



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Masterarbeit

Medieninformatik

Fachbereich VI – Informatik und Medien

Untersuchung der Konfliktmanagementstrategien verschiedener offlinefähiger Systeme

Berlin, den 4. März 2018

Autorin:

Jacoba BRANDNER

Matrikelnummer:

833753

Betreuer:

Herr Prof. Dr. Hartmut SCHIRMACHER

Gutachter:

Herr Prof. Dr. John DOE

Abstract

In dieser Arbeit wird eine Anwendung entwickelt, die ...

Hierzu wird analysiert welche Studien, Projekte oder Anwendungen es zu diesem Thema bereits gibt. Es wird diskutiert,.... Technischen und physikalischen Grundlagen erklärt. Hierb werden Definitionen und Entwicklungswerkzeuge beschrieben und ein Überblick über mögliche Einsatzgebiete gegeben. Für das Verständnis der Umsetzung ist die Klärung der ... erforderlich. Anschließend werden die möglichen Szenarien erarbeitet, woraus sich die Anforderungen an Funktionalität und Design für die Anwendung ergeben. Die Konzipierung und Implementierung des exemplarischen Prototyps bilden den Kern dieser Arbeit, wobei dieser Prototyp in Architektur, Funktionalität und Design erläutert und schließlich in mehreren Testreihen evaluiert wird.

Abstract

This work includes the development of an application ...

Therefore it is analyzed which researches, projects and applications do already exist. It is going to be discussed ... technical base ... At this point definitions and developing tools are described and an overview of potential domains is given.

For comprehension the purification of the theoretical basis of the computation is necessary. Afterwards possible scenarios are worked out what from requirements of functionality and design result.

The conception and implementation of the showcase prototype is the core of this work. This is exemplified in architecture, functionality and design and finally evaluated in several test series.

INHALT

1 Einführung	5
1.1 Motivation	5
1.2 Zielstellung	5
2 Bestehende offlinefähige Systeme / Konzepte	6
2.1 redux-offline	6
2.2 Realm	6
2.3 hoodie	6
3 Grundlagen	7
3.1 Offline First	7
3.2 Konfliktmanagementstrategien in verteilten Systemen	7
3.2.1 Operational Transformation	7
3.2.2 Conflict-free replicated data type	7
3.2.3 Last-Write-Wins (LWW)	8
3.3 CouchDB/PouchDB	9
3.3.1 CouchDB	9
3.3.2 PouchDB	9
4 Konzeption	10
4.1 Anwendungsaufbau	10
4.2 Architektur	10
4.3 Testfälle	10
4.4 Entwicklungsumgebung	10
5 Der Prototyp der Anwendung	12
5.1 Simulator Projektstruktur	12
5.2 Hauptkomponente... wie passiert was	12
5.3 Installationsanleitung	12
6 Evaluation	13
6.1 Systemtest und Ergebnisse	13

7 Zusammenfassung und Ausblick	14
Abkürzungen	15
Abbildungsverzeichnis	16
Literaturverzeichnis	16
Anhang	18

1 EINFÜHRUNG

We live in a disconnected & battery powered world, but our technology and best practices are a leftover from the always connected & steadily powered past. [off]

1.1 MOTIVATION

1.2 ZIELSTELLUNG

2 BESTEHENDE OFFLINEFÄHIGE SYSTEME / KONZEPTE

2.1 REDUX-OFFLINE

2.2 REALM

[rea]

2.3 HOODIE

[hoo]

3 GRUNDLAGEN

3.1 OFFLINE FIRST

1. Separate Apps from Data
2. Deliver App Code (and make it cachable) appcache & ServiceWorkers
3. Save Data Offline localStrogare / localForage, IndexedDB, other Wrapper
4. Detect Connectivity navigator.onLine (Lie-fi)
5. Sync Data - Build upon existing solutions – CouchDB/PouchDB | remoteStorage

3.2 KONFLIKTMANAGEMENTSTRATEGIEN IN VERTEILTEN SYSTEMEN

Im Folgenden werden die drei Strategien Operational Transformation (OT), Conflict-free replicated data type (CRDT) und Last-Write-Wins (LWW) vorgestellt.

3.2.1 OPERATIONAL TRANSFORMATION

3.2.2 CONFLICT-FREE REPLICATED DATA TYPE

CRDTs sind Objekte, die ohne teure Synchronisation aktualisiert werden können. Sie konvergieren schließlich, wenn alle gleichzeitigen Aktualisierungen kommutativ¹ sind und wenn alle Aktualisierungen schließlich von jeder Replik ausgeführt werden [SPBZ11a]. Um diese Garantien zu geben, müssen diese Objekte bestimmte Kriterien erfüllen, welche im Folgenden beschrieben werden [SPBZ11b].

ZUSTANDBASIERTER ANSATZ

Wenn ein Replikat ein Update von einem Client empfängt, aktualisiert es zuerst seinen lokalen Status und dann, einige Zeit später, seinen **vollständigen Status**. So sendet je-

¹Unverändert bei Vertauschen der Operanden

des Replikats gelegentlich seinen vollständigen Status an ein anderes Replikat im System. Um ein Replikat, das den Status eines anderen Replikats empfängt, wendet eine **Zusammenführungsfunktion** (merge) an, um den empfangenen Status mit dem lokalen Status zusammenzuführen. Entsprechend sendet dieses Replikat gelegentlich auch seinen Status an ein anderes Replikat, sodass jedes Update schließlich alle Replikate im System erreicht.

OPERATIONSBASIERTER ANSATZ

Bei diesem Ansatz sendet ein Replikat seinen vollständigen Status (kann groß sein) nicht an ein anderes Replikat. Stattdessen sendet es nur den **Aktualisierungsvorgang** an **alle** anderen Replikate im System und erwartet von ihnen, dass sie das Update auf sich anwenden.

