SIX WEEKS SUMMER TRAINING

REPORT

On

(Big Data: HADOOP)

Submitted by

(Udipto Goswami)

Registration No: 11401209

Programme Name: BTECH-MTECH (Dual) CSE

Under the Guidance of

Name of the Industry Coordinator: Reetika Lakha

School of Computer Science & Engineering

Lovely Professional University, Phagwara

(June-July, 2017)

DECLARATION

I hereby declare that I have completed my six weeks summer training at IBM-CEP Jalandhar (name and location of organization) from 10 June 2017 (start date) to 13 July 2017 (end date) under the guidance of Reetika Lakha (Name of Industry coordinator). I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of BTECH-MTECH (Dual) CSE (Relevant Degree), Lovely Professional University, Phagwara.

(Signature of student)

Udipto Goswami
Registration no: 11401209

Date:

ACKNOWLEDGEMENT

This training was supported by International Business Machine Career Education Partner (IBMCEP) and Lovely Professional University (LPU). I thank IBMCEP for providing insight and expertise that assisted me throughout the following six weeks and also in the proof of concept, although every concept given may not be agreed upon. I would also like to thank LPU for giving me the opportunity to indulge in this six week summer training. I'm sure the technology I learned will benefit me in every aspect of my career.

I thank Miss Reetika Lakha for her guidance and teachings regarding the topic. Her vision and teaching significantly improved this training and the proof of concept. Without her this may not have seemed possible.

TABLE OF CONTENTS

INTRODUCTION

The world is changing. With it the people and the technology. With the growing technology we are also witnessing the growth of massive amount of data emerging from various sources like sensors, mobile phones computers etc. All these data are called the Big Data which is growing exponentially. To handle this kind of data, traditional methods of data handling fails therefore something new and sophisticated is needed. Something which could assure to manage and provide meaningful results from the large datasets. This is where the technology I learned comes into play, HADOOP. It is a java based framework open source programming framework which supports the processing and storage of extremely large data sets in a distributed computing environment.

I spent my last six weeks learning this piece of technology and building a working proof of concept applying it. My proof of concept is based on the print industry’s popularly used printing technique called as Rotogravure printing. This technique uses cylindrical rolls of copper layered metal to print high resolution designs. Lately a lot of industries like Coca cola, Lays and Vaseline etc. are using this techniques and these to maintain the quality of design they keep a database of the cylinders too. This is the data I have focused to work on.

My aim is to analyse the dataset and derive some meaningful information out of it which would help the industries in maximizing profits and managing various resources associated to printing. The primary tools of the Hadoop ecosystem which I have used in this proof of concept are HIVE and PIG. To organize the data in graphical formats, I have used MS word.

PROFILE OF THE PROBLEM

In this section we shall discuss the history of the problem and how the problem has taken shape. The whole idea of this section is to know about our dataset and why we need it. Since the invention of the Rotogravure cylinders, the printing industry has expanded its spans widely. Many industries which developed products and hired other printing people to print for them are now using their own units for printing. Technology of Rotogravure engraving printing allows company to print at a very cheap investment and print very accurate and precise designs.

This is how the whole dataset is formed. Large industries when started using Rotogravure cylinders, manufacturers and clients began to keep records of their product. They did it in order to have a proper resource planning form historical data. Since large industries are using it therefore, the data is also very huge in number with many attributes and properties. All of this as discussed are important at some point. So therefore, the database grew exponentially.

The past also says that while starting any industry for the first time, it is very hard when it comes to managements and resource planning. In most of the situations we end up in severe losses due to which business has to be shut down. But the root of those kind of results are poor market research. If an industry does it proper market research before starting its business with the proper datasets, one can assure of minimum losses. Also proper resource management could be ensured.

So coming to this data set. It acts like the historical dataset which could be taken into account for an industry while market research. With information like paper types, ink types, solvent used, chrome content etc. we can assure or proper planning and thus an industry might survive its starting phase. But it large of be fair. Large enough that storage databases like MySQL or ORACLE will crash while query processing. Therefore efficient system like HADOOP is applied which is very helpful while processing.

EXISTING SYSTEM

This section will discuss about the profile of the existing system of the database for Rotogravure cylinders. As mentioned above the whole system was imported from a simple .txt file which had 40 comma separated columns. From that we can tell that the whole database for maintaining the data of cylinders are in simple structured database like MySQL or oracle. Glancing at the data for the first time, anyone would say that it is a good thing. Stored in structured database means ease of access only JSON files. But, that not it.

The data we downloaded may be structured but, it is not normalised. That is just one problem which gives us irrelevant datasets about the data stored. Another problem is that all total there are 40 attributes of the table. Normalising them will mean to reconfigure everything. Also we cannot make sure that we will be able to extract and preserve that data integrity. Above it, the biggest problem is that data being huge in size. It's about 541 records stored. Many of those records even contain NULL values which even are hard to ignore and resolves. Imagine managing records about 541 with 40 attributes. How much about it be clumsy. After that if you wish to query out the results of the problems which we discussed above, imagine the time it would take while processing. One query could take up to 10 mins to execute. 10 mins is a lot more when we have large numbers of problems statements to execute.

In a gist, the traditional database fails here while storing the data and processing it in the same time. We will encounter frequent crashes of the database, unable to process errors. Also the limitation of storage capacity of a database will be an issue when were are dealing with a data which grows exponentially. The data which we downloaded is just of 541 which produces errors like this imagine what a 10 time's bigger dataset could cause.

As mentioned earlier, some key tools which are used in this analysis includes HIVE. So, comparing MySQL with HIVE gives us the following points:

WHEN TO USE HIVE:

- **If you have large (think terabytes/petabytes) datasets to query:** Hive is designed specifically for analytics on large datasets and works well for a range of complex queries. Hive is the most approachable way to quickly (relatively) query and inspect datasets already stored in Hadoop.
- **If extensibility is important:** Hive has a range of user function APIs that can be used to build custom behaviour in to the query engine.

WHEN TO USE MYSQL:

- **If performance is key:** If you need to pull data frequently and quickly, such as to support an application that uses online analytical processing (OLAP), MySQL performs much better. Hive isn’t designed to be an online transactional platform, and thus performs much more slowly than MySQL.
- **If your datasets are relatively small (gigabytes):** Hive works very well in large datasets, but MySQL performs much better with smaller datasets and can be optimized in a range of ways.
- **If you need to update and modify a large number of records frequently:** MySQL does this kind of activity all day long. Hive, on the other hand, doesn’t really do this well (or at all, depending). And if you need an interactive experience, use MySQL.

