Statistical Analaysis on Airline Data using Hadoop MapReduce and R

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Suman Saurabh, Subodh Rawani, Sachin Mittal, Abhishek Kumar
Y11UC231, Y11UC230, Y11U189, Y11UC010
`suman.saurabh@lnmiit.ac.in`

Under Guidance of
Dr. Ravi Prakash Gorthi

Department of Computer Science and Engineering
The LNM Institute of Information Technology, Jaipur

January 2015

Statistical Analaysis on Airline Data using Hadoop MapReduce and R

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Computer Science and Engineering

by

Suman Saurabh, Subodh Rawani, Sachin Mittal, Abhishek Kumar
Y11UC231, Y11UC230, Y11U189, Y11UC010
suman.saurabh@lnmiit.ac.in

Under Guidance of
Dr. Ravi Prakash Gorthi

Department of Computer Science and Engineering
The LNM Institute of Information Technology, Jaipur

January 2015

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the project entitled Statistical Analysis of Airline Data using Hadoop MapReduce and R submitted by Suman Saurabh(Y11UC231), Subodh Rawani(Y11UC230), Sachin Mittal(Y11UC189), Abhishek Kumar(Y11UC010) in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by him/her at the Department of Computer Science and Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2014-2015 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my/our opinion, this thesis is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

Date

Adviser: Dr. Ravi Prakash Gorthi

To People who are struck in Traffic

Acknowledgments

I would like to begin by thanking Dr. Ravi Prakash Gorthi, whose uending patience and trust made this work a reality. Working under him will always be wonderful learning experience.

We team members Suman, Sachin, Abhishek and Subodh whose admirable support kept the team spirit alive.

I would also like to thank my friend Abhinash Kumar Jha for useful suggestions and sharing his statistical mind.

And ofcourse to our parents whose consatant motivation has made this four year journey along with thesis admirable.

Abstract

Have you ever been stuck in a traffic and wondered you could have predicted the traffic pattern if you'd had more data? Respectfully digital India is evolving and getting enough traffic data of either Road, Railway or Airline would take years.

So to present our analysis on traffic we utilized Airline Data presented at ASA Data Expo 2009. This dataset is constructed from information made available by the Bureau of Transportation Statistics, USA. It consists of more than 120 million records corresponding to each commercial airline flight in the United States between 1987 and 2008[**?**].

As datasets get larger, real-time visualization becomes more difficult. Supposedly a dataset with a billion entries. If we compute a summary of the dataset and visualize it we will either need non-trivial parallel rendering algorithms or significant time to produce a drawing. This solutions would not scale well. To perform analysis we need to mine relevant data using MapReduce programming.

Airline dataset, because of it's large size we used Hadoop MapReduce to mine the relevant data. Graphical summary were than built in R and analysis were performed on it. It revealed the changing patterns in the flight traffic, cancellation, delays with a spot light on days after 9/11.

Before delving into analytics of airline data, we had a literary overview of how Data Analytics is transforming the digital world using the tools like MapReduce, Amazon EMC. The ongoing importance of consuming and extrapolating Total Data for new business processes and analytics approach has brought us tools like Hadoop, Spark that provides pragmatic, cost-effective, scalable infrastructure for building analytic solution on Big Data.

Big Data has not only changed the tools one can use for predictive analytics, it also changed the entire way of thinking about knowledge extraction and interpretation. Traditionally, data science has always been dominated by trial-and-error analysis, an approach that becomes impossible when datasets are large and heterogeneous.

Contents

Chapter Page

List of Figures

List of Tables

Table Page

Chapter *1*

Introduction

1.1 The Area of Work

1.1.1 What is Big Data& Why it matters

Big data is an evolving term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. Although big data doesn't refer to any specific quantity, the term is often used when speaking about petabytes and exabytes of data.

Big data can be characterized by 3Vs: the extreme volume of data, the wide variety of types of data and the velocity at which the data must be must processed.

Volume Big data implies enormous volumes of data. It used to be employees created data. Now that data is generated by machines, networks and human interaction on systems like social media the volume of data to be analyzed is massive **Variety**. Variety refers to the many sources and types of data both structured and unstructured. We used to store data from sources like spreadsheets and databases. Now data comes in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. This variety of unstructured data creates problems for storage, mining and analyzing data **Velocity**.

Big Data Velocity deals with the pace at which data flows in from sources like business processes, machines, networks and human interaction with things like social media sites, mobile devices, etc. The flow of data is massive and continuous. This real-time data can help researchers and businesses make valuable decisions that provide strategic competitive advantages and ROI if you are able to handle the velocity.

A number of recent technology advancements enable organizations to make the most of big data and big data analytics:

1 Cheap, abundant storage.

2 Faster processors.

3 Affordable open source, distributed big data platforms, such as Hadoop.

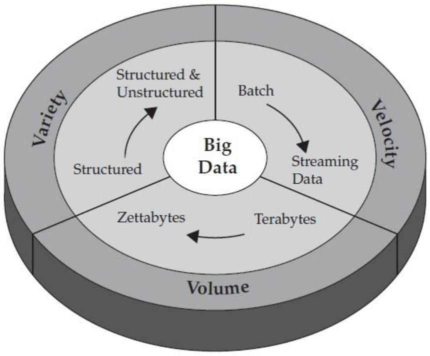

Figure 1-1 *IBM characterizes Big Data by its volume, velocity, and variety—or simply, V^3.*

Figure 1.1

1.1.2 Hadoop

[?] Apache Hadoop is an open source framework written in java for distributed storage and distributed processing of very large data sets on computer clusters built on commodity hardware. It is software framework for storing and processing big data. It accomplishes two tasks massive data storage and faster processing. Hadoop was initially inspired by papers published by Google outlining its approach to handling an avalanche of data, and has since become the de facto standard for storing, processing and analyzing hundreds of terabytes, and even petabytes of data.

Instead of relying on expensive, proprietary hardware and different systems to store and process data, Hadoop enables distributed parallel processing of huge amounts of data across inexpensive, industry-standard servers that both store and process the data, and can scale without limits. With Hadoop no data is considered to be big .It is designed up from a single server to thousands of machines. Rather than relying on high-end hardware, the resiliency of these clusters comes from the software's ability to detect and handle failures at the application layer. And in todays world where lots of amount of data is being created everyday , Hadoops breakthrough advantages mean that businesses and organizations can now find value in data that was recently considered useless. All the modules in hadoop are designed with a hardware (of individual machines, or racks of machines) are commonplace and thus should be automatically handled in software by the framework.

The core of Apache hadoop consists of a Storage part HDFS(Hadoop Distributed File System) and has a processing part (Map Reduce) . Hadoop splits files into large blocks and distributes them amongst the nodes in the cluster.To process the large amount of data hadoop MapReduce transfer codes in the form package to nodes for the parallel processing based on the data each node needs to process. This approach takes in account of data locality to manipulate the data on hand to allow the data to beprocessedfaster and more efficiently than it would be in a more conventionalsupercomputer architecturethat relies on aparallel file systemwhere computation and data are connected via high-speed networking. The Apache hadoop software library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The Project includes these modules:

1. **Hadoop Common** : The common utilities that support the other Hadoop modules i.e the requirements by the other modules.

2. **Hadoop Distributed File System (HDFS)** : It is a distributed file system that stores huge amount of data on commodity machines. A distributed file system that provides high-throughput access to application data.

3.Hadoop YARN : It is a resource management platform i.e. a framework for job scheduling and cluster resource management.

4. **Hadoop MapReduce** : A YARN-based system for parallel processing of large data sets.

Hadoop was created by Doug Cutting, the creator of Apache Lucene, the widely used text search library. Hadoop has its origins in Apache Nutch, an open source web search engine, itself a part of the Lucene project. Nutch was started in 2002, and a working crawler and search system quickly emerged. However their architecture wouldnt scale to the billions of pages on the Web. In 2003 Google published paper on Googles Distributed Filesystem (GFS) which was being used in production at Google. Hence in 2004 they implemented Nutch Distributed Filesystem (NDFS) using GFS architecture that would solve their storage needs for very large files generated as a part of the web crawl and indexing process. In 2004, Google published the paper that introduced MapReduce to the world. NDFS and the MapReduce implementation in Nutch were applicable beyond the realm of search, and in February 2006 they moved out of Nutch to form an independent subproject of Lucene called Hadoop.

