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 14. November 2014

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

Zweitprüfer: M. Sc. Volkmar Herbst

Abstrakt

Danksagung

Vorwort

Glossar

Bonitur landwirtschaftliche Beurteilung des Ackers und der Pflanzen zum Zwecke der Planung des Einsatzes von Dünger, Pestiziden, Fungiziden und Herbiziden

GeoServer freier OGC konformer Mapserver der Open Source Geospatial Foundation, geschrieben in Java

UMN MapServer Mapserver des OGC unter MIT Lizenz, Erstentwicklung durch Universität von Minnesota, welcher als CGI Modul und für verschiedene Sprachen bereitsteht

Abkürzungsverzeichnis

ACID Atomicity, Consistency, Isolation und Durability

BASE Basically Available, Soft state, Eventual consistency

BSON Binary JSON

CAP Consistency, Availability and Partition Tolerance

DBMS Datenbankmanagementsystem

DBS Datenbanksystem

GIS Geoinformationssystem

GPL General Public License

HDFS Hadoop File System

JSON JavaScript Object Notation

LGPL Lesser General Public License

MVCC Multi Version Currency Control

Abbildungsverzeichnis

Tabellenverzeichnis

Inhaltsverzeichnis

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

Die aktuellen Werkzeuge kommen an ihre Grenzen wenn große Datenmengen zur Laufzeit bearbeitet werden müssen. Es ist zu untersuchen welche Vorteile andere Datenhaltungssysteme bieten bzw. welche alternativen Herangehensweisen wie NoSQL, die Verwendung von caching und die verteilte Datenhaltung verwendet werden können. Dafür sind existierende Geoinformationssysteme (GISs) zu untersuchen und deren Eignung für den in Kapitel 3 beschriebenen Anwendungsfall festzustellen. Der Schwerpunkt der

Untersuchung sind die Möglichkeiten und Leistungsfähigkeit der räumlichen Datenverarbeitung. 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[1] 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.

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

2 Grundlagen

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 Intergritätsbedingungen 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.

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. [Edl11]

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.*[1]

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)*
> *3. Jeder Knoten ist entweder eine Blattseite oder hat höchstens 2m + 1 Kinder (maximale Belegung)*[2]

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 logm(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.

[1] [Kud07] S. 288
[2] ebenda

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.*[3]

Kudraß ergänzt dies durch folgende Unterscheidungen:

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.

[3] [Kud07] S. 394

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 Sgated nothing systeme, wobei die beteiligten lokalen DBMS eine höhere Autonomie aufweisen, d.h. dass jeweils eine eigene lokale Datenbank mit lokalem DB-schema vorliegt.[4]

2.1.5 Replikationsverfahren

Synchron

Asynchron

Kaskadiert

[4] [Kud07] S. 398

2.2 geografische Datenverarbeitung

2.2.1 Bezugssysteme

> *Räumliche Bezugssysteme (spatial reference systems) erlauben die Interpretation der gespeicherten Koordinaten als Beschreibung von Lage- und Ausdehnungsinformationen in einem (realen) Datenraum. Ein räumliches Bezugssystem besteht aus einem Koordinatensystem (coordinate system), einem Geltungsbereich und Angaben, die es erlauben, Daten aus unterschiedlichen Koordinatensystemen auf ein globales System abzubilden.*[5]

Kudraß allgemeine Definition wird durch [Lan13] S. 141 ff. mit folgendem ergänzt: Man unterscheidet Koordinatensysteme nach kartesisch, homogen, Kugeltransformation und Ellipsoidentransformation, wobei den kartesischen einer hoher Stellenwert zugegordnet wird.

Allen Bezugssystemen wird zur Identifikation ein weltweit eindeutiger Code zugeordnet. Dieser ist ein von der European Petroleum Survey Group Geodesy vergebener so genannter EPSG Code.

Das auf einem Ellipsoiden basierende Bezugssystem World Geodetic System von 1984[6] wird von der Agricon GmbH verwendet.

