HERITAGE INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

B. Tech FINAL YEAR PROJECT

to obtain the title of

Bachelor In Technology

Specialty : Computer Science

Guided by

Prof. Sandip Samaddar

Web Integrated Transport Support System

Prince Kumar Rohit Kumar Subhodeep Sahoo Ravi Raj

Acknowledgments

We thank Dr. Pranay Chaudhuri, Principle of Heritage Institute of Technology for allowing us to do this project and providing us with all the facilities we needed to complete our project.

We are grateful to Prof. Sandip Samaddar, our project advisor for his patient guidance, encouragement and advice he has provided throughout our time as his students. We have been extremely lucky to have a supervisor who cared so much about our work, and who responded to our questions and queries so promptly. As our teacher and mentor, he has taught us more than we could ever give him credit for. We also owe gratitude to the Head of Department Mr. Subhashis Majumder for allowing us to undertake this project.

We thank our teachers and laboratory assistants at the Heritage Institute of Technology for their continuous inspiration. Last but not the least we thank all friends for their cooperation and encouragement that they have bestowed on us.

Prince Kumar

Rohit Kumar

Subhodeep Sahoo

Ravi Raj

Bonafide Certificate

Certified that the project report titled **"Web Integrated Transport Support System"** is the bonafide work of Prince Kumar, B.Tech. 4th Year, Roll Number - 12615001095, Rohit Kumar, B.Tech. 4th Year, Roll Number – 12615001115,Subhodeep Sahoo B.Tech. 4th Year, Roll Number - 12615001161,Ravi Raj, B.Tech, 4th year, Roll Number - 12615001107 who carried out the project under my supervision.

Sandip Samaddar
Assistant Professor
Computer Science and Engineering
Heritage Institute of Technology

Dr. Subhashis Majumder
Professor and HoD
Computer Science and Engineering
Heritage Institute of Technology

Examiner Signature

Contents

1. Abstract 5

2. Introduction 6

2.1 Problem Definition 6

3. Design and Architecture 7

3.1 Block Diagram and Schema 7

3.2 Web Scraping 9

3.2.1 Data Collection 9

3.2.2 Beautiful Soup 10

3.2.3 Selenium Web Driver 10

3.3 Hadoop 10

3.3.1 Hadoop Distributed File System 10

3.4 Apache Spark 11

3.4.1 Spark SQL 11

3.5 Automation 12

3.5.1 Crontab 12

3.5.2 Secure copy protocol 13

3.6 Full Stack Development 14

3.6.1 Front End 14

3.6.2 Back End (Django) 15

4. Operational Results 16

5. Conclusion 22

6. References 23

1. Abstract

In this age of digitalization, people depend heavily upon online ticket reservation and websites for tours and travels .Web Integrated Transport Support System is one such platform where users can plan their travel. Our main objective is to design a web portal that will use web crawled data kept in the hadoop environment to display various modes of transport between a source and destination . We basically focus on generating results for buses, trains and flights at the same time in a single page . We begin by implementing web scraping to form the dataset which is moved to Hadoop distributed file system , performing query operations with the assistance of Apache Spark . Thereupon, we execute a Django script which brings required data on the screen of user . We have implemented these using Beautiful Soup(a python package) to web scrap the data , Apache Spark to filter the data , Hadoop multi-node environment to keep the dataset ,Selenium web driver to automate the process and Django framework with Bootstrap for full stack development.

2. Introduction

The growth of the Internet has led to an increased dependency of customers upon online ticket reservation for their decision-making choices. Whether it is for viewing availabilities, bookings - we all turn to the Internet for opinions, choices, viewpoints and variety of alternatives". Millions of people globally use online ticket reservation systems every day to decide which is the “Best possible route” at the “Best possible price”. As such, we aim to understand the requirements of customers - whether they are willing to save fare? Whether they are willing to save time? Does it give us any insight into user behaviour? All these questions have been glanced at and worked upon and we present our findings herewith.

