

Homework 03

Umang M Mehta

Friday 1:35pm – 3:15pm

Pseudo-Code for PLAIN

```
public static class TokenizerMapper
    extends Mapper<Object, Text, Text, Text>
public void map(key,value,context)
{
    Sting[] lines = value.toString();
    Extracting all the requierd fileds of Flights into String variables;
    if((year == 2007 && month >=6)|| (year==2008 && month<=5))
    {
        check whether the field are null or not
        if(fileds != null)
        {
            if(origin == ORD && destination != JFK)
            {
                { Creating a String Builder object and appending all the fields comma separated
                emit(destination,LegOne_Fields)
                }
            }
            if(origin != ORD && destination == JFK)
            {
                { Creating a String Builder object and appending all the fields comma separated
                emit(origin,LegOne_Fields)
                }
            }
        }
    }
    // The join is between the destination of Legone flight and origin of Legtwo flight
    }
}
}
```

```

public static class FlightReducer
    extends Reducer<Text,Text,Text,Text>
{ ArrayList LegOne,LegTwo;

public void reducer(key,value,context)
{
Iterating over all the values and spliting it into
String [] while checking for dummy value passed and then
appending to the Array List

Iterating over all the values of arrayList for LegOne and
checking for a match and condition in ArrayList LegTwo
if it satisfies the conditions emitting it.

emit(Dummykey, totaldelay)

}

class CalculateDelayMapper
{
    public map()
    {counter to increment count of flights
    calculating total delay time for all the reduce tasks
    }

    cleanup()
    {emit(Dummy key,total delay of flights)
    }
}

```

```

class AverageReducer
{
emit(Dummy Key, total delay of all flights from the reduced tasks)
}

```

Did your PLAIN program beat Pig?

Yes the PLAIN Program beat the Pig Program.

How did the differences in the Pig programs affect runtime?

FILTERFIRST was fastest of all the pig programs which was followed by JOINFIRST-VERSION 2 and JOINFIRST-VERSION 1

Can you explain why these runtime results happened?

FILTERFIRST was the fastest of all the programs because all the necessary conditions were filtered first before joining. The difference between the other two was mainly because in version 1 of JOINFIRST we check for both flights in the range while in version 2 we only check whether the flight 1 is in range.

Run-Time Calculation:

Programs	Run-Time
1. PLAIN	488 seconds
2. JOINFIRST-VERSION 1	529 seconds
3. JOINFIRST-VERSION 2	505 seconds
4. FILTERFIRST	509 seconds

Average Flight Delay Computation:

Programs	Average Flight Delay Reported
1. PLAIN	50.67124150519758
2. JOINFIRST-VERSION 1	50.67124150519758

3. JOINFIRST-VERSION 2	50.67124150519758
4. FILTERFIRST	50.67124150519758