

BestShift



Machbarkeitsstudie

Abteilung der Informationstechnologie am
Technologischem Gewerbemuseum

October 2, 2015

Contents

Table of Contents	ii
1 Introduction	1
1.1 Projekt-Team	1
1.2 Projektbeschreibung	1
2 Voruntersuchung	3
2.1 Ist-Zustand	3
2.2 Soll-Zustand	3
2.3 Nicht-Ziele	3
2.4 Limitierung	3
2.5 Netzwerkabhängigkeit	3
2.6 Nutzungsbedingungen	3
3 Produktfunktionen	5
4 Produktdaten	7
5 Technische Machbarkeit	9
5.1 Varianten	9
5.2 Programmiersprachen	9
5.3 Datenbank	9
5.4 Nutzwertanalyse	9
5.5 Betriebssystem	9
5.6 Conclusio	9
6 Projektmanagement	11
7 Summary	13
8 Datenmanagement & Datenanalyse	15
8.1 Datenmanagement	15
8.1.1 Vergleich NoSQL und relational [1]	15

Contents

8.1.2	NoSQL Datenbanken	18
8.1.3	Rationale Datenbanken	18
8.1.4	Data Mining & Data Analysis	19
8.1.5	Conclusio	20

1 Introduction

1.1 Projekt-Team

1.2 Projektbeschreibung

2 Voruntersuchung

2.1 Ist-Zustand

2.2 Soll-Zustand

2.3 Nicht-Ziele

2.4 Limitierung

2.5 Netzwerkabhängigkeit

2.6 Nutzungsbedingungen

3 Produktfunktionen

4 Produktdaten

5 Technische Machbarkeit

5.1 Varianten

5.2 Programmiersprachen

5.3 Datenbank

5.4 Nutzwertanalyse

5.5 Betriebssystem

5.6 Conclusio

6 Projektmanagement

7 Summary

8 Datenmanagement & Datenanalyse

In dem folgenden Kapitel werden einige Technologien, Libraries Packages, sowie generelle Software gegenübergestellt. Diese werden dann subjektiv bewertet, und die am besten geeignetsten werden für das Projekt verwendet.

8.1 Datenmanagement

8.1.1 Vergleich NoSQL und relational [1]

SQL Datenbanken

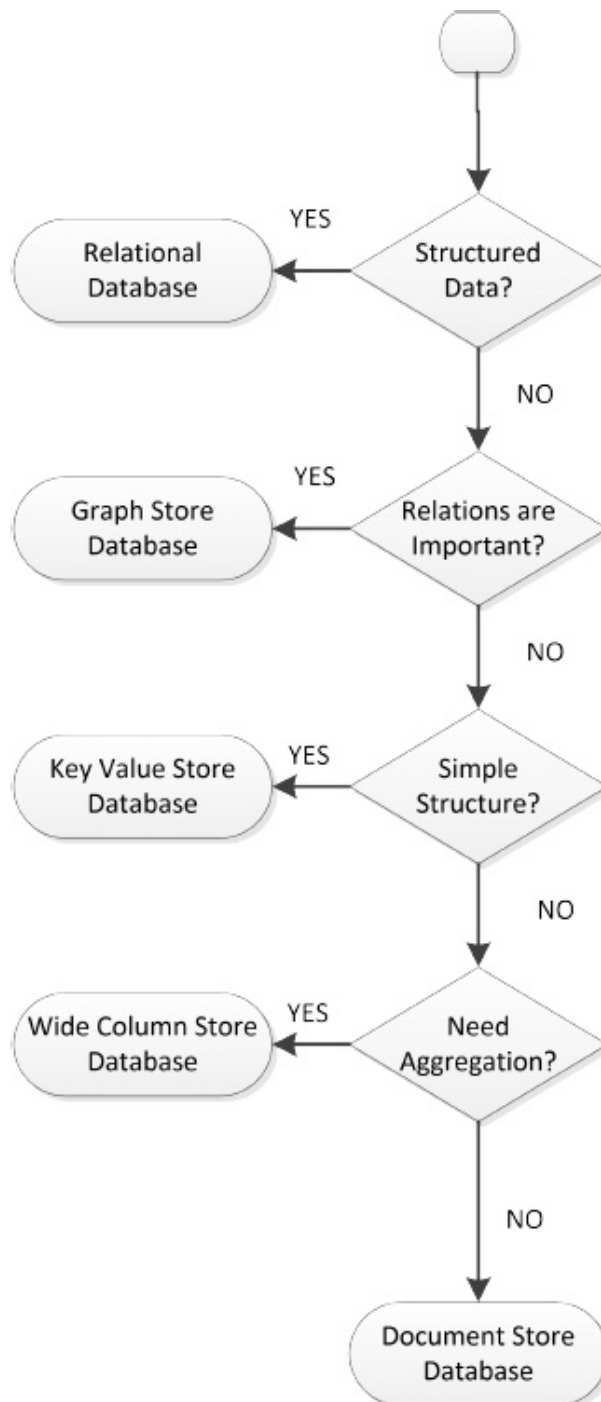
- Typen: Nur eine Art mit kleinen Unterschieden
- Data Storage Model: Individuelle Einträge werden als Reihen (Rows) in Tabellen gespeichert, wobei jede Zelle spezifische Daten über diesen Eintrag beinhaltet.
- Schemas: Struktur und daten type sind im vorhinein festgelegt. Um diese Datenbank Struktur zu ändern muss diese zunächst offline gesetzt werden.
- Skalierbarkeit: Ein einzelner Server muss hierfür mehr Rechenleistung erhalten um größeren Ansprüchen nachzukommen. Es ist prinzipiell möglich SQL Datenbanken auf ein verteiltes System zu erstellen, hierfür werden aber sehr gute Kenntnisse benötigt.
- Data Manipulation: Mittels SELECT, INSERT oder UPDATE
- Konsistenz: Gute konsistenz kann prinzipiell in allen gänglichen DBMS konfiguriert werden.