We have seems the comparison between HIVE and MySQL, let us take a look at another key tool used which is PIG. The following is a comparison between them:

PIG VS SQL:

The DBMS systems that SQL operates on, are considered to be faster than MapReduce (operated on by Pig through the Pig Latin platform). However, it is the loading of data that is more challenging in case of RDBMS, making the set up difficult. Pig Latin offers a number of advantages in terms of declaring execution plans, ETL routines and pipeline modification.

SQL is declarative and Pig Latin is procedural to a large extent. What we mean by this is in SQL, we largely specify “what” is to be accomplished and in Pig, we mention “how” a task is to be performed. A script written in Pig is essentially converted to a MapReduce job in the background before it is executed. A Pig script is shorter than the corresponding MapReduce job, which significantly cuts down development time.

PROBLEM ANALYSIS

In this section, we are describing in detail about our problem statements and why should we focus on them. Before we begin, we have already discussed about why we need this analysis and how it could help any industry using the Rotogravure cylinders. The bigger problem which the existing system has is of the management and performance. Using the traditional methods and technology for analysis results is much more processing time. Therefore HADOOP ecosystem is taken so as to speed up the analysis of the data. Keeping then in mind, the following are the problem statements analysing which we hope to find some meaning full results:

- **Find the top 10 most recent cylinders:** As the problem statement states, we intend to find the result of top 10 cylinders who recently ordered cylinders from a manufacturer. The purpose for this problem statement could be said as to keep in track which company is adapting the following technology. There may be other reasons why this data could be beneficial, for example, if it is found later that the manufacturer who completed orders of the recent cylinders, turns out all the cylinders are containing bands, i.e. manufacturing defects and hence using them in production line will cause a severe error while printing. In this case this data of 10 ten most recent cylinders would help manufacturer to narrow it down about which cylinder is having bands and which not.
- **Find the top 3 most used solvents:** This problem as it suggests will result us the top 3 most used solvent in the printing industries. There are many solvent types which are used while printing process is carried out. They are used since they have the capability to dissolve another solutions in them. In printing industry solvent is used to dissolve the pigment and aid in drying the ink quickly. Various type of solvent are present, one with low boiling point which evaporated quickly leaving the pigment on the surface. Other solvents are absorbed rapidly by the paper, again leaving the pigment on the surface of the substrate. It totally depends on the client where he is using it. So the top 3 most used solvent will tell an industry about which is the most popular one to use for maximum profit and good output. In this way efficient capacity planning could be done with the resources an industry owns.
- **Find the customers who uses top most used solvents:** This problem statement will tell us about the customers which uses the top 3 most used solvents. The problem also tells us

about the popularity of the solvent in the market. This data will help the manufacturer of the solvent to know their customers end to end. This may build up a good business relationship among them. Also end to end feedback and query about the solvent could be taken for further improvements. Maximum customer satisfaction could be provided by this data. Also discounts could be demanded by one when a strong bond is build up.

- **Find the top 3 paper mill location who ordered cylinders recently:** As this problem goes, it gives us the resultant of the top 3 paper mill locations who ordered the cylinders of rotogravure in recent times. This data might be helpful in determining which paper mill location recently ordered cylinders. If cylinders are banded, defected then the recent list may help the manufacturer. We can also determine the popularity of mills, which mill indulges in max production in rotogravure printing techniques.
- **Find the top 3 most used paper types:** This problem is one interesting one for the resource planning of an industry which involves the printing. The top3 most used papers types will give us the most popular and on demand paper types. Using which we will make sure in an industry that customers will choose only these particular types as they are most preferred. This way investments on specific types could be done.
- **Find the records of the cylinder with max chrome content:** The cylinder surface is then chrome plated in a chrome bath or tank, which has the identical appearance to that of a copper bath and finished (polished) immediately after the cylinder has been engraved. This is done to preserve the integrity of the engraving. This ensure the smoothness of the cylinder and maximum durability. The max chrome content will mean that the cylinder is in it 100% condition. The chrome content deteriorates with time and also while repairs. This information will help in determining the manufacturer of the cylinders to determine which cylinders produces are 100% products. Also this will allow to set a benchmark of the chrome content. How much should be kept as a standard mark.
- **Find the count number of band type as: band and no band:** Bands are as discussed above are defects arises in a cylinder due to constant use or manufacturing defects. Bands could ruin the whole design while printing. Therefore determining the number of defected pieces and good pieces produced, will help the manufacturer to optimize its machinery. Also future fault managements could be done.
- **Find the max wax numeric:** This is something which might be used by the client who uses the cylinders. It tells us the max wax say the max optical density. Also more durable designs are produced. So this database keeps the max wax content used. According to it,

cylinders are made such that the wax won't harm the chrome plating and the integrity of the design.

- **Find and count ink types:** This problem statement counts all the ink types used by any client for printing purposes. Determining the counts will give us an estimate of which ink types should be preferred by any other industry. This data serves like a survey result. It tells us what the area where profit could be maximised. In this case, preferred ink type tells us the most popular ink types to be invested on.
- **Find the cylinders with max viscosity:** This is the data of cylinders operating on maximum viscosity. The viscosity of the fluid will determine the rotating speed of the cylinder so as to not to heat up the cylinder mechanism or destroy the design. All press equipment, including viscosity control systems, need to be much more reliable, require significantly less maintenance, enable shorter make-ready times, and be able to operate in a world where pressman skills are decreasing. This data fill significantly help in setting the benchmarks for the equipment using.

So, the following were the problem statements for the area of operation of the data of Rotogravure cylinders. The problems stated above are such which provides some numbers which could be helpful for the client who uses them and the manufacturer who produces the cylinders. The ultimate goal of the whole problems is to ensure fine finished outputs with maximum customer satisfaction.

SOFTWARE REQUIREMENT ANALYSIS

In this section we will go through the requirement analysis of our new system. The last few section have given us the idea of how the existing system used for managing the data is not beneficial in case of big data. Due to several valid reasons we conclude that storage and processing data in traditional system in inappropriate. This is where HADOOP along with its tools come in.

