Diplomarbeit

Höhere Technische Bundeslehranstalt Leonding
Abteilung für Informatik

AKKalytics

Eingereicht von: **Felix Kronsteinerr, 5AHIF**
Adzamija Alen, 5AHIF
Klatzer Emanuel, 5AHIF

Datum: **4. April 2018**

Betreuer: **Gerhard Aistleitner**

Projektpartner: **MIC**

Declaration of Academic Honesty

Hereby, I declare that I have composed the presented paper independently on my own and without any other resources than the ones indicated. All thoughts taken directly or indirectly from external sources are properly denoted as such.

This paper has neither been previously submitted to another authority nor has it been published yet.

Leonding, April 4, 2018

Felix Kronsteinerr, Adzamija Alen, Klatzer Emanuel

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorgelegte Diplomarbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Gedanken, die aus fremden Quellen direkt oder indirekt übernommen wurden, sind als solche gekennzeichnet.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Leonding, am 4. April 2018

Felix Kronsteinerr, Adzamija Alen, Klatzer Emanuel

Zusammenfassung

Here it is described what the thesis is all about. The abstract shall be brief and concise and its size shall not go beyond one page. Furthermore it has no chapters, sections etc. Paragraphs can be used to structure the abstract. If necessary one can also use bullet point lists but care must be taken that also in this part of the text full sentences and a clearly readable structure are required.

Concerning the content the following points shall be covered.

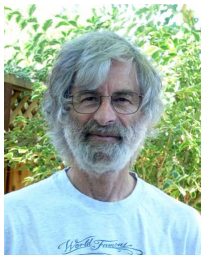
1. *Definition of the project:* What do we currently know about the topic or on which results can the work be based? What is the goal of the project? Who can use the results of the project?

2. *Implementation:* What are the tools and methods used to implement the project?

3. *Results:* What is the final result of the project?

This list does not mean that the abstract must strictly follow this structure. Rather it should be understood in that way that these points shall be described such that the reader is animated to dig further into the thesis.

Finally it is required to add a representative image which describes your project best. The image here shows Leslie Lamport the inventor of LaTeX.



Zusammenfassung

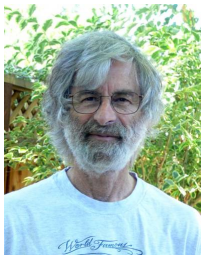
An dieser Stelle wird beschrieben, worum es in der Diplomarbeit geht. Die Zusammenfassung soll kurz und prägnant sein und den Umfang einer Seite nicht übersteigen. Weiters ist zu beachten, dass hier keine Kapitel oder Abschnitte zur Strukturierung verwendet werden. Die Verwendung von Absätzen ist zulässig. Wenn notwendig, können auch Aufzählungslisten verwendet werden. Dabei ist aber zu beachten, dass auch in der Zusammenfassung vollständige Sätze gefordert sind.

Bezüglich des Inhalts sollen folgende Punkte in der Zusammenfassung vorkommen:

- *Aufgabenstellung:* Von welchem Wissenstand kann man im Umfeld der Aufgabenstellung ausgehen? Was ist das Ziel des Projekts? Wer kann die Ergebnisse der Arbeit benutzen?
- *Umsetzung:* Welche fachtheoretischen oder -praktischen Methoden wurden bei der Umsetzung verwendet?
- *Ergebnisse:* Was ist das endgültige Ergebnis der Arbeit?

Diese Liste soll als Sammlung von inhaltlichen Punkten für die Zusammenfassung verstanden werden. Die konkrete Gliederung und Reihung der Punkte ist den Autoren überlassen. Zu beachten ist, dass der/die LeserIn beim Lesen dieses Teils Lust bekommt, diese Arbeit weiter zu lesen.

Abschließend soll die Zusammenfassung noch ein Foto zeigen, das das beschriebene Projekt am besten repräsentiert. Das folgende Bild zeigt Leslie Lamport, den Erfinder von LaTeX.