2.2.2 Datenformate

> *Geoobjekte sind räumliche Elemente, die zusätzlich zu Sachinformationen geometrische und topologische Eigenschaften besitzen und zeitlichen Veränderungen unterliegen können. Kennzeichnend für Geoobjekte sind somit Geometrie, Topologie, Thematik und Dynamik.*[7]

De Lange definiert räumliche Objekte bzw. Geoobjekte ausreichend. Ein Geoobjekt enthält als Geometrie eine oder mehrere zwei- oder dreidimensionale Koordinaten, was

[5] [Kud07] S. 506
[6] EPSG:4326
[7] [Lan13] S. 133

die Lage, den Umfang und die Ausdehnung beschreibt. Zur Topologie zählt de Lange Umgebungen, Nachbarschaften, Teilmengen und Überlagerungen. Weiterhin werden Geoobjekte mit Sachinformationen gespeichert und je nach Anwendungsfall versioniert.[vgl. [Lan13] S. 133]

einfache Geoobjekte

Ein Punkt besteht aus einer zwei- oder dreidimensionalen Koordinate und beliebigen Sach-, Topologie- und Dynamikinformationen. Mehrere Punkte bilden Linien. Bildet eine Linie eine geschlossene Fläche, handelt es sich um ein Polygon.

Vektorenmodell

Es besteht die Möglichkeit eine Menge von Punkten als Vektoren aufzufassen und daraus topologische Objekte entstehen zu lassen. Um damit geografisch zu modellieren, ist eine Diskretisierung d.h. eine Zuordnung der Vektoren notwendig.

Rastermodell

Ein Raster löst einen rechteckigen Bereich mit in einem Koordinatensystem gleichmäßig angeordneten quadratischen Bildelementen bzw. Pixeln fester Größe auf. Geodaten werden ergo mit einer indizierten Matrix abgebildet. Ein dreidimensionales Raster heißt Voxel.

> *Ein Punkt wird näherungsweise durch ein einzelnes Pixel dargestellt. Ein Linienzug wird durch entsprechende Anordnungen zusammenhängender Pixel angenähert erfasst. Linienzüge können dann z.B. durch Folgen von Indexpaaren (Zeile, Spalte) der zugehörigen Pixel beschrieben werden. Eine Fläche ist ebenfalls durch zusammenhängende Pixel darstellbar. Somit sind keine weiteren Zusatzinformationen zur Modellierung von Flächen wie im Vektormodell notwendig [...].*[8]

[8] [Lan13] S. 136

Shapefile

2.2.3 Operationen

Aggregation

Filterung

Geostatistik

Interpolation

Resampling

2.2.4 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 [...].*[9]

[9] [Lan13] S. 337

2.2.5 GDAL

2.2.6 PostGIS

2.2.7 GeoTools

2.3 NoSQL

2.3.1 Definition

NoSQL steht für eine Bewegung des letzten Jahrzehnts, 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 zuerst öffentlich für Lösungen dieser Bewegung verwendet [vgl. [Eva09]] und ist seitdem ein Sammelbegriff für eine hohe Anzahl an Systemen.

NoSQL GIS

In Bezug auf NoSQL kann GIS wie in 2.2.4 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. Für dieses Kapitel diente wesentlich [Jun13] als Quelle.

Key Value Datenbank

Key Value Datenbanken speichern Daten in Tupeln aus Schlüssel und Wert. Der Key ist eine Zeichenkette oder ein Hashwert und der Datentyp von Value ist beliebig im Rahmen der Datentypen der Datenbank. Datenzugriff erfolgt über Key. Es existiert keine einheitliche Abfragesprache.

Erste Datenbanken die zu NoSQL zugeordnet werden sind Key Value Datenbanken. Konkret DBM und BerkleyDB. Aktuelle Vertreter sind Amazon Dynamo, Riak, Voldemort und Redis.

