Llama

A Big Data Integration and Analysis System

Authors:

- *Gaurav Kumar*
- *Héctor Ugarte*
- *Miguel Mármol*
- *Tina Boroukhian*

Mentor:

Mohamed Nadjib MAMI (PhD Student)

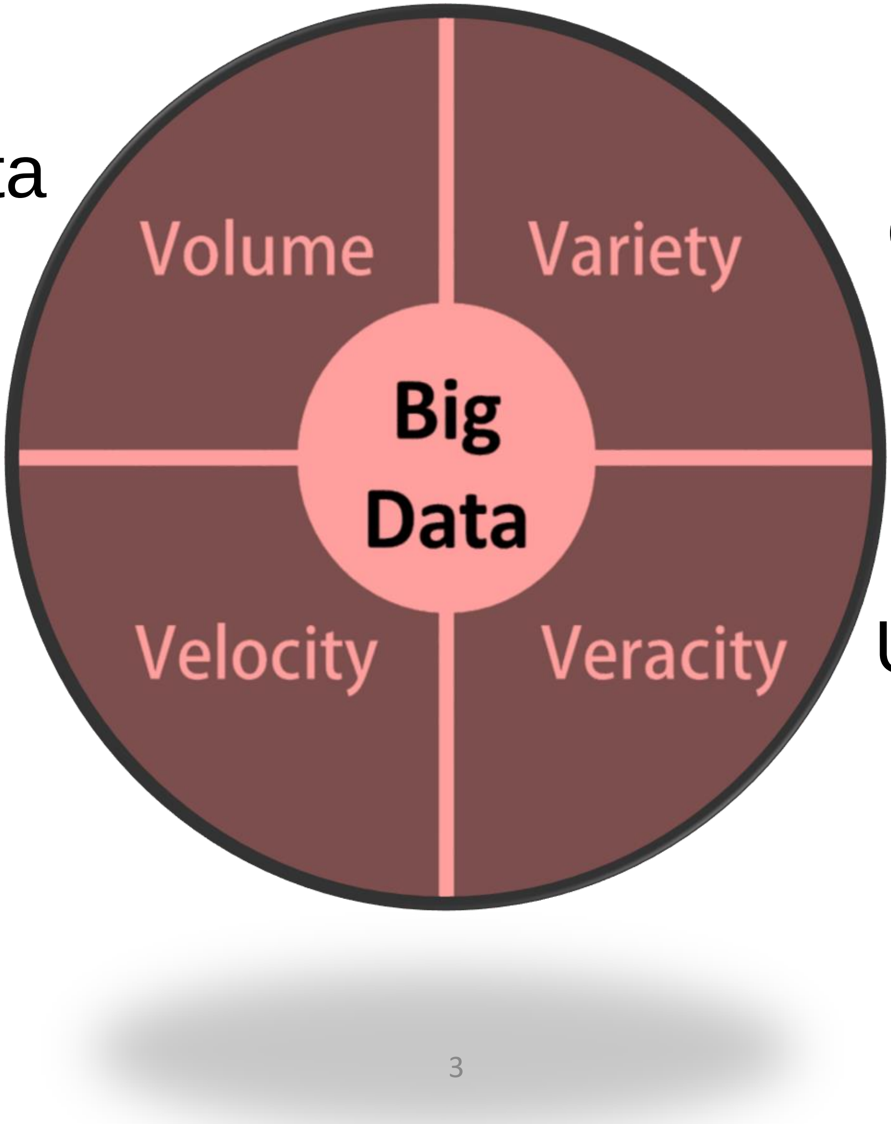
Summer Semester
2015

1. Introduction

What is Big Data?

Scale of Data

Different forms of Data



Uncertainty of Data

Data stream

Data Integration

- It consists of data extraction, transformation and loading operations that transform data state from one to another that is more suitable for decisional analysis.

- Examples of transformations:
  - Cleaning
  - Reformatting
  - Normalization
  - Filtering

Data Analysis

- Analysing big data is applied to find meaning and discover hidden relationships in integrated data.

- Create prediction models for better understanding of current data and its projection in the future.