Acknowledgments

If you feel like saying thanks to your grandma and/or other relatives.

Inhaltsverzeichnis

Kapitel 1

Einleitung

1.1 Ausgangslage

Common word processors do not prepare print-like documents in so far as these programs do not reflect the rules of professional printing which have been grown over centuries. These rules contain clear requirements for balancing page layouts, the amount of white space on pages, font-handling, etc. Donald Knuth's TeX package (see [**?**]) is a word processor which conforms to these printing rules. This package was enhanced by Leslie Lamport by providing more text structuring commands. He called his package LaTeX [**?**].

When preparing a thesis, we want not only to have our content on a top level, we also want to commit to a high level of formal criteria. Therefore, we request our students to use one of these professional printing production environments like TeX or LaTeX.

Furthermore students should train their scientific writing skills. This includes a clear and structured break-down of their ideas, a high-level and clear wording, and the training of transparent citations of ideas from other sources than from theirs. A good source for more information concerning technical and scientific writing can be found in [**?**].

1.2 Ziele

The general goals and objectives of the project are described here. Care must be taken that the goals documented here are purely project goals and have nothing to do with individual goals of the team members. If individual goals should be part of the thesis they are listed in appendix B.

1.3 Übersicht

Details of the diploma thesis have to be aligned between student and supervisor. This should be a basic structure to facilitate the first steps when students start to write their theses.

Abbildung 1.1: Don Knuth, the inventor of TEX

Never forget to add some illustrative images. Images must not be messed up with your normal text. They are encapsulated in floating bodies and referenced in your text. An example can be seen in figure 11.1. As you can see, figures are placed by default on top of the page nearby the place where they are referenced the first time. Furthermore you can see that a list of figures is maintained automatically which can be included easily by typing the command `\listoffigures` into your document.

1.4 Basic Terminology

As usual the very basic terminology is briefly explained here. Most probably the explanations here only scratch a surface level. More detailed explanations of terminology goes into chapter **??**.

1.5 Related Work and Projects

Here a survey of other work in and around the area of the thesis is given. The reader shall see that the authors of the thesis know their field well and understand the developments there. Furthermore here is a good place to show what relevance the thesis in its field has.

1.6 Structure of the Thesis

Finally the reader is given a brief description what (s)he can expect in the thesis. Each chapter is introduced with a paragraph roughly describing its content.

Kapitel 2

Aufgabenstellung

Kapitel 3

Lösungsansatz

Kapitel 4

Architektur

Kapitel 5

Datenquellen und Ihr Aufbau

Kapitel 6

Extract

Kapitel 7

Transform

Kapitel 8

Load

Kapitel 9

Visualisierung

9.1 Kapitel1

9.2 Kapitel2

9.3 Kapitel3

9.4 Kapitel4

Kapitel 10

Die Arbeitsumgebung

10.1 AWS[AA]

10.1.1 Allgemeine Beschreibung

“Amazon Web Services (AWS) ist mit mehr als 175 Services, die umfangreiche Funktionen bieten und in global verteilten Rechenzentren bereitgestellt werden, die weltweit umfassendste und am häufigsten genutzte Cloud-Plattform. Millionen von Kunden – darunter einige der am schnellsten wachsenden Start-up-Unternehmen und der größten Konzerne sowie wichtige Behörden – vertrauen auf AWS, wenn es darum geht, agiler zu werden, Kosten zu senken und Innovationen schneller zu realisieren.„ [noab]

10.1.2 Geschichte

Seit 2006 in dem Jahr als AWS gestartet ist standen Unternehmen mehr im Fokus als Endanwender. Es stellt für Entwickler eine Struktur da um auf Abruf gleich arbeiten zu können.