1.1.2.1 Why use Hadoop?

Hadoop changes the economics and dynamic scale of large scale computing. There are following characteristics of Hadoop.

1. **Scalable** A cluster can be expanded by adding new servers or resources without having to move, reformat, or change the dependent analytic workflows or applications.

2. **Flexible** Hadoop is schema-less and can absorb any type of data,structured or not, from any number of sources. Data from multiple sources can be joined and aggregated in arbitrary ways enabling deeper analyses than any one system can provide i.e different data set can be combined of different formats to give a efficient result.

3. **Cost effective** Hadoop brings massively parallel computing to commodity servers. The result is a sizeable decrease in the cost per terabyte of storage, which in turn makes it affordable to model all your data.

4. Fault tolerant When you lose a node, the system redirects work to another location of the data and continues processing without missing a beat.

5. Massive storage .The Hadoop framework can store huge amounts of data by breaking the data into blocks and storing it on clusters of lower-cost commodity hardware.

6. Distributed Data is divided and stored across multiple computers, and computations can be run in parallel across multiple connected machines. i.e. collection of large datasets which allows to find out useful information.

1.1.2.2 Hadoop Ecosystem

Hadoop is supplemented by an ecosystem of Apache open-source projects that extend the value of Hadoop and improve its usability. Some of these Apache opensource software projects are:

1. Pig A programming language designed to handle any type of data, helping users to focus more on analyzing large data sets and less on writing map programs and reduce programs.

2. Hive A Hadoop runtime component that allows those fluent with SQL to write Hive Query Language (HQL) statements, which are similar to SQL statements. These are broken down into MapReduce jobs and executed across the cluster.

3. Flume A distributed, reliable and available service for efficiently collecting, aggregating and moving large amounts of log data. Its main goal is to deliver data from applications to the HDFS.

4. HBase A column-oriented non-relational (noSQL) database that runs on top of HDFS and is often used for sparse data sets.

5. Mahout A Scalable machine learning and data mining library.

6. Spark A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.

7. Cassandra A scalable multi-master database with no single points of failure.

1.1.2.3 Hadoop Architecture

Hadoop consists of the Hadoop common package which provides file system and OS level architecture a map reduce engine and a Hadoop distributed file system (HDFS) . The Hadoop Common package contains the necessaryJava ARchive (JAR)files and scripts needed to start Hadoop.

For effective scheduling of the work , every Hadoop compatible file system should provide location awareness the name of the rack where work node is located. The goal is to reduce the impact of a rack power outage or switch failure, so that even if these events occur, the data may still be readable. A small

Hadoop cluster has a single master node and multiple worker nodes. The master node includes a Job Tracker , Task Tracker, Name Node and Data Node.

A slave or worker node behaves as both a Data Node and Task Tracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications. Hadoop requirement is Java Runtime Environment version 1.6 or higher. In a large cluster of data HDFS is controlled by Name Node server to host the file system index, and a secondary Name Node that can generate snapshots of the name node's memory structures. It prevents file-system corruption and reduces loss of data. Job tracker server manages scheduling of jobs in a server. In cluster Hadoop map reduce engine is being used against an alternating file system the name node, secondary name node, data node architecture of HDFS are updated by file system Specific equivalent.

1.1.3 MapReduce

MapReduce is a programming paradigm for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. The framework is divided into two parts: Map, allows to parcels out work to different nodes in the distributed cluster. Reduce, collates the work and resolves the results into a single value.

MapReduce framework consists of a single masterJobTrackerand one slaveTaskTrackerper cluster-node. Master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. Although the Hadoop framework is implemented in Java, MapReduce applications can be written in Python, Ruby, R, C++. Eg. Hadoop Streaming, Hadoop Pipes.

1.1.3.1 Hadoop-MapReduce Architecture

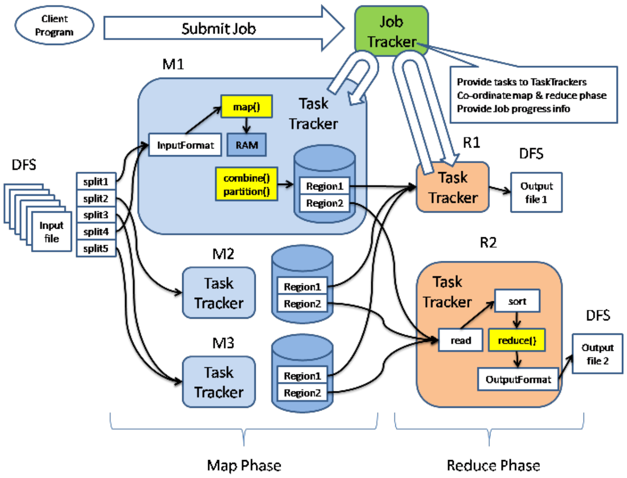

Figure 1.2

Hadoop MapReduce is a software framework for easily writing applications which process vast amounts of data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner. A MapReduce job usually splits the input data-set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to the reduce tasks. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks. The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. The MapReduce framework operates exclusively on ¡key, value¿ pairs, that is, the framework views the input to the job as a set of ¡key, value¿ pairs and produces a set of ¡key, value¿ pairs as the output of the job, conceivably of different types.

Map : Each worker node applies the "map()" function to the local data, and writes the output to a temporary storage. A master node arrange that for redundant copies of input data, only one is processed.

Shuffle : Worker nodes redistribute data based on the output keys (produced by the "map()" function), such that all data belonging to one key is located on the same worker node.

Reduce : Worker nodes now process each group of output data, per key, in parallel.

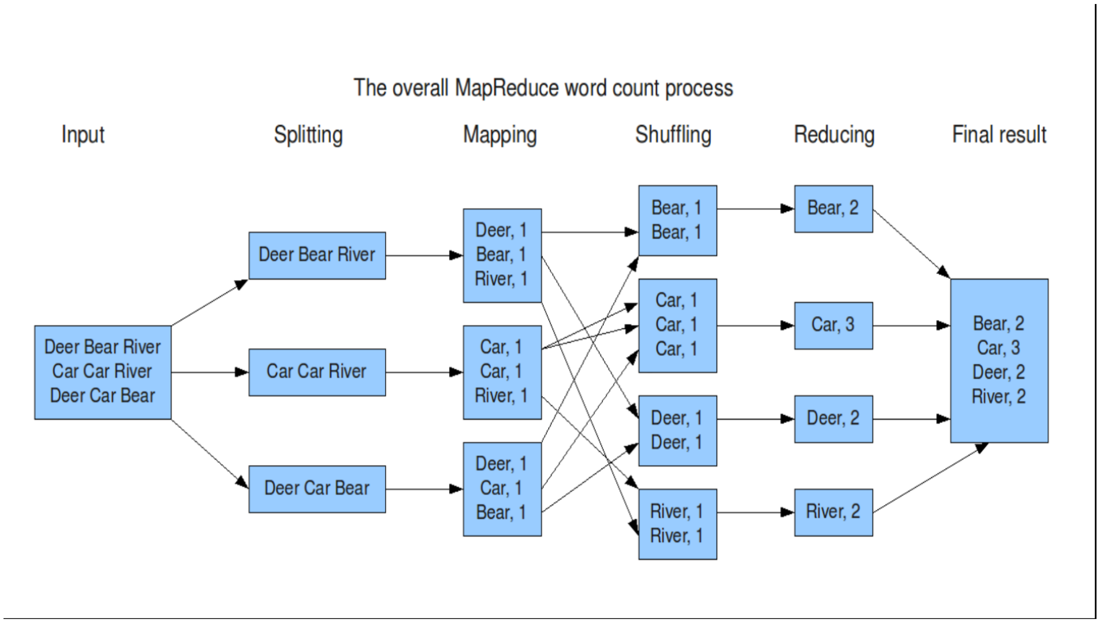

Figure 1.3

1.1.3.2 Map Reduce in Amazon Aws EC2

Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and cost-effectively process vast amounts of data. Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and cost-effectively process vast amounts of data. Amazon EMR simplifies big data processing, providing a managed Hadoop framework that makes it easy, fast, and cost-effective for you to distribute and process vast amounts of your data across dynamically scalable Amazon EC2 instances. The EC2 instances used to run an Elastic MapReduce job flow fall in to one of three categories orinstance groups:

Master The Master instance group contains a single EC2 instance. This instance schedules Hadoop tasks on the Core and Task nodes.