Integration and Analysis of Big Data

- Size, form and speed of data make it a difficult task to integrate and analyze data
  - Time consumption
  - Resource limitation
- The technological advances and availability of low cost resources solved much of the problem
  - Decrease costs (storage, processing and developers)
  - Save time

2. Project Overview

Objectives

- Provide a distributed storage of large volumes of data
- Offer users a set of transformations to clean and prepare the stored data
- Run queries on the cleaned and prepared data to
    - Discover hidden relations
    - Forecast future (Making strategic decision)

Architecture

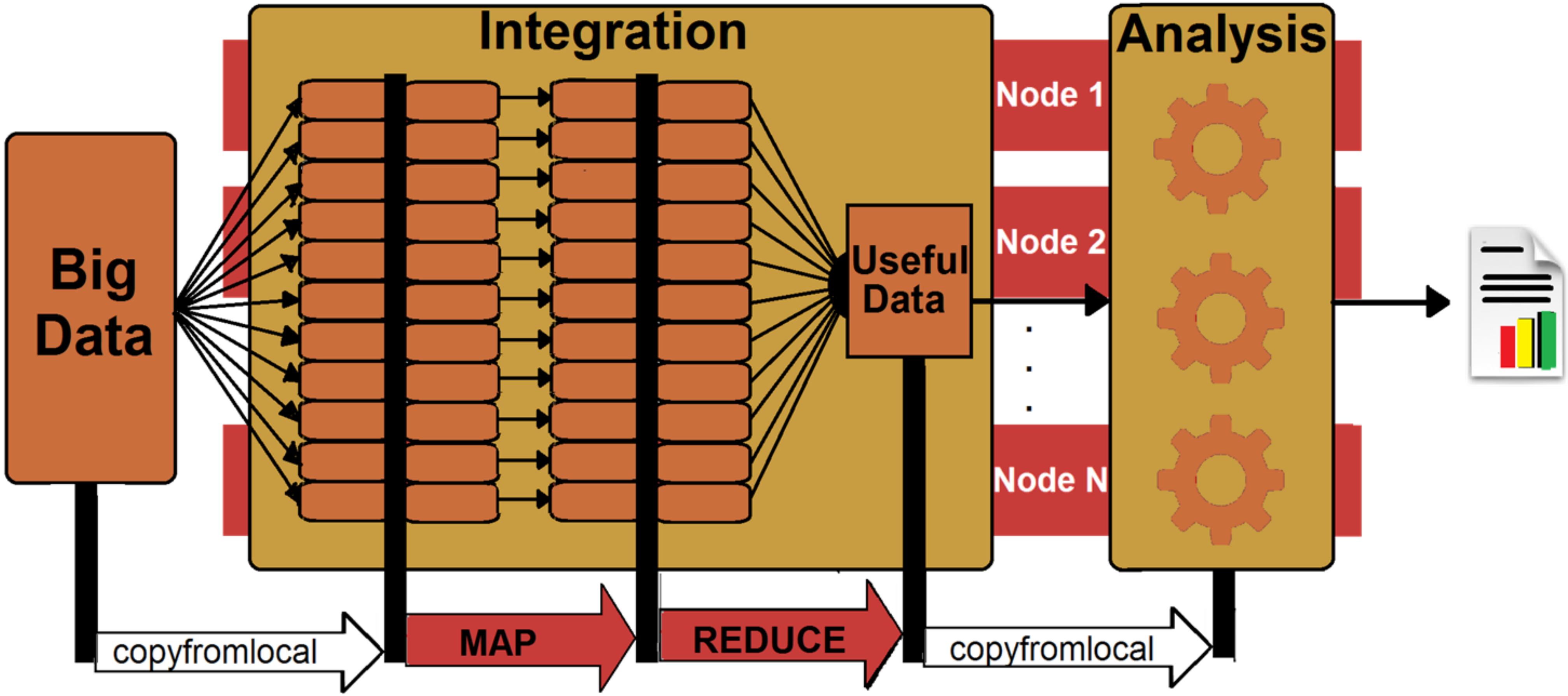


Figure 1: Architecture of Integration and Analyzing of Data [3]

Requirements

Functional:

R01. Get raw sample.

R02. Move data to HDFS.

R03. Clean data

R04. Reduce columns

R05. Analyze data

R06. Get results from HDFS to local system.

Non Functional:

- Scalability on multiple machines
- High Performance - short response time even with high volumes of data.
- Fault tolerance.
- Work in distributed environment.

What is Llama?

Llama is Data Integration and Analysis system made of two components:

1. Data Integrator: Llama first loads structured plain files. Then data loaded is cleaned, reformatted and filtered using a series of user-selected transformations. Integration jobs can be specified in two ways:

(1) Via a Graphical User Interface.
(2) Writing a script in IJSL (for Integration Job Specification Language).

If the first method is used, at the end of the integration phase, the specified job can be exported as an IJSL script that can be used to reproduce the same job or add on it.