Diese Datenbanken eignen sich für heterogene Daten, horizontale Skalierung und Schemafreiheit, da diese einfach strukturierten Daten sich in keiner Relation zueinander befinden.

Dokumentenbasierende Datenbank

Hierbei werden strukturierte Daten, hier Dokumente, unter einem Hash abgelegt und sind über diesen abrufbar. Diese Dokumente sind im großteil der dokumentenbasierten Datenbanken versioniert. Häufige Formate sind JavaScript Object Notation (JSON), Binary JSON (BSON) und YAML.

Ziel ist hier schemafreie Daten zu speichern und den Zugriff zu skalieren. Dabei können zumeist keine Joins verwendbar.

Bekannte Vertreter sind MongoDB, CouchDB und Terrastore.

Spaltenorientierte Datenbank

Im Gegensatz zu zeilenorientierten Datenbanken legen spalteniorientierte Datenbanken ihre Werte, hier Attribute einer Tabelle, spaltenweise ab.

Dies eignet sich für OLAP und Data Warehouse, da Spalteneinfügungen kostengünstig und Garbage Collection effektiv ist. Dagegen besteht ein hoher Aufwand beim Lesen und Schreiben von zusammengehörigen Spaltendaten.

Googles Big Table erweitert diesen Ansatz und beschreibt es in dessen Paper wie folgt:

> *A Bigtable is a sparse, distributed, persistent multi-dimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an uninterpreted array of bytes.*[10]

Die mehrdimensionalen Tabellen oder Maps sind vom Format:
$n * [Domain/Keyspace]\ x\ [item/Column\ Family]\ x\ [Key\ x] * n * [key + Value]$

Googles Ansatz wurde OpenSource in HBase und Cassandra umgesetzt. Die konkrete Implementierung von Google wurde jedoch nicht veröffentlicht. HBase verwendet folgendes Format: Pro Table Zugriff auf Zeilen per Rowkey, diese enthalten Column Familys oder Spalten welche wiederum eine Map namens Column Qualifier mit Tupeln aus der Version als Schlüssel und ein Byte-Array als Wert besitzt.[vgl. [Hä13] S. 13]

Graphenbasierte 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. Knoten sind dabei strukturierte Objekte und Kanten Beziehungen zwischen den Objekten. Diese strukturierten Objekte sind Key Value Tupel. Kanten können typisiert sein. Somit lassen sich direkt Beziehungen zwischen Daten definieren, was sich für semantic web, social network, Bioinformatik und Internetrouting eignet.

[10] [FC06] S. 1

Diese Datenbanken sind nur optional mit einem Schema versehen und besitzen keine einheitliche Abfragesprache. Auch sind im allgemeinen keine Joins vorgesehen.

2.3.3 Hadoop

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

Das 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[11] ausgelegt ist. Es besitzt eine hohe Fehlertoleranz und ist für den Einsatz auf kostengünstiger Hardware ausgelegt. Hoher Datendurchsatz und die Verwendung großer Dateien[12] sind wesentliche Merkmale.[vgl. [Bor07] S. 3] Die Datei-Blöcke 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 map-Funktion verändert und mit reduce-Funktion aggregiert. Ein Master weist die Daten und Funktionen den Slaves[13] 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. In Abbildung 2.1 ist der beschriebe Ablauf dargestellt. Auch hier werden keine besonderen Anforderungen an die Hardware gestellt.[vgl. [JD04] S. 3]

Hadoop besitzt eine Master-Slave Architektur, wobei der Name-Node[14] 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

[11] die in einem verteilten System teilnehmenden Computer heißen Knoten
[12] eine Datei kann mehrere Gigabyte bis mehrere Terrabyte groß sein und wird in Blöcke gleicher Länge aufgeteilt
[13] In diesem Zusammenhang auch Worker genannt
[14] damit ist der Master-Knoten gemeint, auch Jobtracker genannt