The requirement analysis of this new system is divided into two significant parts, one considering the storage of the data and second the processing and execution of the queries. We shall discuss them briefly.

Considering Storage: Our existing system works on a single node cluster. That means only one single system stores the whole data. Doing so, it ensures the whole information remains on a single system where the changes could be done in order to apply in to all other access points. This is however the reason why larger datasets will create problem. Since the data is large, it is not wise to store it in a single node. Distribution over a series of commodity hardware are more effective in result. The following point will clarify more about the storage aspect:

- **Data stored should be easily assessable:** The final retrieval of data should be very simple such that even a non-technical person dealing with the set should be able to understand the access.
- **Data store should maintain reliability:** There should not be errors or wrong data while storing. Ensuring the integrity of the data is important.
- **Maximum availability:** This is something which the traditional storage was not able to do. Maximum availability can be assured by creating duplicates over the network of commodity hardware. One feature which is most demanded these days.
- **Fault Tolerance:** This is another property necessary. Faults may hamper the whole integrity of the database. So there should be capability that the new system should tolerate the faults and be able to recover from it.

Considering Processing: Since our existing system stored and processes everything on a single node cluster therefore, the processing of the data will take much more time. Especially when dealing with the dataset large, the following points should be considered:

- **Fast Processing:** The existing system is unable to fast process things due to storage and process at the same location. But distribution allows multiple nodes to process and store at the same time. Therefore, a significant fast output is expected.
- **Fault Log:** There should be failure record in order to determine at which point did a query didn't commit. This is necessary in order to get accurate outputs.
- **Genuine Outputs:** There should be truth to the data produced. The data should not be a complete irrelevant result. It is expected that the results are true and what the user asked for.

From the above discussed points, we are clear about the requirements related to technical aspects. However there are other aspects which are needed to be considered while setting up the whole dataset into a completely new system, like cost and maintenance etc. Let us quickly look at them too.

Cost Effectiveness: This is the main concern when a major transition like this s to be done. A lot of planning and money is involved and it is important to analyse all aspects. Hadoop's cost per terabyte is much less than high end data warehouse database server like Teradata and Oracle's Exadata. Hadoop's storage costs are also substantially less than many high and mid-level storage area network (SAN) solutions. Since Hadoop is fundamentally a distributed file storage system, it can cost effectively replace other data archiving and/or data storage solutions. Hadoop also has the additional benefit of being able to query its stored data.

Maintenance: It is very important to maintain such large datasets. Maintenance is the phase which is most vulnerable and constant efforts are needed.

The following are the operations which are frequently required while dealing with HADOOP ecosystems.

- **File System Checks:** We should check health of HDFS periodically by running fsck command.
- **HDFS Balance Utility:** Over the period of time data becomes un-balanced across all the Data nodes in the cluster, this could be because of maintenance activity on specific Data node, power failure, hardware failures, kernel panic, unexpected reboots etc. In this case because of data locality, Data nodes which are having more data will get churned and ultimately un-balanced cluster can directly affect your MapReduce job performance.

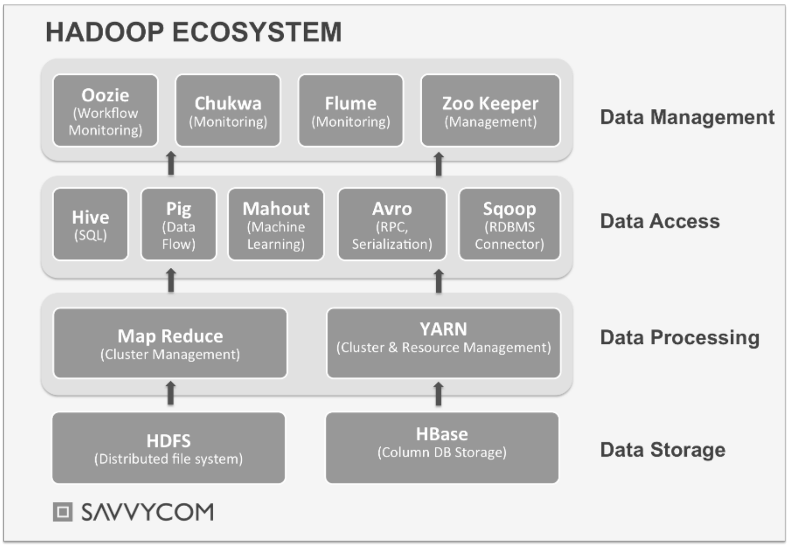
- **Adding new node to cluster:** We should always maintain the list of Data nodes which are authorized to communicate with Name node, it can be achieved by setting dfs.hosts property in hdfs-site.xml.
- **Decommissioning a node from cluster:** It's a bad idea to stop single or multiple Data node daemons or shutdown them gracefully though HDFS is fault tolerant.
- **HDFS Metadata Backup:** fsimage has metadata about your Hadoop file system and if for some reason it gets corrupted then your cluster is un-usable, it's very important to keep periodic backups of filesystem fsimage.

The following points summarises the software requirements of the new system. Based on this our whole system is designed.

DESIGN

In this section we will discuss the design of the proof of concept made. However, since it is only a proof of concept against the HADOOP ecosystem therefore, there is no interactive front end. There is only the HADOOP ecosystem using which we fetch the result we desire. The main motive of it is to justify the learning and explain the working. The design which you will see here is of the HADOOP system and the HDFS to store files and process queries.

