

Screenshots and Supporting Materials for DSND Project 2

Step 1:

```
root@54512c05aa06:/home/workspace# kafka-console-consumer --bootstrap-server localhost:9092 --topic "org.sfpd.calls-for-service" --from-beginning
{"offense_date": "2018-12-31T00:00:00.000", "call_time": "22:17"}
{"crime_id": "183653763", "original_crime_type_name": "Traffic Stop", "report_date": "2018-12-31T00:00:00.000", "call_date": "2018-12-31T00:00:00.000", "offense_date": "2018-12-31T00:00:00.000", "call_time": "23:57", "call_date_time": "2018-12-31T23:57:00.000", "disposition": "ADM", "address": "Geary Bl/divisadero St", "city": "San Francisco", "state": "CA", "agency_id": "1", "address_type": "Intersection", "common_location": ""}
{"crime_id": "183653756", "original_crime_type_name": "Traf Violation Cite", "report_date": "2018-12-31T00:00:00.000"}
```

Step 2:

```
Batch: 0
2020-03-15 17:01:10 INFO CodeGenerator:54 - Code generated in 18.342308 ms
2020-03-15 17:01:10 INFO CodeGenerator:54 - Code generated in 9.179744 ms
+-----+-----+
|original_crime_type_name|disposition|count|
+-----+-----+
|null                    |null       |200  |
+-----+-----+
2020-03-15 17:01:10 INFO WriteToDataSourceV2Exec:54 - Data source writer
org.apache.spark.sql.execution.streaming.sources.MicroBatchWriter@2b3d9dc6 committed.
2020-03-15 17:01:10 INFO SparkContext:54 - Starting job: start at NativeMethodAccessorImpl.java:0
2020-03-15 17:01:10 INFO DAGScheduler:54 - Job 1 finished: start at NativeMethodAccessorImpl.java:0, took 0.000072 s
2020-03-15 17:01:10 INFO MicroBatchExecution:54 - Streaming query made progress: {
  "id" : "19a576be-b6f0-40bc-8b5a-366a02c3c2d6",
  "runId" : "c72626b0-4e4a-4a4e-b09e-a0d60efaddb9",
  "name" : null,
  "timestamp" : "2020-03-15T17:00:58.309Z",
  "batchId" : 0,
  "numInputRows" : 200,
  "processedRowsPerSecond" : 15.973165082661128,
  "durationMs" : {
    "addBatch" : 11541,
    "getBatch" : 247,
    "getOffset" : 347,
    "queryPlanning" : 319,
    "triggerExecution" : 12519,
    "walCommit" : 46
  },
  "stateOperators" : [ {
    "numRowsTotal" : 1,
    "numRowsUpdated" : 1,
    "memoryUsedBytes" : 12879
  } ],
  "sources" : [ {
    "description" : "KafkaSource[Subscribe[org.sfpd.calls-for-service]]",
    "startOffset" : null,
    "endOffset" : {
      "org.sfpd.calls-for-service" : {
        "0" : 200
      }
    },
    "numInputRows" : 200,
    "processedRowsPerSecond" : 15.973165082661128
  } ],
  "sink" : {
    "description" : "org.apache.spark.sql.execution.streaming.ConsoleSinkProvider@748cbae4"
  }
}
```

```

2020-03-15 17:01:17 INFO WriteToDataSourceV2Exec:54 - Data source writer
org.apache.spark.sql.execution.streaming.sources.MicroBatchWriter@67a1e5b2 is committing.
-----
Batch: 1
-----
+-----+-----+-----+
|original_crime_type_name|disposition|count|
+-----+-----+-----+
|null                    |null      |400  |
+-----+-----+-----+

2020-03-15 17:01:17 INFO WriteToDataSourceV2Exec:54 - Data source writer
org.apache.spark.sql.execution.streaming.sources.MicroBatchWriter@67a1e5b2 committed.
2020-03-15 17:01:17 INFO SparkContext:54 - Starting job: start at NativeMethodAccessorImpl.java:0
2020-03-15 17:01:17 INFO DAGScheduler:54 - Job 3 finished: start at NativeMethodAccessorImpl.java:0, took 0.000042 s
2020-03-15 17:01:17 INFO MicroBatchExecution:54 - Streaming query made progress: {
  "id" : "19a576be-b6f0-40bc-8b5a-366a02c3c2d6",
  "runId" : "c72626b0-4e4a-4a4e-b09e-a0d60efaddb9",
  "name" : null,
  "timestamp" : "2020-03-15T17:01:10.946Z",
  "batchId" : 1,
  "numInputRows" : 200,
  "inputRowsPerSecond" : 15.826541109440532,
  "processedRowsPerSecond" : 31.700744967506736,
  "durationMs" : {
    "addBatch" : 6138,
    "getBatch" : 9,
    "getOffset" : 7,
    "queryPlanning" : 57,
    "triggerExecution" : 6309,
    "walCommit" : 94
  },
  "stateOperators" : [ {
    "numRowsTotal" : 1,
    "numRowsUpdated" : 1,
    "memoryUsedBytes" : 17599
  } ],
  "sources" : [ {
    "description" : "KafkaSource[Subscribe[org.sfpd.calls-for-service]]",
    "startOffset" : {
      "org.sfpd.calls-for-service" : {
        "0" : 200
      }
    },
    "endOffset" : {
      "org.sfpd.calls-for-service" : {
        "0" : 400
      }
    }
  },
  "numInputRows" : 200,
  "inputRowsPerSecond" : 15.826541109440532,
  "processedRowsPerSecond" : 31.700744967506736
} ],
  "sink" : {
    "description" : "org.apache.spark.sql.execution.streaming.ConsoleSinkProvider@748cbae4"
  }
}
2020-03-15 17:01:20 INFO MicroBatchExecution:54 - Committed offsets for batch 2. Metadata

