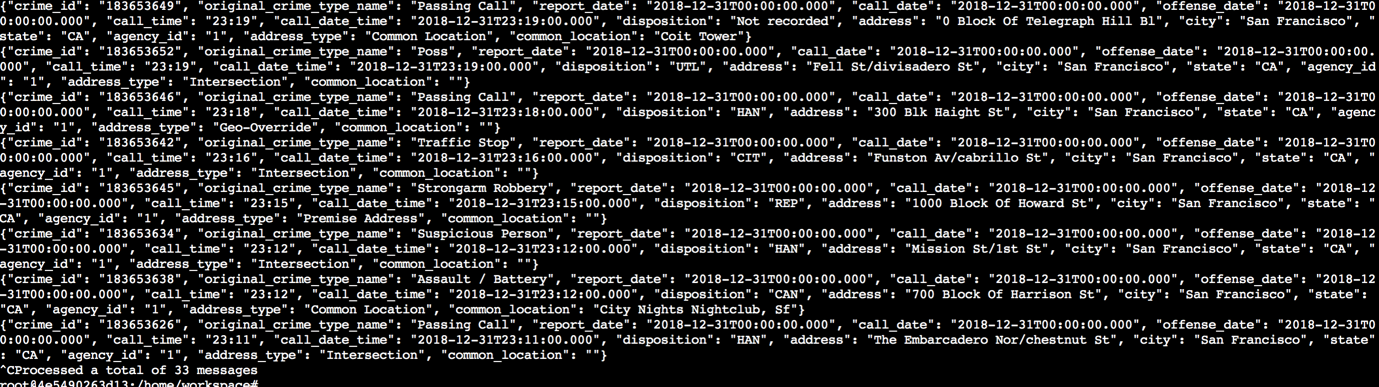
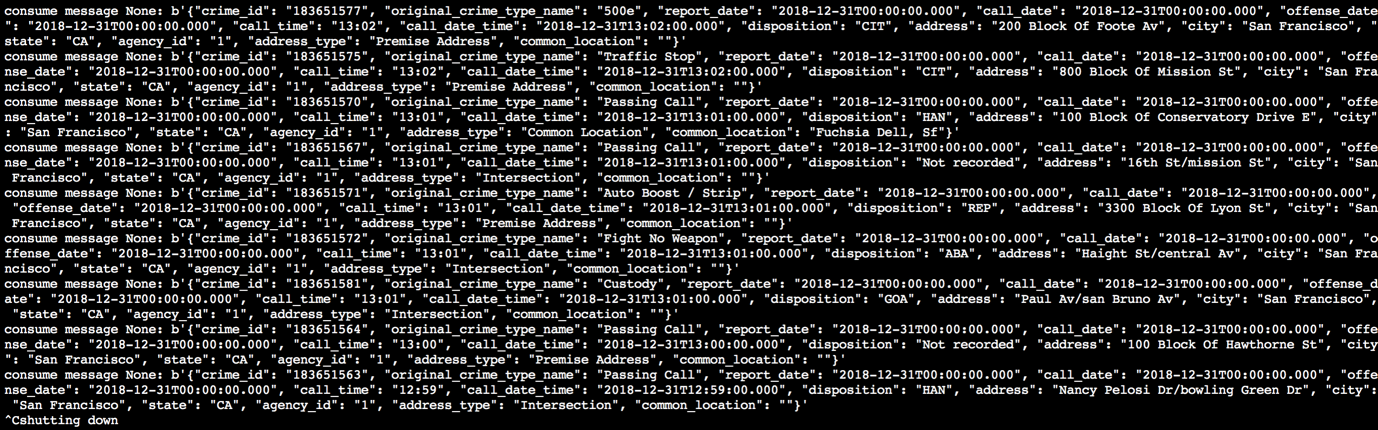