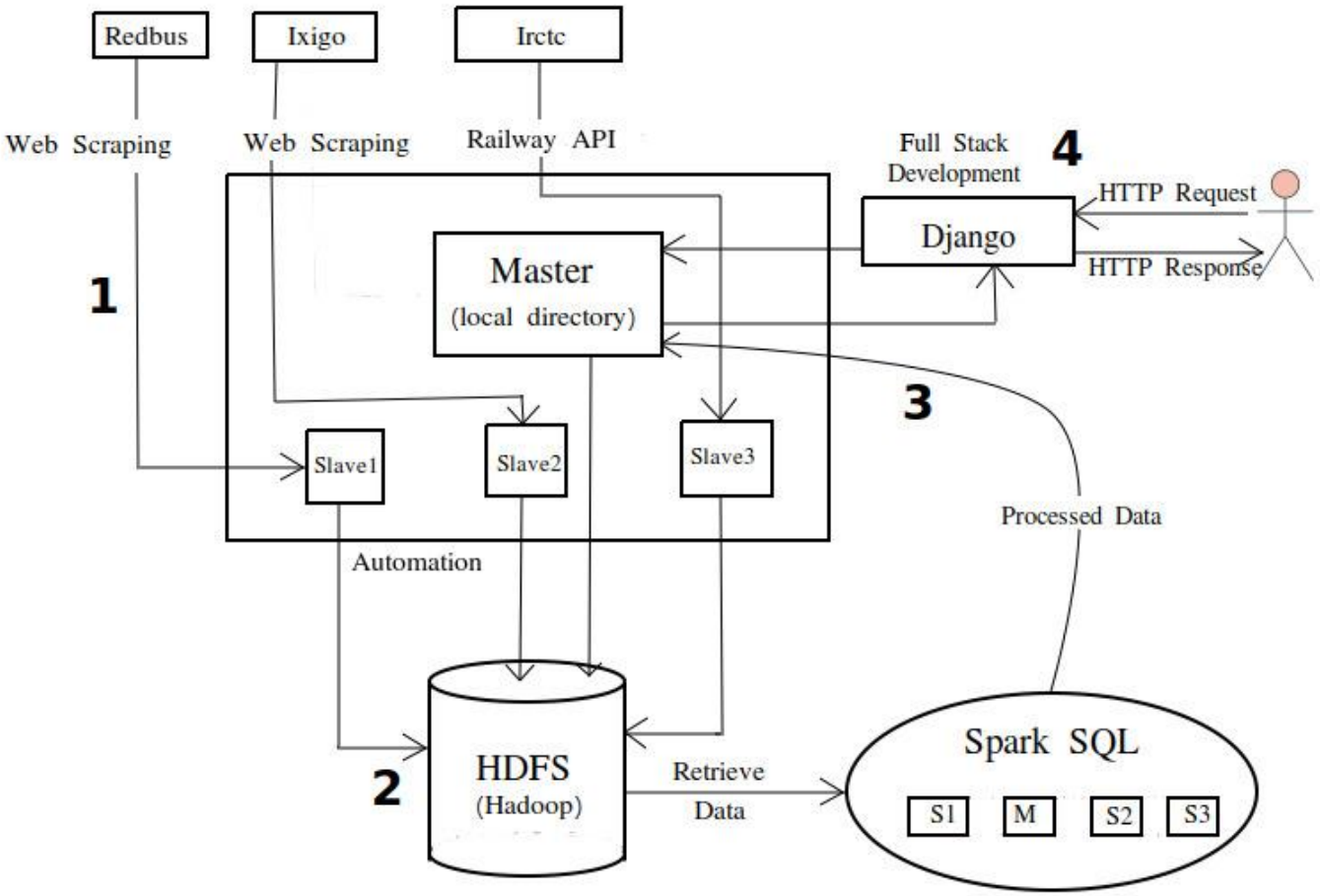
2.1 Problem Definition

Web Integrated Transport Support System is being challenged at every moment to cope up with real time synchronization as and when tickets are booked or cancelled ,it has to be updated as soon as possible on Internet. Every search has a priority to display results with respect to increasing fare or time ,at times when the locations are remote , it must ensure that best modes are displayed at the top in terms of services and punctuality. It is itself a great task to show all the modes (flights,trains and buses) at a single click ,there upon ,keep a constant check on the authenticity of displayed data as “Stale data has no value ”.The control of enormous amount of data is not advisable under the realms of “Relational Database Management System”, so the best way to tackle it with Hadoop multi-node cluster environment.

3. Design and Architecture

Data flow architecture is implemented when input data has to be transformed into output data through a series of computational manipulative components . Blocks are known as filters and the arrow notations are pipes. Pipes are used to transmit data from one component to the next.

3.1 Block Diagram and Schema



The working flow of the above diagram is explained as follows.

Our whole project revolves around crawling which is asynchronous in nature. Hence it is impossible to show the results in runtime. If user will enquire something, he will have to wait for significant amount of time to get the results,we are eliminating this issue and providing a hassle free experience . So we have pre-crawled it and stored the data such that when user enquire it feels like he is getting results at runtime with no delay.

We are using hadoop architecture in which one is master and other three are slaves as shown in the diagram. Apache Spark is installed on top of this architecture.

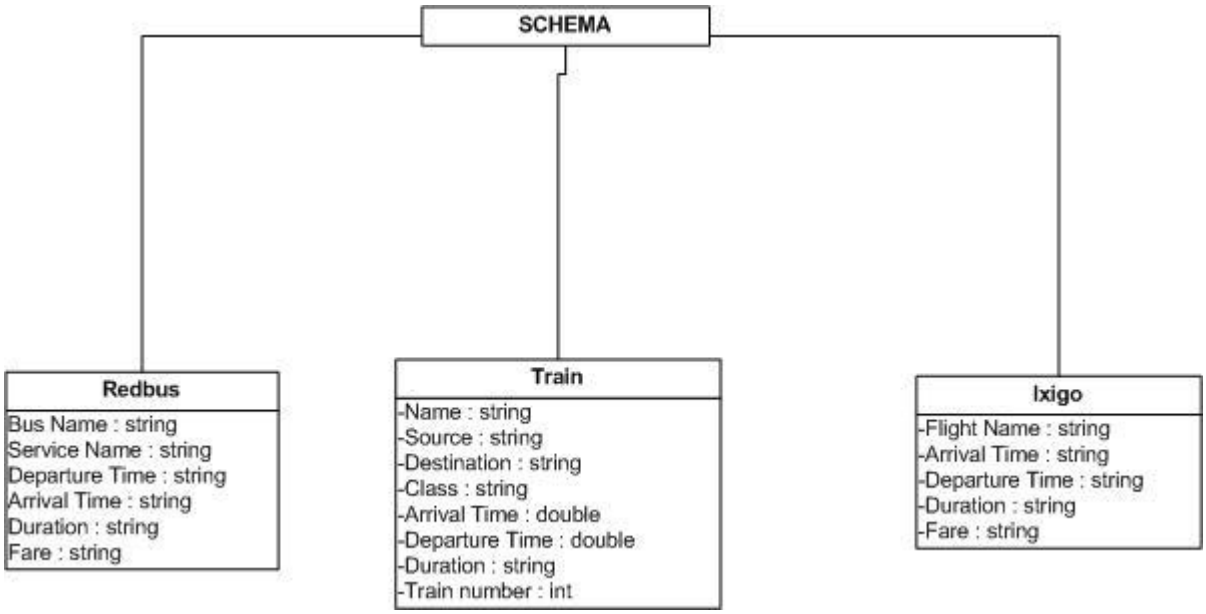
Slave 1's job is to scrap the data of redbus.in and store the raw data in the HDFS. Similarly Slave 2's job will be to scrap ixigo.com. When the data is stored in HDFS , spark-submit query will be executed just after it by each slave who will filter the raw data and sort it according to fare ,thereby transferring the results to the master using scp command. We are storing the data in flat file format of .csv . Now the problem of uniqueness arrive which is solved by renaming the file with its source, destination and date in particular folders residing in the local directory of master.