Abbildung 10.1: AWS

10.2 Linux

10.3 Portainer

10.4 IntelliJ

10.5 Vim

10.6 Docker

Kapitel 11

Technologien

11.1 Kapitel1

11.2 Kapitel2

11.3 NoSql Datenbanken [AA]

11.3.1 Allgemeine Beschreibung

NoSQL steht für Not only SQL und beschreibt Datenbanksysteme, die einen nicht-relationalen Ansatz haben. Datenbanken, die diesem Modell zugrunde liegen, sind horizontal skalierbar und lassen sich für Big-Data-Anwendungen einsetzen. Da NoSQL, Not only SQL bedeutet kann man nicht grundsätzlich auf die Datenbanksprache SQL (Structured Query Language) verzichten. Viele dieser Systeme setzen zwar komplett auf nicht relationale Funktionen, doch existieren auch NoSQL Datenbanken, die nur bestimmte Elemente von SQL-Systemen unberücksichtigt lassen.

Während relationale Datenbanken Tabellen mit Zeilen und Spalten für die Datenspeicherung verwenden, nutzen NoSQL Datenbanken zum Beispiel, Objekte, Dokumente, Liste etc. für die Organisation der Daten. Diese Systeme haben eins gemeinsam, dass sie optimiert sind für Aufgaben, bei denen SQL Systeme an ihre Grenzen stoßen.

Aufgrund des Aufbaus skalieren NoSQL-Datenbanken horizontal. In vielen Fällen sind diese Open-Source-Software, doch gibt es einige Lösungen, die auf eine kommerzielle Lösung basieren. Durch das Fehlen eines Schemas, sind NoSQL Systeme flexibel einsetzbar und eigenen sich für große Datenmengen, wie sie aus Big-Data-Anwendungen kommen. Die Architektur ist auf Skalierbarkeit und Performance ausgelegt. Fast alle NoSQL Ansätze und Modelle lassen sich in vier Hauptkategorien einteilen. Diese sind Key-Value Datenbanken. Dokumentenorientierte Datenbanken, Spartenorientierte Datenbanken und Graphen Datenbanken.

Folgende Eigenschaften zeichnen die NoSQL Systeme aus das Vermeiden von unnötiger Komplexität, eine hohe Performance, die horizontale Skalierbarkeit, die Vermeidung von relationalen Ansätzen des Datenmappings, die Unterstützung der aktuellen Hardwaregenerationen und die Einfachheit in der Installationen und Konfiguration von verteilten Clustern.

11.3.2 Dokumentenorientierte Datenbanken

Dokumentorientierte Datenbanken oder auch Document Store genannt finden Verwendung in der Verwaltung von semistrukturierten Daten. Mit semistrukturierten Daten sind Daten gemeint, die keine feste Struktur haben, sondern die Struktur selbst in den Daten hervorgeht. Durch die fehlende klare Struktur in den Daten sind diese nicht für relationale Datenbanken geeignet, weil sich die Informationen nicht in Tabellen einordnen lassen.

Die Datenbank erstellt dann einen Schlüssel zu jedem Dokument damit alles zugeordnet ist. Im Dokument selbst, dass zum Beispiel mit JSON, YAML oder XML formatiert ist sind die eigentlichen Daten gespeichert. Da die Datenbank kein bestimmtes Schema hat, kann man auch verschiedene Dokumenttypen gemeinsam in einer Dokumentorientierte Datenbank einbinden. Änderungen der Daten in den Dokumenten müssen nicht der Datenbank mitgeteilt werden.

Daten unterschiedlichster Formate und ohne ein gemeinsames Schema können in ein Document Store unterbringen. In der Praxis aber verwendet man in der Regel nur ein Dateiformat für die Dokumenten und man baut die Informationen in einer festen Struktur auf. Durch diese festgelegten Regeln ist es einfacher die Daten zu verarbeiten und die Arbeit mit der Datenbank selbst. Durch diese Ordnung können Suchanfragen an die Datenbank viel besser verarbeitet werden. In einer dokumentbasierten Datenbank können die gleichen Aktionen durchgeführt werden wie auch in einer relationalen Datenbank. Daten lassen sich einfügen, löschen, abfragen und ändern.