NoSQL Datenbanken

- Typen: Viele verschiedene, bspw. key-value, document-based, oder graph datenbank.
- Data Storage Model: Hängt vom Typ der Datenbank ab.
- Schemas: Typischerweise dynamisch. Einträge können on-the-fly hinzugefügt werden.
- Skalierbarkeit: Bei Bedarf kann ein Administrator einfach mehrere Cloud Instanzen hinzufügen. Die Datenbank an sich verteilt die Information auf die notwendige Server Anzahl
- Data Manipulation: Durch Objektorientierte APIs
- Konsistenz: Abhängig vom Product

Relational vs NoSQL

Anhand dem folgenden Graphik kann man prinzipiell eine Entscheidung treffen
- diese sollte dann den Ansprechungen entsprechen.



8.1.2 NoSQL Datenbanken

	Cassandra	HBase	MongoDB	Riak	MySQL Cluster	Couchbase
Performance						
Supported languages						
Documentation						
Recent aktivität						
Community						
Usage examples						

8.1.3 Rationale Datenbanken

	Sybase	IBM DB2	Oracle	Microsoft SQL Server	MySQL	PostgreSQL
Performance						
Supported languages						
Documentation						
Recent aktivität						
Community						
Usage examples						

8.1.4 Data Mining & Data Analysis

Durch die Sensoren im Auto sowie durch die zusätzlich durch den CarPC angebrachten wird eine enorme Menge an Daten geliefert. All diese Daten ergeben jedoch erst einen Sinn; wenn wir sie mit geeigneten Verfahren analysieren und auswerten können.

Frameworks

	RapidMiner	WEKA	R-Programming	Orange	KNIME	NLTK
Performance						
Supported languages						
Documentation						
Recent activity						
Community						
Usage examples						

8.1.5 Conclusio

Anhand der Evaluierung wurde die Auswahl auf die folgenden zwei DBMS Systeme beschränkt: Couchbase (NoSQL) und PostgreSQL (Relational). Da die Anforderungen momentan noch nicht fixiert sind wurde eine relationale und eine NoSQL Variante gewählt (STAND: 02. Okt 2015). In den folgendem Abschnitt gehen wir ein wenig detaillierter auf die beiden Systeme ein.

Name	Couchbase	PostgreSQL
Description	JSON-based document store derived from CouchDB with Memcached-compatible interface	Based on the object relational DBMS Postgres
Database model	Document store	Relational DBMS
Website	www.couchbase.com	www.postgresql.org
Technical documentation	www.couchbase.com/docs	www.postgresql.docs/manuals
Developer	Couchbase, Inc.	PostgreSQL Global Development Group
Initial release	2011	1989
Current release	3.0.3, March 2015	9.4.4, June 2015
License	Open Source	Open Source
Server operating systems	Linux OS X Windows	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows
Data scheme	schema-free	yes
Typing (data types)	no	yes
Secondary indexes	yes	yes
SQL	no	yes

Table 8.1: Couchbase und PostgreSQL (1)

8.1. Datenmanagement

Name	Couchbase	PostgreSQL
APIS and other access methods	Memcached protocol RESTful HTTP API	native C library Streaming API for large objects ADO.NET JDBC ODBC
Supported programming languages	.NET C Clojure ColdFusion Erlang Go Java JavaScript Perl PHP Python Ruby Scala Tcl	.NET C C++ Java Perl Python Tcl
Server-side scripts	View functions in JavaScript	user defined functions
Triggers	yes	yes
Partitioning methods	Sharding	no, but can be realised using table inheritance
Replication methods	Master-master replication Master-slave replication	Master-slave replication
MapReduce	yes	no
Consistency concepts	Eventual Consistency Immediate Consistency	Immediate Consistency
Foreign keys	no	yes
Transaction concepts	no	ACID
Concurrency	yes	yes
Durability	yes	yes
User concepts	simple password-based access control per bucket	fine grained access rights according to SQL-standard

Table 8.2: Couchbase und PostgreSQL (2)