

Apache Hadoop

Hauptseminar "Cloud-Plattformen und Big Data"

Dozent Steffen Rupp

von

René Gentzen

`rene.gentzen@mni.thm.de`

im WS22/23

Inhaltsverzeichnis

1	Hadoop Grundlagen	1
1.1	Hadoop Historie	1
1.1.1	Anforderungen von Big Data	1
1.1.2	Vertikale Skalierung	2
1.1.3	Horizontale Skalierung	2
1.1.4	GFS und MapReduce	3
1.2	Hadoop 1.0 Aufbau	3
1.2.1	HDFS in depth theoretisch	3
1.2.2	MapReduce in depth theoretisch	3
1.3	Hadoop 2.x mit YARN	3
1.3.1	Neue Möglichkeiten mit YARN	3
1.3.2	Neuer Aufbau, neue Verpflichtungen	3
1.3.3	Code einer YARN Application	3
1.4	Hadoop 3.x anreißen	3
2	Das Hadoop Ecosystem	5
2.1	Storage	6
2.1.1	HDFS	6
2.1.2	HBase	6
2.2	Management / Configuration	6
2.2.1	YARN	6
2.2.2	Oozie	6
2.2.3	ZooKeeper	6
2.2.4	Ambari	6
2.3	Datentransfer	6
2.3.1	Sqoop	6
2.3.2	Kafka	6
2.3.3	AVRO	6
2.4	Processing	6
2.4.1	MapReduce	6
2.4.2	Pig	6
2.4.3	Hive	6
2.4.4	Flume	6
2.4.5	Mahout	6
2.4.6	Spark	6
2.4.7	Solr	6

3	Hadoop Operations	7
3.1	Hadoop Setup	7
3.1.1	Single Node Setup	7
3.1.2	Pseudo-Distributed Setup	7
3.1.3	Fully-distributed Cluster	7
3.1.4	Docker Images	7
3.1.5	VM Distributionen	7
3.1.6	Hadoop in der Cloud	7
3.2	Praxis	7
3.2.1	Start eines Single Node Clusters lokal	7
3.2.2	Erste Übung zum Umgang mit dem HDFS	7
3.2.3	MapReduce Workflow: Erstellung eines MapReduce Jobs, Kopieren auf den Name Node und Ausführung	7
3.2.4	YARN Application auf einem Cluster in der Cloud laufen lassen	7
4	ETL mit Pig	9
4.1	Anwendungsfälle	9
4.2	Architektur	9
4.3	Pig Latin	9
4.4	Praxis	9
4.4.1	Hinzufügen zum Cluster	9
4.4.2	Anwendung auf dem Cluster	9
5	Data Ingestion	11
5.1	Sqoop	11
5.2	Flume	11
6	Datawarehousing mit Hive	13
6.1	Anwendungsfälle	13
6.1.1	Unterschiede zu Pig	13
6.2	Architektur	13
6.3	Interaktion	13
6.3.1	HiveQL	13
6.3.2	CLI	13
6.3.3	Java API	13
6.4	Praxis	13
6.4.1	Hinzufügen zum Cluster	13
6.4.2	Einrichtung einer Datenbank	13
6.4.3	Einlesen von Daten im CLI	13
6.4.4	Einlesen von Daten mit Sqoop	13
6.4.5	Absetzen einer Query	13
7	NoSQL mit HBase	15
7.1	Anwendungsfälle	15
7.1.1	CAP-Theorem	15

7.1.2	ACID und BASE	15
7.2	Architektur	15
7.3	Interaktion	15
7.3.1	HBase Shell	15
7.3.2	Java API	15
7.4	Praxis	15
7.4.1	Hinzufügen zum Cluster	15
7.4.2	Einrichtung einer Datenbank	15
7.4.3	Einlesen von Daten	15
7.4.4	Datenmigration aus einem RDBMS	15
7.4.5	Absetzen einer Query	15
8	Streaming mit Kafka	17
8.1	Anwendungsfälle	17
8.2	Architektur	17
8.3	Interaktion	17
8.3.1	Who knows	17
8.4	Praxis	17
8.4.1	Hinzufügen zum Cluster	17
8.4.2	Maybe, vielleicht kann man ja was zeigen	17
9	Hadoop heute	19
9.1	Aktuelle Anwendungsbeispiele zu Hadoop	19
9.1.1	AirBnB	19
9.2	Apache Spark als Gold Standard	19
9.2.1	Kann eh alles besser	19
	Literatur	21

1 Hadoop Grundlagen

“Die Apache Hadoop Softwarebibliothek ist ein Framework, das die über Computercluster verteilte Verarbeitung großer Datensätze mit einfachen Programmiermodellen ermöglicht. Es ist so konzipiert, dass es von einzelnen Servern bis hin zu Tausenden von Rechnern skaliert werden kann, von denen jeder lokale Rechenleistung und Speicherplatz bietet. Anstatt sich auf Hardware zu verlassen, um eine hohe Verfügbarkeit zu gewährleisten, ist die Bibliothek selbst so konzipiert, dass sie Ausfälle auf der Anwendungsebene erkennt und bewältigt, so dass ein hochverfügbarer Dienst auf einem Cluster von Computern bereitgestellt wird, von denen jeder für sich für Ausfälle anfällig sein kann.”[1]

So beschreibt (übersetzt aus dem Englischen) die Apache Software Foundation ihr Top Level Projekt Apache™ Hadoop®. Diese Arbeit wird einen pragmatischen Überblick über Hadoop und die Komponenten im Hadoop Ecosystem geben. Dabei soll der Fokus nicht auf technischen Details, sondern auf Anwendungsorientiertheit liegen. Es sollen konkrete Anwendungsfälle zu den einzelnen Komponenten besprochen werden und nicht abstrakte Architekturkonzepte.