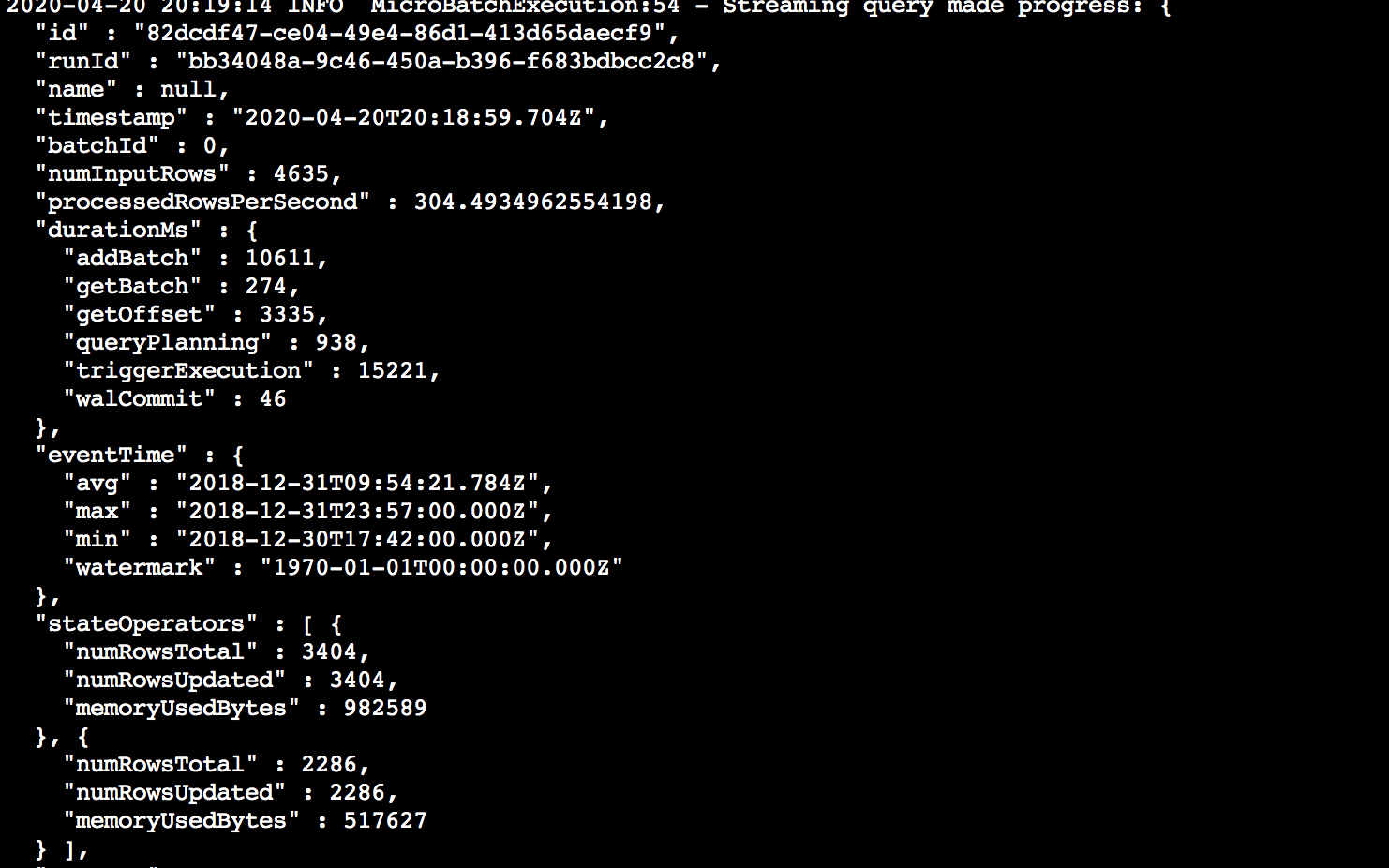
**kafka-consumer-console output**