Core The Core instance group contains one or more EC2 instances. These instances useHDFSto store the data for the job flow. They also run mapper and reducer tasks as specified in the job flow. This group can be expanded in order to accelerate a running job flow.

Task The Task instance group contains zero or more EC2 instances and runs mapper and reduce tasks. Since they dont store any data, this group can expand or contract during the course of a job flow.

1.1.3.3 MapReduce MapR

MapR is a third-party application offering an open, enterprise-grade distribution that makes Hadoop easier to use and more dependable. For ease of use, MapR provides network file system (NFS) and open database connectivity (ODBC) interfaces, a comprehensive management suite, and automatic compression. For dependability, MapR provides high availability with a self-healing no-NameNode architecture, and data protection with snapshots, disaster recovery, and with cross-cluster mirroring.

Allthree MapR Editionsfor Hadoop (M7, M5 and M3) are available within the EMR service. MapR M7 is the fastest and most reliable distribution for Apache Hadoop, and includes an enterprise-grade online database that adds the speed, scalability, and flexibility of NoSQL databases. MapR M7 is the only distribution built for running both operational and analytical workloads in the same cluster.

1.1.4 Hadoop Distributions

These products have been emerged out from Hadoop and incresingly become fast, reliable and scalable.

1.1.4.1 Cloudera Hadoop

Cloudera Inc. was founded by big data geniuses from Facebook, Google, Oracle and Yahoo in 2008. It was the first company to develop and distribute Apache Hadoop-based software and still has the largest user base with most number of clients. Although the core of the distribution is based on Apache

Hadoop, it also provides a proprietary Cloudera Management Suite to automate the installation process and provide other services to enhance convenience of users which include reducing deployment time, displaying real time nodes count, etc.

1.1.4.2 Hortonworks Hadoop

Hortonworks, founded in 2011, has quickly emerged as one of the leading vendors of Hadoop. The distribution provides open source platform based on Apache Hadoop for analysing, storing and managing big data. Hortonworks is the only commercial vendor to distribute complete open source Apache Hadoop without additional proprietary software. Hortonworks distribution HDP2.0 can be directly downloaded from their website free of cost and is easy to install. The engineers of Hortonworks are behind most of Hadoops recent innovations including Yarn, which is better than MapReduce in the sense that it will enable inclusion of more data processing frameworks.

1.1.4.3 Apache Spark

Unlike Hadoop which is a batch processing system, it provides Real-Time Analytics.

1. **Fast Analytics and Stream Processing** Apache Spark is an open source, parallel data processing framework that complements ApacheHadoopto make it easy to develop fast, unified Big Data applications combining batch, streaming, and interactive analytics on all your data

2. **Fast, Powerful Data Processing** For analysts and data scientists who rely on iterative algorithms (e.g. clustering/classification), Spark is 10-100x faster than MapReduce delivering faster time to insight on more data, resulting in better business decisions and user outcomes. Spark is

   Fast : Data processing up to 100x faster than MapReduce, both in-memory and on disk

   Powerful :Write sophisticated parallel applications quickly in Java, Scala, or Python without having to think in terms of only map and reduce operators

   Integrated :Spark is deeply integrated with CDH, able to read any data in HDFS and deployed through Cloudera Manager

1.1.5 Data Analytics and Visualization

Data analytics (DA) is the science of examining raw data with the purpose of drawing conclusions about that information. Data analytics is used in many industries to allow companies and organization to make better business decisions and in the sciences to verify or disprove existing models or theories. Data analytics is distinguished from data mining by the scope, purpose and focus of the analysis. Data miners sort through huge data sets using sophisticated software to identify undiscovered patterns and establish hidden relationships. Data analytics focuses on inference, the process of deriving a conclusion based solely on what is already known by the researcher.

Data visualization is the presentation of data in a pictorial or graphical format. For centuries, people have depended on visual representations such as charts and maps to understand information more easily and quickly.

As more and more data is collected and analyzed, decision makers at all levels welcome data visualization software that enables them to see analytical results presented visually, find relevance among the millions of variables, communicate concepts and hypotheses to others, and even predict the future.

Because of the way the human brain processes information, it is faster for people to grasp the meaning of many data points when they are displayed in charts and graphs rather than poring over piles of spreadsheets or reading pages and pages of reports.

Interactive data visualization goes a step further moving beyond the display of static graphics and spreadsheets to using computers and mobile devices to drill down into charts and graphs for more details, and interactively (and immediately) changing what data you see and how it is processed.

1.1.5.1 Tools for Data Visualization

There are abundant visulization tools but most widely used opensource tools are:

ggplot ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.

googleVis googleVis is an R package providing an interface between R and Google Charts. The functions of the package allow the user to visualise data with the Google Chart Tools without uploading their data to Google. The output of googleVis functions is html code that contains the data and references to JavaScript functions hosted by Google. To view the output a browser with Flash and Internet connection is required, the actual chart is rendered in the browser.

D3 D3 allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, you can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive SVG bar chart with smooth transitions and interaction.

1.2 Problem Addressed

Aviation and air travel has established itself as a key economic and social resource in modern times. As the world population increases and becomes ever more interconnected, the demand for air travel will only increase. Currently there are over 100,000 commercial aviation flights and over 200,000 general aviation flights within the national airspace system (NAS) every day. This does not include military sorties or other, special purpose, flights within the NAS. The number of passengers flying to or from the

U.S. is expected to grow an average of 4.5% annually, with cargo amounts showing a similar increase, while general aviation is expected to grow 1% annually. In addition, there is increasing interest, from both government and commercial sectors, in integrating unmanned aerial vehicles (UAVs) into the NAS. Though full UAV integration poses its own unique set of complications, nevertheless it is only a matter of time before they contribute to the air traffic over the NAS. This constant increase in air traffic within the increasingly congested NAS will require new methods and techniques to efficiently accommodate new traffic.

To address these issues, the US Congress approved plans for the development of the Next Generation Air Transportation System (NextGen). It is an overhaul of the current NAS with the goals of allowing more aircraft to safely fly closer together with more direct routes. It is scheduled for implementation in stages between 2012 and 2025 with 5 major elements: (i) Automatic dependent surveillance-broadcast (ADS-B) will replace radar systems with satellite based global positioning information for each aircraft. This infor- mation will be broadcast in realtime to airports another aircraft within a 150 mile radius allowing them to fly closer without jeopardizing safety. (ii) Systemwide information management (SWIM) is a consolidation of multiple information systems into a single coherent system and will reduce redundancy and facilitate information sharing. (iii) NextGen data communication will add data links between aircraft and air traffic controllers to the current two-way voice communication. (iv) NextGen network enabled weather is an ambitious effort to fuse data from tens of thousands of ground, air, and space based sensors into a single national weather information system to provide realtime weather information. (v) NAS voice switch (NVS) will replace multiple existing voice switching systems into a single consolidated air/ground and ground/ground voice communication system.The NextGen system will provide the infrastructure to allow aircraft to safely fly closer together thereby making more efficient use of limited airspace. It will allow aircraft to use more direct routes instead of being constrained to predetermined sky highways thereby reducing congestion and reduce fuel costs. With pieces of the NextGen infrastructure coming into place, there is an opportunity to further their benefits by developing software tools that provide added value.

