VTA Service Analysis

Submitted by

Amy Cherian

Pragati Shrivastava

Lasya Boddapati

Introduction

- GOALS:
    - Analyze the VTA Ridership Data available on VTA Open Data website.
    - Evaluate Service Productivity of each Line on Monthly and Weekly basis.
    - Page rank Analysis of the service

- DATASET FEATURES
    - 1GB Ridership Data from Jan 2014 to September 2014
    - Attributes used in the analysis
        - Date
        - Line Number - The bus number that services a specific route
        - Service Number - The day of the trip(Weekday, Weekend or Holiday)
        - Direction Number - Number assigned to the direction the trip is operating
        - On - Number of people counted boarding at the stop
        - Off - Number of people counted alighting at the stop
        - Trip ID - Unique ID number for a given trip
        - Stop name
        - Sequence Number - The sequence of the stop along the route in question

Design / Implementation

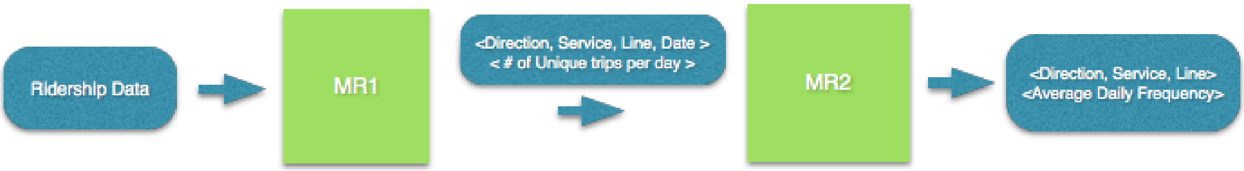
- Graph Representation of the service:
    - Stops correspond to Nodes
    - Edges correspond to the connections between 2 stops
    - Node weights calculated using the below formula ->

      *Node weight = weekday_weight * productivity_on_weekday + weekend_weight * productivity_on_weekend*

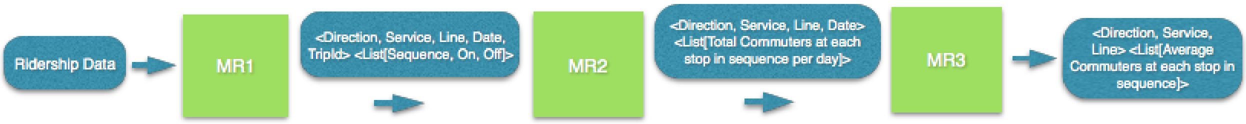
      where *weekday_weight = (5/7), weekend_weight = (2/7*) and

      productivity = Commuters per stop / Frequency of lines passing through the stop

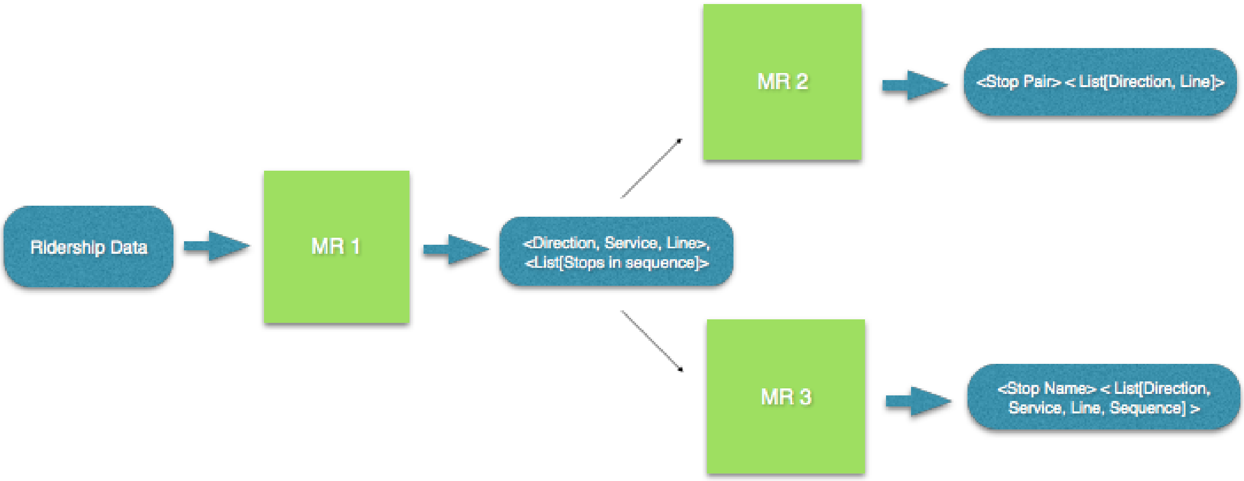
- Adjacency List to represent the graph using all of the below
  - 2 Map Reduce steps to compute the average per line frequency
    - Number of unique trip id's in a day per line (for a particular direction and service number) averaged over all days


