

**Hochschule für Technik, Wirtschaft und Kultur Leipzig**

Fakultät Informatik, Mathematik und Naturwissenschaften

Masterstudiengang Informatik

Masterarbeit

zur Erlangung der akademischen Grades

**Master of Science (M.Sc.)**

**Untersuchung und Optimierung  
verteilter Geografischer  
Informationssysteme zur  
Verarbeitung Agrartechnischer  
Kennzahlen**

Eingereicht von: Kurt Junghanns

Matrikelnummer: 59886

Leipzig 10. November 2014

Erstprüfer: Prof. Dr. rer. nat. Thomas Riechert

Zweitprüfer: M. Sc. Volkmar Herbst

# Abstrakt

# Danksagung

# Vorwort

# Glossar

**Computer** is a programmable machine that receives input, stores and manipulates data, and provides output in a useful format

# Abkürzungsverzeichnis

**ACID** Atomicity, Consistency, Isolation und Durability

**BASE** Basically Available, Soft state, Eventual consistency

**CAP** Consistency, Availability and Partition Tolerance

**DBMS** Datenbankmanagementsystem

**DBS** Datenbanksystem

**GIS** Geoinformationssystem

**HDFS** Hadoop File System

**MVCC** Multi Version Currency Control

# **Abbildungsverzeichnis**

# **Tabellenverzeichnis**



# Inhaltsverzeichnis

<b>Glossar</b>	<b>iv</b>
<b>Abkürzungsverzeichnis</b>	<b>v</b>
<b>Abbildungsverzeichnis</b>	<b>vi</b>
<b>Tabellenverzeichnis</b>	<b>vii</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung . . . . .	1
1.3 Aufbau der Arbeit . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Datenbank . . . . .	3
2.1.1 Begriffsdefinitionen . . . . .	3
2.1.2 Indexstrukturen . . . . .	5
2.1.3 Mehrrechner-Datenbanksystem . . . . .	6
2.1.4 Verteiltes Datenbanksystem . . . . .	7
2.1.5 Replikationsverfahren . . . . .	7
2.2 geografische Datenverarbeitung . . . . .	8
2.2.1 Bezugssysteme . . . . .	8
2.2.2 Datenformate . . . . .	8
2.2.3 GIS . . . . .	8
2.2.4 PostGIS . . . . .	8
2.2.5 GeoTools . . . . .	8

## *Inhaltsverzeichnis*

2.3	NoSQL . . . . .	8
2.3.1	Definition . . . . .	8
2.3.2	Kategorisierung . . . . .	9
2.3.3	Hadoop . . . . .	10
2.3.4	Accumulo . . . . .	11
2.3.5	MongoDB . . . . .	11
2.3.6	CouchDB . . . . .	11
2.3.7	Neo4J . . . . .	11
2.3.8	Rasdaman . . . . .	12
2.3.9	Spacebase . . . . .	12
2.3.10	Geomesa . . . . .	12
2.3.11	ESRI GIS Tools for Hadoop . . . . .	13
2.4	Leistungstests . . . . .	14
<b>3</b>	<b>methodisches Vorgehen</b>	<b>15</b>
<b>4</b>	<b>Ausgangsszenario</b>	<b>16</b>
4.1	Anforderungen . . . . .	16
4.2	Ist-Stand . . . . .	16
<b>5</b>	<b>System 1</b>	<b>17</b>
5.1	Aufbau . . . . .	17
5.2	Installation . . . . .	17
5.3	Datenimport . . . . .	17
5.4	Verarbeitung . . . . .	17
5.5	Schnittstelle . . . . .	17
5.6	Leistungstests . . . . .	17
<b>6</b>	<b>Gegenüberstellung</b>	<b>18</b>
6.1	Kosten . . . . .	18
6.2	Umfang . . . . .	18
6.3	Leistung . . . . .	18

## *Inhaltsverzeichnis*

<b>7</b>	<b>Fazit</b>	<b>19</b>
7.1	Zusammenfassung . . . . .	19
7.2	Wertung . . . . .	19
7.3	Ausblick . . . . .	19
	<b>Literaturverzeichnis</b>	<b>I</b>