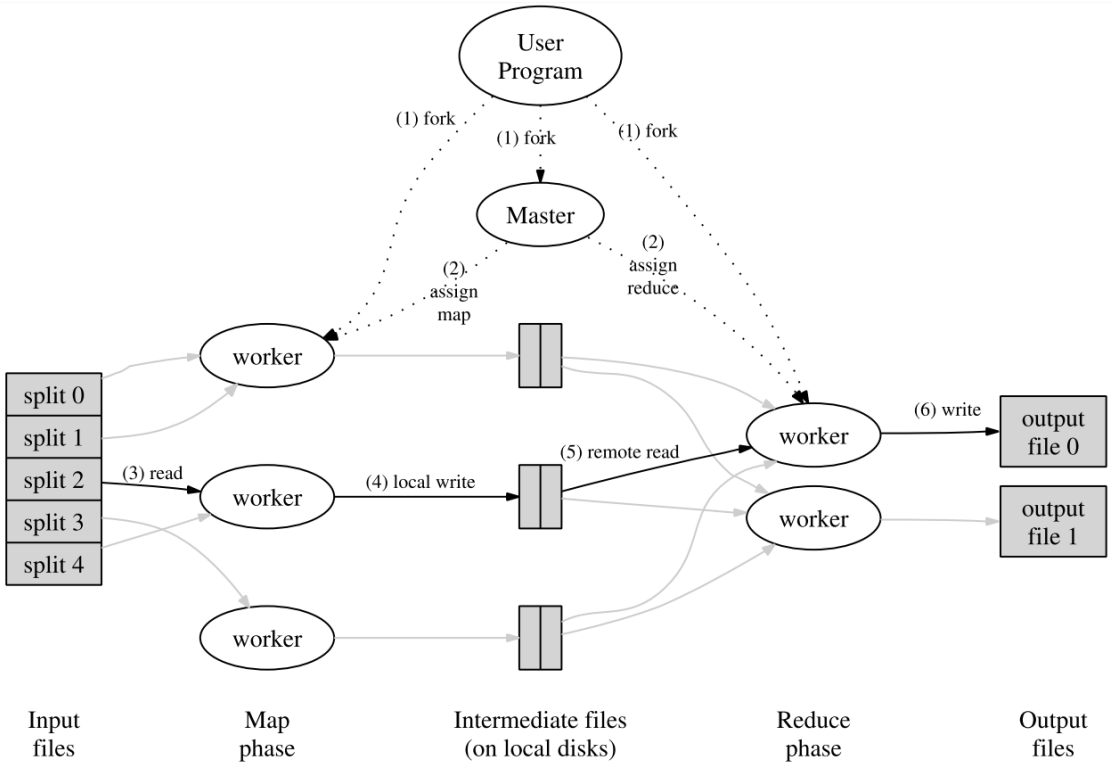

Abbildung 2.1: Übersicht der Ausführung von Googles MapReduce, Quelle: [JD04] S. 3

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

2.3.8 Rasdaman

Rasdaman ist ein Array-Datenbanksystem speziell zum speichern und verarbeiten von Rasterdaten. Es erweitert eine relationale Datenbank und wird mit multi-dimensionalität der Daten, einer eigenen SQL ähnlichen Abfragesprache, Parallelisierung und Skalierbarkeit in beliebigen Maßstab sowie OGC konformen Diensten beworben. Es ist als Client bzw. API unter der Lesser General Public License (LGPL) 3 und als Server unter der General Public License (GPL) 3 für Linux, MacOS und Solaris verfügbar. Als OGC konforme Dienste werden WMS 1.3, WCS 2.0, WCS-T 1.4, WCPS 1.0 und WPS 1.0 bereitgestellt. Die API kann in Java, C++ und über die eigene Abfragesprache rasql verwendet werden.[vgl. [Liv14]]

