Big Data Analytics Assignment 2 solution
Parth Patel
Dt: 03/08/2016

Problem 1 :

Show me the DB statistics. Do you need to create index for some field(s) of this dataset?

Solution :

Created a new DB which only hosts the flight data for the year 2008 as can be confirmed by the number of collections field value. The data was imported from the 2008 air data csv file,using the mongoimport command. Imported a total of `imported 7009728 documents.`

```
Parths-MacBook-Pro:Documents Parth$ mongoimport -d bigDataLab -c flight_2008 --type csv --file /Users/Parth/Desktop/Z604BigDataLab/2008.csv --headerline
2016-03-08T16:00:53.173-0500    connected to: localhost
2016-03-08T16:00:56.160-0500    [........................] bigDataLab.flight_2008      8.7 MB/657.5 MB (1.3%)
2016-03-08T16:00:59.164-0500    [........................] bigDataLab.flight_2008      16.8 MB/657.5 MB (2.6%)
2016-03-08T16:01:02.160-0500    [........................] bigDataLab.flight_2008      25.5 MB/657.5 MB (3.9%)
2016-03-08T16:01:05.164-0500    [#.......................] bigDataLab.flight_2008      34.5 MB/657.5 MB (5.2%)
2016-03-08T16:01:08.165-0500    [#.......................] bigDataLab.flight_2008      42.9 MB/657.5 MB (6.5%)
2016-03-08T16:01:11.160-0500    [#.......................] bigDataLab.flight_2008      52.3 MB/657.5 MB (8.0%)
2016-03-08T16:01:14.164-0500    [##......................] bigDataLab.flight_2008      60.7 MB/657.5 MB (9.2%)
2016-03-08T16:01:17.160-0500    [##......................] bigDataLab.flight_2008      69.5 MB/657.5 MB (10.6%)
2016-03-08T16:01:20.164-0500    [##......................] bigDataLab.flight_2008      77.4 MB/657.5 MB (11.8%)
2016-03-08T16:01:23.162-0500    [###.....................] bigDataLab.flight_2008      85.9 MB/657.5 MB (13.1%)
2016-03-08T16:01:26.160-0500    [###.....................] bigDataLab.flight_2008      95.2 MB/657.5 MB (14.5%)
2016-03-08T16:01:29.161-0500    [###.....................] bigDataLab.flight_2008      103.6 MB/657.5 MB (15.8%)
2016-03-08T16:01:32.161-0500    [####....................] bigDataLab.flight_2008      112.7 MB/657.5 MB (17.1%)
2016-03-08T16:01:35.161-0500    [####....................] bigDataLab.flight_2008      120.9 MB/657.5 MB (18.4%)
2016-03-08T16:01:38.162-0500    [####....................] bigDataLab.flight_2008      129.7 MB/657.5 MB (19.7%)
2016-03-08T16:01:41.160-0500    [#####...................] bigDataLab.flight_2008      137.7 MB/657.5 MB (21.0%)
2016-03-08T16:01:44.160-0500    [#####...................] bigDataLab.flight_2008      146.5 MB/657.5 MB (22.3%)
2016-03-08T16:01:47.161-0500    [#####...................] bigDataLab.flight_2008      154.9 MB/657.5 MB (23.6%)
2016-03-08T16:01:50.160-0500    [#####...................] bigDataLab.flight_2008      164.2 MB/657.5 MB (25.0%)
2016-03-08T16:01:53.162-0500    [######..................] bigDataLab.flight_2008      172.7 MB/657.5 MB (26.3%)
2016-03-08T16:01:56.160-0500    [######..................] bigDataLab.flight_2008      181.3 MB/657.5 MB (27.6%)
2016-03-08T16:01:59.165-0500    [######..................] bigDataLab.flight_2008      188.6 MB/657.5 MB (28.7%)
2016-03-08T16:02:02.160-0500    [#######.................] bigDataLab.flight_2008      195.2 MB/657.5 MB (29.7%)
2016-03-08T16:02:05.167-0500    [#######.................] bigDataLab.flight_2008      203.9 MB/657.5 MB (31.0%)
2016-03-08T16:02:08.161-0500    [#######.................] bigDataLab.flight_2008      211.5 MB/657.5 MB (32.2%)
2016-03-08T16:02:11.162-0500    [#######.................] bigDataLab.flight_2008      216.7 MB/657.5 MB (33.0%)
2016-03-08T16:02:14.161-0500    [########................] bigDataLab.flight_2008      224.9 MB/657.5 MB (34.2%)
2016-03-08T16:02:17.161-0500    [########................] bigDataLab.flight_2008      232.7 MB/657.5 MB (35.4%)
2016-03-08T16:02:20.163-0500    [########................] bigDataLab.flight_2008      240.9 MB/657.5 MB (36.6%)
2016-03-08T16:02:23.161-0500    [########................] bigDataLab.flight_2008      247.5 MB/657.5 MB (37.6%)
2016-03-08T16:02:26.160-0500    [########................] bigDataLab.flight_2008      256.1 MB/657.5 MB (39.0%)
2016-03-08T16:02:29.160-0500    [########................] bigDataLab.flight_2008      265.3 MB/657.5 MB (40.4%)
2016-03-08T16:02:32.162-0500    [########................] bigDataLab.flight_2008      273.8 MB/657.5 MB (41.6%)
2016-03-08T16:02:35.161-0500    [#########...............] bigDataLab.flight_2008      282.2 MB/657.5 MB (42.9%)
2016-03-08T16:02:38.160-0500    [#########...............] bigDataLab.flight_2008      288.4 MB/657.5 MB (43.9%)
2016-03-08T16:02:41.160-0500    [#########...............] bigDataLab.flight_2008      296.2 MB/657.5 MB (45.0%)
2016-03-08T16:02:44.162-0500    [##########..............] bigDataLab.flight_2008      303.6 MB/657.5 MB (46.2%)
2016-03-08T16:02:47.160-0500    [##########..............] bigDataLab.flight_2008      311.9 MB/657.5 MB (47.4%)