Ridership Data
MR1
<Direction, Service, Line, Date >
< # of Unique trips per day >
MR2
<Direction, Service, Line>
<Average Daily Frequency>

  - 3 Map Reduce steps to compute average per stop commuters
    - Per each line (or a particular direction and service number), number of commuters that use each stop in a day, averaged over all days


Ridership Data
MR1
<Direction, Service, Line, Date, TripId> <List[Sequence, On, Off]>
MR2
<Direction, Service, Line, Date>
<List[Total Commuters at each stop in sequence per day]>
MR3
<Direction, Service, Line> <List[Average Commuters at each stop in sequence]>

  - 3 Map Reduce programs to generate the stops list and stop pairs list (which includes the different connecting them)

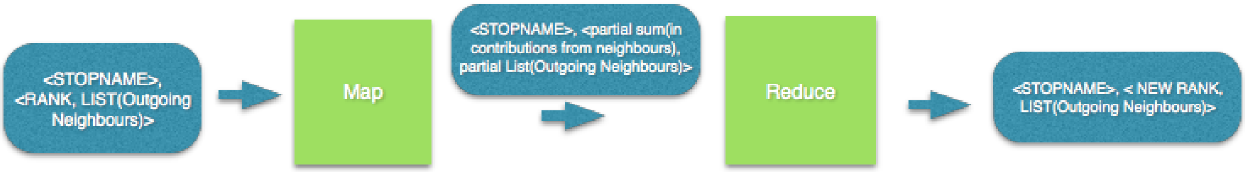

Ridership Data
MR 1
<Direction, Service, Line>,
<List[Stops in sequence]>
MR 2
<Stop Pair> < List[Direction, Line]>
MR 3
<Stop Name> < List[Direction, Service, Line, Sequence] >

- Page Rank Map Reduce implementation
  - 1 Map Reduce step to convert input adjacency list into the following representation:

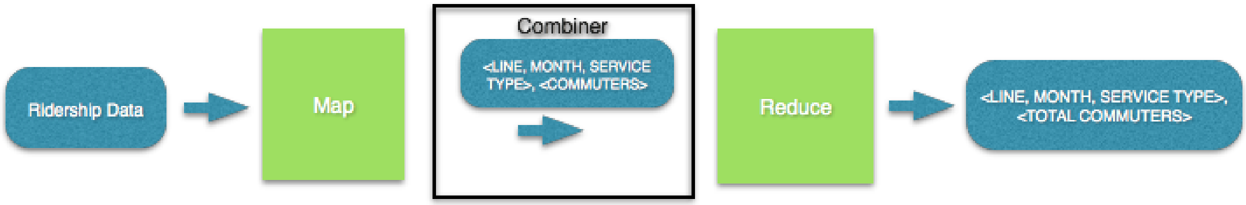
    <STOPNAME> <initial rank , list(Outgoing Neighbours)>

Here the initial rank is assigned by normalizing the stop weights.