Es besteht die Möglichkeit, Rasdaman zu einer bestehenden PostgreSQL zu installieren und direkten Datenaustausch zwischen den beiden Systemen zu ermöglichen. Weiterhin kann Rasdaman in Verbindung mit GDAL verwendet werden. Momentan existiert eine Community und eine Enterprise Variante. Dabei verfügt die Enterprise Variante über mehr Features wie beispielsweise Datenkomprimierung, Serververwaltung per Webbrowser, Laufzeitoptimierungen und verschiedene Datenbankschnittstellen. Von der verwendeten Datenbank wird BLOB als Datenbankinterner Datentyp verwendet.[vgl. [Ras14]]

2.3.9 Spacebase

Spacebase ist eine in Java programmierte geografische Datenbank mit Betonung auf Echtzeit und geringe Latenzen des Unternehmens Parallel Universe. Die Datenbank wird ausschließlich im Arbeitsspeicher gehalten und ausgeführt. Außerdem ist sie für die verteilte Nutzung auf mehreren Computern konzipiert. Räumliche Daten werden in 2D oder 3D mit einem dazugehörigen begrenzenden Rechteck im R-Baum gespeichert. SpaceBase ist für eine sehr große Anzahl an Abrufen und Veränderungen der Daten in Echtzeit geeignet. Für die räumliche Verarbeitung stehen Bereichsabfragen, Überschneidungsabfragen und allgemeine Join Abfragen zur Verfügung. Diese beschränken sich aber auf das begrenzende Rechteck, da die eigentliche Geometrie nur über eine Referenz im Datenobjekt abrufbar ist. Ergänzend besteht die Möglichkeit eigene Abfragen

zu formulieren und dabei die referenzierte Geometrie zu verwenden. Es wird jedoch darauf hingewiesen, dass complexe Operationen auf diese Geometrie die Laufzeit der Berechnungen wesentlich erhöht. Zum Erreichen der geringen Latenzen werden räumliche Berechnungen parallelisiert und auf die Instanzen verteilt. Es stehen APIs für Java, C++, Erlang, Python, Ruby und Node.js zur Verfügung.[vgl. [Uni14]]

2.3.10 Geomesa

Geomesa ist eine freie[15] geografische Datenbank der Firma LocationTech mit den Möglichkeiten der verteilten Verarbeitung und Versionierung der Daten. Es erweitert Accumulo[16], unterstützt die GeoTools API und bietet ein Plugin für den Mapserver GeoServer an. Die Daten werden nach Geohash[17] verwaltet.[vgl. [Fox14]]
Geomesa wird in Verbindung mit stream processing[18] und batch processing[19] verwendet.

Datenimport wird ingest genannt, erfolgt über die Kommandazeile und unterstützt die Datenformate CSV, TSV und SHP. CSV, TSV, Shapefile, GeoJSON, and GML können dagegen exportiert werden.

- Anderer Dateiimport mit GeoTools - Verarbeitung nur über externe Tools (Spark, geotools)

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/1NOOppk8MfDs8Q-QcUidZCSZK7YYwd9RjJoHV1V4Yq_w/edit?pli=1#slide=id.p :
-

[15] Apache License Version 2.0
[16] siehe 2.3.4
[17] siehe 2.1.2
[18] bspw. Spark oder Storm
[19] bspw. Pig oder Cascading

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 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

- Laufzeittests von Operationen und Berechnungen in form von Lasttests - Leistung= erfolgreichen Verarbeitungsvorgängen pro Zeiteinheit unter gegebenen Vorrausetzungen (selbe algorithmen, ähnliche Umgebung) - Lasttests zur Persistierung von Eingangsdaten - Lasttests zur Bereitstellung von verarbeiteten und eingabedaten (an UMN[plugin, WMS, Bilder oder über pgsql] oder allgemein) - Überwachung der Systemauslastung ist notwendig -