2016-03-08T16:02:50.162-0500    [##########..............] bigDataLab.flight_2008      320.4 MB/657.5 MB (48.7%)
2016-03-08T16:02:53.161-0500    [###########.............] bigDataLab.flight_2008      329.1 MB/657.5 MB (50.0%)
2016-03-08T16:02:56.161-0500    [###########.............] bigDataLab.flight_2008      337.4 MB/657.5 MB (51.3%)
2016-03-08T16:02:59.160-0500    [###########.............] bigDataLab.flight_2008      345.9 MB/657.5 MB (52.6%)
2016-03-08T16:03:02.162-0500    [###########.............] bigDataLab.flight_2008      355.0 MB/657.5 MB (54.0%)
2016-03-08T16:03:05.162-0500    [############............] bigDataLab.flight_2008      363.4 MB/657.5 MB (55.3%)
2016-03-08T16:03:08.161-0500    [############............] bigDataLab.flight_2008      370.9 MB/657.5 MB (56.4%)
```

```
2016-03-08T16:03:11.162-0500    [############..........] bigDataLab.flight_2008       379.4 MB/657.5 MB (57.7%)
2016-03-08T16:03:14.160-0500    [############..........] bigDataLab.flight_2008       387.8 MB/657.5 MB (59.0%)
2016-03-08T16:03:17.161-0500    [#############.........] bigDataLab.flight_2008       395.9 MB/657.5 MB (60.2%)
2016-03-08T16:03:20.161-0500    [#############.........] bigDataLab.flight_2008       402.8 MB/657.5 MB (61.3%)
2016-03-08T16:03:23.164-0500    [#############.........] bigDataLab.flight_2008       408.4 MB/657.5 MB (62.1%)
2016-03-08T16:03:26.160-0500    [##############........] bigDataLab.flight_2008       414.0 MB/657.5 MB (63.0%)
2016-03-08T16:03:29.160-0500    [##############........] bigDataLab.flight_2008       421.5 MB/657.5 MB (64.1%)
2016-03-08T16:03:32.160-0500    [##############........] bigDataLab.flight_2008       430.5 MB/657.5 MB (65.5%)
2016-03-08T16:03:35.163-0500    [###############.......] bigDataLab.flight_2008       439.3 MB/657.5 MB (66.8%)
2016-03-08T16:03:38.167-0500    [###############.......] bigDataLab.flight_2008       446.8 MB/657.5 MB (68.0%)
2016-03-08T16:03:41.160-0500    [###############.......] bigDataLab.flight_2008       455.2 MB/657.5 MB (69.2%)
2016-03-08T16:03:44.162-0500    [################......] bigDataLab.flight_2008       463.6 MB/657.5 MB (70.5%)
2016-03-08T16:03:47.161-0500    [################......] bigDataLab.flight_2008       470.0 MB/657.5 MB (71.5%)
2016-03-08T16:03:50.160-0500    [################......] bigDataLab.flight_2008       475.8 MB/657.5 MB (72.4%)
2016-03-08T16:03:53.162-0500    [################......] bigDataLab.flight_2008       484.3 MB/657.5 MB (73.7%)
2016-03-08T16:03:56.161-0500    [################......] bigDataLab.flight_2008       492.6 MB/657.5 MB (74.9%)
2016-03-08T16:03:59.164-0500    [#################.....] bigDataLab.flight_2008       501.2 MB/657.5 MB (76.2%)
2016-03-08T16:04:02.162-0500    [#################.....] bigDataLab.flight_2008       509.7 MB/657.5 MB (77.5%)
2016-03-08T16:04:05.164-0500    [#################.....] bigDataLab.flight_2008       518.2 MB/657.5 MB (78.8%)
2016-03-08T16:04:08.162-0500    [#################.....] bigDataLab.flight_2008       526.2 MB/657.5 MB (80.0%)
2016-03-08T16:04:11.160-0500    [#################.....] bigDataLab.flight_2008       532.6 MB/657.5 MB (81.0%)
2016-03-08T16:04:14.160-0500    [#################.....] bigDataLab.flight_2008       540.7 MB/657.5 MB (82.2%)
2016-03-08T16:04:17.162-0500    [##################....] bigDataLab.flight_2008       549.6 MB/657.5 MB (83.6%)
2016-03-08T16:04:20.162-0500    [##################....] bigDataLab.flight_2008       558.1 MB/657.5 MB (84.9%)
2016-03-08T16:04:23.161-0500    [##################....] bigDataLab.flight_2008       567.4 MB/657.5 MB (86.3%)
2016-03-08T16:04:26.161-0500    [###################...] bigDataLab.flight_2008       576.4 MB/657.5 MB (87.7%)
2016-03-08T16:04:29.160-0500    [###################...] bigDataLab.flight_2008       584.8 MB/657.5 MB (88.9%)
2016-03-08T16:04:32.161-0500    [###################...] bigDataLab.flight_2008       593.2 MB/657.5 MB (90.2%)
2016-03-08T16:04:35.160-0500    [###################...] bigDataLab.flight_2008       601.8 MB/657.5 MB (91.5%)
2016-03-08T16:04:38.161-0500    [####################..] bigDataLab.flight_2008       611.2 MB/657.5 MB (93.0%)
2016-03-08T16:04:41.160-0500    [####################..] bigDataLab.flight_2008       619.6 MB/657.5 MB (94.2%)
2016-03-08T16:04:44.163-0500    [####################..] bigDataLab.flight_2008       628.2 MB/657.5 MB (95.5%)
2016-03-08T16:04:47.161-0500    [#####################.] bigDataLab.flight_2008       635.8 MB/657.5 MB (96.7%)
2016-03-08T16:04:50.161-0500    [#####################.] bigDataLab.flight_2008       644.8 MB/657.5 MB (98.1%)
2016-03-08T16:04:53.161-0500    [#####################.] bigDataLab.flight_2008       652.8 MB/657.5 MB (99.3%)
2016-03-08T16:04:55.298-0500    [######################] bigDataLab.flight_2008       657.5 MB/657.5 MB (100.0%)
2016-03-08T16:04:55.298-0500    imported 7009728 documents
Parths-MacBook-Pro:Documents Parth$
```

