

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 28. Oktober 2014

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

Zweitprüfer: M. Sc. Volkmar Herbst

Abstrakt

Danksagung

Vorwort

Abbildungsverzeichnis

Tabellenverzeichnis

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vi
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	1
1.3 Aufbau der Arbeit	2
2 Grundlagen	3
2.1 Datenbank	3
2.1.1 Begriffsdefinitionen	3
2.1.2 Indexstrukturen	4
2.1.3 Mehrrechner-Datenbanksystem	4
2.1.4 Verteiltes Datenbanksystem	4
2.1.5 Replikationsverfahren	4
2.2 geografische Datenverarbeitung	5
2.2.1 Bezugssysteme	5
2.2.2 Datenformate	5
2.2.3 GIS	5
2.2.4 PostGIS	5
2.2.5 GeoTools	5
2.3 NoSQL	5
2.3.1 Definition	5
2.3.2 Kategorisierung	6
2.3.3 Hadoop	6

Inhaltsverzeichnis

2.3.4	Accumulo	7
2.3.5	MongoDB	7
2.3.6	CouchDB	7
2.3.7	Neo4J	7
2.3.8	Rasdaman	8
2.3.9	Spacebase	8
2.3.10	Geomesa	8
2.3.11	ESRI GIS Tools for Hadoop	9
2.4	Leistungstests	10
3	methodisches Vorgehen	11
4	Ausgangsszenario	12
4.1	Anforderungen	12
4.2	Ist-Stand	12
5	System 1	13
5.1	Aufbau	13
5.2	Installation	13
5.3	Datenimport	13
5.4	Verarbeitung	13
5.5	Schnittstelle	13
5.6	Leistungstests	13
6	Gegenüberstellung	14
6.1	Kosten	14
6.2	Umfang	14
6.3	Leistung	14
7	Fazit	15
7.1	Zusammenfassung	15
7.2	Wertung	15
7.3	Ausblick	15

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 Installation¹ 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.

¹Dabei kann eine Installation aus mehreren Systemen bestehen und eigens implementierte Funktionalitäten enthalten

2 Grundlagen

Computer

2.1 Datenbank

2.1.1 Begriffsdefinitionen

ACID

Atomicity, Consistency, Isolation und Durability (ACID)

MVCC

Multi Version Currency Control (MVCC)

BASE

Basically Available, Soft state, Eventual consistency (BASE)

2.1.2 Indexstrukturen

R-Baum

B-Baum

LSM-Baum

Geohash

2.1.3 Mehrrechner-Datenbanksystem

2.1.4 Verteiltes Datenbanksystem

2.1.5 Replikationsverfahren

Synchron

Asynchron

Kaskadiert

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 angestrebt wird. Dies wird durch andere Abfragesprachen, nicht relationale Datenbanksysteme oder Neudefinitionen von Begriffen wie der Konsistenz zum Ausdruck gebracht. Auf einer Messe zu aktuellen Trends im Datenbankbereich wurde der Begriff NoSQL zuerst öffentlich verwendet.[vgl.]

NoSQL GIS

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.

Key Value Datenbank

Dokumentenbasierende Datenbank

Spaltenorientierte Datenbank

Graphendatanbank 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 Computern¹ ausgelegt ist. Es besitzt eine hohe Fehlertoleranz und ist für den

¹die in einem verteilten System teilnehmenden Computer heißen Knoten

2 Grundlagen

Einsatz auf kostengünstiger Hardware ausgelegt. Hoher Datendurchsatz und die Verwendung großer Dateien² sind wesentliche Merkmale.[vgl. [?] S. 3] Die Datei-Blöcke werden redundant auf die Knoten verteilt und sind mit Hilfe des Name-Node abrufbar.[vgl. [?] 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 map-Funktion verändert und mit reduce-Funktion aggregiert. Ein Master weist die Daten und Funktionen den Slaves³ 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. [?] S. 3]

Hadoop besitzt eine Master-Slave Architektur, wobei der Name-Node⁴ 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

2.3.5 MongoDB

2.3.6 CouchDB

2.3.7 Neo4J

²eine Datei kann mehrere Gigabyte bis mehrere Terrabyte groß sein und wird in Blöcke gleicher Länge aufgeteilt

³In diesem Zusammenhang auch Worker genannt

⁴damit ist der Master-Knoten gemeint, auch Jobtracker genannt

2.3.8 Rasdaman

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 :

- 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://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."[?]

- 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."[?]

- 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."[?]

- 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."[?] (Benötigt ArcGIS)

- Apache License, Version 2.0

2.4 Leistungstests

- siehe BA - in Absprache mit Prof. Riechert

3 methodisches Vorgehen

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

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, 28. Oktober 2014

Unterschrift