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

Aktuelle Möglichkeiten der Datenerfassung über Sensoren und moderne Probenahmegeräte führen zu mehr und mehr Datensätzen, die für einen Landwirtschaftsbetrieb ausgewertet werden müssen. Darüber hinaus besteht die Notwendigkeit, Daten Jahresübergreifend und betriebsübergreifend auszuwerten, um pflanzenbauliche Zusammenhänge über statistische Methoden untersuchen zu können. In den letzten 3 Jahren wurde beispielsweise nur zum Thema N-Versorgung[1] für einen Betrieb etwa 800 Datensätze mit 1,9 Mio Einträgen erfasst. Alle diese Daten haben einen räumlichen Bezug, sie müssen weiterverarbeitet, kartographisch aufbereitet und dargestellt werden.
Daraus ergeben sich verschiedenen Anforderungen an die Technologie, die für die Verarbeitung, Analyse und Darstellung verwendet wird:

- PostgreSQL mit PostGIS zum Datenimport und -export nutzbar
- Gruppieren und Filtern mit geringer Laufzeit
- parallele Berechnung[2] über große Datenmengen mit geringer Laufzeit
- Räumliche Berechnungen wie Verschneiden, Berechnen von Overlays
- Unterschiedliche Prinzipien der Kartengenerierung, hier dynamisches rendern aus dem Datenbestand zur Laufzeit oder dynamisches rendern bei Dateneingang wodurch vorgerenderte Karten bereitstehen
- nutzbare Schnittstelle zur Darstellung mit dem UMN MapServer

[1] Stickstoffdüngung und -aufnahme
[2] hier Statistik bzw. Geostatistik sowie Interpolation

Konkret handelt es sich bei den Eingangsdaten um folgende:

Pflanzenbauliche Daten [3] Punktdaten

Basisdaten wie Feldgrenzen Vektordaten

Externe Satelliteninformationen und Multispektralanalysen Rasterdaten

4.2 Ist-Stand

[3] Sensoren, Bodenuntersuchung, Bonitur, Logger

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

[Edl11] EDLICH, Stefan (Hrsg.): *NoSQL : Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken*. 2., aktualisierte und erw. Aufl. München : Hanser, 2011. – XVIII, 392 S. http://deposit.d-nb.de/cgi-bin/dokserv?id=3791928&prov=M&dok%5Fvar=1&dok%5Fext=htm. – ISBN 9783446427532

[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. Version: 9 2009

[FC06] FAY CHANG, Sanjay Ghemawat Wilson C. Hsieh Deborah A. Wallach Mike Burrows Tushar Chandra Andrew Fikes Robert E. G. Jeffrey Dean D. Jeffrey Dean: Bigtable: A Distributed Storage System for Structured Data / Google, Inc. 2006. – Forschungsbericht

[Fox14] FOX, Anthony: *GeoMesa: Scaling up Geospatial Analysis*. http://www.eclipse.org/community/eclipse_newsletter/2014/march/article3.php. Version: 3 2014

[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

[Jun13] JUNGHANNS, Kurt: *NOSQL*. 5 2013. – Präsentationsfolien zum Oberseminar Datenbanken - aktuelle Trends

[Kud07] KUDRASS, Thomas (Hrsg.): *Taschenbuch Datenbanken : Mit ... 28 Tabellen*. München : Fachbuchverl. Leipzig im Carl-Hanser-Verl., 2007. – 582 S. http://swbplus.bsz-bw.de/bsz260569755inh.htm. – ISBN 3446409440

[Lan13] LANGE, Norbert d.: *Geoinformatik in Theorie und Praxis*. 3. Springer, 2013

[Liv14] LIVE, OSGeo: *Rasdaman*. http://live.osgeo.org/de/overview/rasdaman_overview.html. Version: 2014

[Ras14] RASDAMAN: *Welcome to rasdaman – the World's Leading Array Database*. http://www.rasdaman.org/. Version: 2014

[Uni14] UNIVERSE, Parallel: *Overview*. http://docs.paralleluniverse.co/spacebase/. Version: 2014

Eidesstatliche Erklärung

Ich versichere, dass die Masterarbeit mit dem Titel „Untersuchung und Optimierung verteilter Geografischer Informationssysteme zur Verarbeitung Agrartechnischer Kennzahlen“ 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, 14. November 2014 Unterschrift