Map Reduce

Was ist MapReduce und wofür Apache Hadoop

Inhalt

- Was ist MapReduce
- Was macht Map Reduce
- Wie arbeitet Map Reduce
- Apache Hadoop
- Vor-/Nachteile

Was ist MapReduce?

- Theoretisches **Programmiermodell**
- zur **Berechnung und Analyse** von Daten
- von Google
- für **riesige Datenmengen**
- auf **Computerclustern**

Wie macht das MapReduce?

1. Phase: Map - Teilen der Input-Daten in Key/Value-Paare
2. Phase: Shuffle - Grouppieren der Daten und Verteilen auf Rechenprozesse
3. Phase: Reduce - Erzeugen von Zwischenergebnissen je Gruppe

Beispiel

- Finden der Anzahl der verkauften Produkte je Land
- Anhand Beispiel: https://www.guru99.com/create-your-first-hadoop-program.html

(Folgende Codezeilen sind nur Ausschnitte)

produkte.csv

Jede Zeile = 1 Produktverkauf

7. Spalte = Verkaufsland

```
Transaction_date,Product,Price,Payment_Type,Name,City,State,Country,Account_Created,Last
1/2/09 6:17,Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom,1/2/09 6:0
1/2/09 4:53,Product1,1200,Visa,Betina,Parkville,MO,United States,1/2/09 4:42,1/2/09 7:49
1/3/09 14:44,Product1,1200,Visa,Gouya,Echuca,Victoria,Australia,9/25/05 21:13,1/3/09 14:
1/4/09 12:56,Product2,3600,Visa,Gerd W ,Cahaba Heights,AL,United States,11/15/08 15:47,1
1/4/09 13:19,Product1,1200,Visa,LAURENCE,Mickleton,NJ,United States,9/24/08 15:19,1/4/09
```

SalesMapper.java

1. Eingabe von produkte.csv
2. Trennen anhand Komma
3. Erzeugen von Key/Value aus 7. Spalte
    iv. Key: Ländername
    v. Value: "1"

```
public void map(LongWritable key, Text value, OutputCollector <Text, IntWritable> output

                String valueString = value.toString();
                String[] SingleCountryData = valueString.split(",");
                output.collect(new Text(SingleCountryData[7]), one);
        }
```

SalesCountryReducer.java

1. Addieren von Value(1) je Land in While-Schleife
2. output.collect beinhaltet [Land][Gesamtanzahl]

```
        public void reduce(Text t_key, Iterator<IntWritable> values, OutputCollector<Tex
                Text key = t_key;
                int frequencyForCountry = 0;
                while (values.hasNext()) {
                        // replace type of value with the actual type of our value
                        IntWritable value = (IntWritable) values.next();
                        frequencyForCountry += value.get();

                }
                output.collect(key, new IntWritable(frequencyForCountry));
        }
```

Programmablauf

1. Einlesen von produkte.csv
2. Mappen: SalesCountry.SalesMapper
3. Reducen: SalesCountry.SalesCountryReducer
4. Ergebnis speichern

Aber: Wo ist das Shuffle und der Computercluster?

Apache Hadoop

- Software **Framework** zur verteilten Datenverarbeitung
- Übernimmt die Shuffle-Phase für Anwender
  - **Verteilt** die Rohdaten an Nodes
  - **Koordiniert** die Verarbeitung an Nodes
- **HDFS**: Selbe Rohdaten liegen auf mehreren Nodes gleichzeitig
  - reduziert Datentransfer im Shuffle-Prozess

Wie sieht Implementierung aus?

SalesCountryDriver.java:

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;
public class SalesCountryDriver {
        // Set a name of the Job
        job_conf.setJobName("SalePerCountry");
        // Specify names of Mapper and Reducer Class
        job_conf.setMapperClass(SalesCountry.SalesMapper.class);
        job_conf.setReducerClass(SalesCountry.SalesCountryReducer.class);
        // Run the job
        JobClient.runJob(job_conf);
}
```

```
$HADOOP_HOME/bin/hadoop jar ProductSalePerCountry.jar /inputMapReduce /mapreduce_output_
```

Vor-/Nachteile

Vorteile

- einfache nutzbar durch Framework
- Hadoop OpenSource
- skalierbar und fehlertollerant durch Cluster + HDFS

Nachteile

- keine dynamischen Berechnungen(siehe statischer Mapper+Reducer)

Alternativen/Weiterentwicklungen

- Apache Spark/Flink

Danke für die Aufmerksamkeit! 👍

Fragen gerne

Handout gibts auch.