2. Data Analyzer: Once ready, the new data is analyzed by means of SQL queries. The results can be stored for further analysis.

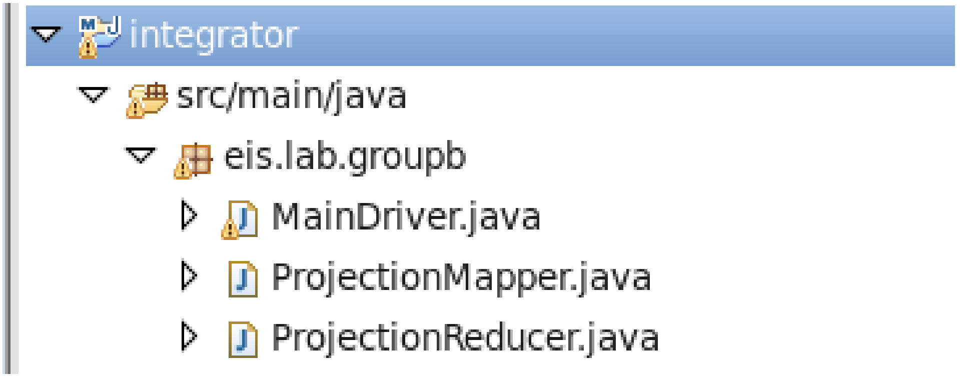
Challenges

- Hadoop Configuration
- Spark Configuration
- Cluster Creation
- Hardware limitations
  - Slow PCs
  - Hard drive space

3. Implementation

Integration Phase

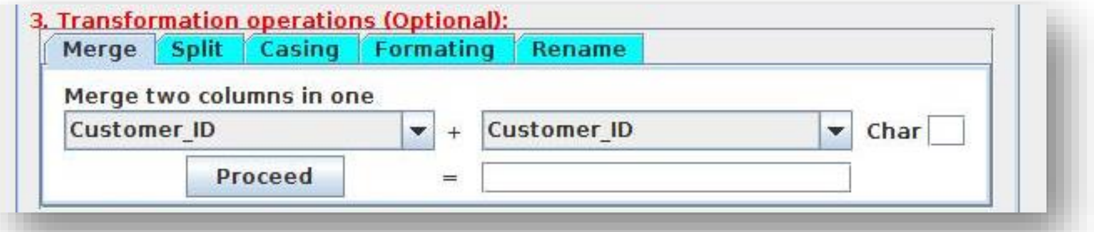
- Write Driver
- Writing one Mapper
  - Transformations
  - Restrictions
- Writing one Reducer
  - Adds a Header to the output file
  - Files will be stored in HDFS
  - Working first on a single node and then on a cluster



```
integrator
  src/main/java
    eis.lab.groupb
      MainDriver.java
      ProjectionMapper.java
      ProjectionReducer.java
```

Integration Phase

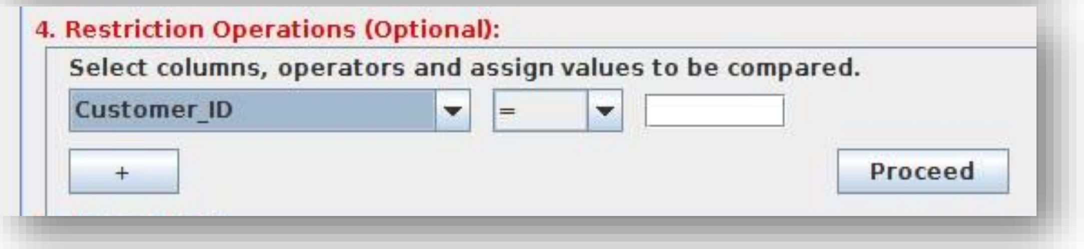
- Implementing five different types of Transformation operations
  - Merge columns
  - Split columns
  - Change the letter case of columns values
  - Change formatting of date columns
  - Rename files headers



3. Transformation operations (Optional):

Merge two columns in one

Customer_ID + Customer_ID Char



4. Restriction Operations (Optional):

Select columns, operators and assign values to be compared.