For trains we have the railway api which is integrated in the Django and will work in runtime. Some time it fails then the fault tolerance is also applied by showing the previously searched recent data.

When the user will enter the source , destination and date. Our program will use these and will show proper result.

Each of the components of block diagram is explained further.

Schema



3.2 Web scraping

It is a technique employed to extract large amounts of data from websites whereby the data is extracted and saved to a local file in your computer or to a database in table (spreadsheet) format. Data displayed by most websites can only be viewed using a web browser. They do not offer the functionality to save a copy of this data for personal use. The only option then is to manually copy and paste the data - a very tedious job which can take many hours or sometimes days to complete. Web Scraping is the technique of automating this process, so that instead of manually copying the data from websites, the Web Scraping software will perform the same task within a fraction of the time.

We are doing web scraping by the help of python library Beautiful Soup and Selenium web driver to run javascript and automate browsing as explained further.

3.2.1 Data Collection

Our study relies on a large-scale crawl from Redbus website, Ixigo.com and Railway API. The data set encompasses Transport name , Arrival time, Duration, Departure time , fare and availability with respect to source and destination . The data we have crawled is as follows -

1. All buses between a source and destination on particular date with the following attributes - Bus name, Arrival Time, Departure time,Duration and fare.

2. All flights between a source and destination on particular date with the following attributes - Flight name, Arrival Time, Departure Time,Duration and fare.

3. Railway API.

https://api.railwayapi.com/v2/between/source/<stn code>/dest/<stn code>/date/<dd-mm-yyyy>/apikey/<apikey>/

3.2.2 Beautiful Soup

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work

3.2.3 Selenium Web Driver

Selenium automates browsers. That's it! What you do with that power is entirely up to you. Primarily, it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can be automated as well.Selenium has the support of some of the largest browser vendors who have taken steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

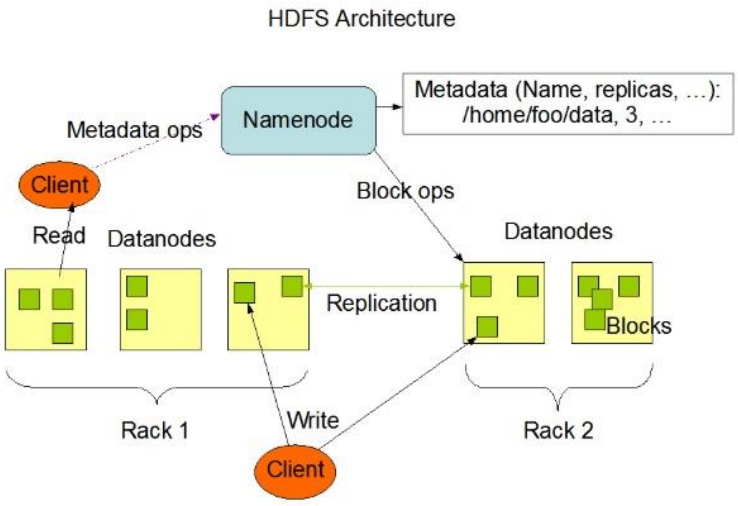
3.3 Hadoop

Apache Hadoop is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.
Why are we using Hadoop ?

1. Hadoop is Easily Scalable
2. Hadoop is Fault Tolerant
3. Hadoop Ecosystem is Robust
4. Hadoop is Very Cost Effective

3.3.1 Hadoop distributed file system

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant.



3.4 Apache Spark

Apache Spark is an open-source distributed general-purpose cluster-computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, which has maintained it since.

Why are we using Spark ?

1. Swift Processing
2. Dynamic in Nature
3. In-Memory Computation in Spark
4. Reusability
5. Fault Tolerance in Spark
6. Real-Time Stream Processing
7. Support Multiple Languages
8. Support for Sophisticated Analysis
9. Integrated with Hadoop

3.4.1 Spark SQL

It is a component on top of Spark Core that introduces a new data abstraction called SchemaRDD, which provides support for structured and semi-structured data.

Resilient Distributed Datasets (RDD) is a fundamental data structure of Spark. It is an immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.

3.5 Automation

Automation is the technology by which a process or procedure is performed with minimal human assistance.

3.5.1 Crontab

Schedule a command to run at a later time.

Syntax

crontab [-u user] file
crontab [-u user] { -l | -r | -e }

DESCRIPTION

Crontab is the program used to install, deinstall or list the tables used to drive the cron daemon in Vixie Cron.
Each user can have their own crontab, and though these are files in /var, they are not intended to be edited directly.
If the -u option is given, it specifies the name of the user whose crontab is to be tweaked.
If this option is not given, crontab examines "your" crontab, i.e., the crontab of the person executing the command.