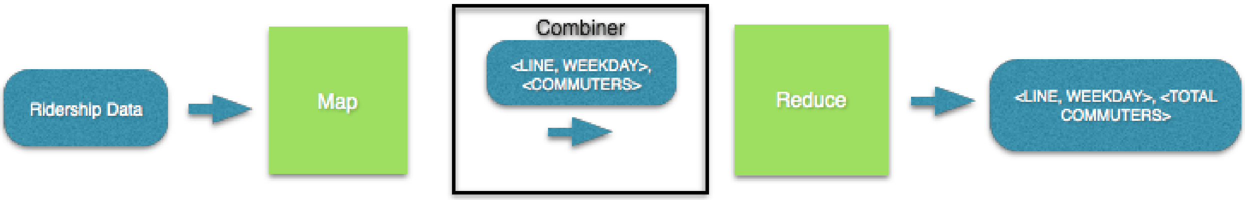
- 1 Map Reduce step to compute the new rank (the iterative step)


<STOPNAME>, <RANK, LIST(Outgoing Neighbours)>
Map
<STOPNAME>, <partial sum(in contributions from neighbours), partial List(Outgoing Neighbours)>
Reduce
<STOPNAME>, < NEW RANK, LIST(Outgoing Neighbours)>

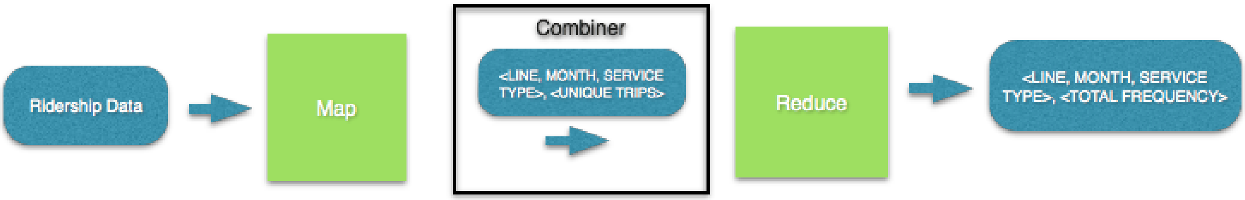
- MONTHLY PER LINE RIDERSHIP


Ridership Data
Map
Combiner
<LINE, MONTH, SERVICE TYPE>, <COMMUTERS>
Reduce
<LINE, MONTH, SERVICE TYPE>, <TOTAL COMMUTERS>

- WEEKLY PER LINE RIDERSHIP


Ridership Data
Map
Combiner
<LINE, WEEKDAY>, <COMMUTERS>
Reduce
<LINE, WEEKDAY>, <TOTAL COMMUTERS>

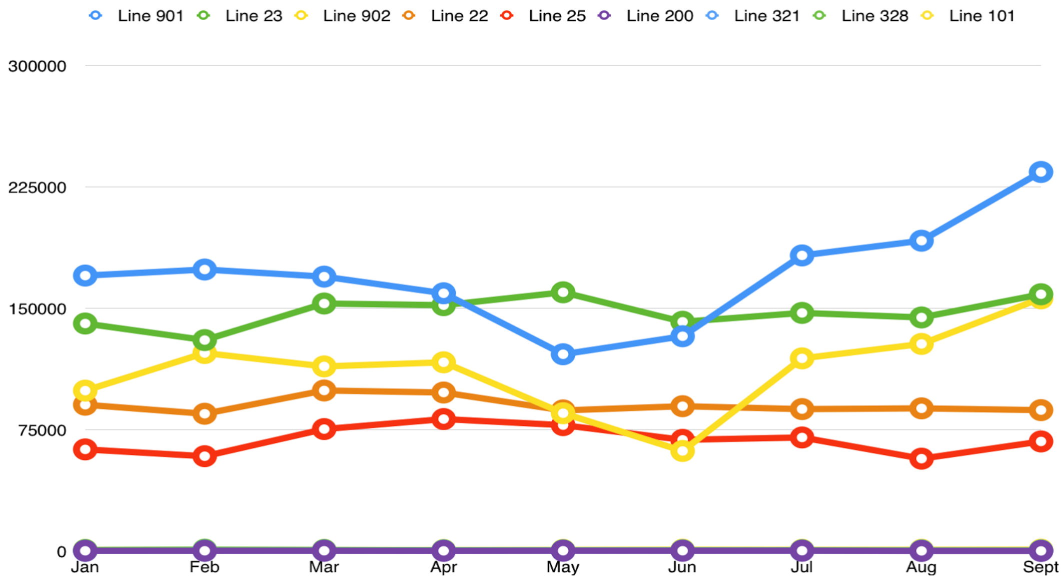
- MONTHLY PER LINE FREQUENCY


Ridership Data
Map
Combiner
<LINE, MONTH, SERVICE TYPE>, <UNIQUE TRIPS>
Reduce
<LINE, MONTH, SERVICE TYPE>, <TOTAL FREQUENCY>