# 1 Einleitung

## 1.1 Motivation

Die Agri Con GmbH verwaltet als Akteur im Bereich „Precision Farming“ täglich mehrere Millionen geografische Punktdaten. Diese Daten werden von aktiven Landwirtschaftsmaschinen und durch die Verarbeitung durch firmeninterne und firmenexterne Mitarbeiter sowie Systeme erzeugt. Weiterhin fallen dadurch indirekt Vektor- und Rasterdaten an, welche gespeichert und anschließend verarbeitet werden müssen. Aus den Quelldaten werden Vektordaten für beispielsweise Verteilung der Grunddüngung erzeugt. Rasterdaten werden für „N-Düngung“ verwendet, was unter anderem die Biomasse, die Nährstoffaufnahme und die Nährstoffverteilung beinhaltet. Diese Menge an Daten ist essentiell für den Betrieb, weshalb diese strukturiert gespeichert und kostengünstig verarbeitet werden müssen. Nicht nur Agri Con steht vor dieser Notwendigkeit, sondern der Großteil der Unternehmen, die sich mit komplexen Geodaten beschäftigen, wie Monsanto, Google, Facebook, ESRI, OpenGEO, etc.

## 1.2 Zielsetzung

Eine PostgreSQL Installation auf einem Computersystem stößt bei der aktuellen Nutzung durch Agri Con an die Leistungsgrenze. Aus diesem Grund ist die Speicherung und die Verarbeitung der Quelldaten in ein anderes System auszulagern. Dafür sind existierende Geoinformationssystem (GIS) zu untersuchen und deren Eignung für den in Kapitel 3 beschriebenen Anwendungsfall festzustellen. Der Schwerpunkt der Untersuchung sind die Möglichkeiten und Leistungsfähigkeit der räumlichen Datenverarbei-

## *1 Einleitung*

tung. Dabei werden NoSQL und Open-Source Systeme höher gewichtet. Aus geeigneten Systemen werden bis zu 3 ausgewählt. Die Auswahl wird speziell untersucht und eine prototypische Installation<sup>1</sup> erstellt. Somit sollen die Systeme mit dem Ist-Stand unter den Faktoren Kosten, Funktionalität und Leistungsfähigkeit verglichen werden.

### **1.3 Aufbau der Arbeit**

Zu Beginn werden theoretischen Grundlagen zu Datenbanken, geographischer Datenverarbeitung, NoSQL und Leistungstests festgehalten. Anschließend definiert Kapitel 3 das Ausgangsszenario, für welches die Systeme analysiert und getestet werden sollen. Die darauf folgenden Kapitel stellen die ausgewählten Systeme unter den Gesichtspunkten Aufbau, Installation, Datenimport, Verarbeitung, Schnittstelle und Leistungstest dar.

Das vorletzte Kapitel stellt die vorgestellten Systeme direkt gegenüber und führt die Daten zu Kosten, Umfang und Leistung auf. Die Thesis endet mit einer Zusammenfassung, einer Empfehlung bzw. Wertung der Ergebnisse und einem Ausblick auf die zukünftige Handhabung der räumlichen Daten bei Agri Con.

---

<sup>1</sup>Dabei kann eine Installation aus mehreren Systemen bestehen und eigens implementierte Funktionalitäten enthalten

## 2 Grundlagen

Computer

### 2.1 Datenbank

#### 2.1.1 Begriffsdefinitionen

##### ACID

Die bekanntesten Vertreter von relationalen Datenbanksystemen wie Oracle, MySQL und PostgreSQL arbeiten transaktional nach Atomicity, Consistency, Isolation und Durability (ACID). ACID kann nach [Kud07] S. 262 wie folgt definiert werden:

##### Atomarität

Transaktionen sind atomar, wodurch ein Abbruch einer Transaktion deren enthaltenen Operationen rückgängig macht.

##### Konsistenz

Das Ende oder der Abbruch einer Transaktion geht immer mit Nachbedingung aller Integritätsbedingungen einher.

##### Isolation

Transaktionen verschiedener Benutzer beeinflussen sich nicht gegenseitig.

##### Dauerhaftigkeit

Jede Änderung einer Transaktion ist nach Ende dieser auf die Festplatte geschrieben und nicht mehr im Puffer vorhanden.

## 2 Grundlagen

Die Definition dieses anerkannten Begriffes ist für Kapitel 2.3 notwendig.

