# Airfare Prediction

Wooseok Kim

# What I did

- Serial Part Implementation
- Parallel part (using MapReduce) Implementation
  - No error on eclipse
  - Error on Terminal
- What I wanted to do
  - ~~Time comparison (Serial vs Parallel)~~
  - ~~Airfare comparison (Serial vs Parallel)~~

# Data-Driven Approaches

- Two-parameter Exponential Smoothing
  - One-parameter exponential smoothing equation simply calculates the average value
  - If the series has a trend, an average slope can be estimated as well
  - $L_n$: avg value or length of Seasonality, $T_n$: Trend
  - $F_{n+1} = L_n + T_n$
  - $L_n = \alpha * y_n + (1 - \alpha) * (L_{n-1} + T_{n-1})$
  - $T_n = \beta * (L_n - L_{n-1}) + (1 - \beta) * T_{n-1}$

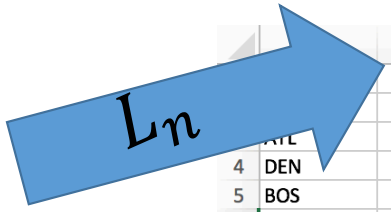
# Data-Driven Approaches

- Two-parameter Exponential Smoothing
  - $F_{n+1} = L_n + T_n\ ,\ L_n = \alpha * y_n + (1-\alpha) * (L_{n-1} + T_{n-1})$
  - $T_n = \beta * (L_n - L_{n-1}) + (1-\beta) * T_{n-1}$
  - Assume that $\alpha$=0.3, $\beta$=0.6, Forecast for Q2 2016 = 320

| | Airfare Avg (ORD) | $L_n$ | $T_n$ | $F_{n+1}$ |
|---|---|---|---|---|
| Q2 2016 | 314.45 | 320 | 0 | |
| Q3 2016 | 308.14 | 318.335 = 0.3*314.45 + 0.7*320 | -0.999 = 0.6(318.335-320)+0.4*0 | 317.336 = 318.335 + (-0.999) |
| Q4 2016 | 313.45 | 314.276 = 0.3*308.14 + 0.7*(318.335-0.999) | -2.835 = 0.6(314.276-318.335)+0.4(-0.999) | 311.441 = 314.276+(-2.835) |

# Serial Part

- Data file
  - Input file (.csv file)
  - Q1_2011 ~ Q2_2016




| | | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| | | 380 | 378.55 | 399.16 | 391.31 | 407.25 | 377.36 |
| | | 390 | 388.95 | 382.13 | 370.96 | 373.97 | 379.33 |
| | [illegible] | 350 | 347.38 | 375.79 | 315.55 | 328.7 | 365.41 |
| 4 | DEN | 315 | 315.92 | 317.77 | 359.74 | 378.08 | 320.96 |
| 5 | BOS | 360 | 360.27 | 361.9 | 400.73 | 353.42 | 392.29 |
| 6 | SFO | 345 | 345.2 | 415.89 | 406.57 | 418.73 | 377.89 |
| 7 | DFW | 385 | 383.18 | 414.73 | 342.99 | 420.88 | 394.06 |
| 8 | SEA | 385 | 386.31 | 369.95 | 368.26 | 367.9 | 359.65 |
| 9 | LGA | 430 | 431.15 | 354.67 | 340.61 | 396.23 | 449.42 |
| 10 | JFK | 350 | 348.99 | 396.85 | 387.2 | 361.3 | 367.14 |
| 11 | PHX | 415 | 413.21 | 326.69 | 316.6 | 324.23 | 454.39 |
| 12 | EWR | 320 | 321.1 | 469.31 | 314.12 | 321.95 | 326.75 |
| 13 | BWI | 365 | 366.03 | 334.31 | 467.03 | 382.32 | 470.78 |
| 14 | PHL | 470 | 469.82 | 388.72 | 383.63 | 473.62 | 406.03 |
| 15 | MCO | 365 | 363.51 | 279.78 | 262.77 | 274.23 | 404.77 |
| 16 | DTW | 305 | 304.15 | 385.09 | 268.44 | 424.77 | 330.25 |
| 17 | MSP | 265 | 267.39 | 405.96 | 406.85 | 386.58 | 289.39 |
| 18 | LAS | 270 | 269.49 | 284.54 | 352.49 | 354.66 | 282.9 |
| 19 | FLL | 275 | 274.12 | 266.38 | 379.09 | 266.88 | 267.31 |
| 20 | SAN | 335 | 333.59 | 357.67 | 256.42 | 269.21 | 336.35 |
| 21 | DCA | 300 | 298.07 | 386.43 | 468.95 | 381.16 | 298.23 |
| 22 | IAH | 380 | 378.22 | 472.76 | 369.97 | 313.94 | 390.59 |
| 23 | TPA | 475 | 476.6 | 309.84 | 305.39 | 289.99 | 334.33 |
| 24 | MDW | 305 | 306.13 | 307.33 | 288.33 | 370.76 | 357.43 |