Cron Table Format
* * * * * Command_to_execute
- ? ? ? -
| | | | |
| | | | +?? Day of week (0?6) (Sunday=0) or Sun, Mon, Tue,...
| | | +???- Month (1?12) or Jan, Feb,...
| | +????-? Day of month (1?31)
| +??????? Hour (0?23)

+????????- Minute (0?59)

MIN HOUR DOM MON DOW CMD

Field	Description	Allowed Value
MIN	Minute field	0 to 59
HOUR	Hour field	0 to 23
DOM	Day of Month	1-31
MON	Month field	1-12
DOW	Day Of Week	0-6
CMD	Command	Any command to be executed.

Example: To schedule a background Cron job for every 10 minutes.
*/10 * * * * /home/maverick/check-disk-space

3.5.2 Secure Copy Protocol(SCP)

It is a command line utility that allows you to securely copy files and directories between two locations.

With scp, you can copy a file or directory:

From your local system to a remote system.
From a remote system to your local system.
Between two remote systems from your local system.

scp [OPTION] [user@]SRC_HOST:]file1 [user@]DEST_HOST:]file2
Copy
OPTION - scp options such as cipher, ssh configuration, ssh port, limit, recursive copy ..etc
[user@]SRC_HOST:]file1 - Source file.
[user@]DEST_HOST:]file2 - Destination file
Local files should be specified using an absolute or relative path while remote file names should include a user and host specification.
scp provides a number of options that control every aspect of its behavior. The most widely used options are:

-P Specifies the remote host ssh port.
-p Preserves files modification and access times.
-q Use this option if you want to suppress the progress meter and non-error messages.

-C. This option will force scp to compresses the data as it is sent to the destination machine.
-r This option will tell scp to recursively copy directories.

Copy a Local File to a Remote System with the scp Command
To copy a file from a local to a remote system run the following command:

scp file.txt remote_username@10.10.0.2:/remote/directory

You will be prompted to enter the user password and the transfer process will start.

You may also want to set up an SSH key-based authentication and connect to your Linux servers without entering a password.

We are using crontab to automate the shell script commands which in turn execute the several important commands
1. Loading the bash environment
2. Web Crawling using python3
3. Hadoop commands
4. Spark-submit
5. some os commands related to file management
6. SCP for giving data to the the serve

3.6 Full Stack Development

It refers to the development of both front end and back end portions of an application. This web development process involves - Presentation layer(front end part that deals with the user interface), Business Logic Layer(back end part that deals with data validation). It takes care of all the steps from the conception of an idea to the actual finished product.

3.6.1 Front End

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.
The objective of designing a site is to ensure that when the users open up the site they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site.

3.6.2 Back End (Django)

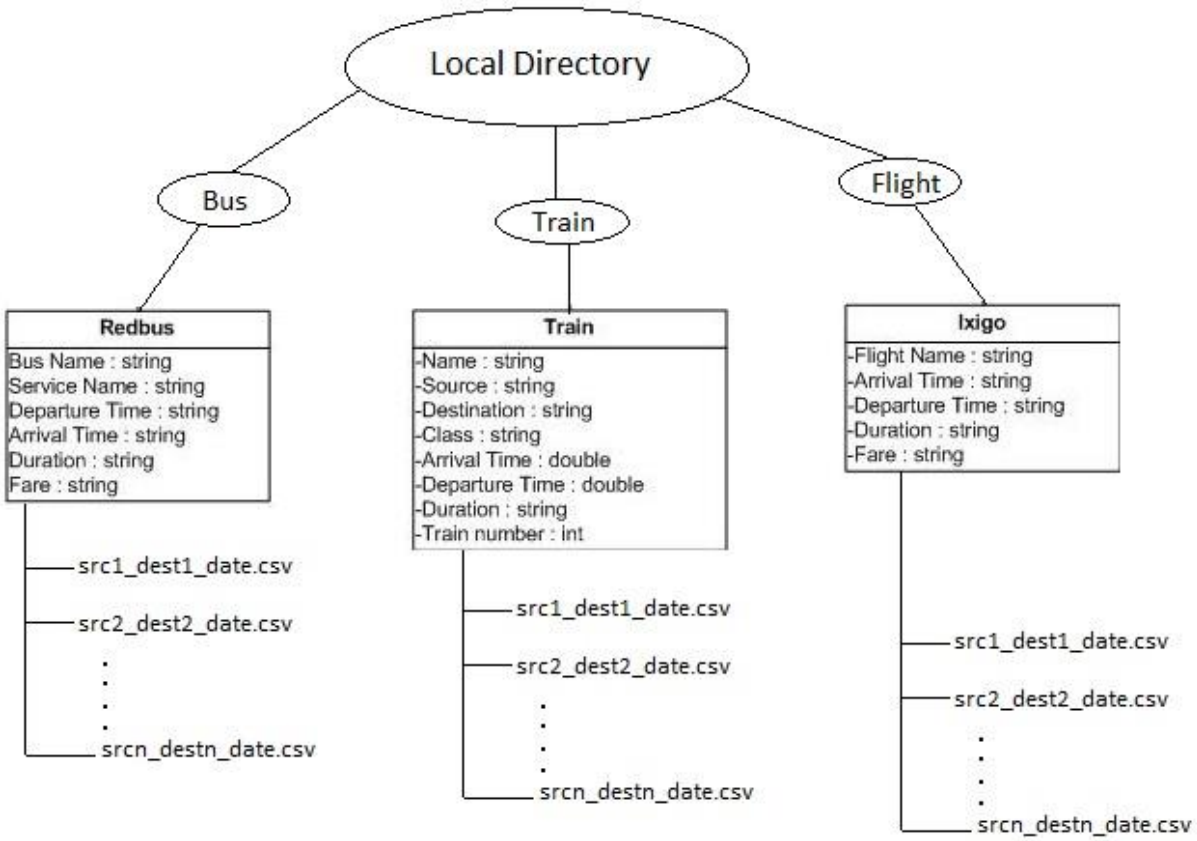
Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel.