This paper focuses on visual analysis tools to study the changes on air traffic congestion in span of 21 years which would allow policy makers to see the effects of changing the aircraft separation volume on congestion. The same tool can also be used as a decision aid for processing requests for unmanned aerial vehicle operations. Specifically, this paper will discuss methods and tools used to calculate and render air traffic densities over areas of interest, as well as methods for aggregating such traffic densities over different time scales to extract fluctuations and periodic cycles in traffic patterns. We apply these tools to study the effects of possible modifications to the current en-route aircraft separation requirements. These modification, which are based on the characteristics of large fixed wing aircraft, has the potential of increasing the amount of available air space, allowing for future increases in overall air traffic numbers. In addition, we apply the same suite of tools to provide a quick visual inspection of planned UAV operation under different aircraft separation requirements. The studies conducted in this paper are based on a data set which is constructed from information made available by the Bureau of Transportation Statistics,

There are over 300,000 flightswithin the United States every day. In the future, daily air traffic number of all varieties are expected to continue rising. In addition, there is increasing interest in integrating unmanned aerial vehicles, for both government and commercial interests, into the national airspace system (NAS). This large growth in aviation operations will only increase traffic within the already limited NAS, leading to higher congestion and less free airspace. In this report, we present visual analysis tools to study the changes on air traffic congestion in span of 21 years. The tools support visualization of time-varying air traffic density over an area of interest using different time granularity. We use this visual analysis platform to investigate how changing the aircraft separation volume can reduce congestion while maintaining key safety requirements. The same tool can also be used as a decision aid for processing requests for unmanned aerial vehicle operations.

To present our analysis on traffic we utilized Airline Data presented at ASA Data Expo 2009. This dataset is constructed from information made available by the Bureau of Transportation Statistics, USA. It consists of more than 120 million records corresponding to each commercial airline flight in the United States between 1987 and 2008. As datasets gets larger, real-time visualization becomes more difficult. Supposedly a dataset with a billion entries. If we compute a summary of the dataset and visualize it we will either need non-trivial parallel rendering algorithms or significant time to produce a drawing. This solutions would not scale well. To perform analysis we need to mine relevant data using MapReduce programming.

1.2.1 Data

The data comes originally from RITA where it is described in detail. It can download the data there, or from the bzipped csv files listed below. These files have derivable variables removed, are packaged in yearly chunks and have been more heavily compressed than the originals.

Variable	Description	Variable	Description
Year	1987-2008	DepDelay	departure delay, in minutes
Month	1-12	Origin	origin IATA airport code
DayofMonth	1-31	Dest	destination IATA airport code
DayOfWeek	1 (Monday) - 7 (Sunday)	Distance	in miles
DepTime	actual departure time (local, hhmm)	TaxiIn	taxi in time, in minutes
CRSDepTime	scheduled departure time (local, hhmm)	TaxiOut	taxi out time in minutes
ArrTime	actual arrival time (local, hhmm)	Cancelled	was the flight cancelled?
CRSArrTime	scheduled arrival time (local, hhmm)	CancellationCode	reason for cancellation (A = ca
UniqueCarrier	unique carrier code	Diverted	1 = yes, 0 = no
FlightNum	flight number	CarrierDelay	in minutes
TailNum	plane tail number	WeatherDelay	in minutes
ActualElapsedTime	in minutes	NASDelay	in minutes
CRSElapsedTime	in minutes	SecurityDelay	in minutes
AirTime	in minutes	LateAircraftDelay	in minutes
ArrDelay	arrival delay, in minutes		

1.2.2 Goal

This is intentionally vague in order to allow different entries to focus on different aspects of the data, but here are a few ideas to that we focussed on :

- Summarize data by time periods, airport, and carrier
- Temporal effects
  - Are some time periods more prone to delays than others?
  - Relationships between delays and *Seasonal factors*: winter, summer, holidays *Weather factors*: Blizzards and severe weather *Daily factors*: Time of day, day of week
- Spatial effects
  - Are some airports more prone to delays than others?
- Carrier effects
  - Are some carriers more prone to delays than others?
- Analysis of traffic on New York and Chicago a densly populated metropolitan cities in USA?

1.2.3 Tools Used

Rhipe packages are used for the development of the MapReduce modal. R is used for Visualization along with googleVis and Shiny app to make it interactive.

1.2.3.1 R

R is a programming language and software environment for statistical computing and graphics. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

1.2.3.2 Rhipe

RHIPE is an R package that provides a way to use Hadoop from R. It can be used on its own or as part of the Tessera environment. RHIPE (hree-pay') is the R and Hadoop Integrated Programming Environment. RHIPE is a merger of R and Hadoop. R is the widely used, highly acclaimed interactive language and environment for data analysis. Hadoop consists of the Hadoop Distributed File System (HDFS) and the MapReduce distributed compute engine. RHIPE allows an analyst to carry out D&R analysis of complex big data wholly from within R. RHIPE communicates with Hadoop to carry out the big, parallel computations.

1.2.3.3 ggplot

ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts. It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.

1.3 Existing System

The main thrust of this paper is on visual analysis of air traffic data. Hence, this section focuses on work related to visualizing air traffic data. One of the most popular technique for visualizing air traffic data is to represent the trajectory of each aircraft as an animated particle. Many such visualizations are available on the web via sites such as youtube. A version that was designed by Aaron Koblin demonstrates several techniques and embellishments for presenting the flight trajectories. More recently, the discrete nature of the flight tracks were smoothed out to obtain a continuous estimate of air traffic density using a view dependent kernel density estimator. Representing air traffic data as a density plot is not new. Kellner [8] also used density plots of the arrival and departure rates of aircraft at different airports to assess their capacity. This paper will use similar techniques in visualizing the air traffic data. More importantly, our work examines the impact of varying minimum aircraft separation policy on air traffic density, and also examines if a flight plan, e.g. of a UAV operation request, will endanger existing flight patterns.

There are many factors affecting air traffic congestion and airport capacity. One of those that is controllable and fall under policy decisions is the specification of minimum separation between aircraft. Currently, this is set to 5 nautical miles horizontally, and 1,000 feet vertically [4] when the aircraft is en-route. This limit is adjusted as the aircraft approaches an airport and can drop to 3 miles horizontally on landing approaches to airports. The relative weight class of the leading and following aircraft are also taken into con- sideration in such situations in order to reduce risks due to wake turbulence [3]. The en-route limit accounts for aircraft speed (typical passenger jets fly at average speed of 500 miles per hour or just over 8 miles per minutes), weather impact on visibility, and wake turbulence from leading aircraft, among other factors. With the touted capabilities of ADS-B, the NextGen enabled weather system, and integrated information system, one can theoretically safely reduce the minimum separation requirements between aircraft. This paper provides visual analysis tools to examine the effects of different shapes and parameters describing the minimum separation volume between aircraft.

With regards to UAV operation, they are more generally referred to as Unmanned Aircraft Systems (UAS)[7, 2]. Over the past few years, interest in UAS has rapidly increased. This is because of the possibilities they offer to both government and commercial interests. They would enable a broad range of satellite-like abilities, but at a much lower cost. Aerial photography, communications, environmental monitoring, and security are some of the abilities that UAS deployment could make possible on a large scale. Currently, UAS are predominantly used by the Department of Defense and the Department of Homeland Security, and often outside of national air space (NAS). A handful of UAS are allowed to operate inside our NAS, though almost exclusively for national security or research purposes. However, each UAS operation must be pre-approved by the FAA on a case by case basis. This process is very tedious and does not scale well to large numbers of flights. There are a few studies on risk managment of operating UAS. A recent study uses a site-specific non- uniform probabilistic background air traffic to study the risks [11]. Using the visual analysis tools presented in this paper, checking whether the flight plan for a UAS will allow for a safe operation within the NAS can be accomplished expeditiously.

1.3.1 The Oracle Airline Data Model

The Oracle Airline Data Model is a powerful logical and physical data model that will help airlines effectively store, manage, and analyze airline data that currently resides in passenger service systems (includes reservation systems and departure control systems), global distribution system (GDS), loyalty management systems, and customer data warehouses. It provides a single scalable repository for transactional and historical data that can be used to provide real-time business intelligence and strategic insights youre your airline. Using sophisticated trending and data mining capabilities based on Oracles OLAP and data mining technology, airline personnel will now have the data analysis capabilities to develop Airline -specific insights that are relevant, actionable, and can improve both top-line and bottom-line results.

