and or operations of the system in order to verify its property against an implementation.

1.3 RESEARCH OBJECTIVES

The researcher aims to apply the TLA+ to the CRUD key operations of the storage engine. To verify the properties of the implementation against the specification, during the development and testing phase of the storage engine. With the acronym CRUD the researcher aims to:

1. C = Create: This will be the write action and subsequent write operations
2. R = Read: This will be the read request sent by a client
3. U = Update: This will be a write new version operation
4. D = Soft Delete: This will be the stop write operation on a given version tree only.

With reference to the above the researcher intends to:

(a) Specify the CRUD operations of the storage engine in TLA+.

(b) Evaluate the design using the specification

(c) Verify an existing implementation against the specification

Chapter 2

Literature Review

Here you review the state of the art relevant to your thesis proposal. The idea is to present the major ideas in the state of the art right up to, but not including, your own personal brilliant ideas.

Critical analysis and comparisons should be made by pointing out the weakness of existing solutions and strengths of your proposal. You organize this section by idea, and not by author or by publication.

In certain situations, a background of the underlying concepts is required for better understanding of the research problem and also to improve the flow of the thesis. This could either be made an introductory part of this section or separately written in a prior section.

Reference Type	**Citation**
Article	[**Hayes1993**]
Book	[**Lamport_2005**]
InProceedings	[**Nicholls_1987**]
InCollection	[**Lund_2019**]
PhD Dissertation	[**Konnov_2019**]
Masters Thesis	[**Lamport_2005**]
Technical Report	[**Engberg1993**]
Misc	[**Hayes1993**]

Table 2.1: Citation Styles.

Chapter 3

Storage Engine

A brief description of the storage engine an example use, replication processes and operations specific to this document

3.1 Overview

The Storage engine itself is not is not what this research paper is focused. However This paper relates to the storage engine by addressing some core issues in the development process to avoid potentially severe costs and termination due to one or more errors or conflicts at a later stage in the development process.

3.1.1 Functionality

The researcher assumes that the storage engine should be able to;

- Store and Retrieve data.
- Modify existing Data
- Keep all versions of the modification
- Delete the modification process
- Keep track of all modification
- Retrieve all or a certain version of the modification.
- Create several replica of the data

- implement a B+ Tree Data structure to handle the data
- Keep record of different data objects
- Allow modification of on each object individually.
- Allow concurrent read write operations to occur.
- Furthermore some other complex operations not yet defined in the scope of this paper.

3.2 Use Case 1

3.3 Use Case 2

3.4 Replication

3.5 Key Operations

The key operation can be classified as the basic CRUD operations.

- Create: This is the first write operation.
- Read: This is all the reading operations.
- Update: This is the version modification and also a write operation
- Delete: This should not delete the data, but only prevent any further updates (versions and modifications) from being

Chapter 4

Storage Engine Specification

4.1 Model Description

4.2 Constant and Variables

4.3 Initialization

4.4 Next State

4.5 Operation Specification

4.5.1 Write Operation

4.5.2 Read Operation

4.5.3 Update Operation

4.5.4 Delete Operation

This part of the thesis is much more free-form. It may have several subsections. But it all has only one purpose: to convince the examiners about the answer to the research question(s) or solution to the problem(s) that you set for yourself in the proposal.

Definition 4.1 (Testing 1,2,3)**.** *This definition is placed within a chapter so is its number.*

Figure 4.1: NUST Emblem.

Chapter 5

Verification of Implementation

So show what you did so far (implementation and testing) that is relevant to answering the question(s) or solving the problem(s).

Chapter 6

Conclusion

6.1 Section 1

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

6.1.1 Subsection 1

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit

amet, consectetur.

$$\begin{aligned} A &= \frac{\pi r^2}{2} \\ &= \frac{1}{2}\pi r^2 \end{aligned} \tag{6.1.1}$$

Adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur? [**bruno**, **smith**]

Column 1	Column 2	Column 3	Column 4
1, 1	1, 2	1, 3	1, 4
2, 1	2, 2	2, 3	2, 4
3, 1	3, 2	3, 3	3, 4
4, 1	4, 2	4, 3	4, 4

Table 6.1: The caption of the table goes here.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco[1] laboris nisi ut aliquip ex ea commodo consequat [**bruno**]. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. [**smith**]

$$\begin{aligned} p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3 \\ - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3 \end{aligned} \tag{6.1.2}$$

6.1.2 Subsection 2

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi

[1] Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam.

architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

6.2 Section 2

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

6.3 Future Work

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit

amet, consectetur.

Led ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur.

Appendix A

First Appendix

The separate numbering of appendices is also supported by LaTeX. The *appendix* macro can be used to indicate that following chapters are to be numbered as appendices. Only use the *appendix* macro once for all appendices.