CS744A1

Hao Fu, Kan Wu, Huayu Zhang

September 2017

1 Part A

I upgraded Spark to 2.2.0 as it is the stable version for structured streaming. The CPU/mem configuration is in Table 1. To git rid of the annoying logs, I set the log level as WARN.

spark.driver.memory	8g
spark.executor.cores	4
spark.executor.memory	8g
spark.task.cpus	1

Table 1: CPU/mem configuration

Question 1. The key is to count the RT, MT, RE within a 60-minute window. The is done by

```
val windowedCounts = fileStreamDf.groupBy(
window($"timestamp", "60 minutes", "30 minutes"), $"interaction"
).count().orderBy("window")
```

To print the complete table, I set the numRows 563500 (number of files × maximum number of entries in each file) as an upper bound. The output mode is set "complete".

Question 2. The critical part is to select userB from MT entries.

```
val selectedUser = fileStreamDf.select("userB").where("interaction = 'MT'")
```

To process the data every 10 seconds. I use the Trigger class in the query.

```
val query = selectedUser.writeStream.format("csv")
...
.trigger(Trigger.ProcessingTime("10 seconds"));
```

The output mode is set "append" as I do not need to repeat previous items.

Question 3. I generated the list data as all odd numbers from 1 to 100000. The important thing is to inner join the list data with the stream by "userA".

```
val filteredCounts = fileStreamDf.join(whiteList, "userA").groupBy("userA").count()
```