The Oracle Airline Data Model provides detail transaction storage and advanced analysis into a full range of airline subject areas, including reservations, sales, operations, loyalty, and finance. Using reservation data, the data model can provide detailed insight into passenger bookings by time period, fare class, and flight. It provides insights into channel performance looking at bookings, cancellations, and revenues through travel agency, OTA, ticket counter, call center, and web channels. It allows you to analyze passenger revenues by geography, time period, and flight. Finally it provides insights into loyalty program member activity through a variety of reports. The data model fits the needs of large network carriers and low-cost carriers.

Chapter *2*

Literature survey

2.1 Introduction

Air traffic data usually consists of a collection of flight trajectories of different aircraft. Each flight trajectory usually contains information about the type of aircraft, origin and destination airports, followed by a series of entries that records the time, location, and altitude of the aircraft. The flight tracks are usually recorded in 10 second intervals. Other information such as date, heading, velocity, etc. are generally recorded as well, but were not available in the data set used in our study. The data set used to test and demonstrate our visual analysis tool has an area of interest that is New York an Chicago. Two of the largest metropolitan cities in United States of America. It includes all flight path information from flights that took place from the begining of Jan 1987 to the end of Dec 2008. There are 349,992 unique flight path records in this particular data set. This data set is comprised of uniquely identified flight paths, each containing latitude, longitude, and altitude information at 10 second intervals for the duration of the flight within the area of interest. The time of day and month in which the flights took place are specified. However, the specific date the flight took place is not included.

2.1.1 Flight Data Monitoring

[**?**] Many airlines collect and analyze flight data of routine flights. The process is generally referred as flight data monitoring, which involves data acquisition, transmission, storage and analysis, which are described in detail in this section. By reviewing a number of software tools for flight data analysis, a benchmark of current flight data analysis methods was established. Improvement opportunities were identified from the literature review, which motivated this research.

2.1.2 Anomaly Detection

The approach for anomally detection is as described in[1]. The approach that detects abnormal flights from routine airline operations using FDR data and asks domain experts to interpret the results and operational implications. Thus, anomaly detection algorithms will be developed to detect anomalies from FDR data. Anomaly detection refers to the problem of detecting an observation (or patterns of observations) that is inconsistent with the majority members of the dataset. It is also referred to as novelty detection, anomaly detection, fault detection, deviation detection, or exception mining in different application domains. A significant number of anomaly detection techniques have been developed. While some of the techniques are generic and can be applied to different application problems, many of them are focused on solving particular types of problems in an application domain.

2.1.3 General Anomaly Detection Techniques

Many anomaly detection techniques have been developed to address anomaly detection problems in many application domains. Three main approaches have been taken: statistical approach, classification approach, and clustering approach. The categories are not mutually exclusive as some of the techniques adopt concepts from more than one basic approach. For eg. there is anomaly in the flight traffic pattern at the end of year 2001 related to the 9/11 terrorist attack.

Chapter *3*

Proposed Work

In this thesis, we investigate and visualize data for domestic flights for US airline data in focus with flight orignated at New York and Chicago, the two most densly populated cities in United States of America.The data set contains more than 120 million flight records from October 1987 to December 2008. The thesis reflects the process followed by analysts working with big data: sampling is used to generate hypotheses that are then tested against the complete dataset. The work is similar to the work done by Michael T Crotty[**?**]

The computation for the comparison of their informal ”rule” and analyses of the distribution of the population values requires coding MapReduce programming modal. R to be used as a tool as a major component for data mining and visulization as R is one of the most powerful tool for statistical data analysis. Many have argued that statistics students need additional facility to express statistical computations. By introducing students to commonplace tools for data management, visualization, and reproducible analysis in data science and applying these to real-world scenarios, prepares them to think statistically. The statistical data analysis cycle involves the formulation of questions, collection of data, analysis, and interpretation of results. Data preparation and manipulation is not just a first step, but a key component of this cycle. When working with data, analysts must first determine what is needed, describe this solution in terms that a computer can understand, and execute the code.

For analysing airline data we first looked at the variables provided by the data and reports that they cab predict. Here are summary of influential factors for developing predicting system on Airline Data.

3.0.4 On Time Performance

3.0.4.1 Factors to be considered when making a Reservation

Time Factors : Month, Day of Week, Departure Time, Arrival Time

Location’s Factor : Whether departure or arrival airport is large medium or small.

Others : Distance, Taxi-out, Taxi-in

3.0.4.2 Causes of Flight Delay and Cancellations

Air Carrier : Maintenance or Crew problem, Aircraft Cleaning or Baggage Loading.

Extreme Weather : Significant metrological conditions(actual or forecasted), that in the judgement of carrier, delays or prevents the operation of a flight such as tornado, blizzard and hurricane.

National Aviation System : Non- Extreme weather conditions, heavy traffic volume and air-traffic control.

Late- Arriving Aircraft : A prevoius flight with same aircraft arrived late, causing the present flight to depart late.

Security : Delays or Cancellations caused by evacution of terminal or re-boarding of Aircraft becuse of security breach.

Since the of dataset is very large, real-time visualization is not scalable. If we try to compute a summary of the dataset and visualize it we will either need non-trivial parallel rendering algorithms or significant time to produce a drawing.

For analyzing such big dataset MapReduce programming modal has been proposed to mine the relevant information from dataset and perform statisitical modelling. For e.g. To plot the air traffic pattern originating at NewYork, it is scalable to perform mining operation on the data and extract the flight details orignated at New York. We can than use this data to build a graphical models on Cancellation Rate, Delay Rate monthly as well as weekly. This modal can be plotted graphically to understand the anomally present in 20 decades New York air traffic history.

The proposed work in thesis builds visulization modal described in four sections. First section mines the data from 120MM airline traffic using Hadoop MapReduce and R. Details of the package is provided above in Introduction section. Section 2 brings the modal of traffic pattern for top 20 busiest air traffic destinations. Chicago the city with highest airline traffic is palced at the top succeeding Atlanta and Dallas Fort-Worth. Section 3 provides the detail of overall airline traffic orignated at Chicago. This section also visualizes the pattern for flight cancellation and delay in Chicago in respect with its two airports O'Hare International and Chicago Midway. Section 4 modals the similar graph as of Chicago for the airport in New York: LaGuardia and John F Kennedy.

3.1 Mining Data Using Hadoop and Rhipe

Hadoop MapReduce programming modal is built in Java but hadoop provides utility to allow MapReduce programs to run from most of the famous languages. Hadoop streaming is a utility that comes with the Hadoop distribution. The utility allows you to create and run MapReduce jobs with any executable or script as the mapper and/or the reducer in any programming language. Both the mapper and the reducer are executables that read the input from stdin (line by line) and emit the output to stdout. The utility will create a Map/Reduce job, submit the job to an appropriate cluster, and monitor the progress of the job until it completes. resently the supported languages are Python, C, R, Scala,Ruby.

Many pacakages in R are present which allows to build mapreduce programs but due to good doucmentation of Rhipe, it has been used by author to develop mapreduce code to mine data. RHIPE is the R and Hadoop Integrated Programming Environment. RHIPE is a merger of R and Hadoop. R is the widely used, highly acclaimed interactive language and environment for data analysis.Hadoop consists of the Hadoop Distributed File System (HDFS) and the MapReduce distributed compute engine. RHIPE allows an analyst to carry out D&R analysis of complex big data wholly from within R. RHIPE communicates with Hadoop to carry out the big, parallel computations.

For building mapreduce application location data for each row is taken as key value and total number of key count is obtained for ¡key:value¿ pair. The value determines the popularity of the location similar to the word count example. Similarly cancellation and delay rate is calculated for each flight orignated from Chicago and New York.