# Serial Part

- Use the algorithm
- Save outputs as a .txt file
- Can compare predication fare and original fare

# Serial Part

- Algorithm
  - this.f is final outputs

```
public void findAirFare() {


    int j = 0;
    for(int i = 0 ; i < this.count ; ++i) {
        l[i] = fare[i][0]; t[i] = 0.0;
        for(j = 1 ; j < this.fare[0].length ; ++j) {
            this.l[j] = this.alpha*this.fare[i][j] + (1-this.alpha)*(this.l[j-1] + this.t[j-1]);
            //System.out.println("Airport: " + this.airport[i] + "\tLn : " + this.l[j]);
            this.t[j] = this.beta*(this.l[j]-this.l[j-1]) + (1-this.beta)*this.t[j-1];
            //System.out.println("Airport: " + this.airport[i] + "\tTn : " + this.t[j]);
            this.f[i][0] = this.l[j] + this.t[j];

        }
```

# Serial Part

- Output – Text file

```
public void printAirFare(PrintWriter bw) throws IOException{
    for(int i = 0 ; i < this.count ; ++i) {

        bw.write(String.valueOf(this.airport[i]));
        bw.write(",");
        bw.write(String.valueOf(this.f[i][0] +"\n"));
    }
```

output.txt

```
LAX,333.7159644464051
ORD,300.3565217009047
ATL,353.52932425536363
DEN,304.11509893415786
BOS,328.83644336514067
SFO,345.7203497060547
DFW,347.12266282800925
SEA,361.41748044244946
LGA,371.89798027377
JFK,358.92051483283547
PHX,328.369701030891
EWR,289.5238655121581
BWI,248.50010524362594
PHL,194.39173668958244
MCO,404.0593409459097
DTW,370.95777582345846
MSP,294.0234695959812
LAS,243.14380743174453
FLL,338.8505014450679
SAN,325.1062862935129
DCA,309.541328153208
IAH,295.60122544246815
TPA,461.2207938642094
MDW,289.0692690429785
PDX,275.0247474075691
STL,352.642220021762
MCI,361.23636940509124
CLT,378.25675023940096
SMF,264.4700622086045
OAK,344.79803965642606
```

# Serial Part

| Airport | Prediction fare | Original fare |
|---|---|---|
| LAX | 333.7159644464051 | 365.57 |
| ORD | 300.3565217009047 | 324.54 |
| ATL | 353.52932425536363 | 355.13 |
| DEN | 304.11509893415786 | 297.41 |
| BOS | 328.83644336514067 | 405.95 |
| SFO | 345.7203497060547 | 349.44 |
| DFW | 347.12266282800925 | 345.07 |