Damit sich diese Aktionen durchführen lassen, erhält jedes einzelne Dokument einen eindeutigen Schlüssel. Wie dieser Schlüssel zusammengesetzt wird ist im Grunde egal. Eine einfache Zeichenfolge, der komplette Pfad oder andere Methoden können genutzt werden um das Dokument zu adressieren.

Abbildung 11.1: MongoDB

Mongo DB

Allgemein
„MongoDB ist eine universelle, dokumentbasierte, verteilte Datenbank für die moderne Anwendungsentwicklung und die Cloud, die in puncto Produktivität höchsten Ansprüchen gerecht wird.“ [noaa]

Expressive Object Model
MongoDB unterstützt ein ausdrucksstarkes Objekt Modell. Objekte in der MongoDB können Eigenschaften haben und in sich selbst verkettet sein. Also ein Objekt kann mehrere Stufen haben. Dieses Modell ist sehr objektorientiert und kann jede Objektstruktur leicht repräsentieren. Auch kann jedes Objekt indexiert werden und auch noch auf jeder Stufe.

Sekundäre Indizes
Dieses Konzept macht es leicht jede Eigenschaft eines Objekts zu indexieren egal, ob es verschachtelt ist oder nicht. Das macht es leicht durch diese Indizes abfragen zu machen und Ergebnisse zu finden.

Hohe Erreichbarkeit
MongoDB unterstützt ein Single Master Modell. Das bedeutet das es einen Eltern Knoten gibt und viele Kinder Knoten. Falls der Eltern Knoten kaputtgeht oder einfach nicht mehr funktioniert wird ein Kind der neuen Eltern Knoten. Früher hatte dieser Prozess, 10–40 Sekunden gebraucht, aber seit neuesten alles unter 2 Sekunden wieder geregelt. Doch während der Zeit bis ein neuer Elternknoten gefunden wurde, rennt das Replika Set nicht und nimmt keine writes an.

Schreib Skalierbarkeit
MongoDB mit seinen Single Master Modell kann nur auf seinen Primären Server Schreibbefehlen annehmen. Während die sekundären Server nur für Lesebefehle benutzt wird. Also, wenn man drei Replika Sets hat kann nur der Eltern Knoten schreiben während die Kinder nur Lesen können. Das schränkt die Skalierbarkeit sehr ein. Man könnte mehrere Shards benutzen aber es werden trotzdem nur ein Drittel Schreibbefehle von den

Knoten benutzt. Doch können alle Knoten gestriped werden, dass alle Knoten lesen und schreiben können.

Qurey Language Support
MonogDB hat eine eigene Query Language, kann aber auch ANSI SQL unterstützen.

Benutzbarkeit
MongoDB ist eine leicht zu benutzende Datenbank die auch leicht aufgesetzt wird. Da die Dokumente in der MongoDB fast wie im Code aussehen ist es oft leicht die Objekte zu verstehen und zu benutzen. Auch hat MongoDB eine Vielzahl von Treibern und Applikationen um die Datenbank zu warten und darauf zu arbeiten.

Native Aggregation
Die Datenbank hat ein eingebautes Aggregierungs Framework um die Daten in der Datenbank für eine ETL Pipeline zu transformieren. Dieses Framework ist für kleine und mittlere Daten prozessierende Anfragen zu bearbeiten, aber desto größer die Datenmenge desto komplizierter wird dieser Prozess schwerer zu debuggen. Um größere Anfragen zu debuggen wird daher öfter Spark oder Hadoop benutzt. Dies sind Tools mit ihren eigenen Ressourcen, Skills und viele anderen Faktoren.

Schema-less Model
In MongoDB kann man sich aussuchen, ob ein Schema auf die Dokumente forciert wird oder nicht. Seit Version 3.2 kann auch ein Schema forciert werden. Jedes Dokument kann seine eigene Struktur haben und die Applikationen selbst interpretieren dann die Daten korrekt. Für einige Applikationen ist diese Flexibilität nötig.

