

Hadoop: Finde das längste Wort

Korch, Pascal Florian
s0568195@htw-berlin.de

Kaminski, Elias
s0573558@htw-berlin.de

04.02.2023

1 Einleitung

Hadoop ist ein System zur effizienten Speicherung und Verarbeitung von großen Datenmengen. Als eines der ersten Open Source Projekte in diesem Bereich gilt es als Initiator der Big Data Ära.[4] "Ursprünglich von Google Inc. designed, wurde Apache Hadoop bereits 2008 als Top Level Open Source Projekt der Apache Foundation eingestuft." [4]

1.1 Aufgabenstellung

Die Aufgabe, die hier bearbeitet wurde ist die Nummer 2.a aus dem 2. Teil der Übung. Sie besteht darin 6 Texte verschiedener Sprachen nach dem jeweils längsten auftauchenden Wort zu durchsuchen. Anschließend galt es dieses Wort nach folgendem Schema in ein Dokument zu schreiben: "Sprache – Längstes Wort – Länge".

1.2 Struktur der Arbeit

Da wir beide zu Beginn des Semesters noch nie mit Hadoop gearbeitet haben, investierten wir zunächst Zeit auf die Einrichtung von Hadoop auf unseren Computern. Da einer von uns nur auf Mac arbeiten konnte und der andere sowohl Mac als auch Windows nutzte entschieden wir uns die Einrichtung nur auf unseren Macs zu machen um unnötige Probleme durch den Systemunterschied zu vermeiden. Nachdem das Setup abgeschlossen war experimentierten wir mit dem durch Prof. Kovalenko bereitgestelltem Beispiel. Nach der anschließenden Implementierung nahmen wir Messungen vor und werteten diese im letzten Schritt aus.

2 Setup

Da wir wie bereits erklärt auf MacOS arbeiten installierten wir Hadoop über homebrew:

```
1 brew install hadoop
```

Nach der Installation galt es in den folgenden Files Einstellungen vorzunehmen:[1]

```
1 hadoop-env.sh
2 core-site.xml
3 mapred-site.xml
4 hdfs-site.xml
```

2.1 hadoop-env.sh

In dieser Datei werden Umgebungsvariablen gesetzt und Einstellungen für die Hadoopkonfiguration vorgenommen. Es kann dafür verwendet werden das Verhalten des Hadoop Daemons zu verändern. Beispielsweise kann das Logging verändert oder die Menge des verwendeten Speichers angepasst werden. Die Variable die zuerst gesetzt werden muss ist JAVA_HOME. Sie gibt an wo die zu verwendende Javaversion liegt. Da Hadoop maximal Java 8 unterstützt sollte man bei mehreren Javainstallationen auf die richtige Version achten.[2]

```
1 <!-- Put site-specific property overrides in this file. -->
2 <configuration>
3   <property>
4     <name>hadoop.tmp.dir</name>
5     <value>/usr/local/Cellar/hadoop/hdfs/tmp</value>
6     <description>A base for other temporary directories</
7   </property>
8   <property>
9     <name>fs.default.name</name>
10    <value>hdfs://localhost:8020</value>
11  </property>
12 </configuration>
```

2.2 mapred-site.xml

In mapred-site.xml wird verwendet um die Einstellungen von Map und Reduce zu manipulieren. Normalerweise ist dieses File leer. Alle Änderungen hier werden in hadoop-default.xml übernommen.[3]

```
1 <configuration>
2   <property>
3     <name>mapred.job.tracker</name>
4     <value>localhost:8021</value>
5   </property>
6 </configuration>
```

2.3 hdfs-site.xml

Mithilfe von hdfs-site.xml können Einstellungen an den Daemons des Hadoop File Systems (hdfs) vorgenommen werden.[2]

```
1 <configuration>
2   <property>
3     <name>dfs.replication</name>
```

```

4     <value>1</value>
5   </property>
6 </configuration>

```

3 Implementierung

Für die Implementierung wurde eine lokale Installation von Hadoop gewählt. Der Source Code befindet sich hier: <https://github.com/wolfelias/longest-word>.