### **MVCC**

In grundlegenden relationalen Systemen werden Transaktionen verzögert oder sogar gesperrt, um Konsistenz und Isolation zu gewährleisten. **Multi Version Currency Control (MVCC)** erhöht die Effizienz des blockierenden Verhaltens. Dabei werden von jedem Objekt mehrere Versionen verwaltet. Neue Versionen entstehen durch Änderungen einer anderen. Eine Transaktion verwendet die zu Transaktionsbeginn aktuelle Version. Dadurch werden die allgemeinen Sperrverfahren (siehe [Kud07] S. 266 ff.) verbessert, indem lesende Transaktionen sich nicht gegenseitig blockieren und schreibende gegen lesende Transaktionen nicht mehr synchronisiert werden müssen. (vgl. [Kud07] S. 270)

### **BASE**

**Basically Available, Soft state, Eventual consistency (BASE)** ist ein optimistischer und sperrenfreier Ansatz mit fließender Konsistenz. [SE11]

### **CAP**

**Consistency, Availability and Partition Tolerance (CAP)**

**Partition Tolerance**

**Eventual-Consistency**

**Consistent-Hashing**

### 2.1.2 Indexstrukturen

Indexstrukturen oder allgemein Zugriffsstrukturen dienen dem effizienten Zugriff auf Dateneinträge. Ein Index ist nach [Kud07] S. 284 wie folgt definiert:

*Ein Index ist ein Verzeichnis von Dateneinträgen der Form  $(k, k^*)$ , das den effizienten Zugriff auf allen Einträgen mit einem Suchschlüsselwert  $k$  erlaubt. Dabei bezeichnet  $k$  den Wert eines Suchschlüssels (auch Zugriffsattribut) und  $k^*$  den Verweis auf den Datensatz in der Datei, der  $k$  als Wert des Suchschlüssels enthält.*

Zugriffsstrukturen haben je nach Art und Umfang der Daten sowie entsprechend den Anforderungen an das Datenbanksystem (DBS) unterschiedliche Strukturen. In der einfachsten Struktur unterscheidet man nach Indexen die direkt die Daten beinhalten, auf die Daten zeigen oder eine Menge von Adressen beinhalten. (siehe [Kud07] S. 284)

Im folgenden werden spezielle Indexstrukturen vorgestellt, da dieses Wissen zur Bewertung von DBS herangezogen werden müssen.

#### B-Baum

*Der B-Baum ist ein dynamisch balancierter Indexbaum, bei dem jeder Indexeintrag auf eine Seite der Hauptdatei zeigt.<sup>1</sup>*

Der Baum besitzt die Höhe  $h$  und die Ordnung  $m$  sowie die folgenden Eigenschaften:

1. Jeder Weg von der Wurzel zum Blatt hat die Länge  $h$  (balanciert)
2. Jeder Knoten enthält mindestens  $m$  Elemente (außer der Wurzel) und höchstens  $2m$  Elemente (mindestens halbvolle Belegung)

---

<sup>1</sup>[Kud07] S. 288



## 2 Grundlagen

*3. Jeder Knoten ist entweder eine Blattseite oder hat höchstens  $2m + 1$  Kinder (maximale Belegung)<sup>2</sup>*

Diese Struktur garantiert eine Belegung von 50%. Weiterhin beschreibt  $h$  die Anzahl der Seitenzugriffe als relevantes Maß für die Zugriffskosten und Datensätze  $n$  bedingen den Zugriff in maximal  $\log m(n)$  Seitenzugriffen. (vgl. [Kud07] S. 288)

Eine Spezialisierung stellt der B+-Baum dar. Hierbei befinden sich die Dateneinträge ausschließlich in den Blattknoten. Die Blattknoten sind unidirektional verkettet.

### LSM-Baum

Log structured merge tree

### R-Baum

R-Bäume sind balancierte Bäume und „organisieren  $k$ -dimensionale Rechtecke mithilfe überlappender Blockregionen“ ([Kud07] S. 523) Diese Struktur wird folglich zur räumlichen Datenhaltung eingesetzt, da die Indexierung anhand räumlicher Informationen der Daten erfolgt. Ein Verzeichnisknoten besteht aus einem Tupel (ref, mur). ref steht für den Verweis auf den direkten Nachfahren und mur für das minimal umgebende Rechteck der Kindknoten. Datenknoten enthalten dagegen nur mur als eigentliches Geoobjekt. (vgl. [Kud07] S. 523 ff.)