1.1 Hadoop Historie

1.1.1 Anforderungen von Big Data

“Der Begriff „Big Data“ bezieht sich auf Datenbestände, die so groß, schnelllebig oder komplex sind, dass sie sich mit herkömmlichen Methoden nicht oder nur schwer verarbeiten lassen.”[2]

Schon Anfang der Neunziger war es nicht mehr praktikabel, Webseiten händisch, zum Beispiel in “Web Directories”, zu katalogisieren. Man wollte Nutzern trotzdem die Möglichkeit geben, Informationen durch das Durchsuchen zentraler Anlaufstellen ausfindig zu machen. Automatisierte Tools, die sogenannten “Web Crawler” wurden erfunden, um diese Arbeit zu übernehmen.[3]

Das Internet erlebte in den letzten Jahren des 20. Jahrhunderts ein explosionsartiges Wachstum an Nutzern und Webseiten, und damit auch an Informationen, die katalogisiert werden mussten.[4] Um eine immer größer werdende Menge an Informationen verarbeiten zu können, gibt es zwei Ansätze der Skalierung: Vertikale und horizontale Skalierung. Diese sollen in den folgenden Abschnitten erläutert werden.

1.1.2 Vertikale Skalierung

Bei der vertikalen Skalierung ("scaling up") werden *einem* System mehr Ressourcen wie zum Beispiel größerer Speicher, oder eine schnellere CPU hinzugefügt. Dadurch bekommt man einen Performance-Gewinn: Man kann mehr Daten speichern, oder Berechnungen werden schneller fertig gestellt. Ein großer Vorteil der vertikalen Skalierung ist, dass Anwendungsprogramme in der Regel nicht angepasst werden müssen, um vom diesem Performance-Wachstum zu profitieren. Wenn man eine 5TB große Festplatte gegen eine 10TB Festplatte austauscht, dann hat man den Speicherplatz eines Servers vertikal skaliert. Die darauf laufenden Programme müssen nicht angepasst werden, sondern man kann einfach doppelt so viele Daten speichern.[5]

Vertikale Skalierung hat drei große Nachteile: Erstens kann man nicht unbegrenzt vertikal skalieren. Ein Server kann physisch nur eine begrenzte Anzahl an Hardware aufnehmen. Zweitens wächst die Performance eines Systems bei vertikaler Skalierung höchstens linear[6], die Kosten allerdings nicht. Heutzutage kann man gerade bei Cloud-Anbietern sehr leistungsfähige Systeme bei linearem Preisanstieg mieten.[7] Sucht man aber noch mehr Performance in *einem* System, dann steigen die Kosten exponentiell[8]. Drittens skalieren nicht alle Faktoren in einem System gleich gut vertikal. Die Speicherkapazität von SSDs ist zum Beispiel seit 1978 von 45MB auf 100TB gestiegen (Faktor $2222,22 \cdot 10^3$), während sich die Datenrate nur von 1.5MB/s auf 500/460MB/s (Sequential Read/Write) erhöht hat (Faktor $0,333 \cdot 10^3$).[9][8]

1.1.3 Horizontale Skalierung

Bei der horizontalen Skalierung ("scaling out")

1.1.4 GFS und MapReduce

1.2 Hadoop 1.0 Aufbau

1.2.1 HDFS in depth theoretisch

1.2.2 MapReduce in depth theoretisch

1.3 Hadoop 2.x mit YARN

1.3.1 Neue Möglichkeiten mit YARN

1.3.2 Neuer Aufbau, neue Verpflichtungen

1.3.3 Code einer YARN Application

1.4 Hadoop 3.x anreißen

2 Das Hadoop Ecosystem

2.1 Storage

2.1.1 HDFS

2.1.2 HBase

2.2 Management / Configuration

2.2.1 YARN

2.2.2 Oozie

2.2.3 ZooKeeper

2.2.4 Ambari

2.3 Datentransfer

2.3.1 Sqoop

2.3.2 Kafka

2.3.3 AVRO

2.4 Processing

2.4.1 MapReduce

2.4.2 Pig

2.4.3 Hive

2.4.4 Flume

2.4.5 Mahout

2.4.6 Spark

2.4.7 Solr

3 Hadoop Operations

3.1 Hadoop Setup

3.1.1 Single Node Setup

3.1.2 Pseudo-Distributed Setup

3.1.3 Fully-distributed Cluster

3.1.4 Docker Images

3.1.5 VM Distributionen

3.1.6 Hadoop in der Cloud

HDFS oder Cloud FS

3.2 Praxis

3.2.1 Start eines Single Node Clusters lokal

3.2.2 Erste Übung zum Umgang mit dem HDFS

3.2.3 MapReduce Workflow: Erstellung eines MapReduce Jobs, Kopieren auf den Name Node und Ausführung

3.2.4 YARN Application auf einem Cluster in der Cloud laufen lassen

4 ETL mit Pig

Auch wenn es vermehrt von Spark verdrängt wird

4.1 Anwendungsfälle

Welche neuen Dinge ermöglicht dieses Tool Eine Zeile Pig Latin entspricht vielen Zeilen MapReduce