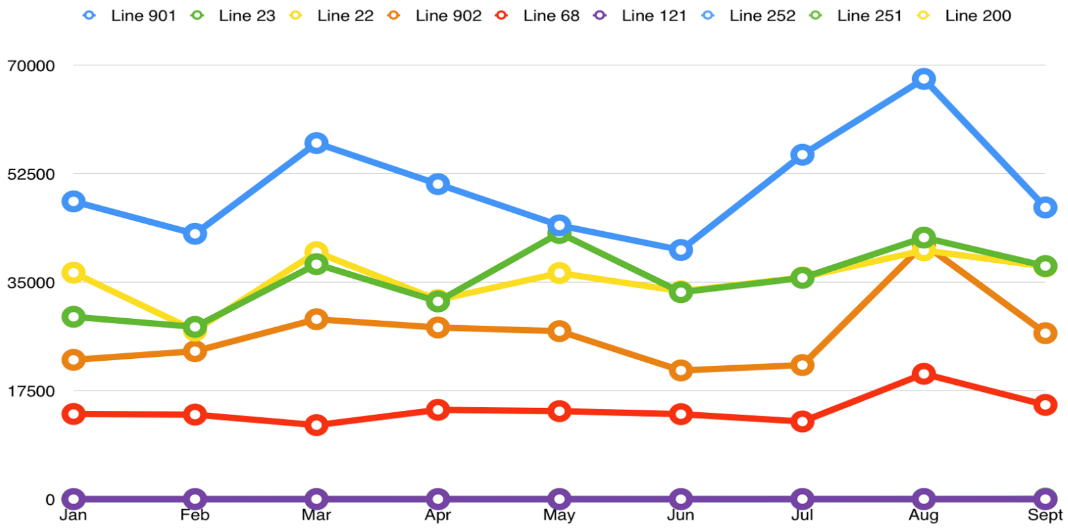
Results

The data analysis paved the path to categorize the findings as follows. The below graphs show the top 5 and least 4 Lines(bus number for a route) for each category.

- ❖ **Per line Monthly Ridership:** The number of commuters for a line on a monthly basis

