# CS598 Capstone Project Task 2

# rcook4

## Integration

Project code can be publicly viewed here: <https://github.com/rcook4/cs598project>

The pipeline was comprised of the following systems in order:

**EBS Snapshot => Kinesis Agent => Kinesis Stream => Kinesis Analytics => Lambda => DynamoDB**

**ESB Snapshot** files were moved into a folder watched by the **Kinesis Agent**. The **Kinesis Agent** then streamed the file records out to a single **Kinesis Stream**. There was a **Kinesis Analytics** application for each question which ran SQL against the stream using a 1-hour tumbling windows. Each **Kinesis Analytics** sent streaming results to the corresponding per question **Lambda** function. Each **Lambda** function streamed values to the corresponding per question **DynamoDB** table.

## Algorithms

To ensure accuracy, I removed data for any cancelled or diverted flights.

What is the departure delay of a canceled flight?

What is the arrival delay of a diverted flight?

Will diverted/cancelled flights be counted as flights to/from an airport?

The first three questions in group two state to compute the top ten answers ordered by “on-time performance”. The given example answers for those questions used **mean-minutes-of-delay metric** but it is only specifically required for question four which I did not chose. Instead of mean delay I decided to use percentage of flights within an acceptable delay threshold. My reasoning was that I consider impactful delays, such as one causing a missed connection, to more detrimental to “on-time performance” than just being a single data point within an average. I was unsure what the threshold should be until I noticed the dataset itself has an indicator column of whether or not the flight had more than a 15-minute delay. It seems the Bureau of Transportation Statistics created this column as a measure to hold flights accountable to, and 15 minutes sounded like a reasonable buffer. My answers to group two questions use **good-percentage metric** for computing “on-time performance”.

Which carrier looks better to you?

origin uniquecarrier avg\_mins neg\_delay short\_delay long\_delay depdel15 good\_pct

BWI PA (1) 4.8 20 64 21 21 80

BWI EA 8.6 48 5145 895 902 85.2

The SQL query algorithm used for all answered questions entails:

1. Tumble a one-hour window from the source stream
2. Aggregate the metric to a destination stream

In the GitHub project there exists a SQL file named by Group, Question, for each query.

The Lambda algorithm used for all answered questions entails:

1. Attempt to create the DynamoDB table
2. For each record perform a put

In the GitHub project there exists a PY file named by Group, Question, for each function.

## Results

|  |  |  |
| --- | --- | --- |
| **G1Q2 RANK** | **AIRLINE** | **GOOD\_PCT** |
| 1 | HA | 93.79 |
| 2 | AQ | 90.58 |
| 3 | PS | 89.04 |
| 4 | ML(1) | 84.59 |
| 5 | WN | 82.84 |
| 6 | OO | 82.24 |
| 7 | EA | 81.60 |
| 8 | 9E | 80.90 |
| 9 | NW | 80.83 |
| 10 | F9 | 80.43 |

