Thèse présentée pour l'obtention du grade de

DOCTEUR de SORBONNE UNIVERSITÉ

Spécialité

Ingénierie / Systèmes Informatiques

École doctorale

Informatique, Télécommunication et Électronique Paris (ED130)

Investigate an approach to distributed storage that ensures consistency semantics tailored to the application, while retaining scalability and availability.

Saalik Hatia

Soutenue publiquement le : *mai 2023*

Devant un jury composé de :

Antoine Miné, Professeur, Sorbonne Université	*Role dans le jury*
Gaël Thomas, Professeur, Sorbonne Université	*Rapporteur*
Marc Shapiro, Directeur de Recherche, Sorbonne Université, LIP6	*Directeur de thèse*

À Alice et Bob

Remerciements

Merci à *Ilyas Toumlilt*[Tou21] pour ce super template.

J'aimerais également remercier *ma famille* pour leur confiance.

Pour tout le reste, *il y a Mastercard*.

Enfin, "*La café, c'est la vie*".

Abstract

...

Keywords: K1, K2, K3, K4, K5

Résumé

...

Mots-clés: K1, K2, K3, K4, K5

Contents

1 **Introduction** 1
- 1.1 Overview 1
- 1.2 Contributions 2
- 1.3 Publications 3
- 1.4 Organization 3

I Background 5

2 **Distrbuted databases** 7
- 2.1 Consistency models 7
- 2.2 Data structures 7
- 2.3 Consistency issues 7

3 **Related Work** 9
- 3.1 Overview 9
  - 3.1.1 Discussion 9

II Contributions 11

4 **Contrib 1** 15
- 4.1 Contrib 1 15

5 **Contrib 2** 17

6 **Contrib 3** 19

7 **System API and Implementation** 21
- 7.1 Example of code input 21

III Experimental Evaluation 23

8 Benchmark app and setup 27

9 Performance Evaluation 29

9.1 Setup 29

9.2 Metrics A 29

9.3 Metrics B 29

9.4 Metrics C 29

9.5 Metrics 4 29

10 Summary and discussion 31

IV Conclusion 33

Bibliography 37

A Résumé 45

Introduction 1

main intro part, in general it should contain at least 4 paragraphs each answers one of the following questions.

P1: What's the thesis main problem?

P2: Why the problem is a problem?

P3: what's the solution?

P4: Why the solution is better than the state of the art?

1.1 Overview

An essential component of a database is its backend, in charge of recording the lowest level of data into a *store*. Although conceptually simple at a high level, actual backends are complex, due to the demands for fast response, high volume, limited footprint, concurrency, distribution, reliability, and so on. For instance, the open-source RocksDB has 350k LOC, Redis is 160k LOC, and the AntidoteDB backend is 17k LOC. Any such complex software has bugs; database backend bugs are critical, possibly violating data integrity or security [**rocksdbbug**].

Using verification tools has the potential to avoid such bugs, but, given the complexity of a modern backend, fully specifying all the moving pieces is a daunting task.

This article reports an incremental approach to the rigorous and modular development of a backend. We first formalise the semantics of transactions above a versioned key-value store; this high-level specification helps to reason about correctness, both informally and with the Coq proof tool. Thanks to explicit versioning, the state at any point in time is deterministic, and all implementations are behaviourally equivalent. We specify a map-based and a journal-based variant.

Reading the specification as a kind of pseudocode, we implement it verbatim, without optimisation. More specifically, we implement a map-based and a journal-based store, both in memory and on disk.

Using them directly is impractical; features such as caching, write-ahead logging, checkpointing, journal truncation, etc., are required to improve performance. Our simplifying insight is that such features can be described and implemented by *composing* instances of the basic variants. In particular, we show how to compose a write-ahead log and to bound storage footprint. In future work, we believe the same approach can justify sharding, and geo-distribution. The formal rules for correct dynamic composition are particularly simple.

In the style of MVCC (multi-value concurrency control), a transaction reads from a causally-consistent snapshot. It terminates by either committing atomically, or by aborting without modifying the store. The transaction model appeals to a store's specialised book-keeping operations (called doUpdate, doCommit and lookup), implicity assuming infinite memory and no failures. To bound a store's memory footprint, we restrict its domain.

This paper focuses on safety properties, and does not consider security issues.

Our contributions can be summarised as follows:

- A formal model of a concurrent, transactional backend store, with three variants: map- and journal-based, and compositional, and a model for the correct composition of stores.
- Interpreting the formal model in terms of system design and implementation challenges; applying it to the implementation to volatile and persistent maps, and a crash-tolerant journal.
- Implementation by composition of a write-ahead log, with caching and bounded storage footprint.
- Experimental evaluation, testing for correctness and showing that our rigorous approach does not preclude performance.

1.2 Contributions

The main results of this dissertation are as follows:

- R1
- R2
- R3
- R4

- R5

Our experimental evaluation shows that:...