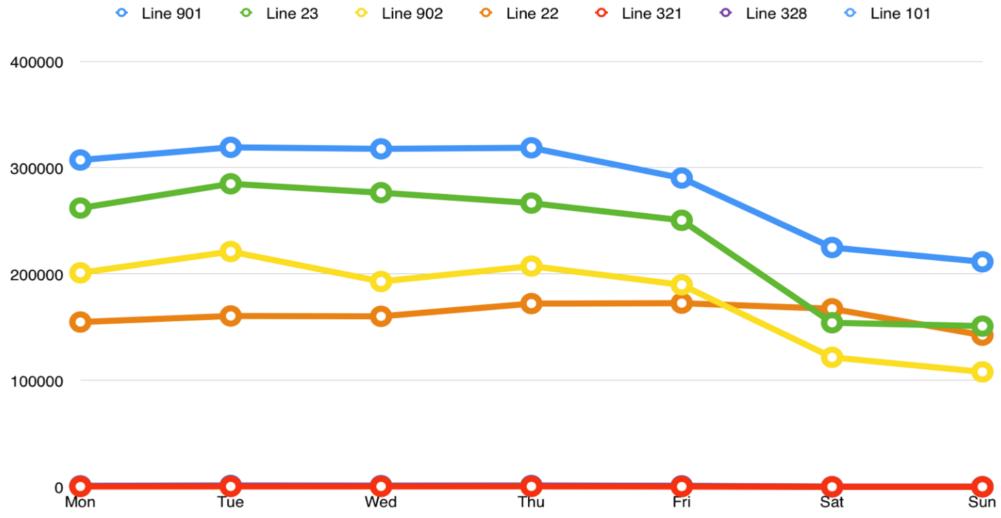


WEEKDAY SERVICE

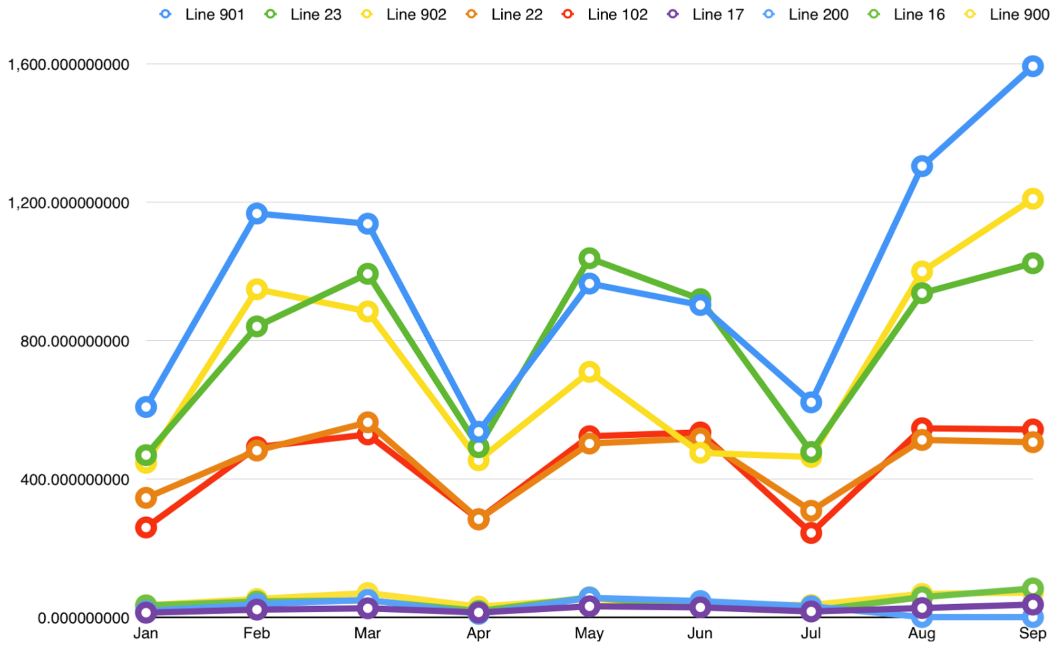


WEEKEND & HOLIDAY SERVICE

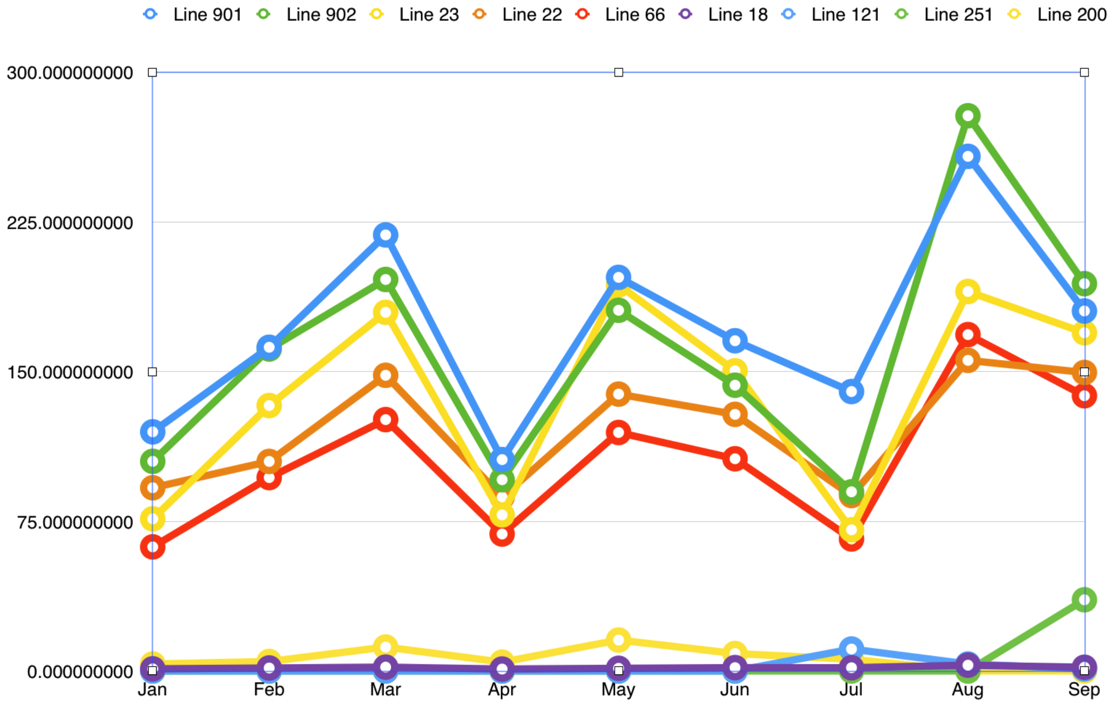
- **Per line Weekly Ridership:** The number of commuters for a line on a weekly basis.