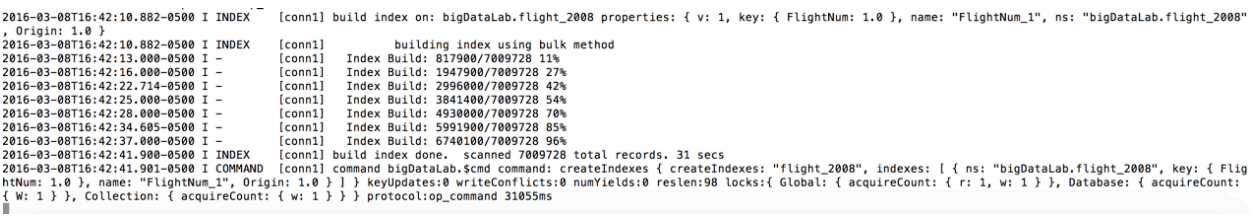
```
> db.stats()
{
        "db" : "bigDataLab",
        "collections" : 1,
        "objects" : 7009728,
        "avgObjSize" : 498.1848970459339,
        "dataSize" : 3492140622,
        "storageSize" : 901554176,
        "numExtents" : 0,
        "indexes" : 1,
        "indexSize" : 63606784,
        "ok" : 1
}
>
```

The stats also show that a default id index is created on the collection which is quite large in size. This is a good indicator that the collection is huge and we should create some more indexes before we go ahead into querying the collection.