**Consumer Server:**

****

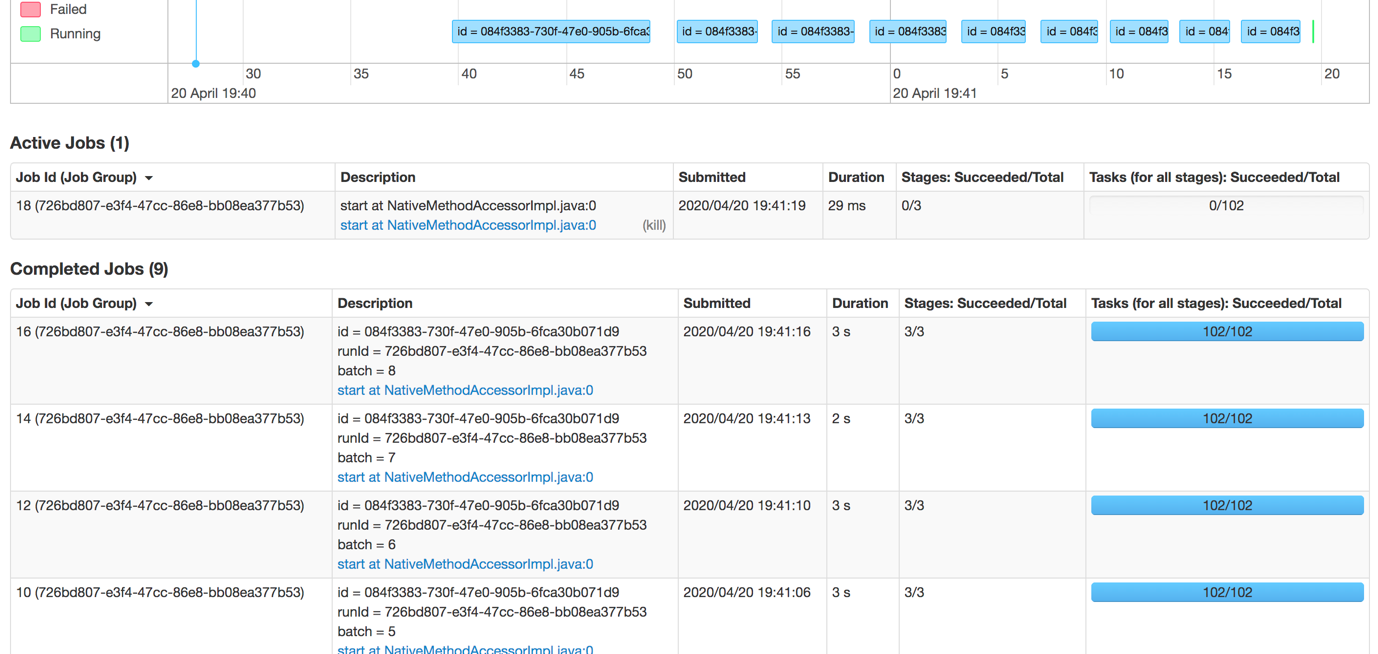
**Spark Job progress report:**

****

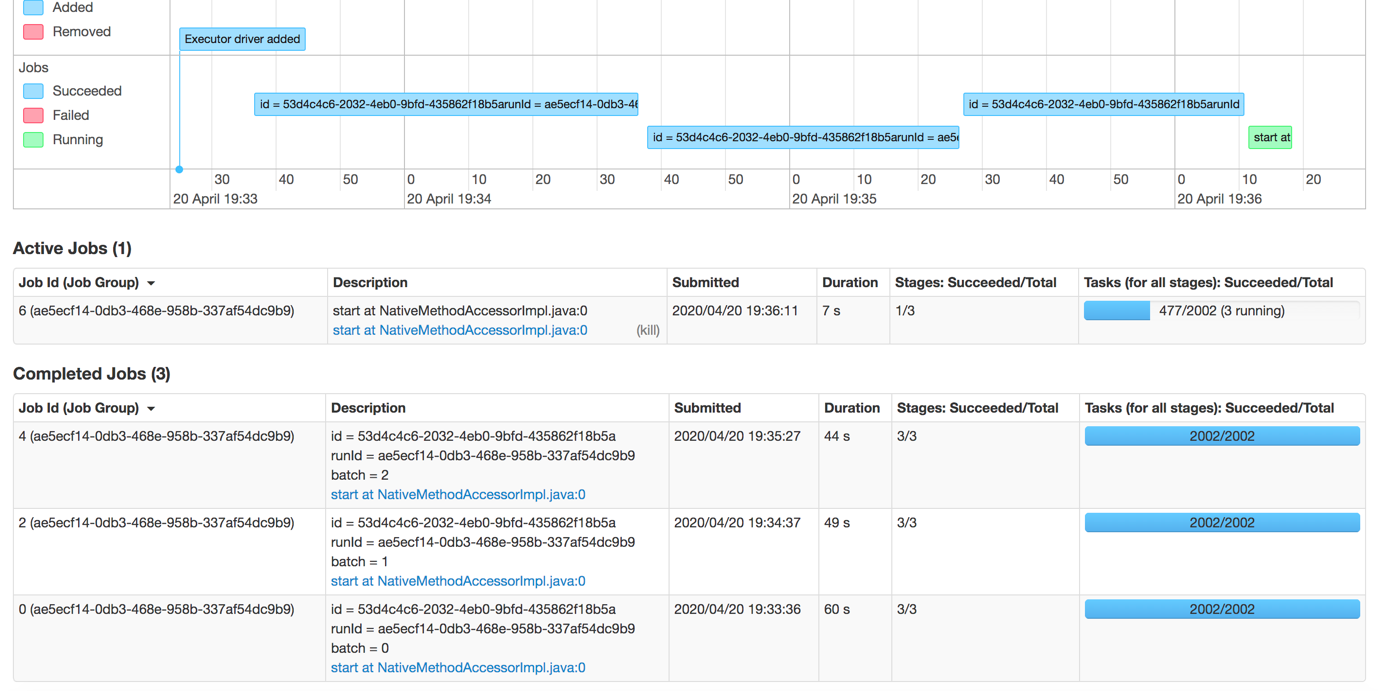


**Spark Streaming UI**

**Optimized performance**



**Non optimized performance**



1. How did changing values on the SparkSession property parameters affect the throughput and latency of the data?

* Number of input rows increases per batch
* Number of processed rows per second increases
* Duration per batch in UI (in seconds)

1. What were the 2-3 most efficient SparkSession property key/value pairs? Through testing multiple variations on values, how can you tell these were the most optimal?

I have tried many configurations as below:

spark.conf.set('spark.executor.memory', '3g')

spark.conf.set("spark.default.parallelism", 100)

spark.conf.set('spark.executor.cores', '3g')

spark.conf.set("spark.streaming.backpressure.enabled", "true")

spark.conf.set("spark.sql.adaptive.enabled", "true")

The only one that made huge difference is:

spark.conf.set("spark.sql.shuffle.partitions", 50)

We can tell it is optimal if duration does not improve any more or get worse behind a certain threshold