Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel, J. Gresens

Übungsblatt 12 zur Vorlesung
Big Data Analytics
im WiSe 2015/2016
Abgabe: 22.01.2016, 420 P

Sollten Probleme auftauchen, wenden Sie sich bitte an die Mailingliste der Veranstaltung:

`bd1516@wr.informatik.uni-hamburg.de`

1 Erste Schritte mit Spark (60 P)

In dieser Aufgabe werden wir zum Einstieg einige einfache Berechnungen mit Spark durchführen:

- Erstellen Sie ein RDD mit dem Daten aus `/user/bigdata/wikipedia-text-tiny-clean500.csv`.
- Berechnen Sie mit Hilfe von RDD Operationen die Anzahl (absolut und relativ zur Gesamtzahl) und die Namen der Artikel, welche das Wort Computer enthalten.
- Speichern Sie das Ergebnis in eine Datei im HDFS.
- Schreiben Sie eine zweite Variante zur Berechnung der Artikelanzahl, die hierfür Sparks Accumulatoren verwendet.

Führen Sie diese Aufgabe zunächst im lokalen Modus aus.
Versuchen Sie dann beim Start von Spark dieses so zu konfigurieren, dass die Zeilen über mehrere Knoten verteilt werden. Dokumentieren Sie Ihre Aufrufparameter.

1.1 Hinweise

Passen Sie unbedingt den Port der Benutzerschnittstelle wie im Beispiel an:

```
PYSPARK_PYTHON=python3 pyspark --conf spark.ui.port=4711
```

Um die Daten über mehrere Knoten verteilen zu können, führen Sie dann Pyspark mit YARN und bspw. bis zu 8 Executor-Prozessen pro Knoten aus.
Zur Dokumentation einzelner Operationen empfielt sich: `http://spark.apache.org/docs/latest/`

Abgabe:

`1-erste-schritte.py`	Ihre Abgabe für die Ausführung mit Pyspark.
`1-erste-schritte.txt`	Dokumentation der Aufrufparameter für pyspark und der Ausgabe.

2 Artikelabstände in der Wikipedia (180 P)

In dieser Aufgabe werden wir den Abstand zwischen zwei Wikipedia-Artikeln berechnen. Hierfür können wir wieder eine Distanzmetrik d basierend auf den Worthäufigkeiten der einzelnen Artikel nutzen. Das Ergebnis der Distanzmetrik $d(A, B)$ für beliebige Artikel A und B soll zwischen 0 und 1 liegen.
Überlegen Sie sich eine Metrik für den Abstand zwischen zwei Artikeln und implementieren Sie diese.
Erfassen und dokumentieren Sie die Laufzeit ihres Programs beim Start mit dem Resourcenmanager YARN und 8 Executors pro Knoten. Geben Sie einige Beispielausgaben für Artikelabstände in der Textdatei mit ab.

2.1 Hinweise

Ein primitives Code-Gerüst ist unter `2-wikipedia-distanz.py` im Archiv enthalten. Nutzen Sie zum Testen zunächst die Datei: `/user/bigdata/wikipedia-text-tiny-clean500.csv`.
Es empfielt sich das erwartete Format der Tupel zu dokumentieren und alle Zwischenergebnisse zu überprüfen.

Abgabe:

`2-wikipedia-distanz.py`	Ihr Programm für die Ausführung mit Spark.
`2-wikipedia-distanz.txt`	Dokumentation der Aufrufparameter für pyspark, ihre Zeitmessungen und einige Beispielausgaben.

3 Machinelles Lernen mit Spark (180 P)

In dieser Aufgabe werden wir mit Spark Artikeln basierend auf den Worthäufigkeiten in der Wikipedia clustern.
Nutzen Sie die Berechnung der Worthäufigkeit von Aufgabe 2 und laden Sie wieder den Datensatz `/user/bigdata/wikipedia-text-tiny-clean500.csv`.
Das Dictionary muss allerdings noch für den KMeans-Algorithmus aus MLlib vorbereitet werden. Testen Sie verschiedene Zahlen von Cluster zwischen 2 und 10 und benutzen Sie den mittleren Abstand der Artikel zu den Clusterzentren als Fehlermetrik.
Analysieren Sie die einzelnen Clusterzentren und Visualisieren Sie die nahe gelegenen Artikel (größere Darstellung je näher der Artikel am Clusterzentrum liegt) wieder mit der Wordcloud.

3.1 Hinweise

Testen Sie ihr Skript in der pyspark-Shell, via `spark-submit` gestartete Skripte können unter Umständen im Falle eines Fehlers einfach anhalten und werden nicht abgebrochen.
Sie finden weitere Dokumentation zu KMeans unter `https://spark.apache.org/docs/1.4.1/mllib-clustering.html#k-means`.

Abgabe:

`3-kmeans.py`	Ihr Programm für die Ausführung mit Spark.
`3-kmeans.txt`	Dokumentation der Aufrufparameter für pyspark.
`3-analyse.pdf`	Ihre Wordclouds und Analyse der Clusterzentren.