# Parallel Part

- Original data modification

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | LAX | Los Angeles I | Los Angeles | CA | 377.36 | 405.23 | 199,374 |
| 2 | ORD | Chicago O'Ha | Chicago-O'Ha | IL | 379.33 | 407.34 | 182,266 |
| 3 | BOS | Logan Intern | Boston | MA | 365.41 | 392.39 | 159,577 |
| 4 | DEN | Denver Inter | Denver | CO | 320.96 | 344.66 | 154,845 |
| 5 | ATL | Hartsfield-Ja | Atlanta | GA | 392.29 | 421.26 | 153,946 |
| 6 | SFO | San Francisc | San Francisc | CA | 377.89 | 405.8 | 141,445 |
| 7 | JFK | John F. Kenn | New York-JFI | NY | 394.06 | 423.16 | 139,845 |
| 8 | SEA | Seattle/Taco | Seattle | WA | 359.65 | 386.21 | 132,642 |
| 9 | DFW | Dallas/Fort V | Dallas-DFW | TX | 449.42 | 482.61 | 131,697 |
| 10 | LGA | LaGuardia | New York-La | NY | 367.14 | 394.25 | 126,637 |
| 11 | MSP | Minneapolis- | Minneapolis | MN | 454.39 | 487.95 | 121,442 |
| 12 | PHX | Phoenix Sky | Phoenix | AZ | 326.75 | 350.88 | 117,913 |
| 13 | EWR | Newark Libe | Newark | NJ | 470.78 | 505.54 | 114,925 |
| 14 | PHL | Philadelphia | Philadelphia | PA | 406.03 | 436.01 | 111,882 |
| 15 | DTW | Detroit Metr | Detroit | MI | 404.77 | 434.66 | 110,677 |
| 16 | BWI | Baltimore/W | Baltimore | MD | 330.25 | 354.64 | 107,484 |
| 17 | MCO | Orlando Inte | Orlando | FL | 289.39 | 310.76 | 103,052 |
| 18 | FLL | Fort Lauderd | Fort Lauderd | FL | 282.9 | 303.79 | 93,554 |
| 19 | LAS | McCarran Int | Las Vegas | NV | 267.31 | 287.05 | 91,043 |
| 20 | SAN | San Diego Int | San Diego | CA | 336.35 | 361.18 | 83,579 |
| 21 | MDW | Chicago Mid | Chicago-Mid | IL | 298.23 | 320.25 | 81,183 |
| 22 | DCA | Ronald Reag | Washington- | DC | 390.59 | 419.43 | 80,688 |
| 23 | TPA | Tampa Interr | Tampa | FL | 334.33 | 359.02 | 75,679 |
| 24 | PDX | Portland Inte | Portland | OR | 357.43 | 383.82 | 72,611 |
| 25 | IAH | George Bush | Houston-Inte | TX | 504.58 | 541.85 | 70,218 |
| 26 | STL | St Louis Lam | St. Louis | MO | 359.11 | 385.62 | 65,690 |
| 27 | MCI | Kansas City I | Kansas City | MO | 340.76 | 365.92 | 63,186 |
| 28 | OAK | Metropolitar | Oakland | CA | 277.62 | 298.12 | 61,234 |

# Parallel Part

- Cloudera
- MapReduce

| Class | Input/Output | Key | Value |
|---|---|---|---|
| Mapper | Input | Offset | Data |
| | Output | AirportName, Fare | Original Airfare |
| Reducer | Input | AirportName, Fare | Original Airfare |
| | Output | AirportName, Fare | Airfare Prediction |

# Parallel Part

- Mapper

```
//map output
private final static DoubleWritable outputValue = new DoubleWritable();
// map output Key
private Text outputKey = new Text();

public void map(DoubleWritable key, Text value, Context context) throws IOEx
    airefare af = new airefare(value);

    outputKey.set(af.airport());

    context.write(outputKey, outputValue);
}
```

```
public airefare(Text text) {
    String[] col = text.toString().split(",");

    this.num = 0;
    this.airportCode = col[1];
    this.airportName = col[2];
    this.airportCity = col[3];
    this.airportState = col[4];
    this.fare = Double.parseDouble(col[5]);
```

# Parallel Part

- Reducer

```
public void reduce (Text key, Iterable<DoubleWritable> values, C

    for(DoubleWritable value : values) {
        fare[count] = value.get();
        ++count;
    }

    calculationFare cf = new calculationFare(fare);
    cf.findAirFare();

    double ticketPrice = cf.getF();
    result.set(ticketPrice);

    context.write(key, result);
}
```

```
public void findAirFare() {

    l[0] = fare[0]; t[0] = 0.0;

    for(int i = 1 ; i < 400 ; ++i) {
        this.l[i] = this.alpha*this.fare[i] + (1-this.alpha)*(this.l[i-1] + this.t[i-1]);
        this.t[i] = this.beta*(this.l[i] - this.l[i-1]) + (1-this.beta)*this.t[i-1];
        this.f = this.l[i] + this.t[i];
    }
```

# Parallel Part

- Upload intput files on Hadoop cluster

# Parallel Part