4.2 Architektur

4.3 Pig Latin

4.4 Praxis

4.4.1 Hinzufügen zum Cluster

4.4.2 Anwendung auf dem Cluster

5 Data Ingestion

5.1 Sqoop

5.2 Flume

6 Datawarehousing mit Hive

6.1 Anwendungsfälle

Welche neuen Dinge ermöglicht dieses Tool

6.1.1 Unterschiede zu Pig

6.2 Architektur

6.3 Interaktion

6.3.1 HiveQL

6.3.2 CLI

6.3.3 Java API

6.4 Praxis

6.4.1 Hinzufügen zum Cluster

6.4.2 Einrichtung einer Datenbank

6.4.3 Einlesen von Daten im CLI

6.4.4 Einlesen von Daten mit Sqoop

6.4.5 Absetzen einer Query

7 NoSQL mit HBase

7.1 Anwendungsfälle

Welche neuen Dinge ermöglicht dieses Tool

7.1.1 CAP-Theorem

7.1.2 ACID und BASE

7.2 Architektur

7.3 Interaktion

7.3.1 HBase Shell

7.3.2 Java API

7.4 Praxis

7.4.1 Hinzufügen zum Cluster

7.4.2 Einrichtung einer Datenbank

7.4.3 Einlesen von Daten

7.4.4 Datenmigration aus einem RDBMS

7.4.5 Absetzen einer Query

8 Streaming mit Kafka

8.1 Anwendungsfälle

Welche neuen Dinge ermöglicht dieses Tool Ersetzt durch Spark Streaming

8.2 Architektur

8.3 Interaktion

8.3.1 Who knows

8.4 Praxis

8.4.1 Hinzufügen zum Cluster

8.4.2 Maybe, vielleicht kann man ja was zeigen

9 Hadoop heute

9.1 Aktuelle Anwendungsbeispiele zu Hadoop

9.1.1 AirBnB

9.2 Apache Spark als Gold Standard

9.2.1 Kann eh alles besser

Literatur

1. *Apache Hadoop* [Apache Hadoop Main Page] [online]. [besucht am 2022-12-07]. Abger. unter: <https://hadoop.apache.org/>.
2. *Big Data: was es ist und was man darüber wissen sollte* [online]. [besucht am 2022-12-05]. Abger. unter: https://www.sas.com/de_de/insights/big-data/what-is-big-data.html.
3. GRIFFITHS, Richard T. *Search Engines* [History of the Internet] [online]. 2007-06-21. [besucht am 2022-12-05]. Abger. unter: <https://web.archive.org/web/20070621143859/http://www.internethistory.leidenuniv.nl/index.php3?m=6&c=7#how>.
4. ZAKON, Robert H'obbes'. *Hobbes' Internet Timeline - the definitive ARPAnet & Internet history* [Hobbes' Internet Timeline] [online]. 2018-01-01. [besucht am 2022-12-05]. Abger. unter: <https://www.zakon.org/robert/internet/timeline/#Growth>.
5. BEAUMONT, David. *How to explain vertical and horizontal scaling in the cloud* [Cloud computing news] [online]. 2014-04-09. [besucht am 2022-12-07]. Abger. unter: <https://www.ibm.com/blogs/cloud-computing/2014/04/09/explain-vertical-horizontal-scaling-cloud/>.
6. GUSTAFSON, John L. Amdahl's Law. In: PADUA, David (Hrsg.). *Encyclopedia of Parallel Computing*. Boston, MA: Springer US, 2011, S. 53–60. ISBN 978-0-387-09766-4. Abger. unter DOI: [10.1007/978-0-387-09766-4_77](https://doi.org/10.1007/978-0-387-09766-4_77).
7. *Pricing - Linux Virtual Machines Scale Sets / Microsoft Azure* [online]. [besucht am 2022-12-07]. Abger. unter: <https://azure.microsoft.com/en-us/pricing/details/virtual-machine-scale-sets/linux/>.
8. ATHOW, Desire. *At 100TB, the world's biggest SSD gets an (eye-watering) price tag* [TechRadar] [online]. 2020-07-07. [besucht am 2022-12-07]. Abger. unter: <https://www.techradar.com/news/at-100tb-the-worlds-biggest-ssd-gets-an-eye-watering-price-tag>.
9. *who was who in SSD? - StorageTek* [online]. [besucht am 2022-12-07]. Abger. unter: <http://www.storagesearch.com/storagetek.html>.