The following figure illustrated the HADOOP ecosystem briefly:


HADOOP ECOSYSTEM
Oozie (Workflow Monitoring)
Chukwa (Monitoring)
Flume (Monitoring)
Zoo Keeper (Management)
Data Management
Hive (SQL)
Pig (Data Flow)
Mahout (Machine Learning)
Avro (RPC, Serialization)
Sqoop (RDBMS Connector)
Data Access
Map Reduce (Cluster Management)
YARN (Cluster & Resource Management)
Data Processing
HDFS (Distributed file system)
HBase (Column DB Storage)
Data Storage
SAVVYCOM

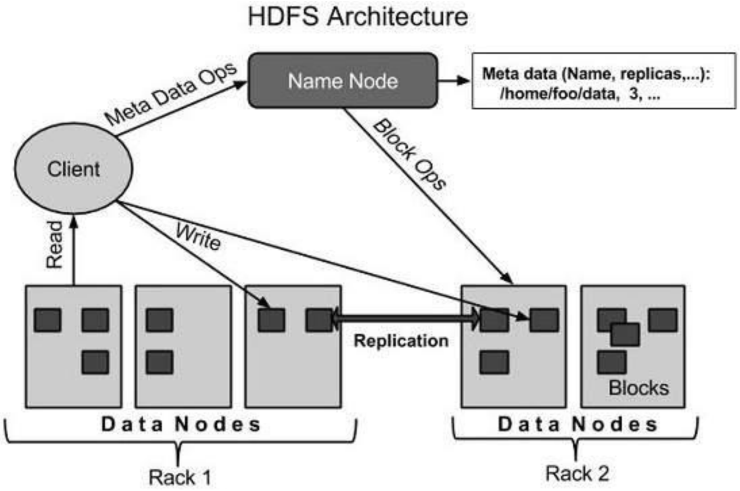
If we take a look at the figure given, it describes the 4 layers of the ecosystem through which the data is accessed. The first layer as names is the management layer. Operated with various tools like Oozie, Flume, and Zoo Keeper. These are responsible for constant monitoring of the flow and integrity of the data in hand.

The second layer is the data access layer where above mentioned tools like HIVE, PIG and Sqoop are there. There are tools highly used for data access. This proof of concept also uses HIVE and PIG for data access. Other tools like Mahout and Avro are there for tasks like Machine Learning and RPC or serialization.

The third layer is the data processing layer. This layer is responsible for matching the records and present in then reduces form as user wants. There are two processes through which we can do that. One from Map Reduce which is used in this proof of concept and other YARN, a cluster and resource management unit.

The forth layer is the storage layer. This layer is responsible for data storage. Again there are two ways, HDFS and HBase. HDFS is the standard storage where the storage is done by distributing the data among commodity hardware. HBase, is another way, where data is stored in a semi-structured way in a column database storage. However, this proof of concept uses only the HDFS.

The next design is of the HDFS Architecture. This explains how the data transfers from layer to layer and the magic happens.



Observe that the client makes a request to the name node which name node is the commodity hardware that contains an operating system and the name node software.

The name node is responsible for various tasks through which it acts as the master node for operations to perform. The following are the functions:

- Manages the file system namespace.
- Regulates client’s access to files.
- It also executes file system operations such as renaming, closing, and opening files and directories.

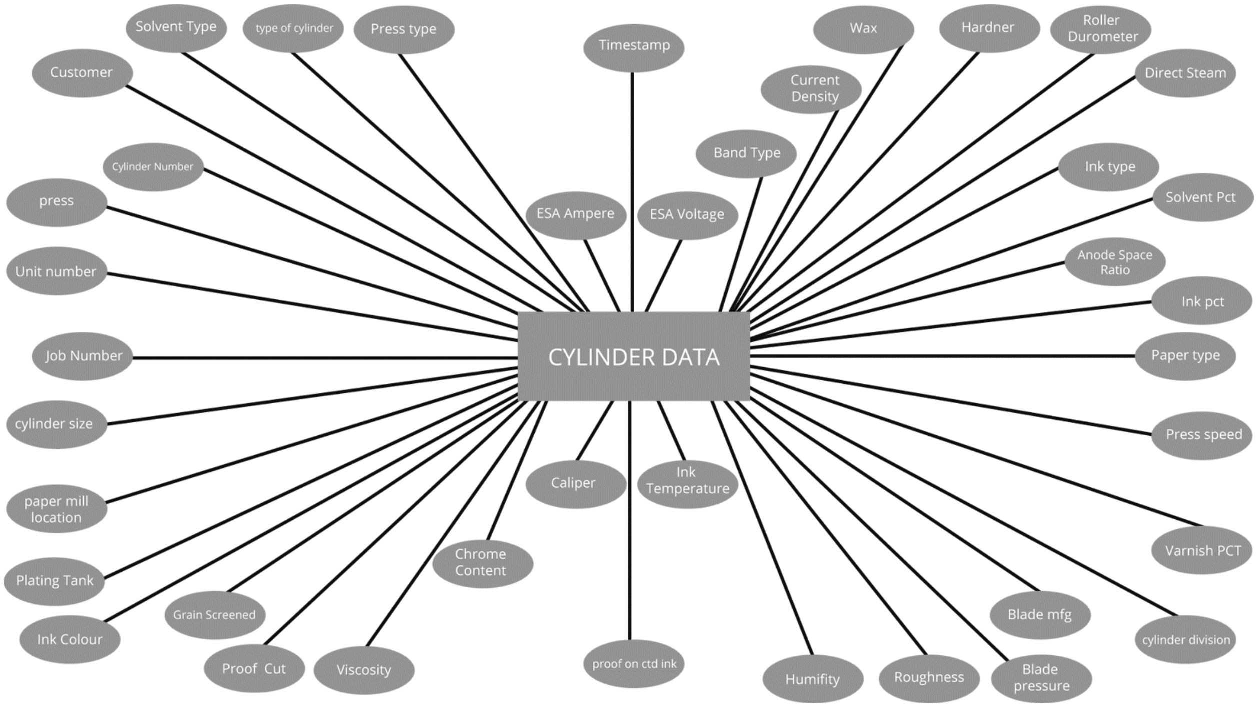
When a request is made the name nodes maps the query to the data nodes in which the data is stored. Then the Mapping happens. The Blocks in which there data are stored are searched and are retrieved back.

The next diagram which is shown is the ER diagram of the database structure we have for Rotogravure cylinders. Since there is only one table therefore only one main diagram. However there are some queries where we used some pre-fetched data those diagrams are also shown.