### Geohash

## 2.1.3 Mehrrechner-Datenbanksystem

*Bei einem Mehrrechner-Datenbanksystem (MDBS) werden die Datenbankverwaltungsfunktionen auf mehreren Prozessoren bzw. Rechnern ausgeführt.<sup>3</sup>*

Kudraß ergänzt dies durch folgende Unterscheidungen:

---

<sup>2</sup>ebenda

<sup>3</sup>[Kud07] S. 394

## 2 Grundlagen

**shared everything** Datenbankmanagementsystem (DBMS) befindet sich auf eng gekoppelter Multiprozessor-Umgebung.

**shared nothing** Die Verarbeitung erfolgt durch mehrere Rechner mit jeweils einem DBMS, dabei ist der Externspeicher unter den beteiligten Rechnern partitioniert.

**shared disk** Hierbei handelt es sich um mehrere lokal angeordnete, lose oder nah gekoppelte Rechner mit je einem DBMS und einer gemeinsamen Speicherzuordnung. Lokal verteilte Systeme werden als parallele Datenbanksysteme bezeichnet.

### 2.1.4 Verteiltes Datenbanksystem

*Verteilte Datenbanksysteme (VDBS) sind geografisch verteilte Shared-Nothing Systeme mit homogenen lokalen DBMS, die gemeinsam ein globales konzeptionelles DB-Schema unterstützen. Förderierte Datenbanksysteme (FDBS) sind ebenfalls geografisch verteilte Shared nothing systeme, wobei die beteiligten lokalen DBMS eine höhere Autonomie aufweisen, d.h. dass jeweils eine eigene lokale Datenbank mit lokalem DB-schema vorliegt.<sup>4</sup>*

### 2.1.5 Replikationsverfahren

**Synchron**

**Asynchron**

**Kaskadiert**

---

<sup>4</sup>[Kud07] S. 398

## **2.2 geografische Datenverarbeitung**

### **2.2.1 Bezugssysteme**

### **2.2.2 Datenformate**

**Punkte**

**Vektoren**

**Raster**

**Shapefile**

### **2.2.3 GIS**

### **2.2.4 PostGIS**

### **2.2.5 GeoTools**

## **2.3 NoSQL**

### **2.3.1 Definition**

NoSQL steht für eine Bewegung der letzten 6 Jahre, in welcher die Abkehr von klassischen relationalen Systemen gefordert oder zumindest ein Umdenken bestehender Strukturen, Vorgehen und Grundsätze angestrebt wird. Dies wird durch andere Abfragesprachen, nicht relationale Datenbanksysteme oder Neudefinitionen von Begriffen wie der Konsistenz zum Ausdruck gebracht. Der Ursprung wird in der Literatur verschieden hergeleitet, jedoch wird immer zu den ersten Vertretern der NoSQL Bewegung Systeme mit einer anderen Abfragesprache und einfache Schlüssel-Hash Datenbanken gezählt. Auf einer Messe zu aktuellen Trends im Datenbankbereich wurde der Begriff NoSQL

## 2 Grundlagen

zuerst öffentlich für Lösungen dieser Bewegung verwendet [vgl. [Eva09]] und ist seitdem ein Sammelbegriff für eine hohe Anzahl an Systemen.

### NoSQL GIS

Ein GIS ist wie folgt definiert:

*Ein System, das auf einen Datenbestand zurückgreift und Auswertungen dieser Daten zulässt, so dass Informationen abgeleitet und wiedergegeben werden können, kann allgemein als ein Informationssystem bezeichnet werden. [...]*

*Im Mittelpunkt der Geoinformatik stehen mit den Geoinformationssystemen raumbezogene Informationssysteme, die im Gegensatz zu den übrigen Informationssystemen Geoobjekte der realen Welt modellieren und diese in ein digitales Informationssystem abbilden [...]. Die Gegenstände eines Geoinformationssystems besitzen wie auch bei allen anderen Informationssystemen eine Thematik (und Dynamik). Das Besondere bei Geoinformationssystemen ist, dass Geoobjekte darüber hinaus Geometrie und Topologie als implizite und untrennbare Bestandteile aufweisen! Die Verarbeitung derartiger raumbezogener Informationen erfordert spezielle Werkzeuge bzw. Funktionen, die von den übrigen Informationssystemen nicht bereitgestellt werden [...].<sup>5</sup>*

In Bezug auf NoSQL kann GIS ebenso definiert werden, jedoch muss das zugrunde liegende System nicht relational sein. Im Rahmen dieser Arbeit ist mit GIS ein System oder die Teilsysteme zur räumlichen Datenhaltung, Datenverarbeitung und Bereitstellung gemeint.