3.2 Evaluting Air-Traffic Patterns of USA Flights

What are the busiest cities by total flight traffic. John F Keneddy will feature, but what are the others? For each airport code compute the number of inbound, outbound and all flights. The figure Log of time to complete vs log of total counts plots the traffic pattern.

```
map <– expression({
#For each input record, parse out required fields and output new record:
extractTop20 = function(line) {
  fields <– unlist(strsplit(line, ”\\,”))
  #Skip header lines and bad records:
  if (!(identical(fields[[1]], ”Year”)) \& length(fields) == 29) {
    origin <– fields[[17]]
    destination <– fields[[18]]
    #Skip records where departure dalay is ”NA”:
    if (!(identical(origin, ”NA”))) {
      #field[9] is carrier, field[1] is year, field[2] is month:
      rhcollect(paste(fields[[1]], ”\t”, origin, sep=””), 1)
    }
    if (!(identical(destination, ”NA”))) {
      # field[9] is carrier, field[1] is year, field[2] is month:
     rhcollect(paste(fields[[1]], ”\t”, destination, sep=””),1)
    }
  }
  }
  #Process each record in map input:
  lapply(map.values, extractTop20))

reduce <- expression(
  #intialization of variable pre
  pre = {
    count <- numeric(0)
  },reduce = {
    # Depending on size of input, reduce will get called multiple times
    # for each key, so accumulate intermediate values in delays vector:
    count <- c(count, as.numeric(reduce.values))
  },
  post = {
    # Process all the intermediate values for key:
   tot <- sum(count)
   rhcollect(reduce.key, tot)
  })
```

MapReduce code showed above collects the data in form of key, value pair. Generated results are than employed for further processing and calculating top busiest cities in respect to Airline traffic.

3.3 Evaluating Chicago Air-Traffic Pattern

The busiest airport displayed by Figure 4.1 is Chicago. Chicago O'Hare international is one of the busiest airport on Earth. Chicago traffic data are collected with Hadoop MapReduce technique. Time required for MapReduce operations were approximately 5hrs. The generated result were than used for performing cancellation rate, delay rate weekly as well as monthly.

3.4 Evaluating NewYork City Air-Traffic Pattern

NYC, though placed at 6th as most busiest airport but is one of the most populated cities in United States. Because of the advent of 9/11 in WTC, New York has been to observe the anomally in New York Air traffic.

Chapter *4*

Results

This section discusses the analysis performed on Airline data which was preseted at Data expo 2008. Analysis of data is performed in R and ggplot library is used for presenting the graphical charts.

4.1 Evaluting Traffic Patterns of top Metropolitan Cities

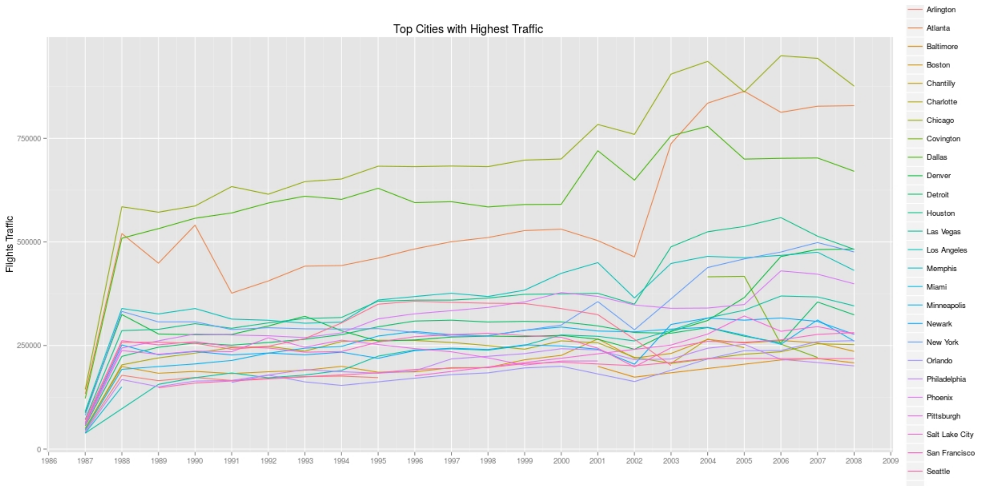

Figure 4.1

We investigated and visualize data for domestic flights that originated or terminated at all cities. We can look at the change in flight traffic patterns because of the increasing populations. As of April 26, 2015, the United States has a total resident population of 320,760,000, making it the third most populous country in the world.[1] It is very urbanized, with 81% residing in cities and suburbs as of 2014 (the worldwide urban rate is 54%).California and Texas are the most populous states,as the mean center of U.S. population has consistently shifted westward and southward.New York City is the most populous city in the United States.

Figure 4.1 displays this three-way comparison of traffic patterns. We have noticed that flight traffic increases with time line but there is a setback in traffic in the year before 2002. Reason for this anomally can be co-realted with 9/11/2001 terrorist attack on WTC. This inturn would have lead to increase in security, more flight regulations and thus heavy decrease in traffic.

4.2 Chicago Traffic Pattern

Chicago, on Lake Michigan in Illinois, is among the largest cities in the U.S. Famed for its bold architecture. The city is also renowned for its museums, including the Art Institute and its expansive collections, including noted Impressionist works.

4.2.1 Total Traffic

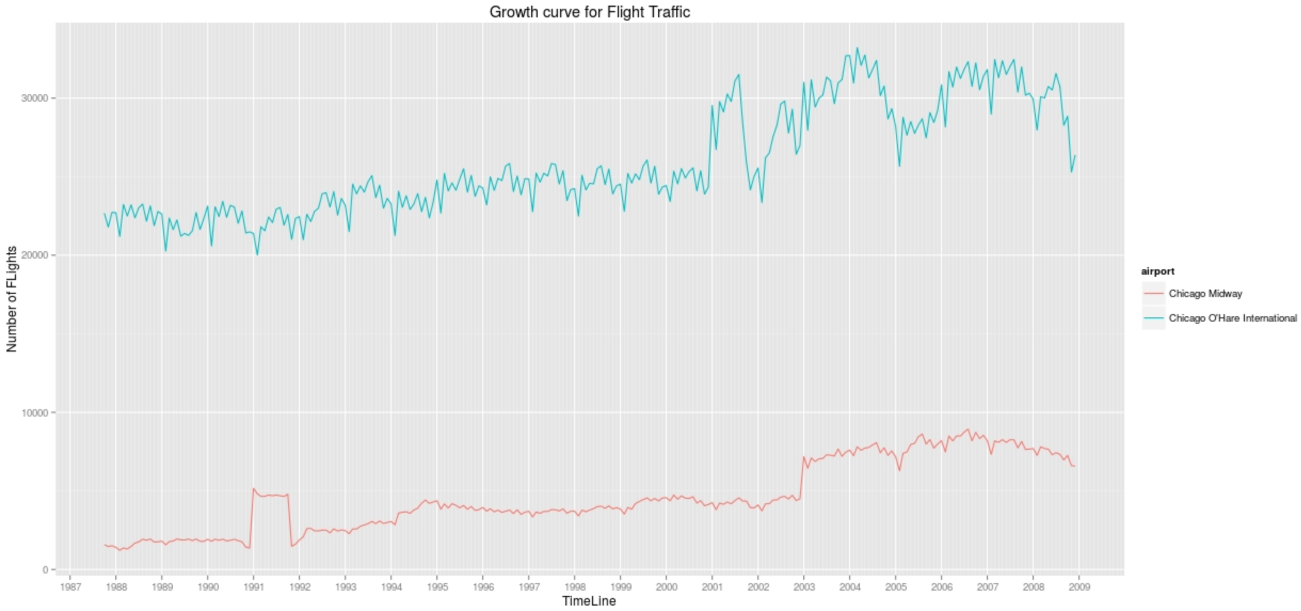

Figure 4.2

Figure 4.2 points the diffrence between two airports of Chicago i.e. Chicago O'Hare and Chicago Midway. Chicago Midway is least used by people of Chicago though O' Hare is one of the busiest airport. We can clearly observer the Chicago Midway traffic growth started from year 2003 when an investment was made to improve the operations on Chicago Midway. Anomally due to 9/11 is less visible in Midway than in O'Hare.

