Big Data Management and Analytics
Assignment 8

Assignment 8-1

(a) Compare and highlight the differences between Spark and Flink

Features	Apache Flink	Apache Spark
Streaming engine	Stream approach: A batch is a finite set of streamed data	Micro-batch approach: A stream is ‚cut' into small batches
Iterative processing	Native iteration support	Non-native iteration, implemented as regular for-loops outside the system
Latency	Low latency, high throughput	High latency compared to Flink
Time management	Out-of-order events, windows, user-defined	Process time-based

Assignment 8-1

(b) In how far is Flink more suitable for streaming tasks?

- provides natively a stream-processing approach
- has a lower latency
- supports more powerful windowing systems
- has explicit time-handling

Assignment 8-2

Building blocks of an Apache Flink program:

FlatMap functions take elements and transform them into zero, one or more elements in a non-nested structure

Represents a collection of elements of the same type

```
import org.apache.flink.api.common.functions.FlatMapFunction;
import org.apache.flink.api.java.DataSet;
import org.apache.flink.api.java.ExecutionEnvironment;
import org.apache.flink.api.java.tuple.Tuple2;
import org.apache.flink.util.Collector;
```

Context in which program is executed

Collects a record and forwards it

Tuples have a fixed length and contain a set of fields which can be of different types

Assignment 8-2

Building blocks of an Apache Flink program:

Create a DataSet of strings by reading out the text file

Create a context object in which the program is executed

```
public class FlinkProgram {
    public static void main(String[] args) throws Exception {
        ExecutionEnvironment env = ExecutionEnvironment.getExecutionEnvironment();

        DataSet<String> rawdata = env.readTextFile("C:\\Users\\kazempour\\Documents\\ttwist.txt");

        DataSet <Tuple2<String, Integer>> result = rawdata
            .flatMap(new Splitter())
            .groupBy(0)
            .sum(1);
        result.print();
    }
```

Create a resulting DataSet consisting of 2-tuples by…

…mapping a splitter method

…summing up (reducing) the number of their occurences

…grouping the resulting tuples according to their words

Assignment 8-2

Building blocks of an Apache Flink program:

```
public static class Splitter implements FlatMapFunction<String, Tuple2<String, Integer>> {
    @Override
    public void flatMap(String line, Collector<Tuple2<String, Integer>> out) {
        for (String wordToken : line.split(" ")) {
            out.collect(new Tuple2<String, Integer>(wordToken, 1));
        }
    }
}
```

Method takes a string and a collector as a 2-tuple and appends a collection filled with 2-tuples of the structure: (wordToken, #ofOfOccurence)

Assignment 8-3

See Java-Code!