1.3 Publications

Some of the results presented in this thesis have been published as follows:

- myFirstPublication
- mySecondPublication

During my thesis, I explored other directions and collaborated in several projects that have helped me to get insights on the challenges of ... These efforts have led me to contribute to the following publications and deliverables:

- TechReport
- ANRProjectDelivrable
- EUProjectDelivrable

1.4 Organization

This thesis is divided into three parts. The rest of this document is organized as follows:

- Part I introduces the common background of our work, formulate the problem, presents the existing solutions and discusses the use-case requirements, this part is divided into three chapters:
  - Chapter 2 provides a complete and up-to-date review of the MyDomain.
  - Chapter **??** presents a use-case point of view of the ...
  - Chapter 3 studies and compares the solutions that have been designed in the state of the art of ...
- Part II, in light of what we saw in the existing work, we will here justify some protocol choices used in our approach...

- Part III provides an experimental evaluation demonstrating the benefits of our approach, compared to other solutions from the state of the art.
- the last part IV we summarize our contribution, and present our vision for the future requirements towards more ...

Part I

Background

2

Distrbuted databases

2.1 Consistency models

2.2 Data structures

State of the art Key Value Store

2.3 Consistency issues

3 Related Work

Intro paragraph

3.1 Overview

...

3.1.1 Discussion

...

Part II

Contributions

In this chapter, we present the design, algorithms and architecture of ...

4 Contrib 1

4.1 Contrib 1

TBD

5

Contrib 2

TBD

6

Contrib 3

TBD

7 System API and Implementation

TBD

7.1 Example of code input

```
 1  let dc_connection = colony_dc.connect(CONFIG.dbURI, CONFIG.credentials);
 2  let cnt = dc_connection.counter("myCounter");
 3  dc_connection.update(
 4    cnt.increment(3)
 5  )
 6
 7  let peer_connection = colony_pop.connect(CONFIG.signalingServers, CONFIG.
        credentials)
 8  let tx = await peer_connection.startTransaction()
 9  let map = tx.gmap("myMap");
10  tx.update([
11    map.register("a").assign(42),
12    map.set("e").addAll([1, 2, 3, 4])
13  ])
14  tx.commit().then(
15    console.log(
16      await peer_connection.gmap("myMap").set("e").read()
17    )
18  )
```

Fig. 7.1.: An example of THESE2OUF program.

The TypeScript example in Figure 7.1 illustrates the API. This application opens a session (Line 1). Then, it creates and increments a CRDT counter object (Lines 2–5). Then it connects to a peer group (line 7), and updates the grow-only map (gmap) `"myMap"` in a transaction (lines 8–12). This map contains references to a register object (key `"a"`) and a set object (key `"e"`). The counter update and the commit are both asynchronous (Lines 9 and 13), returning a promise. At line 13, the client waits for the promise, and displays the content of the set.

Part III

Experimental Evaluation

Don’t forget to put the source code link ;-)

8 Benchmark app and setup

Overview ...

Workload ...

9 Performance Evaluation

The performance evaluation will focus on situations where …

9.1 Setup

…

9.2 Metrics A

…

9.3 Metrics B

…

9.4 Metrics C

…

9.5 Metrics 4

…

10

Summary and discussion

...

Part IV

Conclusion

TBD

Bibliography

[Tou21] Ilyas Toumlilt. “Colony: A Hybrid Consistency System for Highly-Available Collaborative Edge Computing”. PhD thesis. Sorbonne Université, 2021 (cit. on p. iii).

List of Figures

7.1 An example of THESE2OUF program. 21

List of Tables

List of Listings

docs/typescript/src/api.ts 21

A Résumé

The doctoral schools of Sorbonne Université require at least one page of summary in French. (Even if the website says that you need to provide more than one page, one is enough, which I did).

Résumons donc en français. Tout bon résumé commence par une description générale du problème de la thèse. Le problème étant que les écoles doctorales de Sorbonne Université exigent au moins une page de résumé en français. Voici donc un résumé garanti non traduit sur un DeepL.

Le deuxième paragraphe du résumé doit répondre à la question *pourquoi le(s) problème(s) présenté(s) est un vrai problème ?*.

Ensuite, le troisième paragraphe doit répondre à la question *quelles sont les solutions apportées par la thèse à ces problèmes ?* Cette thèse explore ces problèmes en profondeur, en étudiant l'état de l'art lié … et présente la solution *SystèmeCool2Ouf*, conçu pour répondre aux problématiques exposées. L'une des principales exigences est une approche *DurACuir* qui *EstVraimentVraimentDureCar..*. Cependant, cela rend difficile la satisfaction des attentes en matière de *Toute solution à des petits défauts et compromis, on va pas se le cacher*.

Paragraphe 4, *En quoi la solution est meilleure que l'état de l'art ?* Pour répondre à ces défis, nous avons fait le choix d'adopter une approche … en fournissant les plus fortes garanties de …. Un défi connexe est …, que nous avons limité grâce à …

Enfin, traditionnelement on liste une référence aux contribution, Les contributions de cette thèse peuvent être résumées comme suit:

- …;
- …;
- …;

Notre évaluation expérimentale montre que …