We can also observe a sharp dip in flight traffic in the month of December and January, this may be because of the Christmas Holidays and New Year eve where most of the pilots would be on leave or because of highly cold weather.

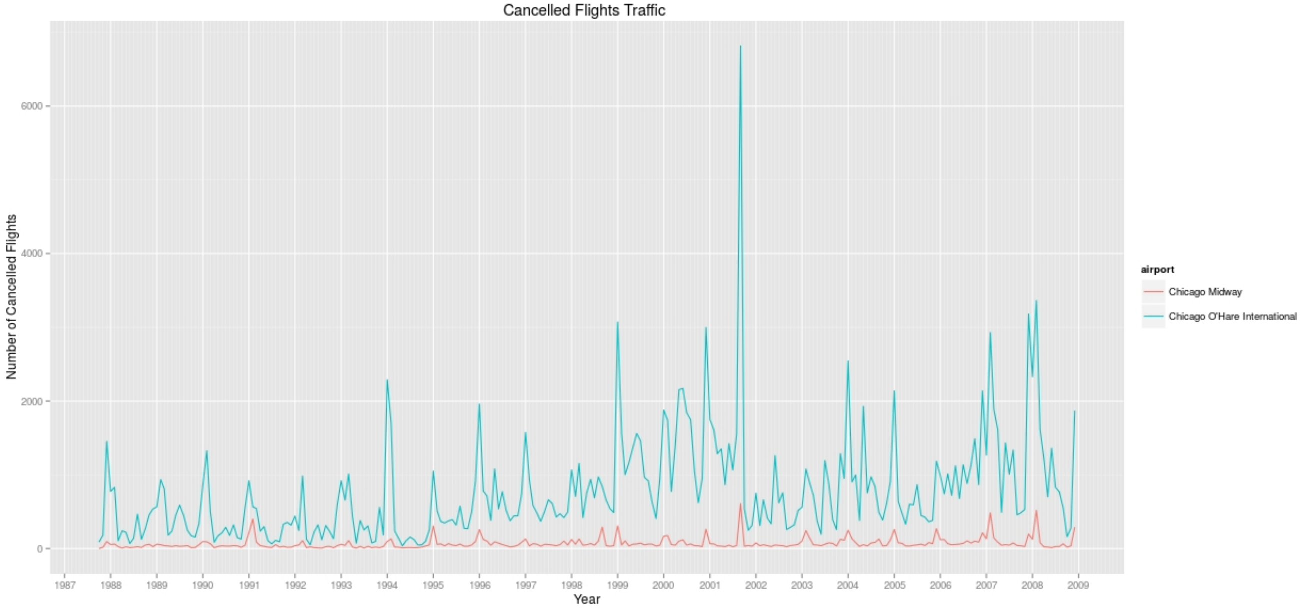

Figure 4.3

4.2.2 Canceled Flights

Figure 4.3 we can observe the cancellation rate is at peak in the month of December and January. While Figure 4.2 showed a drop in the same time line.

4.2.3 Delay Rate

The graph shown in Figure 4.4 delay rate pattern for the city Chicago and the diffrence between Midway and O' Hare.

4.3 New York Traffic Pattern

Home to the Empire State Building, Times Square, Statue of Liberty and other iconic sites, New York City is a fast-paced, globally influential center of art, culture, fashion and finance. The analysis is made bewteen LaGaurdia(LGA) and John F Kennedy(JFK). LGA is a much easier airport to navigate. Unless you're set on taking public transportation to/from the airport its the way to go. LGA is smaller, and thus can be a lot quicker to exit through. It's also much closer to the city than JFK, but this is only useful if you'll be taking a cab. Public transportation from LGA is as good as non-existent.

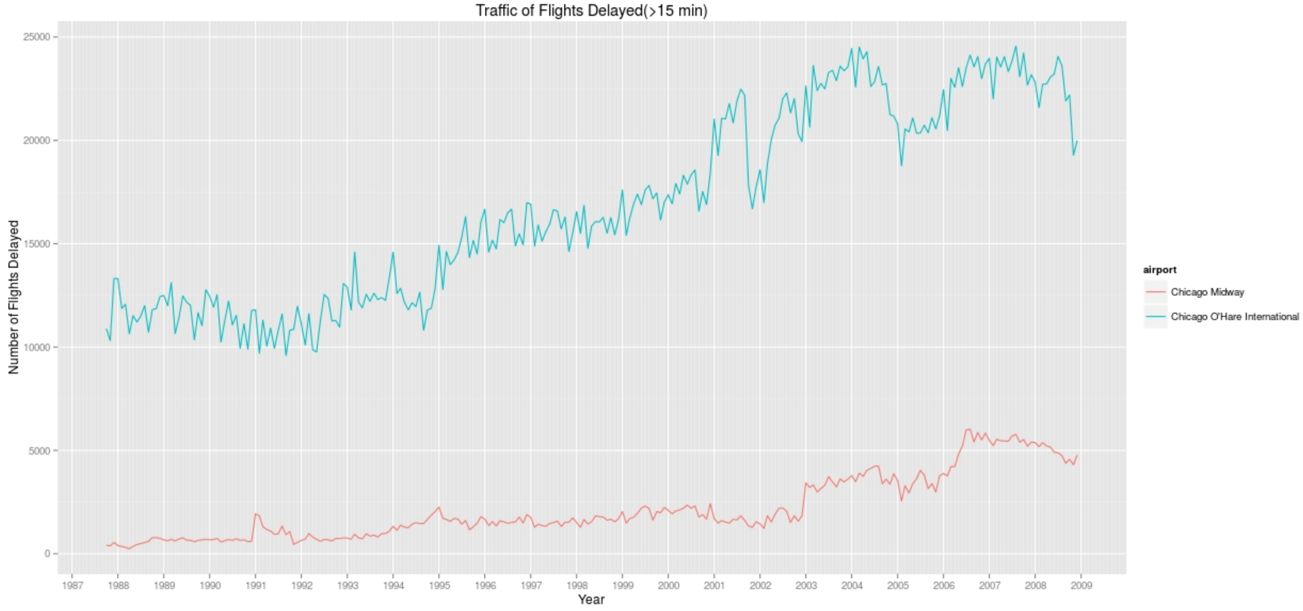

Figure 4.4

4.3.1 Total Traffic

In earlier years traffic of LaGuardia was higher than JFK but after advent 2007 JFK has shown higher traffic growth rate. We can also observe lower slope in 2002 because of 9/11. Figure 4.5 shows steep slope in the start of January 2001 was beacuse of 100th Yankees soccer league.

4.3.2 Cancelled Flights

Figure 4.6 is shows the cacelled flight status. Similar conclusion can be drawn from this graph as of Figure 4.3. High peaks in month of January and 9/11 can be easily observed.

4.3.3 Delayed Flights

Figure 4.7 is imilar to the oberservations made at Chicago. Again similar conclusion can be drawn signifying a general trend in American populations.