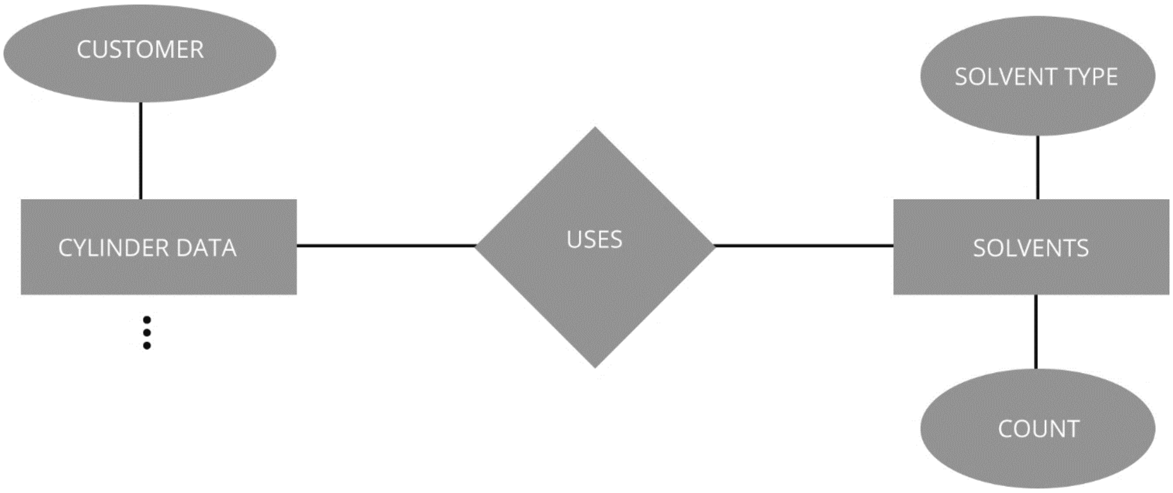
You can see, in this diagram, the attributes of the cylinder data entity. All the properties are represented associated with it.

This is the master record of the cylinder data which is maintained. With 40 attributes it is really difficult to represent it. It roughly looks like the figure given next. The ER diagram representation is shown in the following.

The following is the ER diagram representation of cylinder data:



The next diagram is for Problem statement 3 which states to find the customers who uses top most used solvents.

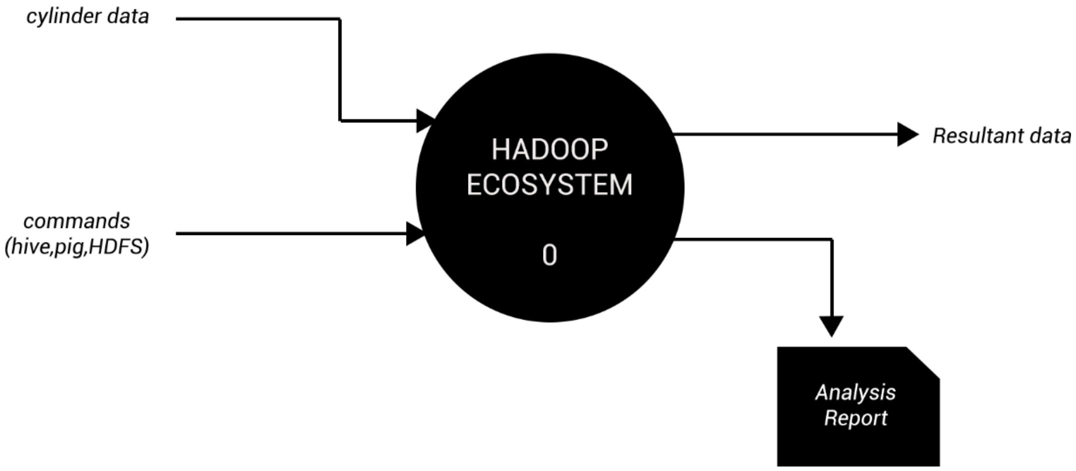


However, there the cylinder data is the previous entity which we just showed in the diagram. Here I’ve minimised the attributes only to represent. This one is the relationship diagram which is formed.

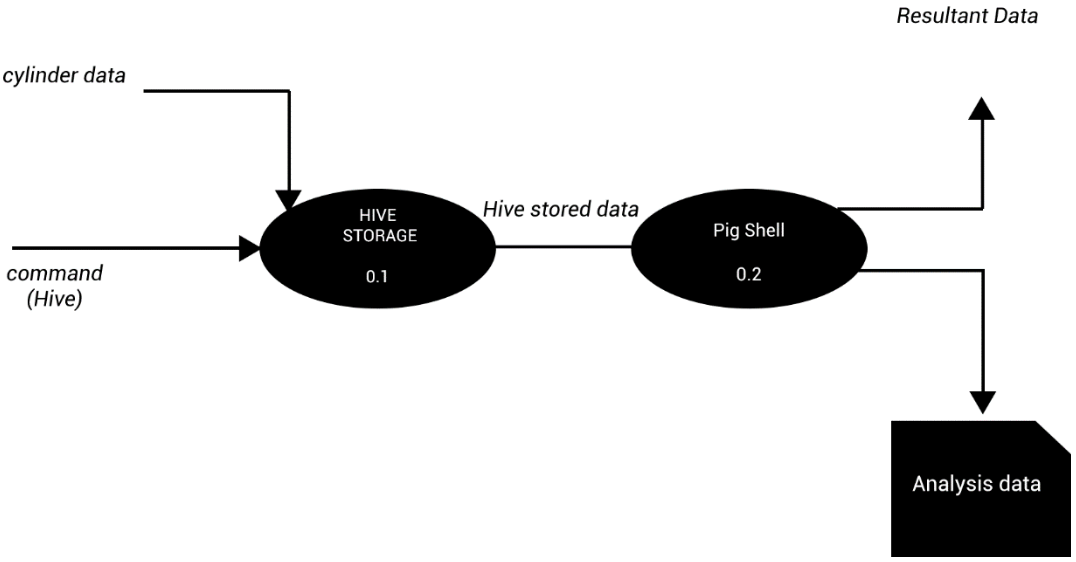
DATA FLOW DIAGRAM:

The following diagram show the processing of the data throughout the Hadoop system.

LEVEL 0 DFD



LEVEL 1 DFD



TESTING

In this section we will discuss some tests which were done to inspect the proper functioning of the environment. Our main aim is to inspect HIVE and PIG commands whether the proper function regarding the command was carried out of not. Our database whether loaded successfully giving output according to our requirement. First is to check if the dataset was downloaded properly. Second is to inspect whether the dataset was successfully imported into HIVE or not. Lastly, we will inspect PIG commands for their results.

