Universität Augsburg
Institut für Informatik
Lehrstuhl für Organic Computing
Prof. Dr. Jörg Hähner

Ansprechpartner
Dominik Rauh
dominik.rauh@informatik.uni-augsburg.de
Eichleitnerstr. 30, Raum 502

Wintersemester 2018/2019

Peer-to-Peer und Cloud Computing

Aufgabenblatt 9

Dieses Übungsblatt ist Teil der Bonusregelung. Sie können Ihre Lösung diesmal, anders als sonst, während der Übung am **Mittwoch, den 06.02.2019,** abnehmen lassen. Die Vorstellung der Ergebnisse wird voraussichtlich ebenfalls in dieser Übung stattfinden.

Implementierungsaufgabe zu MapReduce

Die Betreiber einer digitalen Bibliothek möchten ihren Nutzern einen Überblick über den Inhalt ihrer Dokumente geben. Hierzu wollen sie die zehn häufigsten Substantive des gesamten Dokumentenkorpus extrahieren und in einer Tag-Cloud visualisieren. Die Größe eines Tags soll dabei von seiner Häufigkeit im Korpus abhängig sein. Da der Dokumentenkorpus der Bibliothek sehr groß ist, soll für das Berechnen dieser Häufigkeiten *MapReduce* benutzt werden.

Implementieren Sie eine MapReduce-basierte Anwendung in *Hadoop*, die einen Dokumentenkorpus als Eingabe erhält und eine Menge von Paaren – bestehend aus einem Substantiv und seiner Häufigkeit im Korpus – zurückgibt.

Verwenden Sie für die Lösung dieser Aufgabe das zur Verfügung gestellte Repository[1]. Dieses stellt alle notwendigen Komponenten bereit:

- Ein Maven-Projekt *topk*, das ein Rahmenwerk zur Implementierung eines eigenen Mappers und Reducers enthält: Sie müssen nur noch die beiden mit *TODO* markierten Stellen mit einer geeigneten Implementierung füllen.
- Den Part-of-Speech-Tagger der Stanford NLP Group[2], welcher im Maven-Projekt bereits als Abhängigkeit definiert ist.

[1] `https://git.rz.uni-augsburg.de/oc-dozent/topk.git`
[2] `https://stanfordnlp.github.io/CoreNLP/`

- Einen beispielhafter Korpus aus zehn Wikipedia-Artikeln, der für Ihre Lösung als Eingabe dienen soll.
- Ein *Dockerfile*, welches eine Standalone-Installation von Hadoop bereitstellt, die für das Ausführen Ihrer MapReduce-Pipeline benutzt werden kann.

Sie benötigen für die Lösung dieses Blattes folgende Software:

- Git
- Java und Maven
- Docker

Maven

Das Maven-Projekt kann mithilfe von

```
mvn package
```

gebaut werden; auf diese Weise wird eine Hadoop-fähige jar-Datei unter `target` generiert.

Docker

Der Docker-Container kann mithilfe von

```
docker build -t topk .
```

gebaut, sowie durch

```
docker run -it -v \
    <Pfad zum Maven-Projekt>/target/topk-1.0-SNAPSHOT-jar-with-dependencies.jar\
    :/opt/topk.jar topk
```

gestartet werden. Dabei muss der Pfad *vor* dem Doppelpunkt eventuell an die Dateipfad-Syntax des Hostsystems angepasst werden.

Anschließend befinden Sie sich in der Bash des Docker-Containers, in der Sie mittels

```
hadoop/bin/hadoop jar topk.jar de.uniaugsburg.informatik.oc.TopK
```

einen Hadoop-Durchlauf starten können.

Anschließend sollten im Ordner `output` zwei Dateien zu finden sein:

- `_SUCCESS`, als Zeichen für die erfolgreiche Ausführung, sowie
- `part-r-00000`, welche das Ergebnis enthält.

Weitere eventuell nützliche Hinweise

Den Namen `$n` eines laufenden Containers können Sie durch

```
docker ps
```

herausfinden.

Die Tastenfolge, um von Docker zu detachen ist: Ctrl+p Ctrl+q.

Wieder attachen können Sie mit:

```
docker attach $n
```

Sie können die jar-Datei auch händisch in einen (laufenden) Container mit Namen `$n` kopieren:

```
docker cp \
  topk/target/topk-1.0-SNAPSHOT-jar-with-dependencies.jar $n:/opt/topk.jar
```