Why do we use Django ?

When we are building a website, we always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc.
*
Luckily for us , other people long ago noticed that web developers face similar problems when building a new site, so they teamed up and created frameworks (Django being one of them) that give we ready-made components to use.

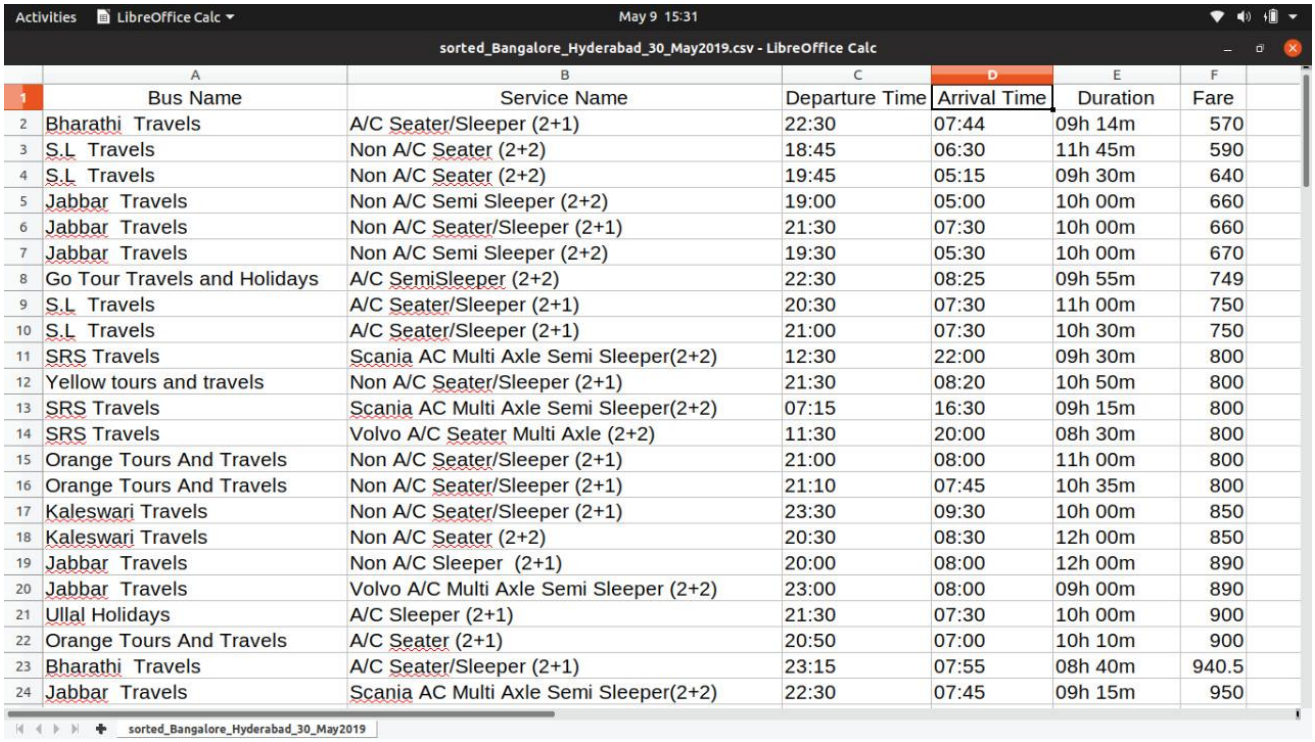
4. Operational Results

4.1 File Structure



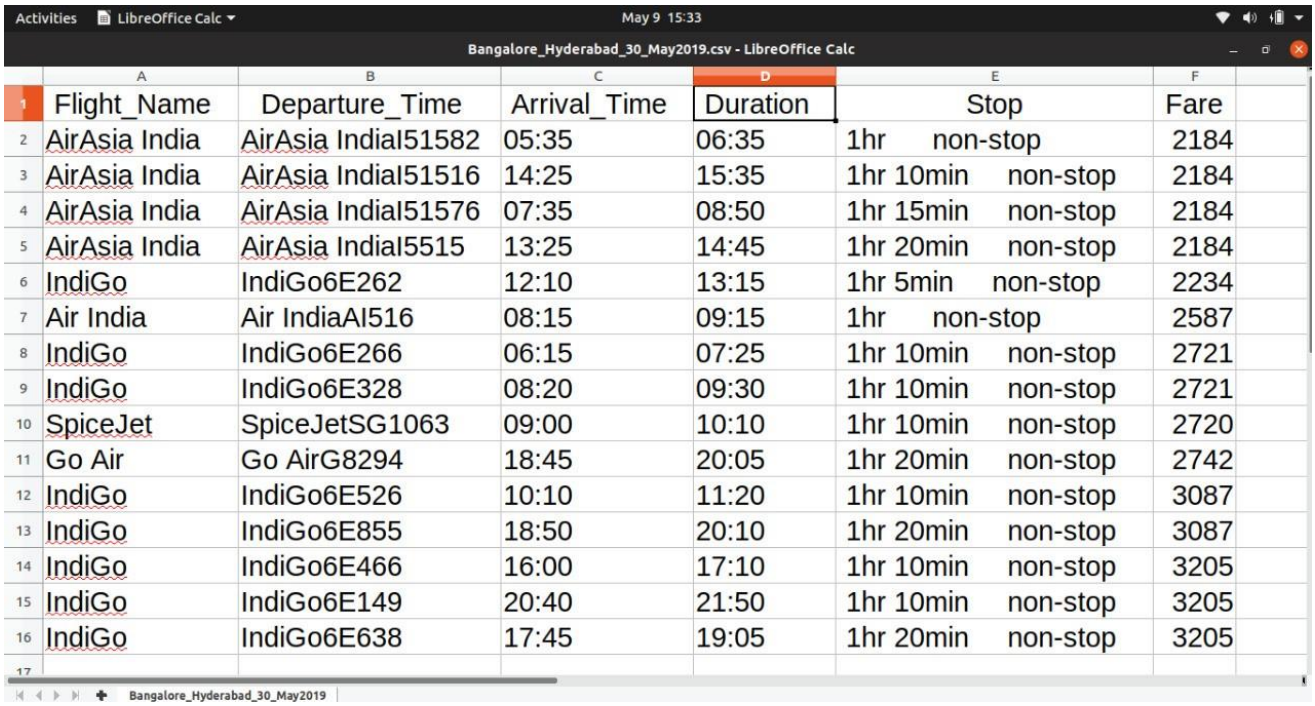
4.3 Illustrating schema of various modes of transport with the help of dataset

4.3.1 Redbus



	A	B	C	D	E	F
1	Bus Name	Service Name	Departure Time	Arrival Time	Duration	Fare
2	Bharathi Travels	A/C Seater/Sleeper (2+1)	22:30	07:44	09h 14m	570
3	S.L Travels	Non A/C Seater (2+2)	18:45	06:30	11h 45m	590
4	S.L Travels	Non A/C Seater (2+2)	19:45	05:15	09h 30m	640
5	Jabbar Travels	Non A/C Semi Sleeper (2+2)	19:00	05:00	10h 00m	660
6	Jabbar Travels	Non A/C Seater/Sleeper (2+1)	21:30	07:30	10h 00m	660
7	Jabbar Travels	Non A/C Semi Sleeper (2+2)	19:30	05:30	10h 00m	670
8	Go Tour Travels and Holidays	A/C SemiSleeper (2+2)	22:30	08:25	09h 55m	749