Its generally a good idea to create index on the columns/keys over which we will frequently query the database. In case of the given data, I believe we should do normal indexes on **FlightNum, Origin Airport code** (based on the second question) since the queries on these

keys will usually be of the **EQUAL TO OR NOT EQUAL TO** kind. Optionally we can also index **UniqueCarrier,TailNum and Dest**. In addition to this we can have indexes on the keys **ArrTime, DepTime, Month, DayofMonth** etc., since these keys would usually have queries of the **LESS THAN GREATER THAN** kind.

For this particular assignment question, I will create indexes on FlightNum, Origin Airport code since it is the immediate requirement.



Problem 2 :

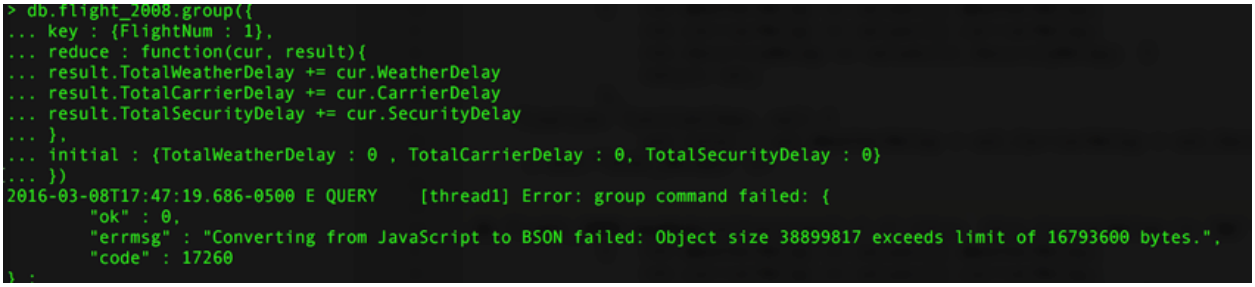
Aggregate WeatherDelay, CarrierDelay, SecurityDelay for each target FlightNum and Origin airport code via MongoDB query (aggregate or mapReduce function? Or Both?). What you found from these query? Give me queries and a brief report.

Solution :

There are two ways to go about solving this problem, one is using aggregation and the other one using map reduce. I prefer using the aggregate function, because it consumes less time compared to map reduce. The primary reason for it being, unlike map reduce the aggregate function performs all the operations in memory. But this depends entirely for the data set at hand, for example, the data for this assignment is considerably huge, and hence the aggregate function throws out of memory error.

Query : db.flight_2008.group (
{
key : { FlightNum : 1},
reduce: function(cur, result){
result.TotalWeatherDelay += cur.WeatherDelay;
result.TotalCarrierDelay += cur.CarrierDelay;

```
                result.TotalSecurityDelay += cur.SecurityDelay},
        initial:{TotalWeatherDelay : 0 , TotalCarrierDelay : 0, TotalSecurityDelay : 0}
    })
```



In order to remove this error, I tried adding a condition which only considers the values which are 0 or more, thus ignoring the NAs

Query : db.flight_2008.group (

```
    {
        key : { FlightNum : 1},
        cond:{CarrierDelay:{$gte:0},SecurityDelay:{$gte:0},WeatherDelay:{$gte:0}},
        reduce: function(cur, result){
                result.TotalWeatherDelay += cur.WeatherDelay;
                result.TotalCarrierDelay += cur.CarrierDelay;
                result.TotalSecurityDelay += cur.SecurityDelay},
        initial:{TotalWeatherDelay:0 ,TotalCarrierDelay:0,TotalSecurityDelay:0}
    })
```

