Übungsblatt: Auswertung von Tweets

Software Architekturen, 6. Semester Bachelor: Wirtschaftsinformatik

FH CAMPUS 02
Informationstechnologien & Wirtschaftsinformatik

Grahsl Hans-Peter, Steyer Manfred

Der zu erstellende MapReduce Job soll lokal (Standalone-Modus) direkt aus Eclipse gestartet werden können.

1. Laden Sie das **Hadoop Starter-Kit** aus Moodle herunter, importieren Sie das ZIP-Archiv in Eclipse als Java Projekt und machen Sie sich kurz mit der Codebase im **Package edu.campus02.iwi.hadoop.twitter** vertraut.

2. Die DriverTweetFeed Klasse sowie simple Wrapper-Klassen zur De/Serialisierung von und nach JSON sind bereits vorhanden.

3. Ihr MapReduce Job soll einfache Berechnungen zur Analyse eines Twitter Feeds durchführen. Die Input-Datei liegt als **zeilenweises JSON-Format** vor (→ *kleine Datei unter data/input/tweets/sample.json, etwas größere mit ~50MB feed.json*). Verwenden Sie in Ihrer Job-Config am einfachsten die **TextInputFormat bzw. TextOutputFormat Klassen**. Die De/Serialisierung von und nach JSON in die bestehenden Wrapper-Klassen machen Sie selbst direkt im Mapper bzw. Reducer unter Verwendung der GSON-Library.

    - Beispielaufruf zur Deserialisierung von JSON in ein Tweet-Objekt
      new Gson().fromJson(*„jsonStringOf1LineFromInputFile“*, Tweet.class)
    - Beispielaufruf zur Serialisierung nach JSON von einem TweetParts-Objekt
      new Gson().toJson(*TweetPartObject*, TweetParts.class)

4. Schreiben Sie einen MapReduce Job, um die folgenden **Tweet-Statistiken pro Sprache (=lang Attribute der Tweet-Klasse)** zu berechnen:
    a. Anzahl der Tweets
    b. Durchschnittliche Anzahl der Zeichen pro Tweet
    c. Prozentsatz an Tweets die URLs (mind. 1) beinhalten
    d. Anzahl an Tweets die mind. 3 Hashtags beinhalten

5. Hinweise:

Ihr *Mapper* ist wie folgt typisiert: `<LongWritable, Text, Text, Text>`

Die *map-Methode* bekommt als value eine gesamte JSON-Zeile der Input-Datei. Diese müssen Sie mittels GSON-Library in ein Tweet-Objekt Deserialisieren (siehe Bsp. oben)

Sie verwenden die Klasse TweetParts, um die Metriken von je einem Tweet zu speichern (num_chars, num_hashtags, num_urls). **OutputKey ist die Sprache, OutputValue der JSON-String des TweetParts Objekts** (serialisiert mittels GSON-Library).

Ihr *Reducer* ist wie folgt typisiert: `<Text, Text, Text, Text>`

Die *reduce*-Methode bekommt pro Key (=Sprache) alle dazugehörigen JSON-Zeilen als Iterable<Text>. Diese müssen sie jeweils wieder deserialisieren, um die benötigten Statistiken zu berechnen, welche in ein TweetAvgResult Objekt gespeichert werden.

OutputKey ist die Sprache, OutputValue der JSON-String des TweetAvgResult Objekts (serialisiert mittels GSON-Library).