9	S.L Travels	A/C Seater/Sleeper (2+1)	20:30	07:30	11h 00m	750
10	S.L Travels	A/C Seater/Sleeper (2+1)	21:00	07:30	10h 30m	750
11	SRS Travels	Scania AC Multi Axle Semi Sleeper(2+2)	12:30	22:00	09h 30m	800
12	Yellow tours and travels	Non A/C Seater/Sleeper (2+1)	21:30	08:20	10h 50m	800
13	SRS Travels	Scania AC Multi Axle Semi Sleeper(2+2)	07:15	16:30	09h 15m	800
14	SRS Travels	Volvo A/C Seater Multi Axle (2+2)	11:30	20:00	08h 30m	800
15	Orange Tours And Travels	Non A/C Seater/Sleeper (2+1)	21:00	08:00	11h 00m	800
16	Orange Tours And Travels	Non A/C Seater/Sleeper (2+1)	21:10	07:45	10h 35m	800
17	Kaleswari Travels	Non A/C Seater/Sleeper (2+1)	23:30	09:30	10h 00m	850
18	Kaleswari Travels	Non A/C Seater (2+2)	20:30	08:30	12h 00m	850
19	Jabbar Travels	Non A/C Sleeper (2+1)	20:00	08:00	12h 00m	890
20	Jabbar Travels	Volvo A/C Multi Axle Semi Sleeper (2+2)	23:00	08:00	09h 00m	890
21	Ullal Holidays	A/C Sleeper (2+1)	21:30	07:30	10h 00m	900
22	Orange Tours And Travels	A/C Seater (2+1)	20:50	07:00	10h 10m	900
23	Bharathi Travels	A/C Seater/Sleeper (2+1)	23:15	07:55	08h 40m	940.5
24	Jabbar Travels	Scania AC Multi Axle Semi Sleeper(2+2)	22:30	07:45	09h 15m	950