TEST CASE 1: HIVE TABLE CREATION:

Once downloaded, we will export it into HIVE but first, we need to create an external HIVE table in order to make it available for PIG.

COMMAND:

```
create external table cylinder_bands
( time_stamp int,cylinder_number string, customer string, job_number int, grain_screened string, int_color string, proof_on_ctd_ink string, blade_mfg string, cylinder_division string, paper_type string, ink_type string, direct_steam string, solvent_type string, type_on_cylinder string, press_type string, press int, unit_number int, cylinder_size string, paper_mill_location string, plating_tank int, proof_cut int, viscosity int, caliper double, ink_temperature double, humifity double, roughness double, blade_pressure int , varnish_pct double, press_speed int, ink_pct double, solvent_pct double, esa_voltage int    , esa_amperage int    , wax double, hardener double, roller_durometer int, current_density int, anode_space_ratio double, chrome_content int, band_type string)
 row format delimited fields terminated by ','location '/user/hive/warehouse/rotogravure.db';
```

From this command we made an external table named cylinder_bands in HIVE in location /user/hive/warehouse/rotogravure.db.

TEST CASE 2: COMFIRMING TABLE CREATION:

For confirming if our database is created successfully or not we can use the following:

COMMAND:

```
describe cylinder_bands;
```

This command will list the inner structure of the table. If the necessary attributes are there then it’s a success. Otherwise drop it and create again.

TEST CASE 3: IMPORTING DATA INTO HIVE

From the .txt file now we are importing the dataset into HIVE table.

COMMAND:

```
LOAD DATA LOCAL inpath '/Rotogravure/cylinder_data.txt' OVERWRITE INTO TABLE
cylinder_bands;
```

This will result OK if it is a success. If failure encountered, then check the names of the folders properly and try again.

TEST CASE 3: ENSURING DATA IS PRESENT

Now that we have imported the data, we just need to check if data is there or not.

COMMAND:

```
select * from cylinder_bands;
```

This command will give us the whole dataset as result. If it doesn’t in the TEST 2 was failed.

TEST CASE 4: IMPORTING INTO PIG:

The hive bit is over, now we check the Pig part in order to perform the operations.

COMMAND:

```
cylinder_data = LOAD '/user/hive/warehouse/rotogravure.db' USING PigStorage(',') AS
 (time_stamp:int, cylinder_number:chararray, customer:chararray, job_number:int
,grain_screened: chararray, int_color:chararray, proof_on_ctd_ink:chararray, blade_mfg:
chararray, cylinder_division:chararray, paper_type:chararray, ink_type: chararray,
direct_steam :chararray, solvent_type: chararray, type_on_cylinder: chararray, press_type
:chararray, press:int, unit_number: int, cylinder_size: chararray, paper_mill_location
:chararray, plating_tank: int, proof_cut :int, viscosity: int, caliper:double,
ink_temperature: double, humifity: double, roughness: double, blade_pressure :int ,
varnish_pct: double, press_speed: int, ink_pct: double, solvent_pct: double, esa_voltage:
int, esa_amperage: int, wax:double, hardener: double, roller_durometer: int,
current_density: int, anode_space_ratio: double, chrome_content: int,
band_type:chararray);
```

This command will import the cylinder dataset which is previously stored in HIVE into PIG under the name *cylinder_data.*

TEST CASE 5: ENSURE PIG VARIABLE CREATION:

To make sure of it we will use the following:

COMMAND:

```
DESCRIBE cylinder_data;
```

This will give us the schema of the PIG variable *cylinder_data*. Ensure all attributes are present. If not them create again.

TEST CASE 6: DATA IMPORT SUCCESSFUL:

TO make sure of the data present use:

COMMAND:

```
dump cylinder_data;
```

This command acts like select query in HIVE, gives us all the data which is present in PIG variable *cylinder_data*.

These were the six primary test cases which are needed to be ensured about our proper import of the datasets. After this there are the commands which are used for the problem statements. For each variable which taken, the best practice will be to describe it once and when dump it to make sure the data integrity.

Now, we shall look into every problem statement’s test:

TEST CASE 7: PROBLEM 1:

COMMAND:

```
order_by_time_stamp = order cylinder_data by time_stamp DESC;
limiting_result= LIMIT order_by 10;
STORE limiting_result INTO '/user/hive/result1';
```

These three commands should give us the first result to the top 10 records recently entered. To ensure its integrity, we will look at the timestamp column. If we see it in descending order or it then it’s a success. Otherwise query failed.

TEST CASE 8: PROBLEM 2:

COMMAND:

```
group_by_solvent = GROUP cylinder_data by solvent_type;
counting_solvents = FOREACH group_by_solvent GENERATE group,
COUNT(cylinder_data.solvent_type);
```

```
order_by_solvents = ORDER counting_solvents by $1 DESC;
limiting_result = LIMIT order_by_solvents 3;
STORE limiting_result INTO '/user/hive/result1';
```

In this set of commands, in every step describe and dump will be a best practice. First check the *group_by_solvent* variable. If it has done its job then proceed to next one, *counting_solvents*. If the generated output consists of the desired structure then proceed to *order_by_solvents*. This should order all solvents by descending order. Next *limiting_result* will limit the result set to only 3. Should it be more than that, then try again. Then at last STORE command which ensures to store the result into HIVE back. Recheck the HIVE folder specified.

TEST CASE 9: PROBLEM 3:

COMMAND:

```
solvents_used = LOAD '/user/hive/Problem_result_2' USING PigStorage('\t')
AS(solvent:chararray, cnt:int);

filtering_cylinder = JOIN cylinder_data by solvent_type, solvents_used by solvent;

filtering_customer = FOREACH filtering_cylinder GENERATE $2,$12;
distincting_result = DISTINCT filtering_customer;
```

Ensure the *solvents_used* variable is not empty after the first command. The *filtering_cylinder* uses JOIN statement with *solvents_used*, so ensure only those data came which are required. *filtering_customer* ensures to generate those columns which contains the customer and solvent type. Lastly *distincting_result* is there for getting distinct results. If the results are not distinct then try again.