Customer_ID =

- Implementing Restriction Operations
  - For numeral values: =, <>, >, <,>=, <=
  - For textual values: EQUAL, NOT EQUAL, CONTAINS

Integration Phase

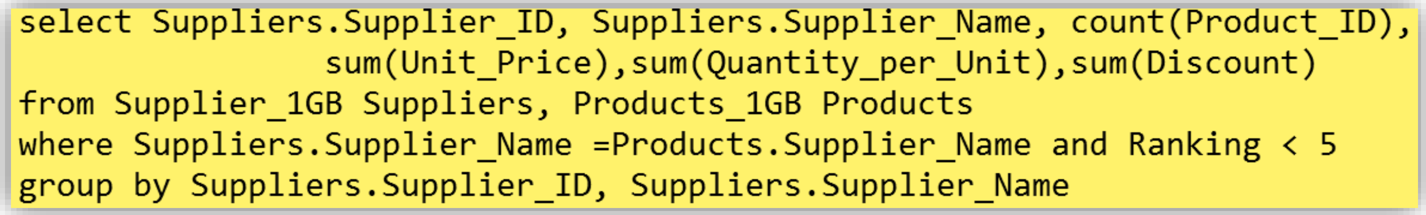
- Integration Job Specification Language (IJSL)

“a declarative language”

```
INPUTFILE Customers.csv
OUTPUTFILE Output.csv
SEPARATOR |
PROJECTEDCOLUMNS 0|2|3
PROJECTEDNAMES Customer_ID|Address|City
RESTRICTION 0|>=|20|0|<|50
```

Analysis Phase

- We are using SQL functionalities
  - Write a SQL query
  - Join multiple files
    - Join condition on common columns
  - Aggregation operations using GROUP BY
  - Add single or multiple WHERE predicates
  - Working as a single node and then on a cluster



```
select Suppliers.Supplier_ID, Suppliers.Supplier_Name, count(Product_ID),
                sum(Unit_Price),sum(Quantity_per_Unit),sum(Discount)
from Supplier_1GB Suppliers, Products_1GB Products
where Suppliers.Supplier_Name =Products.Supplier_Name and Ranking < 5
group by Suppliers.Supplier_ID, Suppliers.Supplier_Name
```

Testing

- Integration Testing
  - The two modules (Integration and Analysis) are working correctly after being integrated in one solution.
- Unit Testing
  - Unit test cases
    - In first case, verifying the list of available files in HDFS
    - In second case, verifying that all the field names that the file has have been retrieved correctly.
    - In third case, verifying the execution of the input query.
    - In fourth case, verifying the generation of the output file after the query has been executed.
- Validation Testing
  - Running several SQL queries and verifying the correctness of the results.

Core Technologies

- **Apache Hadoop** is an open-source framework for distributed storage and processing of very large data sets on a of commodity hardware. [1]
 (Used for Integration phase)

- **Apache Spark** is an open-source cluster computing framework. It employs the concept of RDD which are distributed unit resides primarily in memory, hence its high speed. [2]
 (Used for Analysis phase)

Development Environment

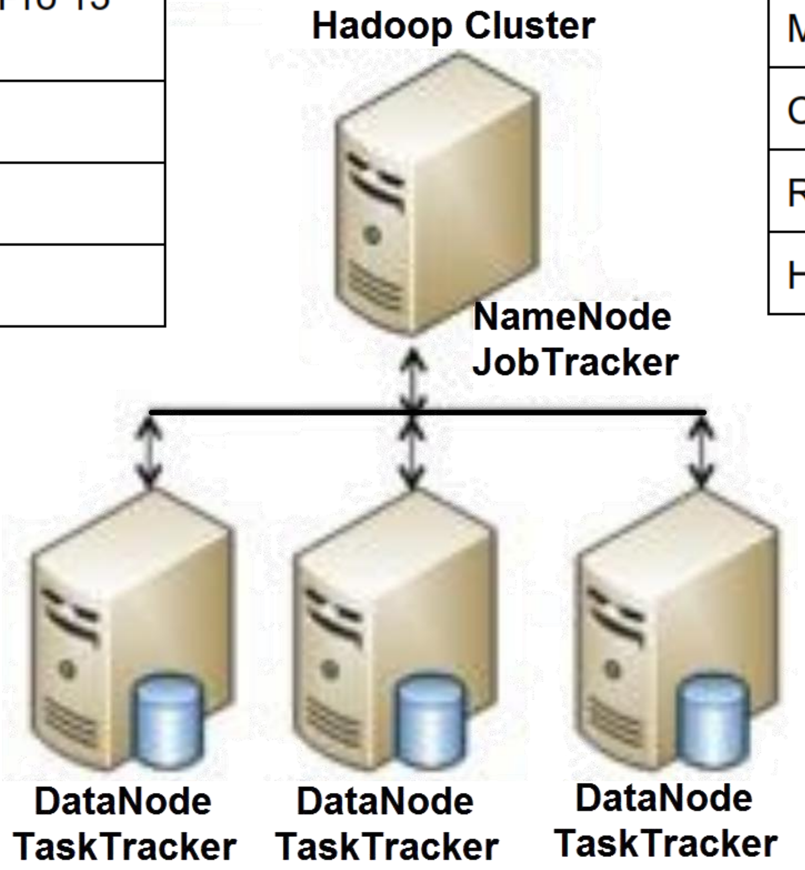
- Ubuntu 14.4
- Java JDK 1.8
- Hadoop 2.6
- Scala 2.10
- Spark 1.4
- Eclipse IDE
- Maven 3.3
- Python 2.7.10