4.3.2 Flight

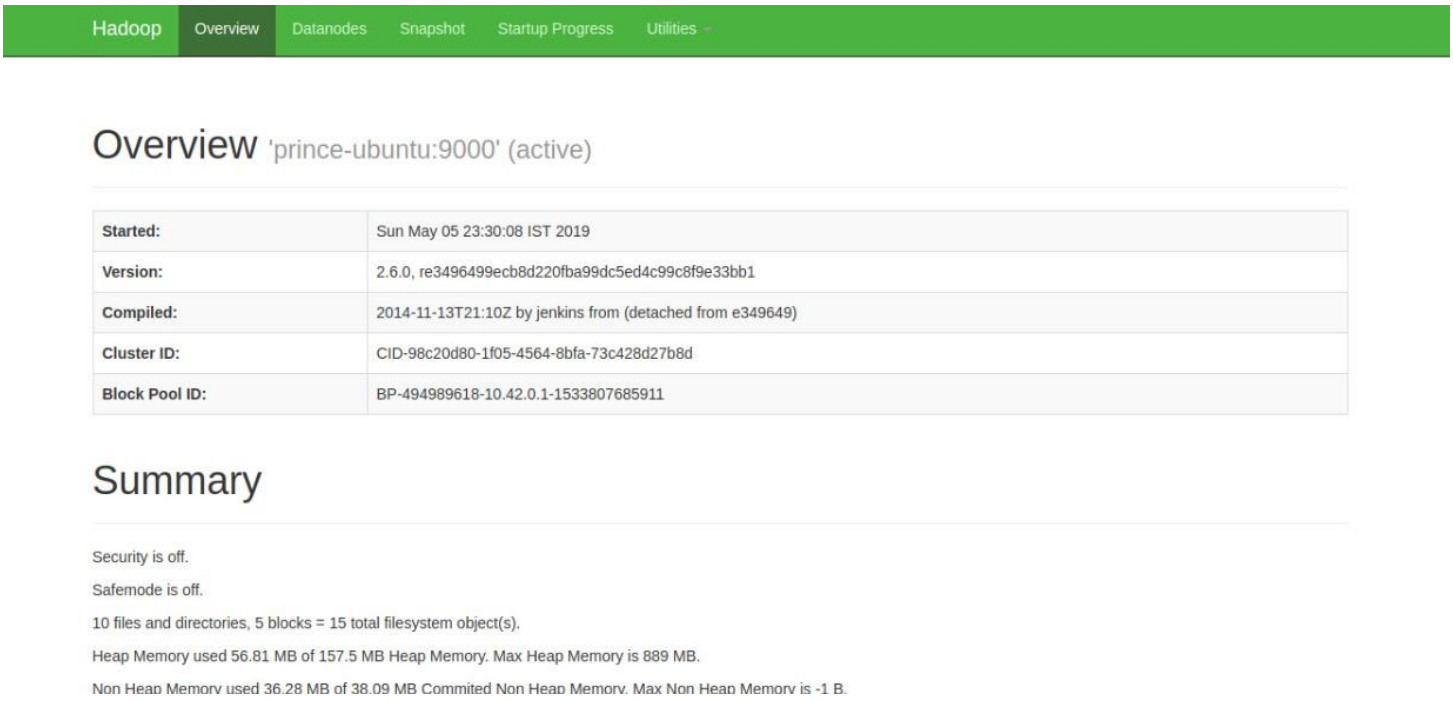


Flight_Name	Departure_Time	Arrival_Time	Duration	Stop	Fare
AirAsia India	AirAsia IndiaI51582	05:35	06:35	1hr non-stop	2184
AirAsia India	AirAsia IndiaI51516	14:25	15:35	1hr 10min non-stop	2184
AirAsia India	AirAsia IndiaI51576	07:35	08:50	1hr 15min non-stop	2184
AirAsia India	AirAsia IndiaI5515	13:25	14:45	1hr 20min non-stop	2184
IndiGo	IndiGo6E262	12:10	13:15	1hr 5min non-stop	2234
Air India	Air IndiaAI516	08:15	09:15	1hr non-stop	2587
IndiGo	IndiGo6E266	06:15	07:25	1hr 10min non-stop	2721
IndiGo	IndiGo6E328	08:20	09:30	1hr 10min non-stop	2721
SpiceJet	SpiceJetSG1063	09:00	10:10	1hr 10min non-stop	2720
Go Air	Go AirG8294	18:45	20:05	1hr 20min non-stop	2742
IndiGo	IndiGo6E526	10:10	11:20	1hr 10min non-stop	3087
IndiGo	IndiGo6E855	18:50	20:10	1hr 20min non-stop	3087
IndiGo	IndiGo6E466	16:00	17:10	1hr 10min non-stop	3205
IndiGo	IndiGo6E149	20:40	21:50	1hr 10min non-stop	3205
IndiGo	IndiGo6E638	17:45	19:05	1hr 20min non-stop	3205

4.3.3 Trains

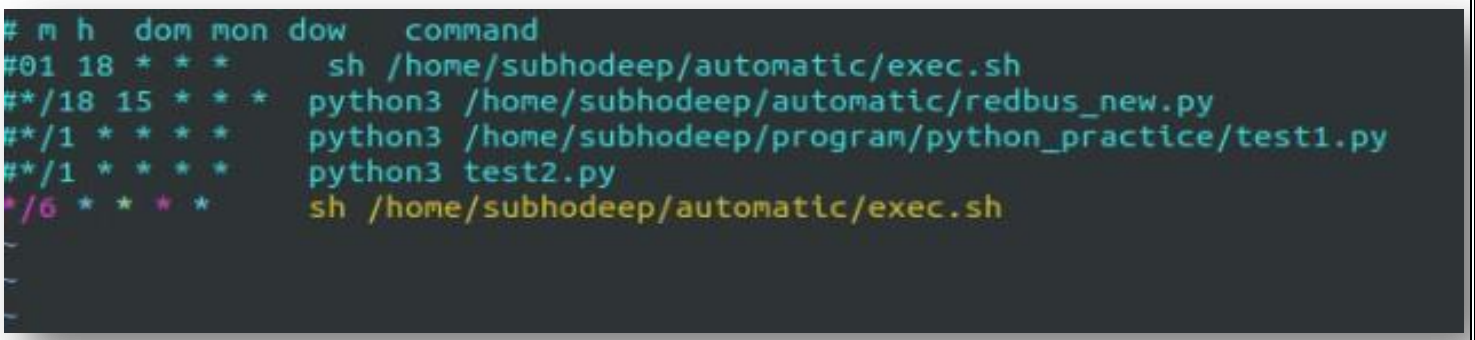

debit: 1
trains:
0:
days:
0:
runs: "Y"
code: "MON"
1: {…}
2: {…}
3: {…}
4: {…}
5: {…}
6: {…}
name: "YPR-SC GARIBRATH EXP"
src_departure_time: "16:45"
travel_time: "11:16"
to_station:
lng: 78.3177875
name: "LINGAMPALLI"
code: "LPI"
lat: 17.4830213
number: "12736"
classes:
0:
code: "SL"
name: "SLEEPER CLASS"