### 2.3.2 Kategorisierung

Edlich unterscheidet, wie andere Autoren, NoSQL Datenbanken nach vier Kategorien. Jedoch kann eine eindeutige Zuteilung nicht für jedes System erfolgen, da Prinzipien verschiedener Kategorien auf eines zutreffen können. Unter <http://nosql-database.org/> ist eine persönliche Übersicht der NoSQL Datenbanken von Herrn Edlich dargestellt.

---

<sup>5</sup>[Lan13] S. 337

### **Key Value Datenbank**

### **Dokumentenbasierende Datenbank**

### **Spaltenorientierte Datenbank**

### **Graphendatenbank Datenbank**

Der bekannteste Vertreter der graphenbasierten Datenbanken ist Neo4J. Alle Daten und deren Beziehungen werden in Form von Graphen persistiert. Ein Graph besteht dabei aus Knoten und gerichteten Kanten. Somit lassen sich direkt Beziehungen zwischen den Daten definieren.

### **2.3.3 Hadoop**

Hadoop ist ein unter der Apache Lizenz 2.0 stehendes Java-Framework zur Datenhaltung und Verarbeitung von großen Datenmengen auf mehreren Computern. Es basiert auf MapReduce und dem Dateisystem HDFS.

Hadoop File System (HDFS) ist ein verteiltes Dateisystem, welches keine besonderen Anforderungen an die Hardware stellt und für die Verwendung von mehreren hundert bis tausend Computern<sup>6</sup> ausgelegt ist. Es besitzt eine hohe Fehlertoleranz und ist für den Einsatz auf kostengünstiger Hardware ausgelegt. Hoher Datendurchsatz und die Verwendung großer Dateien<sup>7</sup> sind wesentliche Merkmale.[vgl. [Bor07] S. 3] Die Dateiblöcke werden redundant auf die Knoten verteilt und sind mit Hilfe des Name-Node abrufbar.[vgl. [Hä13] S. 7]

Die verteilte Verarbeitung übernimmt MapReduce. Entsprechend dem Namen entspringt der Name MapReduce aus der funktionalen Programmierung, in welcher die Funktionen „map“ und „reduce“ zum Einsatz kommen. So werden hier die Daten mit einer

---

<sup>6</sup>die in einem verteilten System teilnehmenden Computer heißen Knoten

<sup>7</sup>eine Datei kann mehrere Gigabyte bis mehrere Terrabyte groß sein und wird in Blöcke gleicher Länge aufgeteilt

## 2 Grundlagen

map-Funktion verändert und mit reduce-Funktion aggregiert. Ein Master weist die Daten und Funktionen den Slaves<sup>8</sup> zu. Die Slaves führen die Funktionen mit den ihnen zugewiesenen Daten aus und speichern ihre Ergebnisse auf deren Festplatte ab. MapReduce wurde von Google definiert. Auch hier werden keine besonderen Anforderungen an die Hardware gestellt.[vgl. [JD04] S. 3]

Hadoop besitzt eine Master-Slave Architektur, wobei der Name-Node<sup>9</sup> ankommende Anfragen bearbeitet und die Slave-Knoten organisiert. Hadoop ist per API verwendbar und bietet sich somit zur Stapelverarbeitung an. Es wird meist nur als Grundgerüst verwendet und mit Datenbanken wie HBase, MongoDB oder PostgreSQL sowie mit Frameworks für die Nutzung wie Hive, Pig, Spark oder Scalding erweitert.

### 2.3.4 Accumulo

[https://en.wikipedia.org/wiki/Apache\\_Accumulo](https://en.wikipedia.org/wiki/Apache_Accumulo)

### 2.3.5 MongoDB

### 2.3.6 CouchDB

### 2.3.7 Neo4J

---

<sup>8</sup>In diesem Zusammenhang auch Worker genannt

<sup>9</sup>damit ist der Master-Knoten gemeint, auch Jobtracker genannt

### 2.3.8 Rasdaman

[http://live.osgeo.org/de/overview/rasdaman\\_overview.html](http://live.osgeo.org/de/overview/rasdaman_overview.html) :

