Ludwig-Maximilians-Universität München
Institut für Informatik
PD Dr. Peer Kröger
Klaus Schmid, Dr. Andreas Züfle

München, 01.07.2015

Datenbanksysteme II

SS 2015

Übungsblatt 8: Big Data

Besprechung: 06.07.-09.07.2015

Aufgabe 8-1 *Big Data*

Kreuzen Sie zutreffende Antworten an.

Warum eignen sich traditionalle Ansätze eher nicht für Big Data?
☐ Big Data passt nicht auf eine einzelne Maschine
☐ Das Lesen aus dem Speicher dauert zu lange
☐ Traditionelle Ansätze skalieren sehr gut mit großen Mengen unstrukturierter Daten
☐ Für Big Data lässt sich kein relationales Schema finden

Welche der folgenden Punkte machen für Big Data Sinn?
☐ Benutzung von komplexer Hardware
☐ Benutzung von komplexer Software
☐ Einfache Möglichkeit der Kapazitätserweiterung
☐ Verwendung von gewöhnlichen User-PCs

Was sind Probleme im Umgang mit Computerclustern?
☐ Viele Maschinen bedeutet erhöhte Ausfallrate
☐ Viele Maschinen bedeutet, dass auch mit langsamen Maschinen umgegangen werden muss
☐ Die Rechenleistung einzelner Maschinen sinkt
☐ Das Verschicken von Daten ist teuer

Aufgabe 8-2 *Spark*

Gegeben sei eine Logdatei im Apache Common Log Format bei der die Einträge wie folgt ausschauen:

`127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] GET /apache.gif HTTP/1.0 200 2326`

Die einzelnen Komponenten eines Logeintrages sind

- `127.0.0.1` : Die IP Adresse des Clients
- `-` : Loginname des Nutzers, unter dem er sich am Clientrechner angemeldet hat
- `frank` : Nutzerkennung die vom Nutzer bei der Authentifizierung auf dem Webserver angegeben wurde
- `[10/Oct/2000:13:55:36 -0700]` : Zeitpunkt der Transaktion
- `"GET /apache.gif HTTP/1.0"` : Anfrageart, angefragte Datei und verwendetes Protokoll
- `200` : Statuscode für den Abschlusszustand der Verbindung
- `2326` : Anzahl der übermittelten Bytes

Nehmen Sie zur Vereinfachung für diese Aufgabe an, dass in der Logdatei ausschließlich Einträge mit den Statuscodes 200 (für OK) und 404 (für Not Found) vorkommen. Außerdem kommt es vor, dass bei Not Found Fehlern als Größe für übertragene Bytes teilweise `0`, teilweise aber fälschlicherweise auch ein Wert größer als 0 in die entsprechenden Logeinträge geschrieben wurde.

(a) Implementieren Sie eine Funktion in Spark, die falsche Einträge, also diejenigen Einträge mit Statuscode 404 und Anzahl übertragener Bytes > 0, verwirft. Sie dürfen annehmen, dass die Datei bereits eingelesen ist und ein RDD `lines` erzeugt wurde. `lines` enthält pro Logeintrag ein `String[]` Array, dass die einzelnen Komponenten eines Logeintrages enthält. Der RDD-Eintrag für das obige Beispiel schaut also wie folgt aus

```
["127.0.0.1", "-", "frank", "[10/Oct/2000:13:55:36 -0700]", "GET /apache.gif HTTP/1.0", "200", "2326"].
```

(b) Als nächstes soll für jede in der Logdatei vorkommende IP Adresse die an sie übertragene Datenmenge ermittelt werden. Beschreiben (oder implementieren) Sie wie eine Lösung mittels Spark aussehen könnte.

(c) Was ist der Hauptvorteil von Apache Spark gegenüber Hadoop MapReduce?

Aufgabe 8-3 *Veracity*

Gegeben sei eine Relation *Studenten*, die Information zu Studenten und deren Position enthält. Folgende Anfrage sucht alle Studenten die sich möglicherweise in die Vorlesung verirrt haben.

```
SELECT DISTINCT Nachname
FROM Studenten stud
WHERE stud.position near "Raum S 002 (Schellingstr. 3 (S) VG)"
```

Das relationale Datenbankmanagementsystem kennt das Problem der Unsicherheit. Es schätzt die Wahrscheinlichkeit von jedem Studenten das Anfrageprädikat

$$\phi := \text{Student befindet sich in Raum S 002 (Schellingstr. 3 (S) VG)}$$

zu erfüllen. Studenten mit eine Wahrscheinlich von 0 werden ignoriert, so dass die Ergebnisrelation folgendermaßen aussieht:

Vorname	$P(\phi)$
Attenberger	0.8
Brunner	0.5
Carl	0.6
Deschler	0.1

Es soll folgende Anfrage beantwortet werden:

```
SELECT count(*)
FROM Studenten stud
WHERE stud.position near "Raum S 002 (Schellingstr. 3 (S) VG)"
```

Als Ergebnis soll also zurückgeben werden:

count(*)	P(count(*))
0	0.036
1	0.238
2	0.440
3	0.262
4	0.024

(a) Geben Sie informell einen Algorithmus an, um obiges Ergebnis in polynomieller Zeit zu berechnen.

(b) Führen Sie ihren Algorithmus durch!