4.4 Hadoop Multinode Environment



4.5 Automation using Crontab

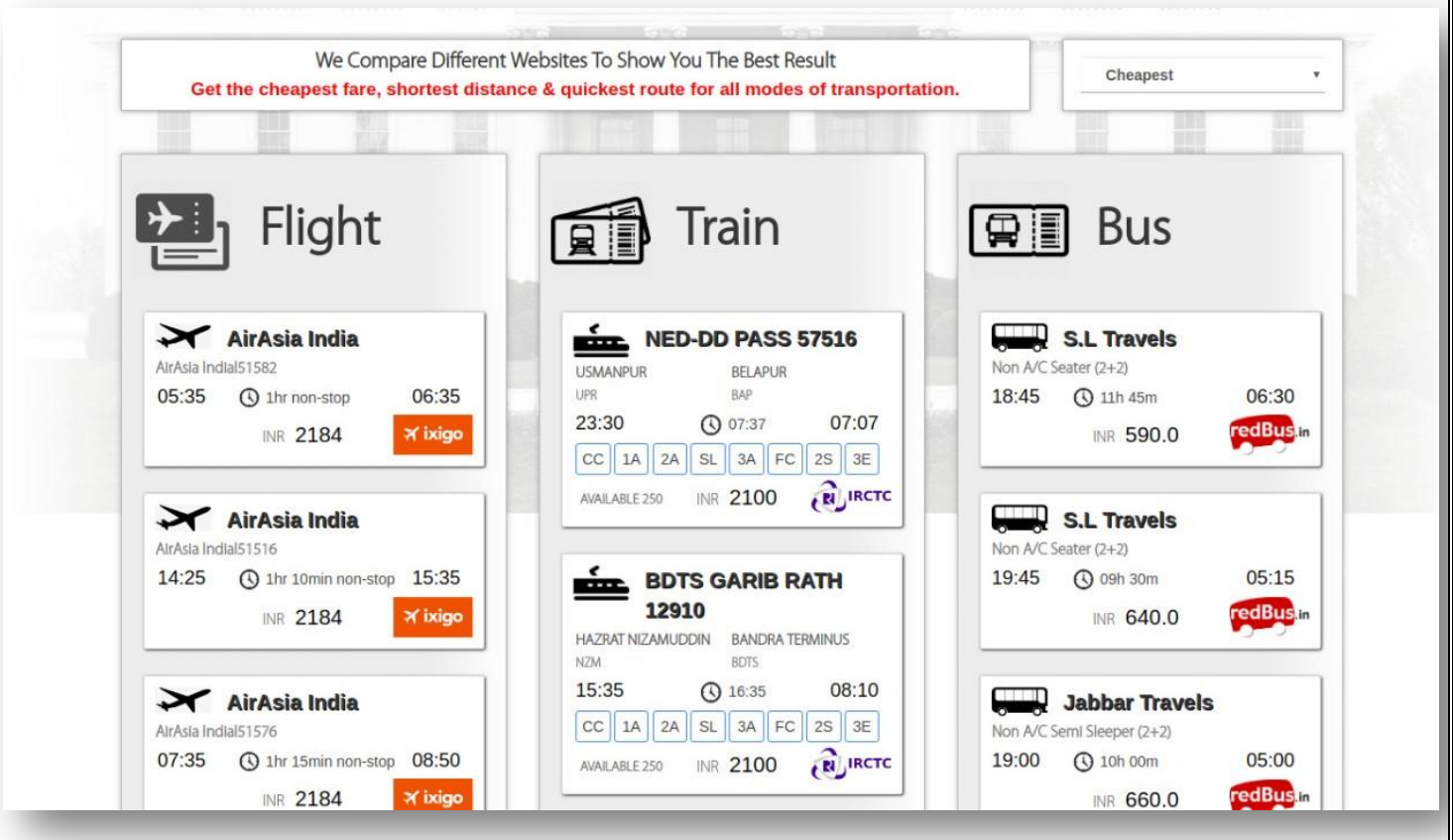
A shell script running in an interval of 6 minutes.



A python script running in an interval of 3 minutes.

```
# m h  dom mon dow   command
#*/1 * * * *    python3 /home/prince/test.py
#*/20 * * * *   python3 /home/prince/programs/web_crawling/cinsta.py
*/3 * * * *     python3 /home/prince/automatic/sync.py
```

4.6 Web Integrated Transport Support System(GoRaahi)



5. Conclusion

"GoRaahi" (a web integrated transport support system) will revolutionize the way we enquire tickets online for travelling from one place to another with quality transport facilities at affordable price which doesn't exceed your budget .It is a modern approach to traditional problems built with latest technologies like Apache Spark and efficient storage platform like Hadoop distributed file system totally centered on our very own built web crawlers.

6. References

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

https://www.seleniumhq.org/projects/webdriver/

https://hadoop.apache.org/

https://docs.djangoproject.com/en/2.2/

https://www.redbus.in/

https://www.ixigo.com/

https://railwayapi.com/api/#train-between-stations

https://crontab.guru/

https://stackoverflow.com/

https://spark.apache.org/sql/