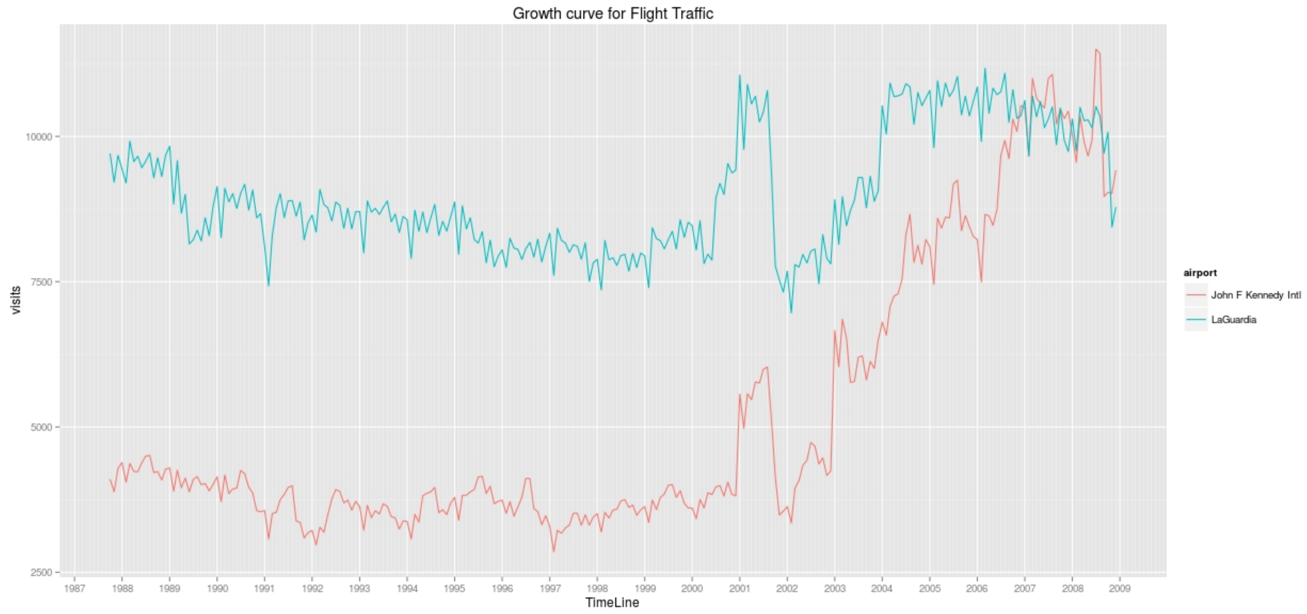

Figure 4.5

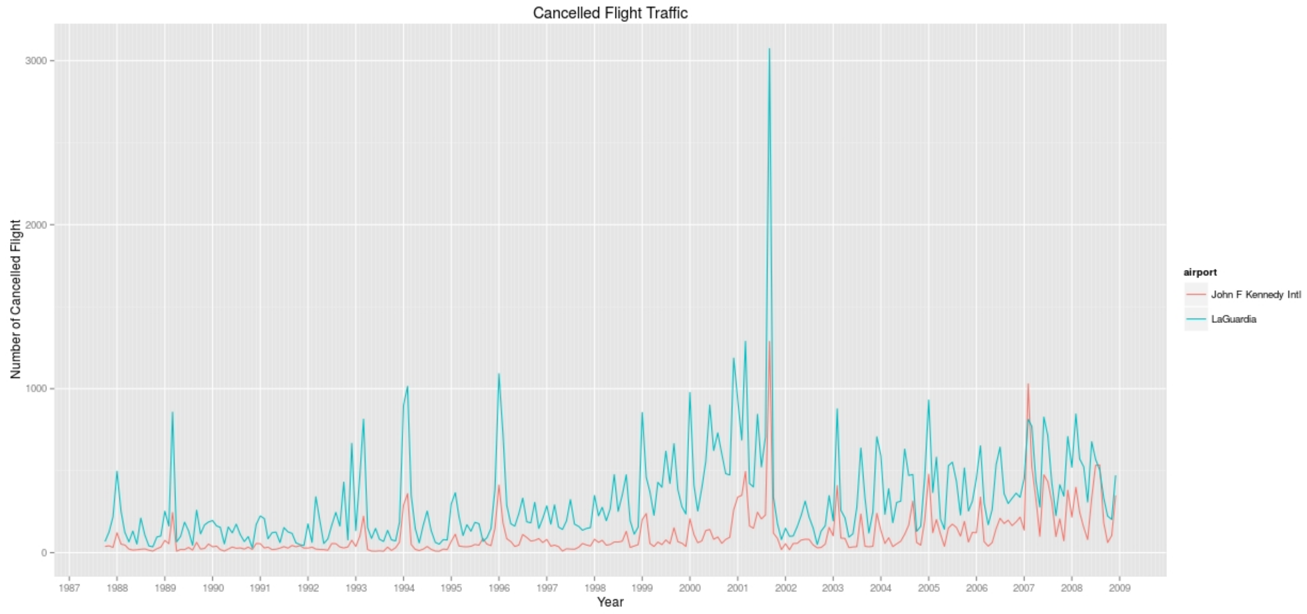

Figure 4.6

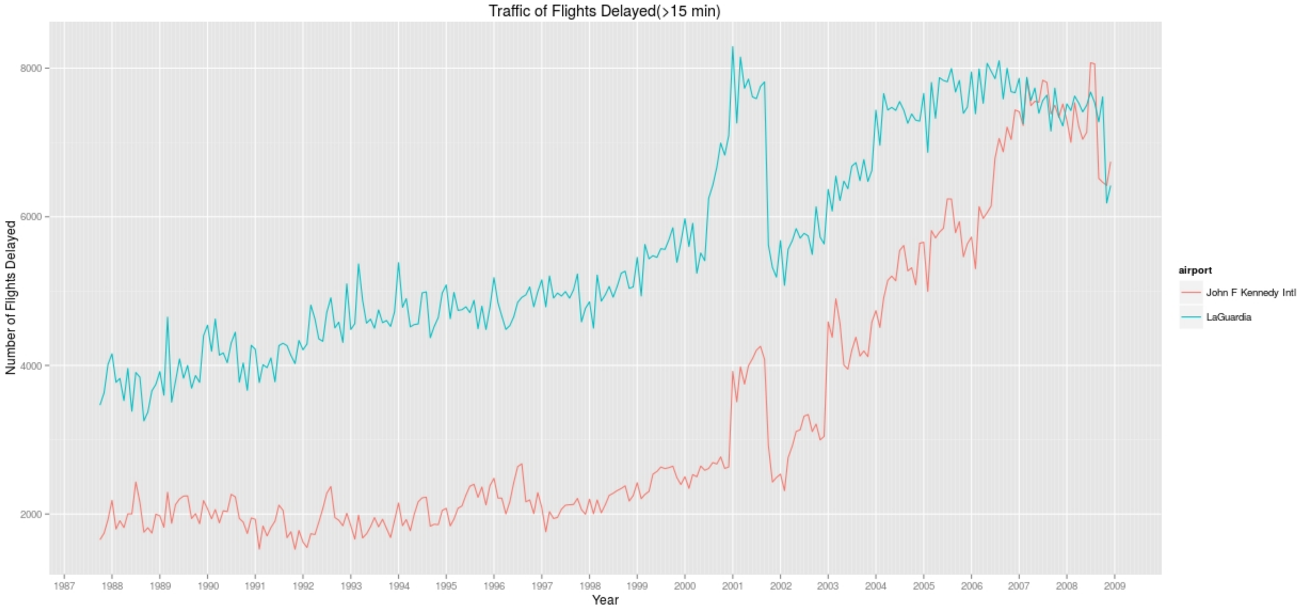
Traffic of Flights Delayed(>15 min)
Number of Flights Delayed
8000
6000
4000
2000
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
Year
airport
John F Kennedy Intl
LaGuardia

Figure 4.7

Chapter *5*

Conclusions and Future Work

Changes in the patterns of air traffic at New York and Chicago over 21 years (from 1987 to 2008) are explored. Most notably rise and fall of curve at 9/11 in delay and total traffic respectively. Befor 1996, the Chicago was completely gone, but a slow recovery started to take place. The levels of air traffic were more or less stable over the five year period ending in 2008. The population growth of the surrounding area only seems to correlate with air traffic levels in the post-hub time period. Also, as more airlines served the airport, more destinations were added. However, delays for both arrivals and departures increased as well.

5.1 Scope of further work

There are many possible future investigations that could be pursued. One would be to compare trends with national trends. The analysis performed here could even be replicated for other airports around the country. It would also be interesting to get passenger data from the airline traffic that could help shed more light on the validity of the speculative explanations for the trends presented their. With regards to the delay data, adding weather data could be very interesting in helping explain some of the variation in delays. There are also more data than delay and cancel rate in the flight data provided by the Data Expo 2009 competition that could be further explored, especially in the on-time performance area. What airline, time of day, day of the week, month of the year should you choose to travel on to minimize the chance of being delayed at USA Air Traffic?