4. Evaluation

Cluster Description

Brand and Model	Apple Macbook Pro 13
CPU	Core i7 2.9 Ghz
RAM	8 GB
Hard Disk	750 GB

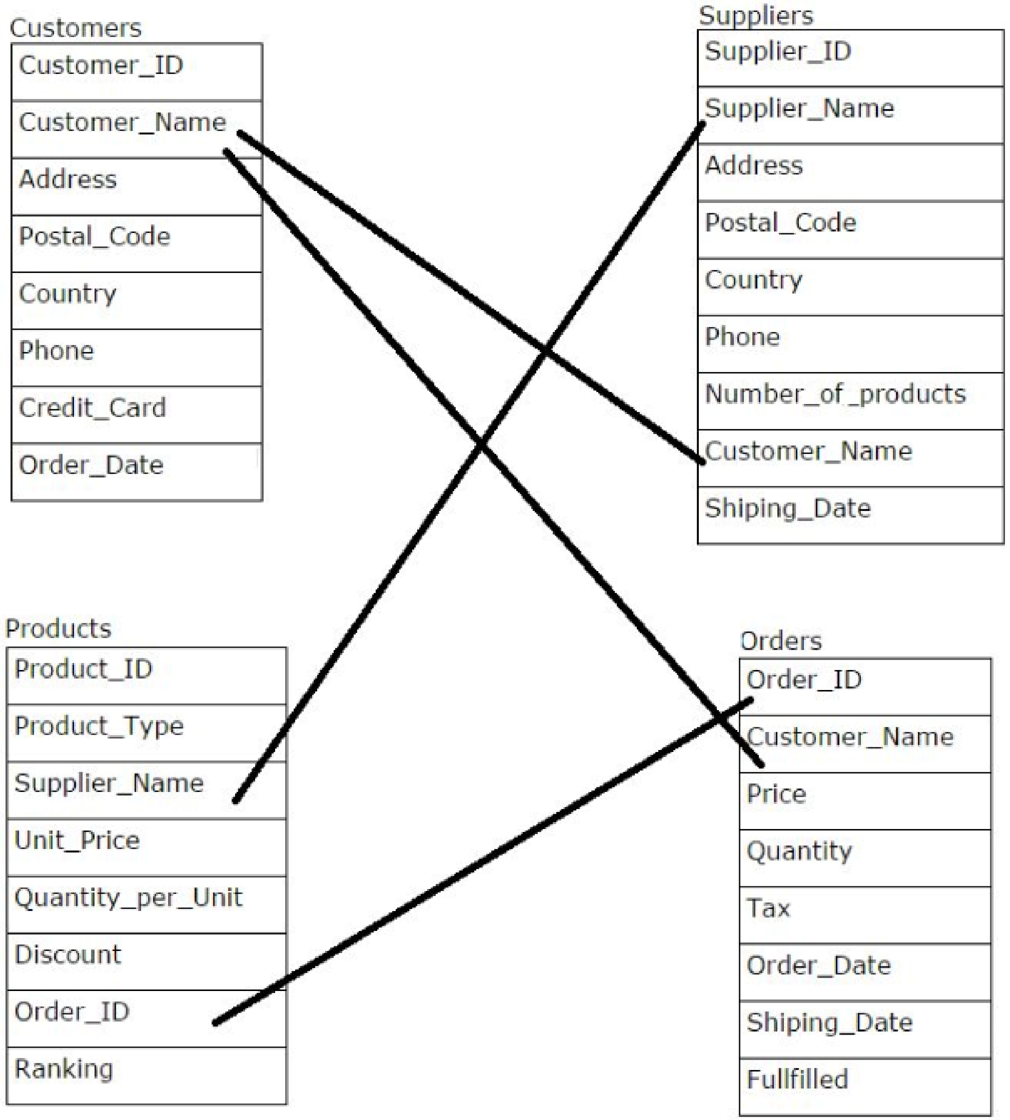
Brand and Model	Asus Zenbook UX303L
CPU	Core i7 2.4Ghz
RAM	8 GB
Hard Disk	1000 GB