And I got the following result :

```
> db.flight_2008.group (
... {
... key : { FlightNum : 1},
... cond:{CarrierDelay:{$gte:0},SecurityDelay:{$gte:0},WeatherDelay:{$gte:0}},
... reduce: function(cur, result){
... result.TotalWeatherDelay += cur.WeatherDelay;
... result.TotalCarrierDelay += cur.CarrierDelay;
... result.TotalSecurityDelay += cur.SecurityDelay},
... initial:{TotalWeatherDelay:0 ,TotalCarrierDelay:0,TotalSecurityDelay:0}
... })
[
        {
                "FlightNum" : 3920,
                "TotalWeatherDelay" : 693,
                "TotalCarrierDelay" : 1527,
                "TotalSecurityDelay" : 0
        },
        {
                "FlightNum" : 509,
                "TotalWeatherDelay" : 577,
                "TotalCarrierDelay" : 5622,
                "TotalSecurityDelay" : 19
        },
        {
                "FlightNum" : 1333,
                "TotalWeatherDelay" : 300,
                "TotalCarrierDelay" : 2064,
                "TotalSecurityDelay" : 56
        },
        {
                "FlightNum" : 675,
                "TotalWeatherDelay" : 152,
                "TotalCarrierDelay" : 5229,
                "TotalSecurityDelay" : 12
        },
        {
                "FlightNum" : 4,
                "TotalWeatherDelay" : 892,
                "TotalCarrierDelay" : 9484,
                "TotalSecurityDelay" : 63
        },
        {
                "FlightNum" : 54,
                "TotalWeatherDelay" : 1329,
                "TotalCarrierDelay" : 10200,
                "TotalSecurityDelay" : 163
```

Finder

Similarly we can do the same thing for Origin values as follows.

```
Query : db.flight_2008.group (
        {
                key : { Origin : 1},
                cond:{CarrierDelay:{$gte:0},SecurityDelay:{$gte:0},WeatherDelay:{$gte:0}},
                reduce: function(cur, result){
                                result.TotalWeatherDelay += cur.WeatherDelay;
                                result.TotalCarrierDelay += cur.CarrierDelay;
                                result.TotalSecurityDelay += cur.SecurityDelay},
                initial:{TotalWeatherDelay:0 ,TotalCarrierDelay:0,TotalSecurityDelay:0}
        })
```

```
> db.flight_2008.group (
... {
... key : { Origin : 1},
... cond:{CarrierDelay:{$gte:0},SecurityDelay:{$gte:0},WeatherDelay:{$gte:0}},
... reduce: function(cur, result){
... result.TotalWeatherDelay += cur.WeatherDelay;
... result.TotalCarrierDelay += cur.CarrierDelay;
... result.TotalSecurityDelay += cur.SecurityDelay},
... initial:{TotalWeatherDelay:0 ,TotalCarrierDelay:0,TotalSecurityDelay:0}
... })
[
        {
                "Origin" : "IND",
                "TotalWeatherDelay" : 16944,
                "TotalCarrierDelay" : 103854,
                "TotalSecurityDelay" : 447
        },
        {
                "Origin" : "ISP",
                "TotalWeatherDelay" : 6632,
                "TotalCarrierDelay" : 21591,
                "TotalSecurityDelay" : 1915
        },
        {
                "Origin" : "JAN",
                "TotalWeatherDelay" : 10941,
                "TotalCarrierDelay" : 37080,
                "TotalSecurityDelay" : 95
        },
        {
                "Origin" : "JAX",
                "TotalWeatherDelay" : 17671,
                "TotalCarrierDelay" : 68175,
                "TotalSecurityDelay" : 127
        },
        {
                "Origin" : "LAS",
                "TotalWeatherDelay" : 46816,
                "TotalCarrierDelay" : 456943,
                "TotalSecurityDelay" : 4023
        },
        {
                "Origin" : "LAX",
                "TotalWeatherDelay" : 32967,
                "TotalCarrierDelay" : 687746,
```

Note : Here the query first sorts the key in ascending order and groups all the documents belonging to the same key, and later performs the required aggregation, in this case sum.

The exact same results could be obtained by a simple aggregate queries as follows:

```
db.flight_2008.aggregate([

                { $group: { _id: "$Origin", TotalWeatherDelay: { $sum: "$WeatherDelay" }
,TotalCarrierDelay: { $sum: "$CarrierDelay" } ,TotalSecurityDelay: { $sum: "$SecurityDelay" } } },
                { $out: "OriginDelayAggregate" }
               ])


db.flight_2008.aggregate([

                { $group: { _id: "$FlightNum", TotalWeatherDelay: { $sum: "$WeatherDelay" }
,TotalCarrierDelay: { $sum: "$CarrierDelay" } ,TotalSecurityDelay: { $sum: "$SecurityDelay" } } },
                { $out: "FlightNumDelayAggregate" }
               ])
```

Note : Here the $out aggregation property lets us save the aggregation result as a separate collection.

We can further break down the given query to consider groupings on FlightNum and Origin separately or considering grouping them together. The difference in these approaches is that, if we deal with them separately we will first get the sum of the WeatherDelay, CarrierDelay and SecurityDelay for groups distinct values of FlightNum and we have to perform same operation for groups of distinct values of Origin as seen above.