TEST CASE 10: PROBLEM 4:

COMMAND:

```
order_by_time_stamp = order cylinder_data by time_stamp DESC;
filtering_data = FILTER order_by_time_stamp by paper_mill_location!='';
limiting_result = LIMIT filtering_data 3;
display_paper_mills = FOREACH limiting_result GENERATE $18;
```

Same procedure should be applied, ensure all the commands are executed properly by describing all the variables. In statements where there are FILTER, LIMIT or FOREACH, pay more attention to the results are up to the expectation or not.

TEST CASE 11: PROBLEM 5:

COMMAND:

group_by_paper = GROUP cylinder_data by paper_type;

counting_paper = FOREACH group_by_paper GENERATE group, COUNT(cylinder_data.paper_type);

order_by_paper = ORDER counting_paper by $1 DESC;

limiting_result = LIMIT order_by_paper 3;

Again ensure all variables executed properly. There are GROUP functions used. Make sure GROUP is made as expected. Also FOREACH ORDER and LIMIT should be double checked.

TEST CASE 12: PROBLEM 6, PROBLEM 7, PROBLEM 8, PROBLEM 9, PROBLEM 10 :

COMMAND:

group_chrome_all = GROUP cylinder_data ALL;

max_chrome = FOREACH group_chrome_all GENERATE cylinder_data.cylinder_number,cylinder_data.customer, MAX(cylinder_data.chrome_content);

All these problems are similar therefore ensure the GROUP function acts accordingly as it is important for the MAX function to be used. If first fails, the second variable will too.

IMPLEMENTATION

In this section we will witness the demonstration of the implementation of the whole concept. For out convenience the implementation is classified into 3 phase. Phase 1 is downloading the data and storing into HIVE. Phase 2 is importing the data to PIG and performing queries. Phase 3 is analysis of the data and designing some graphs.

Let us witness them one by one:

PHASE 1: DOWNLOADING DATA AND IMPORTING TO HIVE:

The following is the link to the dataset:

https://archive.ics.uci.edu/ml/datasets/Cylinder+Bands

Copy all the data in the file paste it using notepad and save it by any suitable name.

After doing that perform the following command:

- *create database rotogravure;*
- *use rotogravure;*
- *create external table cylinder_brands (time_stamp int,cylinder_number string,customer string,job_number int,grain_screened string,int_color string,proof_on_ctd_ink string,blade_mfg string,cylinder_division string,paper_type string,ink_type string,direct_steam string,solvent_type string,type_on_cylinder string,press_type string,press int,unit_number int,cylinder_size string,paper_mill_location string,plating_tank int,proof_cut int,viscosity int,caliper double,ink_temperature double,humifity double,roughness double,blade_pressure int ,varnish_pct double,press_speed int,ink_pct double,solvent_pct double,esa_voltage int ,esa_amperage int ,wax double,hardener double,roller_durometer int,current_density int,anode_space_ratio double,chrome_content int,band_type string) row format delimited fields terminated by ','location '/user/hive/warehouse/rotogravure.db';*
- *LOAD DATA LOCAL inpath '/Rotogravure/cylinder_data.data' OVERWRITE INTO TABLE cylinder_bands;*

These commands wrap up the first phase. Doing this we have successfully created database rotogravure and created a table cylinder_brands.

PHASE 2: IMPORTING FROM HIVE TO PIG AND QUERING:

Perform the following commands for the process:

- *cylinder_data = LOAD '/user/hive/warehouse/rotogravure.db' USING PigStorage(',') AS (time_stamp:int,cylinder_number:chararray,customer:chararray,job_number:int,grain_screened: chararray,int_color:chararray,proof_on_ctd_ink:chararray,blade_mfg: chararray,cylinder_division:chararray,paper_type:chararray,ink_type: chararray,direct_steam :chararray,solvent_type: chararray,type_on_cylinder: chararray,press_type :chararray,press:int,unit_number: int,cylinder_size: chararray,paper_mill_location :chararray,plating_tank: int,proof_cut :int,viscosity: int,caliper:double,ink_temperature: double,humifity: double,roughness: double,blade_pressure :int ,varnish_pct: double,press_speed: int,ink_pct: double,solvent_pct: double,esa_voltage: int,esa_amperage: int,wax:double,hardener: double,roller_durometer: int,current_density: int,anode_space_ratio: double,chrome_content: int,band_type:chararray);*
- *PROBLEM 1: Find the top 10 most recent cylinders:*
  - *order_by_time_stamp = order cylinder_data by time_stamp DESC;*
  - *limiting_result= LIMIT order_by 10;*
  - *STORE limiting_result INTO '/user/hive/result1';*
- *PROBLEM 2: Find the top 3 most used solvents:*
  - *group_by_solvent = GROUP cylinder_data by solvent_type;*
  - *counting_solvents = FOREACH group_by_solvent GENERATE group, COUNT(cylinder_data.solvent_type);*
  - *order_by_solvents = ORDER counting_solvents by $1 DESC;*
  - *limiting_result = LIMIT order_by_solvents 3;*
  - *STORE limiting_result INTO '/user/hive/result1';*
- *PROBLEM 3: Find the customers who uses top most used solvents:*
  - *solvents_used = LOAD '/user/hive/Problem_result_2' USING PigStorage('\t') AS(solvent:chararray, cnt:int);*
  - *filtering_cylinder = JOIN cylinder_data by solvent_type, solvents_used by solvent;*
  - *filtering_customer = FOREACH filtering_cylinder GENERATE $2,$12;*
  - *distincting_result = DISTINCT filtering_customer;*
- *PROBLEM 4: Find the top 3 paper_mill location who ordered cylinders recently:*
  - *order_by_time_stamp = order cylinder_data by time_stamp DESC;*