Brand and Model	Dell Alienware m11x r1
CPU	Core 2 Duo 1.7Ghz
RAM	8 GB
Hard Disk	320 GB

Brand and Model	Dell Latitude 3450
CPU	Core i3 2.4Ghz
RAM	4 GB
Hard Disk	500 GB

Sample Data



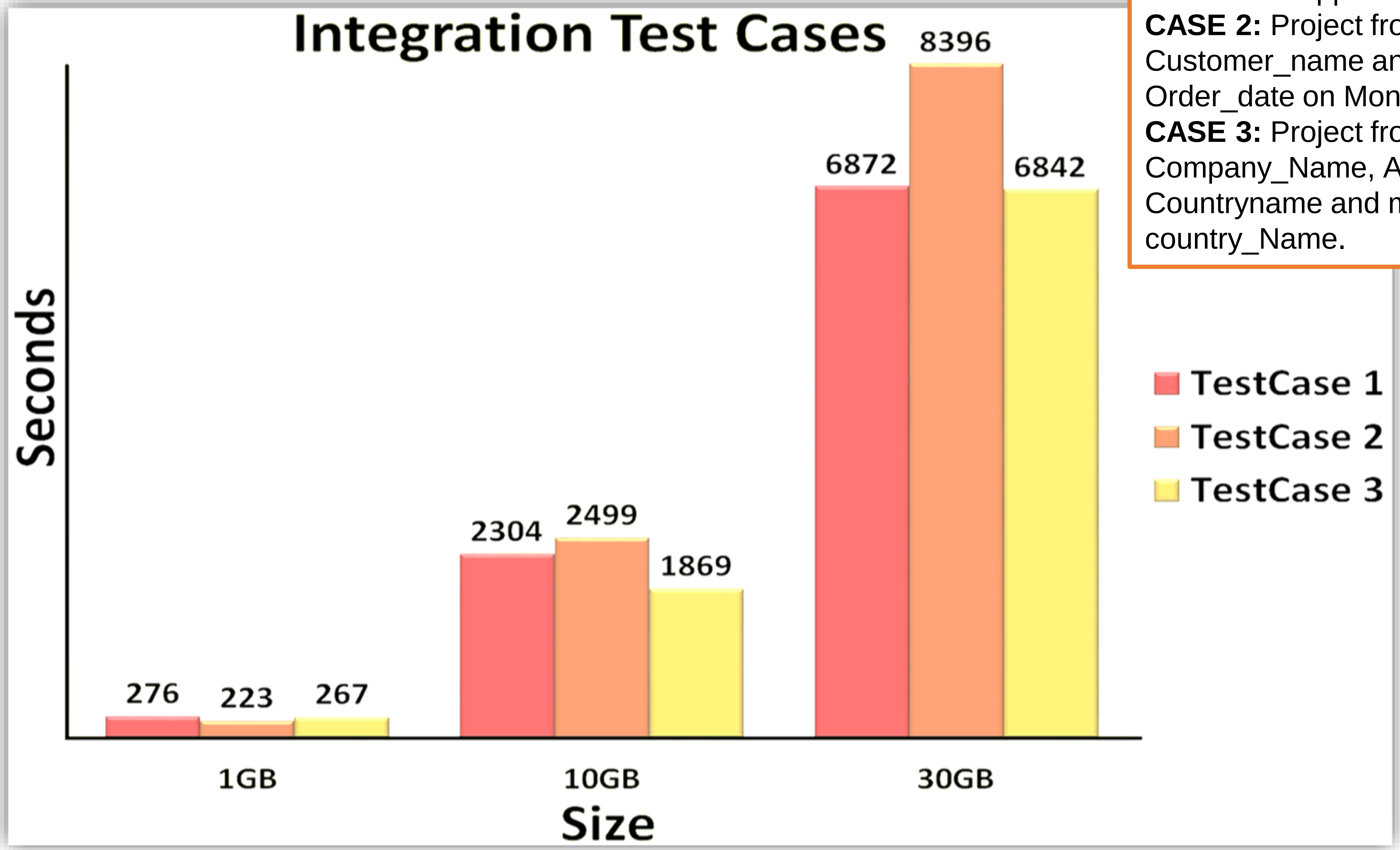
Customers
Customer_ID
Customer_Name
Address
Postal_Code
Country
Phone
Credit_Card
Order_Date

Suppliers
Supplier_ID
Supplier_Name
Address
Postal_Code
Country
Phone
Number_of_products
Customer_Name
Shiping_Date

