Übungsblatt: Basic Warm-Up

Software Architekturen, 6. Semester Bachelor: Wirtschaftsinformatik

FH CAMPUS 02
Informationstechnologien & Wirtschaftsinformatik

Grahsl Hans-Peter, Steyer Manfred

Die zu erstellenden MapReduce Jobs sollen lokal (Standalone-Modus) direkt aus Eclipse gestartet werden können.

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.warmup** vertraut.

Sie arbeiten bei allen Aufgaben dieses Übungsblatts mit einer kleinen Textdatei ~5 MB (Quelle: Gutenberg Book Project), welche Werke von William Shakespeare beinhaltet (siehe Ordner data/input/warmup/pg100.txt)

Beispiel 1:

Die Klasse **AllCapsDriver.java** beinhaltet bereits die typische Grundstruktur für einen einfachen MapReduce Job.

- Starten Sie den aktuellen Job und geben Sie dem Programm die 2 benötigten Aufrufparameter mit <inputDir> <outputDir>
  data/input/warmup/pg100.txt data/output/warmup/caps

  Sehen Sie sich das Ergebnis im Output-Ordner *data/output/warmup/caps* an und versuchen Sie das Default-Verhalten von MapReduce zu beschreiben. Sie haben ja noch keine(!) eigenen Mapper und Reducer geschrieben.

- Orientieren Sie sich am WordCount Beispiel und schreiben Sie nun geeignete Mapper und Reducer Klassen, sodass die gesamte Textdatei in Großbuchstaben ausgegeben werden kann.

  Führen Sie den Job zuerst mit **job.setNumReduceTasks(1)** und danach mit **job.setNumReduceTasks(2)** aus. Vergleichen Sie den Output und beschreiben Sie was passiert ist und warum.

  Hinweis: Verwenden Sie beim Starten unterschiedliche Output-Ordner als Aufrufparameter damit die Ergebnisse vom 1. Durchlauf nicht überschrieben werden, z.B.
  data/input/warmup/pg100.txt data/output/warmup/caps/run1
  data/input/warmup/pg100.txt data/output/warmup/caps/run2

Beispiel 2:

Die Klasse **BigramDriver.java** beinhaltet bereits die typische Grundstruktur für einen einfachen MapReduce Job.

- Schreiben Sie je eine Mapper und Reducer Klasse, um die Worthäufigkeiten von **Bigrams** *(= je 2 unmittelbar benachbarte Wörter in einer Textzeile)* zu bestimmen.

  z.B. Die Input-Zeile „As I was going to St.Ives“ führt zu folgenden 5 Bigrams
  - ⇨ As I
  - ⇨ I was
  - ⇨ was going
  - ⇨ going to
  - ⇨ to St.Ives

  Hinweis: Der Einfachheit halber werden die Wörter im Text bei Whitespaces getrennt. Sie können den Text Value im Mapper wie folgt in einzelne Wörter trennen: `String[] tokens = value.toString().trim().split("\\s+");`
  Ähnlich wie beim WordCount Beispiel ist der Output Key im Mapper ist vom Typ Text und beinhaltet je ein Bigram. Der dazugehörige Value ist fix 1.

- Ändern Sie Ihr Programm, sodass im Ergebnis **nur mehr das Bigram mit der höchsten Häufigkeit** aufscheint.

  Hinweis: Dazu müssen Sie im Reducer in Ihrer reduce Methode beim Summieren aller Values pro Key eruieren, welches Bigram die höchste Summe hat und diese bei Bedarf laufend aktualisieren. D.h. Sie schreiben nicht jeden Key mit zugehörigem Value in den Context sondern nur jenen, der das Bigram mit der von Ihnen bestimmten höchsten Häufigkeit beinhaltet. Dazu verwenden Sie folgende Methode in Ihrer Reducer-Klasse welche am Ende einmal vom Framework aufgerufen wird.

```
@Override
protected void cleanup(Context context) throws IOException,
                                               InterruptedException {
        //TODO: write your bigram with max frequency to context here...
        //context.write(KEY, VALUE);
}
```

Beispiel 3:

Die Klasse **FilterDriver.java** beinhaltet bereits die typische Grundstruktur für einen einfachen MapReduce Job.

- Schreiben Sie je eine Mapper und Reducer Klasse, damit im Ergebnis nur jene Zeilen unverändert aufscheinen, welche die Zeichenkette „torture“ beinhalten. Die Reihenfolge der Zeilen im Ergebnis ist dabei egal.

  Hinweis: Sie können im Mapper mit folgendem Befehl prüfen, ob die Input-Zeile die gewünschte Zeichenkette beinhaltet: `value.toString().contains(“Filterwort”)`

- Welche Funktion übernimmt der Reducer in diesem Beispiel bzw. spielt dieser überhaupt eine wichtige Rolle?