- **Per line Service productivity:** The number of commuters per line divided by the frequency of that line



WEEKDAY SERVICE



WEEKEND & HOLIDAY SERVICE

❖ Page Rank Results

Top 5 Stops

- SAN JOSE CALTRAIN STATION
- SANTA CLARA CALTRAIN STATION
- SANTA CLARA & ALMADEN BLVD
- 2ND & SANTA CLARA
- GREAT MALL / MAIN TRANSIT CENTER

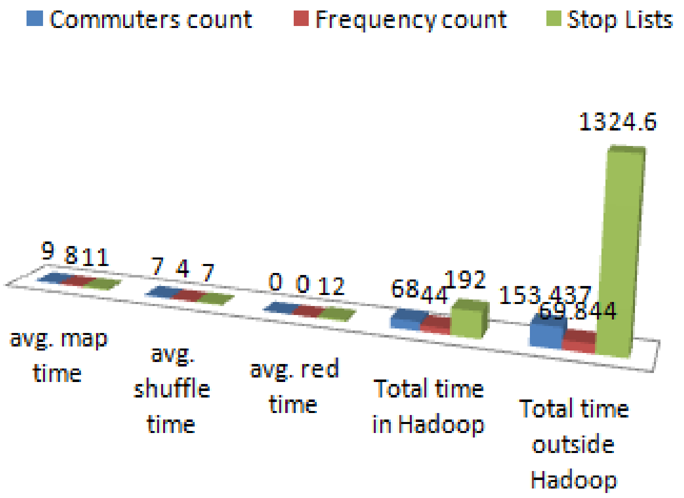
Bottom 5 Stops

- HAMILTON STATION (0)
- WINCHESTER STATION (0)
- CAMPBELL STATION (0)
- BASCOM STATION (0)
- RACE STATION (0)

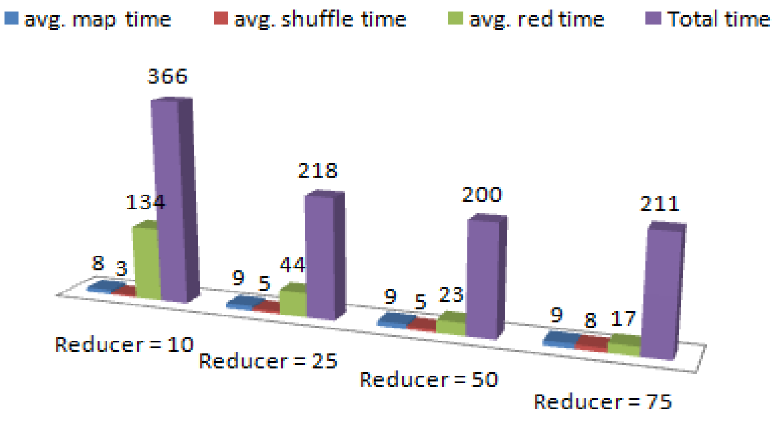
EXPERIMENTS & PERFORMANCE EVALUATIONS

The following experiments were done on the Hadoop ecosystem using mapreduce programs written in python. The cluster size was 24 nodes.

1. Mapreduce time analysis(in secs): The different sets of mapreduce programs were run on the Hadoop ecosystem and also on a single node system and the execution times were compared. The single node systems took significantly longer to complete.



2. Varying the number of reducers for each mapreduce task: As we increased the number of reducers from 10 to 75, the total time for execution was seen to reduce considerably.



RELATED WORK

http://www.vta.org/sfc/servlet.shepherd/version/download/068A0000001FZVM