3.1 Map Funktion

Die Funktion Map wird mit folgender Signatur aufgerufen:

```

1   protected void map(Object key, Text value, Context context)

```

Anhand von des HDFS Contextes wird der Pfad des zu analysierenden Dokumentes analysiert. Aus diesem Pfad wird die Information zur aktuellen Sprache geholt. Das Text-Objekt value enthält den zu analysierenden Text Baustein. Dieser wird zunächst in einzelne Wörter unterteilt. Anschließend wird jedes Wort, konkateniert mit der Sprache, und der Länge des Wortes in den HDFS Context geschrieben. Der dem HDFS Context übergebene Tupel kann also folgendermaßen aussehen:

```

1   context.write(new Text("English_Example", new IntWritable(7)))

```

3.2 Reduce Funktion

Das Ziel der Reduce Funktion ist das Befüllen folgender Hashmap:

```

1   private final HashMap<String, HashMap<String, Integer>>
      allData = new HashMap<>();

```

In dieser Hashmap wird jeder Sprache eine Hashmap zugeordnet, die wiederum mit Wörtern und deren Länge befüllt ist. Ein beispielhafter Eintrag könnte folgendermaßen aussehen:

```
<"English", <"Example",7>>
```

Dafür wird zunächst die Sprache aus dem ersten Element des Tupels geholt. Dafür wird der String bei dem Unterstrich in Sprache und Wort geteilt. Aus "English_Example" wird also ein Array der Form:

```
["English", Example"]
```

3.3 Cleanup Funktion

In der Cleanup Funktion wird durch die Sprachen der oben beschriebenen Hashmap iteriert. Für jede Sprache wird mithilfe eines Comparators das längste Wort gefunden:

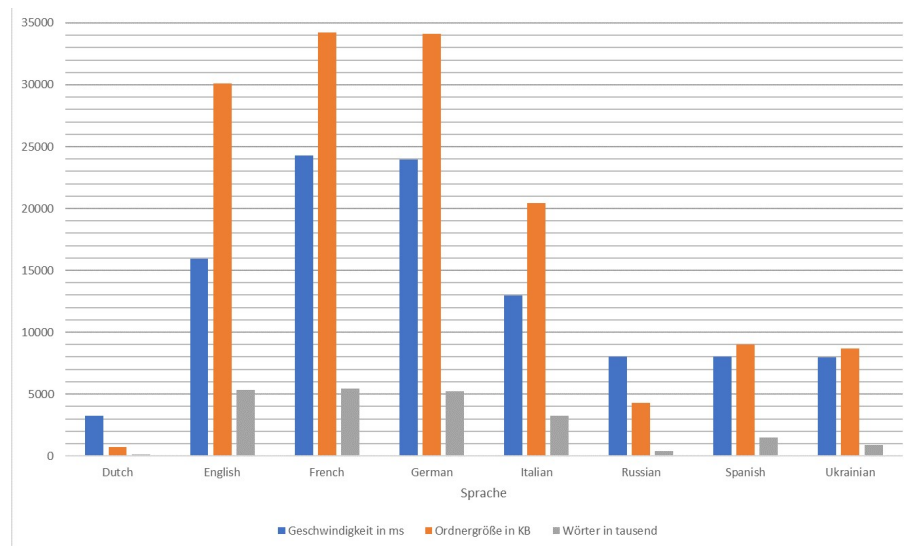
```

1 @Override
2     protected void cleanup(Reducer<Text, IntWritable, Text,
3                             IntWritable>.Context context) throws IOException,
4                             InterruptedException {
5
6         Comparator<Map.Entry<String, Integer>> comparator = (o1, o2) ->
7             o2.getValue().compareTo(o1.getValue());
8
9         for (Map.Entry<String, HashMap<String, Integer>> language :
10             allData.entrySet()) {
11             StringBuilder sb = new StringBuilder();
12
13             List<Map.Entry<String, Integer>> values = new ArrayList<>()
14             language.getValue().entrySet();
15             values.sort(comparator);
16             sb.append(language.getKey());
17             sb.append(" - ");
18             sb.append(values.get(0).getKey());
19             context.write(new Text(sb.toString()), new IntWritable(
20                 values.get(0).getValue()));
21         }

```

Es wurde die Ausführungszeit gemessen, die Hadoop für die jeweiligen Sprachen benötigte. Bei der Messung wurde die Konfiguration in der Main Methode nicht berücksichtigt, sondern nur die Ausführung des Hadoop Jobs. Damit soll ein eindeutigeres Messergebnis erzeugt werden.