Latenzzeit
Als Erstes wird in der Datenbank ein primärer Knoten ausgewählt und die Replika state Maschine aufgebaut. Es gibt kein äußerliches System, dass das Replika Set sagt, was es zu tun hat oder es beobachtet. Das Set selbst sucht den primären Knoten aus, wann es sich selbst replizieren soll, etc. Dadurch ist es viel einfacher und man eliminiert so ganze Klassen und Topologie Probleme.

Als Nächstes erlaubt dir MongoDB den primären Knoten zu fragen ob das replizieren funktioniert hat, oder von sekundäre Knoten. Durch das einbüßen von der Latenzzeit, kann stärker garantiert werden ob ein Datenbank write funktioniert hat oder nicht.

Wenn der primäre Knoten nicht mehr erreicht werden kann, überlegen die sekundären Knoten was zu tun ist, wenn sich eine Mehrheit gefunden hat für einen Knoten wird dieser der neue primäre Knoten. Dies erfolgt durch das höchste optime, dass eine Uhr ist, die in jedem Knoten läuft und wie schon erklärt wird, dort der Knoten genommen der die höchste Anzahl hat. Doch müssen sich auch die Knoten im Klaren Sein das es nur einen primären Knoten pro Cluster geben kann. Währenddessen wird der Minderheit

klar, dass sie kein Quorum mehr haben und so den primären knoten abwählen, und kann so keine writes mehr akzeptieren.

11.3.3 Wide-Column Datenbanken

Cassandra

Allgemein

Expressive Object Model

Sekundäre Indizes

Hohe Erreichbarkeit

Schreib Skalierbarkeit

Qurey Language Support

Benutzbarkeit

Native Gruppierung

Schema-less Model

Latenzzeit

11.3.4 In-Memory Datenbanken

Redis

Allgemein

Aerospike

Allgemein

Tarantool

Allgemein

11.4 Spark und Data Analytics [AA]

11.5 Python [AA]

Kapitel 12

Databus

Kapitel 13

Kotlin

Kapitel 14

Gradle

Kapitel 15

Kafka

Kapitel 16

Portainer

Kapitel 17

Load

Kapitel 18

NO SQL

Kapitel 19

Python

Kapitel 20

Spark

Kapitel 21

Was sind Visualisierungs Applikationen

21.1 Kapitel1

21.2 Kapitel2

21.3 Kapitel3

21.4 Kapitel4

Kapitel 22

Der Unterschied zwischen Javascript und Zero Code

22.1 Kapitel1

22.2 Kapitel2

22.3 Kapitel3

22.4 Kapitel4

Kapitel 23

Vergleich der Visualisierungs Applikationen

23.1 Kapitel1

23.2 Kapitel2

23.3 Kapitel3

23.4 Kapitel4

Kapitel 24

Probleme beim Visualisierungsprozess

24.1 Kapitel1

24.2 Kapitel2

24.3 Kapitel3

24.4 Kapitel4

Kapitel 25

Summary

Here you give a summary of your results and experiences. You can add also some design alternatives you considered, but kicked out later. Furthermore you might have some ideas how to drive the work you accomplished in further directions.

Literaturverzeichnis

[noaa] Die beliebteste Datenbank für moderne Apps. URL: https://www.mongodb.com/de.

[noab] Was ist AWS? Sicheres Cloud Computing mit Amazon Web Services (AWS). URL: https://aws.amazon.com/de/what-is-aws/.

Abbildungsverzeichnis

Tabellenverzeichnis

Project Log Book

Date	Participants	Todos	Due

Anhang A

Additional Information

If needed the appendix is the place where additional information concerning your thesis goes. Examples could be:

- Source Code
- Test Protocols
- Project Proposal
- Project Plan
- Individual Goals
- ...

Again this has to be aligned with the supervisor.

Anhang B

Individual Goals

This is just another example to show what content could go into the appendix.