|  |  |  |
| --- | --- | --- |
| **G1Q3 RANK** | **DAYOFWEEK** | **GOOD\_PCT** |
| 1 | SAT | 82.83 |
| 2 | TUE | 81.16 |
| 3 | MON | 81.98 |
| 4 | SUN | 80.41 |
| 5 | WED | 79.57 |
| 6 | THU | 77.02 |
| 7 | FRI | 76.13 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **G2Q1 RANK** | **SRQ** | **CMH** | **JFK** | **SEA** | **BOS** |
| 1 | AA 92.29 | PI 93.63 | UA 88.71 | OO 90.58 | PA(1) 91.56 |
| 2 | TZ 91.68 | ML(1) 93.55 | AA 84.03 | PS 90.25 | ML(1) 90.76 |
| 3 | UA 91.17 | AA 92.13 | XE 83.50 | HA 88.01 | TZ 88.36 |
| 4 | XE 90.65 | NW 90.48 | DH 82.83 | YV 87.59 | EV 87.22 |
| 5 | DL 90.07 | EA 90.40 | CO 82.63 | NW 87.27 | DL 86.74 |
| 6 | EA 89.54 | DL 89.58 | NW 80.00 | DL 86.89 | XE 85.24 |
| 7 | NW 89.52 | DH 88.11 | DL 78.80 | AA 86.75 | EA 85.15 |
| 8 | US 89.43 | US 87.86 | PA(1) 78.75 | US 86.09 | NW 85.13 |
| 9 | TW 88.75 | TW 87.62 | B6 78.58 | CO 85.15 | AA 84.14 |
| 10 | ML(1) 86.72 | CO 86.26 | US 78.54 | DH 85.13 | 9E 83.68 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **G2Q2 RANK** | **SRQ** | **CMH** | **JFK** | **SEA** | **BOS** |
| 1 | EYW 100.00 | ALB 100.00 | ABQ 100.00 | EUG 100.00 | ONT 100.00 |
| 2 | FLL 100.00 | AUS 100.00 | AGS 100.00 | PSC 92.81 | SWF 100.00 |
| 3 | MCO 95.70 | OMA 100.00 | ANC 100.00 | CVG 91.88 | LGA 92.48 |
| 4 | TPA 95.30 | SYR 100.00 | ISP 100.00 | MEM 89.67 | AUS 91.76 |
| 5 | MEM 95.23 | CLE 93.30 | MYR 100.00 | DTW 88.65 | MSY 89.91 |
| 6 | BNA 94.61 | SDF 92.44 | SWF 100.00 | IND 88.05 | BDL 89.84 |
| 7 | RDU 94.39 | IND 91.49 | RDU 96.23 | CLE 87.69 | OAK 87.67 |
| 8 | IAH 93.56 | SLC 90.68 | MDW 91.59 | LIH 87.57 | LGB 87.40 |
| 9 | MSP 91.35 | DFW 90.64 | CLT 91.30 | IAH 87.45 | SJU 86.86 |
| 10 | RSW 91.34 | BNA 90.53 | GSP 89.35 | MSP 87.32 | SJC 86.85 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **G2Q3 RANK** | **LGA=>BOS** | **BOS=>LGA** | **OKC=>DFW** | **MSP=>ATL** |
| 1 | TW 100.00 | TW 100.00 | TW 94.58 | EA 86.72 |
| 2 | PA(1) 89.22 | DL 88.70 | AA 87.91 | OO 82.00 |
| 3 | US 86.42 | US 84.62 | EV 87.91 | FL 79.93 |
| 4 | DL 86.40 | PA(1) 83.73 | OO 85.61 | DL 76.94 |
| 5 | EA 83.78 | EA 81.23 | DL 85.25 | NW 75.30 |
| 6 | MQ 71.92 | MQ 69.89 | MQ 81.97 | OH 74.95 |
| 7 | NW 69.75 | NW 66.05 | OH 0.00 | EV 72.37 |
| 8 | OH 60.61 | AA 50.00 |  |  |
| 9 | AA 0.00 | OH 37.93 |  |  |
| 10 |  | TZ 0.00 |  |  |

I was unable to get accurate results for G3Q2 and I do understand that will cause me to lose 2 points.

**Quality of Results** - **Good (8 points)** - “Results are mostly correct and appropriate. Specifically, results are incorrect or lack important fields for **1** question.”

## Optimizations

I performed system-level cost and performance optimization by using the serverless technologies of Kinesis Stream, Kinesis Analytics, Lambda and DynamoDB.

I performed application-level cost performance optimizations by writing SQL which both vertically partitioned the stream records to the minimum number of needed columns and reduced the number of stream records through the use of a tumbling window.

## Opinion

It was interesting to learn about these online publicly available transportation datasets. I am not a frequent flyer so I will have to assume that the results are sensible. The specific dataset used was more than a decade old the findings may no longer be current. On a personal level, I tend to be more price concerned than quality concerned so it is unlikely that the results will change my air travel behaviors.

## Comparison

I had significant prior experience with data batch processing but I had no prior experience with data stream processing. I suspect experience had a considerable influenced the building of each data pipeline. I found the batch pipeline easier to build, troubleshoot, modify and got faster results. That said I realize that as the world continues to speed up there will continue to be a drive to reduce the reaction time to data and as such data stream processing currently appears to be the only option for true real-time responses.

## Video Demonstration Link

<https://mediaspace.illinois.edu/media/t/0_2uxnlwel/>

I used the tag "CS598CCC-SUMMER-2019" and the upload is published as "unlisted".