Da es sich um einen Sendevorgang handelt, wenn zwei Updates u_1 und u_2 , bei einem Replikat i angewendet werden und diese Updates an zwei Replikate r_1 und r_2 gesendet werden, können diese Updates in unterschiedlicher Reihenfolge bei diesen Replikaten ankommen. r_1 kann sie in der Reihenfolge u_1, u_2 empfangen, während bei r_2 die Updates in umgekehrter Reihenfolge (u_2, u_1) ankommen können. Sind die Aktualisierungen **kommutativ**, können die Repliken zussammengeführt werden, egal in welcher Reihenfolge die Updates bei ihnen ankommen - der resultierende Zustand ist derselbe. In diesem Modell wird ein Objekt, für das alle gleichzeitigen Aktualisierungen kommutativ sind, CmRDT (commutative replicated data type - kommutativ replizierter Datentyp) genannt.

Beispiel:...

CRDTs befassen sich mit einem interessanten und grundlegendem Problem in verteilten Systemen, haben jedoch eine wichtige Einschränkung: "Da ein CRDT konstruktionsbedingt keinen Konsens verwendet, hat der Ansatz starke Einschränkungen; Dennoch sind einige interessante und nicht-triviale CRDTs bekannt" [SPBZ11b]. Die Einschränkung ist, dass die CRDT-Adresse nur einen Teil des Problemraums betrifft, da nicht alle möglichen Aktualisierungsoperationen kommutativ sind und daher nicht alle Probleme in CRDTs umgewandelt werden können. Auf der anderen Seite können CRDTs für einige Arten von Anwendungen durchaus nützlich sein, da sie eine nette Abstraktion zur Implementierung replizierter verteilter Systeme bieten und gleichzeitig theoretische Konsistenzgarantien bieten.

3.2.3 LAST-WRITE-WINS (LWW)

3.3 COUCHDB/POUCHDB

3.3.1 COUCHDB

3.3.2 POUCHDB

4 KONZEPTION

Die erarbeiteten Anforderungen an ... werden in diesem Kapitel für die Konzeption angewendet. Beginnend mit dem Aufbau der Anwendung werden in den folgenden Abschnitten die Anwendungsfälle, die Architektur und schließlich die Komponenten der Entwicklungsumgebung aufgeführt.

4.1 ANWENDUNGSaufbau

4.2 ARCHITEKTUR

4.3 TESTFÄLLE

Folgende Testfälle werden während der Entwicklung stetig durchgeführt. Das erfolgreiche Bestehen dieser Tests ist eine notwendige Qualitätseigenschaft der zu entwickelnden Applikation.

4.4 ENTWICKLUNGsumgebung

Für die Erstellung der Smartphone-Applikation wurde folgende Soft- und Hardware verwendet:

SOFTWARE

- Sublime Editor 3
- git, Version 2.7.4 zur Versionsverwaltung

HARDWARE

- Tuxedo (Intel® Core™i7-6500U, 2,50GHz x 4, 7,7 GB RAM) als ersten Entwicklungsrechner (Betriebssystem: Ubuntu¹ 16.06, 64-bit-Version)

¹ Download unter <https://www.ubuntu.com/download/desktop>

- Lenovo Thinkpad X200 (Intel® Core™2 Duo, 2,40GHz, 8GB RAM) als zweiten Entwicklungsrechner (Betriebssystem: Debian 7.8, 64-bit-Version)
- Testgeräte...

5 DER PROTOTYP DER ANWENDUNG

5.1 SIMULATOR PROJEKTSTRUKTUR

5.2 HAUPTKOMPONENTE... WIE PASSIERT WAS

5.3 INSTALLATIONSANLEITUNG

6 EVALUATION

Um die Funktionalität des Prototyps zu untersuchen, wurden folgende Testgeräte ausgewählt.

6.1 SYSTEMTEST UND ERGEBNISSE

7 ZUSAMMENFASSUNG UND AUSBLICK

Das System ist bei Bedarf auf verschiedene Weisen erweiterbar...

ABKÜRZUNGEN

CRDT Conflict-free replicated data type

LWW Last-Write-Wins

OT Operational Transformation

ABBILDUNGSVERZEICHNIS

LITERATURVERZEICHNIS

- [hoo] *Hoodie – The Offline First Backend*. <http://hood.ie/>,. – Zugriff: 12.04.2018 2.3
- [off] *Offline First*. <http://offlinefirst.org/>,. – Zugriff: 12.04.2018 1
- [rea] *realm – The new standard in data scnchronization*. <https://realm.io/>,. – Zugriff: 12.04.2018 2.2
- [SPBZ11a] SHAPIRO, Marc ; PREGUIÇA, Nuno ; BAQUERO, Carlos ; ZAWIRSKI, Marek: A comprehensive study of Convergent and Commutative Replicated Data Types / Inria – Centre Paris-Rocquencourt ; INRIA. Version: Januar 2011. <https://hal.inria.fr/inria-00555588>. 2011 (RR-7506). – Research Report. – 50 S. 3.2.2
- [SPBZ11b] SHAPIRO, Marc ; PREGUIÇA, Nuno ; BAQUERO, Carlos ; ZAWIRSKI, Marek: Conflict-free Replicated Data Types. Version: Juli 2011. <https://hal.inria.fr/inria-00609399>. 2011 (RR-7687). – Research Report. – 18 S. 3.2.2, 3.2.2

ANHANG

EIDESSTATTLICHE ERKLÄRUNG

CD-INHALT

Auf der beigefügten CD befinden sich

- Die schriftliche Ausarbeitung dieser Masterrarbeit im PDF-Format
- Das erstellte Projekt