```

consumer_server.py

```

a294-49a5-af76-7eca9a91b84e
root@b53338bc57d7:/home/workspace# python consumer_server.py
consuming messages:
consumed 0 messages
consuming messages:
consumed 0 messages
consuming messages:
consumed 0 messages
consuming messages:
consumed 0 messages
--consumed message: None: b' "    },\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    {\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"crime_id\\": \\"183653665\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"original_crime_type_name\\": \\"Drunk Driver\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"report_date\\": \\"2018-12-31T00:00:00.000\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"call_date\\": \\"2018-12-31T00:00:00.000\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"offense_date\\": \\"2018-12-31T00:00:00.000\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"call_time\\": \\"23:20\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"call_date_time\\": \\"2018-12-31T23:20:00.000\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"disposition\\": \\"ADV\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"address\\": \\"Balboa St/41st Av\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"city\\": \\"San Francisco\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"state\\": \\"CA\\",\n"'
consumed 1 messages
consuming messages:
--consumed message: None: b' "    \\"agency_id\\": \\"1\\",\n"'

```

Performance tuning settings:

log_output_v0.txt :

```

# readStream
.option("maxOffsetsPerTrigger", 200)
.option("maxRatePerPartition", "")

# writeStream
.trigger(processingTime = '20 seconds')

# baseline around 32 processed rows per sec (prps) - Batch 1

```

log_output_v1.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 200)
.option("maxRatePerPartition", "")

# writeStream
.trigger(processingTime = '2 seconds')

# prps ~28
```

log_output_v2.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 200)
.option("maxRatePerPartition", "")

# writeStream
.trigger(processingTime = '200 seconds')

# prps ~15 (Batch 0 – not representative?)
```

log_output_v3.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", "")

# writeStream
.trigger(processingTime = '20 seconds')

# large difference in prps: 28 before, 287 after (Batch 1)
```

log_output_v4.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", "")

# writeStream
.trigger(processingTime = '2 seconds')

# large negative impact difference in prps: 287 down to 1
```

log_output_v5.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", "")

# writeStream
.trigger(processingTime = '200 seconds')

# prps ~155 (Batch 0 - only batch)
```

log_output_v6.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", 100)

# writeStream
.trigger(processingTime = '20 seconds')

# prps ~323
```

log_output_v7.txt :

```
# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", 1000)

# writeStream
.trigger(processingTime = '20 seconds')

# prps ~294
```

log_output_v8.txt:

```
# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", 10)

# writeStream
.trigger(processingTime = '20 seconds')

# prps ~320
```

log_output_v9.txt:

```
# 3 minute run

# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", 100)

# writeStream
.trigger(processingTime = '40 seconds')

# prps ~306
```

log_output_v10.txt:

```
# 3 minute run

# readStream
.option("maxOffsetsPerTrigger", 2000)
.option("maxRatePerPartition", 10)

# writeStream
.trigger(processingTime = '40 seconds')

# prps ~334
```

#U/Udacity/Data Streaming Nanodegree#