Products
Product_ID
Product_Type
Supplier_Name
Unit_Price
Quantity_per_Unit
Discount
Order_ID
Ranking

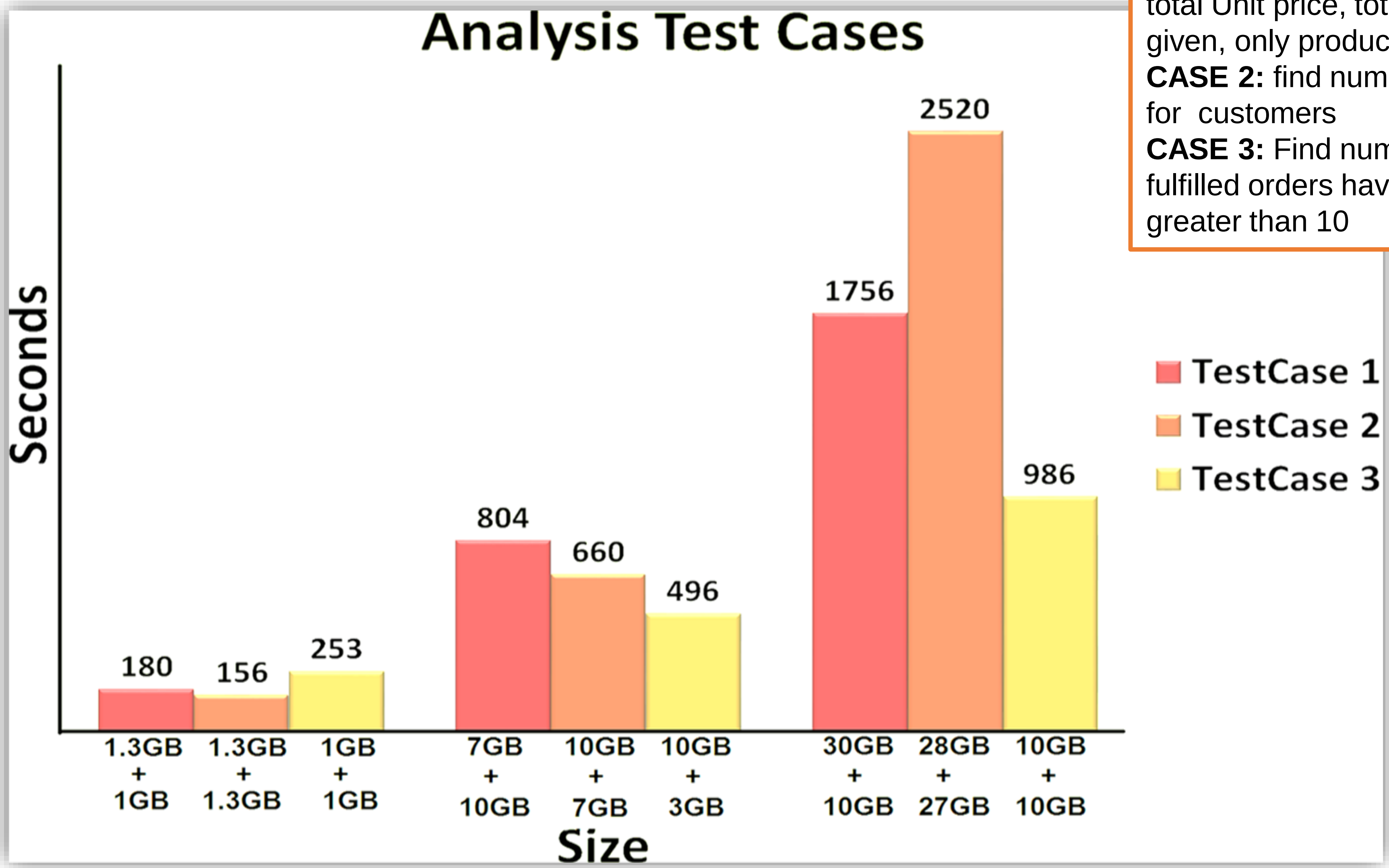
Orders
Order_ID
Customer_Name
Price
Quantity
Tax
Order_Date
Shiping_Date
Fullfilled