4 Auswertung



Verhältnis von Ausführungszeit, Dateigröße und Anzahl an Wörtern der jeweiligen Sprache

Die Auswertung kann in Abbildung 1 betrachtet werden. In der Abbildung wird die Ausführungszeit in einer lokalen Entwicklungsumgebung (2,8 GHz Quad-Core Intel Core i7) in Verhältnis zu der Größe der Dateien bzw. zu der Anzahl an Wörtern jeder Sprache gesetzt. Es zeigte sich, dass die Verarbeitungsgeschwindigkeit von Hadoop nicht nur mit der Anzahl der zu überprüfenden Wörter, sondern auch mit der Dateigröße zusammenhängt. Am Beispiel von Französisch und Englisch sehen wir eine geringere Ausführungsdauer, als bei Französisch, obwohl eine ähnliche Wortanzahl vorlag. Lediglich die Dateigrößen waren unterschiedlich. Außerdem zu sehen war, dass eine Halbierung der Wortzahl nicht zu einer Halbierung der Ausführungszeit führt – hier am Beispiel Italienisch zu Spanisch. Es ist jedoch schwierig, das Ergebnis zu interpretieren. Da in einer lokalen Entwicklungsumgebung viele Parameter eine Rolle spielen und Hadoop eigentlich auf verteilten Systemen ausgeführt wird, kann hier keine relevante Aussage über Verhältnis von Dateigröße und Ausführungszeit getroffen werden.

5 Fazit

Die lokale Installation und Konfiguration von Hadoop gestaltete sich zunächst als schwierig. Wir haben verschiedene Betriebssysteme ausprobiert und uns letztendlich für MAC OS entschieden. Hier gelang die Konfiguration am schnellsten. Laut unseren Recherchen wäre die Verwendung von Docker jedoch noch einfacher gewesen. Bei einer neuen Arbeit mit Hadoop würden wir versuchen diesen Weg zu gehen.

Die Verwendung von Hadoop auf lokalen Entwicklungsumgebungen bzw. nicht-verteilten Systemen macht nur begrenzt Sinn. Die eigentlichen Stärken von Hadoop werden hierbei nicht ausgeschöpft. So waren die verwendeten Datenmengen nicht sehr groß (maximal 34,2 MB) und befanden sich damit nicht im Bereich von Big Data. Interessant wäre es für zukünftige Projekte mit dem Framework, die Anzahl an Textdateien in den Ordnern deutlich zu erhöhen. Ein Vergleich von Ausführungszeiten macht in einem lokalen System unserer Meinung nur begrenzt Sinn. Die lokalen Ausführungszeiten haben bei uns stark variiert. Für eine Einordnung der Laufzeit von Hadoop empfehlen wir die Verwendung von Benchmarktests mit festgelegten Netzwerkbandbreiten bzw. mit festgelegter Hardware.

Litteratur

- [1] Diwakar. *Installing Hadoop on Mac*. Beer&Diapers.ai. Nov. 19, 2019. URL: <https://medium.com/beeranddiapers/installing-hadoop-on-mac-a9a3649dbc4d> (visited on 01/23/2023).
- [2] *GettingStartedWithHadoop - HADOOP2 - Apache Software Foundation*. URL: <https://cwiki.apache.org/confluence/display/HADOOP2/GettingStartedWithHadoop> (visited on 02/06/2023).

- [3] Kalyn Says. *Explaining Hadoop Configuration* — *Edureka.co*. Edureka. Section: Big Data. Sept. 26, 2014. URL: <https://www.edureka.co/blog/explaining-hadoop-configuration/> (visited on 02/06/2023).
- [4] Laurenz Wuttke. *Hadoop einfach erklärt!* datasolut GmbH. Mar. 23, 2022. URL: <https://datasolut.com/apache-hadoop-einfuehrung/> (visited on 02/04/2023).