- Array-Datenbanksystem - PostgreSQL Aufsatz - Multi-Dimensionalität - eigene Anfragesprache - skalierend - unterstützt WCS Core und WCPS - Implementierte Standards: OGC WMS 1.3, WCS 2.0, WCS-T 1.4, WCPS 1.0, WPS 1.0 - Lizenz: Clients und APIs: GNU Lesser General Public License (LGPL) version 3; Server-Engine: GNU General Public License (GPL) version 3 - Unterstützte Plattformen: Linux, MacOS, Solaris - APIs: rasql, C++, Java

<http://www.rasdaman.org/> :

- open-source - "extends standard relational database systems with the ability to store and retrieve multi-dimensional raster data"

<http://www.rasdaman.de/> :

- "erlaubt die Ablage von unbeschränkt grossen multi-dimensionalen Arrays ("Rasterdaten") in einer konventionellen Datenbank"

### 2.3.9 Spacebase

<http://docs.paralleluniverse.co/spacebase/> :

- serverseitig - in-memory - spatial data store - ausgelegt für viele rechner und hohen Durchsatz (real-time) - 2D und 3D Objekte mit 3D bbox - load balancing enthalten - spatial queries möglich - benötigt JVM - API für Java, Ruby, Python, Node.js, C++, Erlang - API stellt nur elementare spatial queries zur verfügung: intersect oder contains - eigene spatial queries können definiert werden

### 2.3.10 Geomesa

- Ingest = Import über Kommandozeile (geomesa-tools) - Ingest von shp, csv und tsv Dateien - Anderer Dateiimport mit GeoTools - Verarbeitung nur über externe Tools (Spark, geotools) - Export: csv, tsv, shp, geojson, gml

## 2 Grundlagen

[http://www.eclipse.org/community/eclipse\\_newsletter/2014/march/article3.php](http://www.eclipse.org/community/eclipse_newsletter/2014/march/article3.php) :

- open-source - build on Accumulo and Hadoop - Supporting the GeoTools API - Geo-Server Plugin - geohash for indexing

<https://www.locationtech.org/proposals/geomesa> :

- outperforming postgis with geoserver

<http://de.slideshare.net/CCRinc/location-techdc-talk2-28465214> - Verwendung fraktaler Kurven - mit Spark und Scalding wesentlich schneller als PostGIS

[https://docs.google.com/presentation/d/1N00ppk8MfDs8Q-QcUIdZCSZK7YYwd9RjJoHV1V4Yq\\_w/edit?pli=1#slide=id.p](https://docs.google.com/presentation/d/1N00ppk8MfDs8Q-QcUIdZCSZK7YYwd9RjJoHV1V4Yq_w/edit?pli=1#slide=id.p) :

-

<https://github.com/locationtech/geomesa>:

- Apache License Version 2.0

### 2.3.11 ESRI GIS Tools for Hadoop

- 4 elements

- Esri Geometry API for Java: "This is a generic library that includes geometry objects, spatial operations, and spatial indexing, it can be used to spatially enable Hadoop. By deploying the Esri geometry API library (as a jar) within Hadoop, developers are able to build Map/Reduce applications that are spatially enabled, by leveraging the Esri Geometry API along with the other Hadoop APIs in their application." [ESR14]

- Spatial Framework for Hadoop: "This library includes the user defined objects that extend Hive with the capabilities of the Esri Geometry API. By enabling this library in Hive, users are able to construct queries that are very SQL like using HQL. In this case, users don't have to write a Map/Reduce application, they can interact with Hive, write their SQL like queries and get answers directly from Hadoop. Queries in this case can include spatial operations and values." [ESR14]

- Geoprocessing Tools for Hadoop: "These tools are specifically used in ArcGIS. Through the tools, users can connect to Hadoop from ArcGIS. Connecting to Hadoop from ArcGIS is really useful to the toolkit users, since they can import their analysis result in



## 2 Grundlagen

ArcGIS for Visualization. They can also do more complex and sophisticated analysis now that they narrowed down their data to a specific subset. Additionally, users can leverage the ArcGIS platform capabilities to publish their maps to web and mobile apps, and can integrate it with BI reports."[ESR14]