Evaluation Results



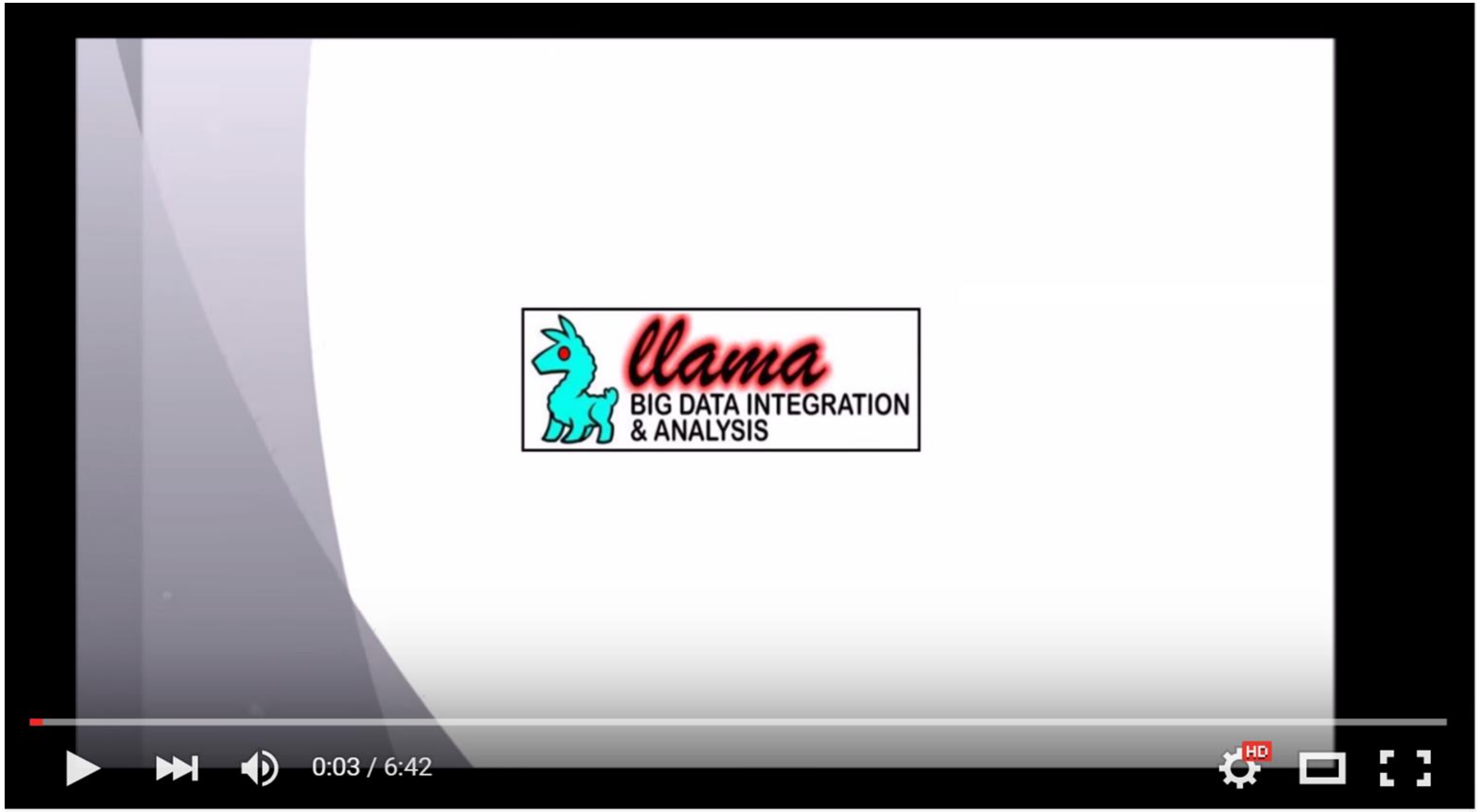
CASE 1: Project from customers table Customer_name and Address and cast Address to Upper case.
CASE 2: Project from Orders table Order_id, Customer_name and Order_date and split Order_date on Month and DayYear.
CASE 3: Project from Supplier Supplier_Id, Company_Name, Address and Countryname and merge address and country_Name.

Evaluation Results



CASE 1: Find all supplier ID and Supplier Name with number of products delivered with total Unit price, total Quantity and total Discount given, only products having ranking <5
CASE 2: find number of available suppliers for customers
CASE 3: Find number of available Products in fulfilled orders having no of products greater than 10

Demo



https://youtu.be/qlHG55S2K7g

References

[1]: "Hadoop." *Wikipedia*. Wikimedia Foundation, n.d. Web. 08 Oct. 2015.

[2]: "Spark." *Wikipedia*. Wikimedia Foundation, n.d. Web. 08 Oct. 2015.

[3]: “Hadoop Architecture”. Wikimedia Foundation, n.d.Web.08 Oct. 2015.