- GIS Tools for Hadoop: "This project is intended as a place to include multiple samples that leverage the toolkit. The samples can leverage the low level libraries, or the Geoprocessing tools. A couple of samples are available to help you test the deployment of the spatial libraries with Hadoop and Hive, and make sure everything runs with no issues before you start leveraging the setup from your HQL queries, or from the GP tools. To check your deployment, for Hive and GP tools usage, the sample point-in-polygon-aggregation-hive can be utilized. The sample leverages the data and lib directories on the same path."[ESR14] (Benötigt ArcGIS)

- Apache License, Version 2.0

### 2.4 Leistungstests

- siehe BA - in Absprache mit Prof. Riechert

### 3 methodisches Vorgehen

- Grundlagen sind notwendig zur Bewertung der Systeme - die exakte Definition der Anforderungen und des Ist-Standes ist Voraussetzung zur Bewertung - die Struktur der Untersuchung der Systeme dient dem einheitlichen Vergleich und der Nachvollziehbarkeit - konkret: Aufbau ist wichtig zur Beurteilung der Leistungsfähigkeit; Installation und Schnittstelle sollen aufzeigen, dass das System unter der definierten Umgebung lauffähig ist; Datenimport zeigt die Integrationsfähigkeit; Verarbeitung als wesentlicher Punkt, neben Aufbau Voraussetzung für Leistung, ; Leistungstests als Quelle zum Vergleich der Leistung - Verweis auf bestehende Vergleiche (Literatur)

# **4 Ausgangsszenario**

## **4.1 Anforderungen**

- UMN ist zur Darstellung zu verwenden - PostgreSQL mit PostGIS zum Datenimport und -export - kartografisches Produkt

## **4.2 Ist-Stand**

# **5 System 1**

## **5.1 Aufbau**

## **5.2 Installation**

## **5.3 Datenimport**

## **5.4 Verarbeitung**

## **5.5 Schnittstelle**

## **5.6 Leistungstests**

# **6 Gegenüberstellung**

## **6.1 Kosten**

## **6.2 Umfang**

## **6.3 Leistung**

# **7 Fazit**

## **7.1 Zusammenfassung**

## **7.2 Wertung**

## **7.3 Ausblick**

- auch Bezug auf Verarbeitung von ganzen Länderdaten mit dem System(en) - Darstellung als wichtige Komponente: Möglichkeiten und Performanz

# Literaturverzeichnis

- [Bor07] BORTHAKUR, Dhruba: The Hadoop Distributed File System, Architecture and Design / The Apache Software Foundation. 2007. – Forschungsbericht
- [ESR14] ESRI: *GIS Tools for Hadoop*. <http://esri.github.io/gis-tools-for-hadoop/>. Version: 2014
- [Eva09] EVANS, Eric: *NOSQL 2009*. [http://blog.sym-link.com/2009/05/12/nosql\\_2009.html](http://blog.sym-link.com/2009/05/12/nosql_2009.html). Version: 9 2009
- [Hä13] HÄBERLEIN, Dan: *Migration und Extrahierung von Datensätzen mittels spaltenorientierten Datenbanken am Beispiel von Apache HBase*, Universität Leipzig, Bachelor Thesis, September 2013. – Wirtschaftswissenschaftliche Fakultät
- [JD04] JEFFREY DEAN, Sanjay G.: MapReduce, Simplified Data Processing on Large Clusters / Google, Inc. 2004. – Forschungsbericht
- [Kud07] KUDRASS, Thomas (Hrsg.): *Taschenbuch Datenbanken*. Hanser, 2007
- [Lan13] LANGE, Norbert de: *Geoinformatik in Theorie und Praxis*. Bd. 3. Springer, 2013
- [SE11] STEFAN EDLICH, Jens Hampe Benjamin Brauer Markus B. Achim Friedland F. Achim Friedland: *NoSQL Einstieg in die Welt Nichtrelationaler Web 2.0 Datenbanken*. Bd. 2. Hanser, 2011

# Eidesstatliche Erklärung

Ich versichere, dass die Masterarbeit mit dem Titel „...“ nicht anderweitig als Prüfungsleistung verwendet wurde und diese Masterarbeit noch nicht veröffentlicht worden ist. Die hier vorgelegte Masterarbeit habe ich selbstständig und ohne fremde Hilfe abgefasst. Ich habe keine anderen Quellen und Hilfsmittel als die angegebenen benutzt. Diesen Werken wörtlich oder sinngemäß entnommene Stellen habe ich als solche gekennzeichnet.

Leipzig, 10. November 2014

Unterschrift