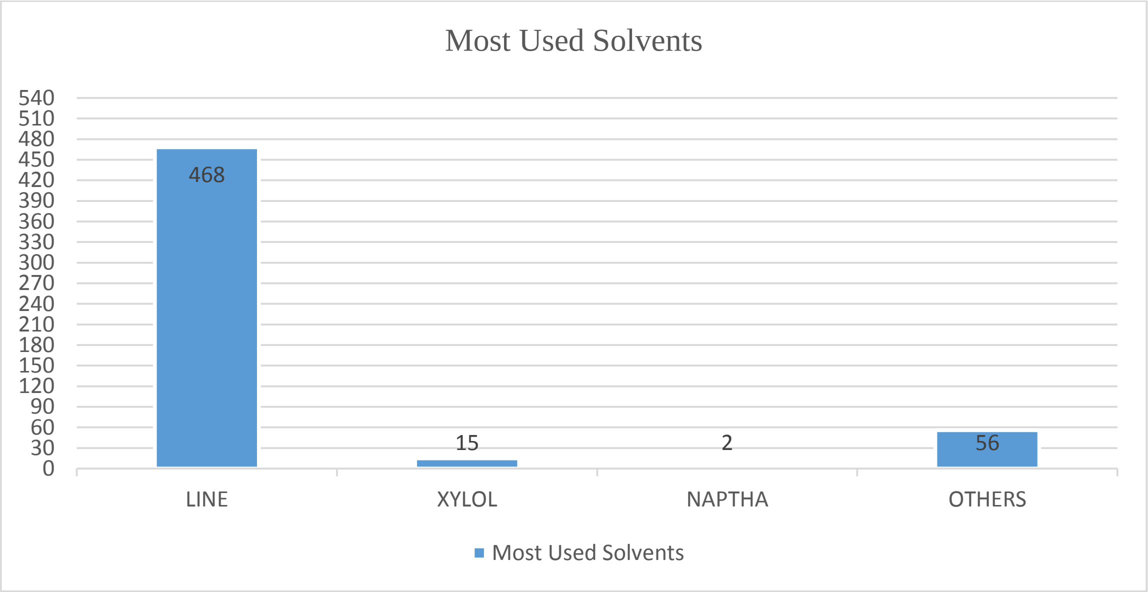
- *filtering_data = FILTER order_by_time_stamp by paper_mill_location!='';*
- *limiting_result = LIMIT filtering_data 3;*
- *display_paper_mills = FOREACH limiting_result GENERATE $18;*

- *PROBLEM 5: Find the top 3 most used paper types:*
  - *group_by_paper = GROUP cylinder_data by paper_type;*
  - *counting_paper = FOREACH group_by_paper GENERATE group, COUNT(cylinder_data.paper_type);*
  - *order_by_paper = ORDER counting_paper by $1 DESC;*
  - *limiting_result = LIMIT order_by_paper 3;*
- *PROBLEM 6: Find the records of the cylinder with max chrome content:*
  - *group_chrome_all = GROUP cylinder_data ALL;*
  - *max_chrome = FOREACH group_chrome_all GENERATE cylinder_data.cylinder_number,cylinder_data.customer, MAX(cylinder_data.chrome_content);*
- *PROBLEM 7: Find the count number of band type as : band and no band:*
  - *group_byBand_type = GROUP cylinder_data by band_type;*
  - *counting_band = FOREACH group_byBand_type GENERATE group,COUNT(cylinder_data.band_type);*
- *PROBLEM 8: Find the max wax numeric:*
  - *group_by_all = GROUP cylinder_data ALL;*
  - *max_wax = FOREACH group_by_all GENERATE cylinder_data.cylinder_number, MAX(cylinder_data.wax);*
- *PROBLEM 9: Find the count ink types:*
  - *group_by_ink = GROUP cylinder_data by ink_type;*
  - *counting_ink = FOREACH group_by_ink GENERATE group, COUNT(cylinder_data.ink_type);*
- *PROBLEM 10: Find the cylinders with max viscosity:*
  - *group_by_all = GROUP cylinder_data ALL;*
  - *max_viscous = FOREACH group_by_all GENERATE cylinder_data.cylinder_number,MAX(cylinder_data.viscosity);*

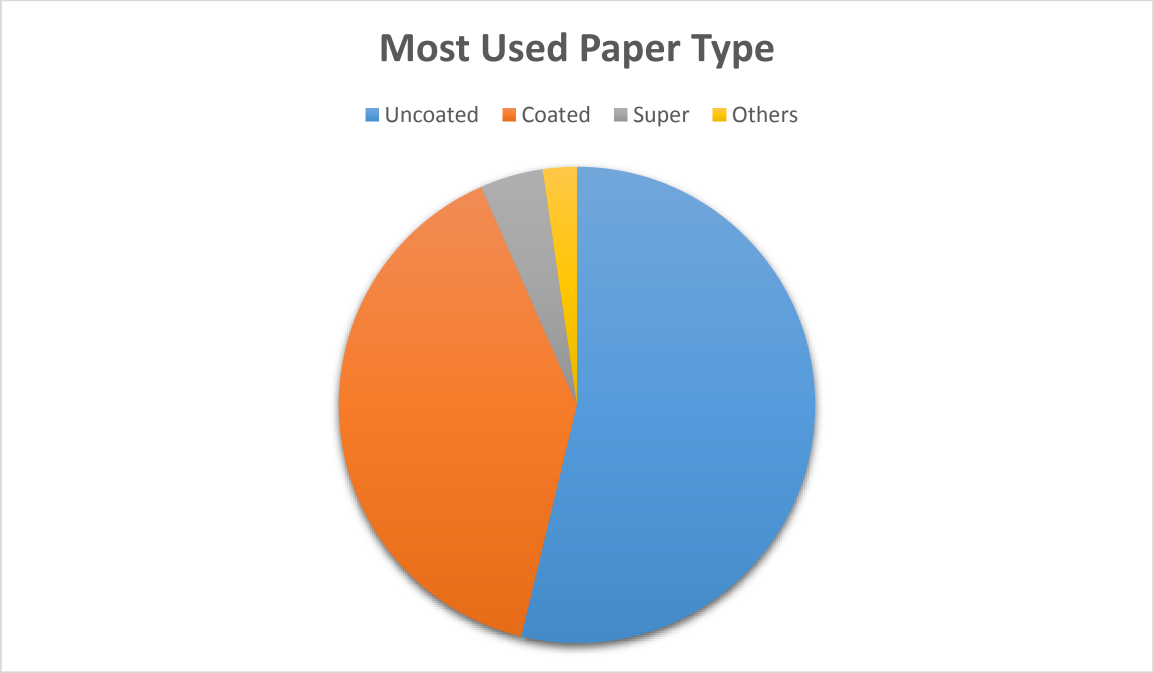
PHASE 3: ANALYSIS OF MEANINGFUL DATA:

Note: Not every data is made into graphs, since there were queries which were some clear result. However those result which were feasible to be in a graphical format are represented as such.

TOP 3 MOST USED SOLVENTS:

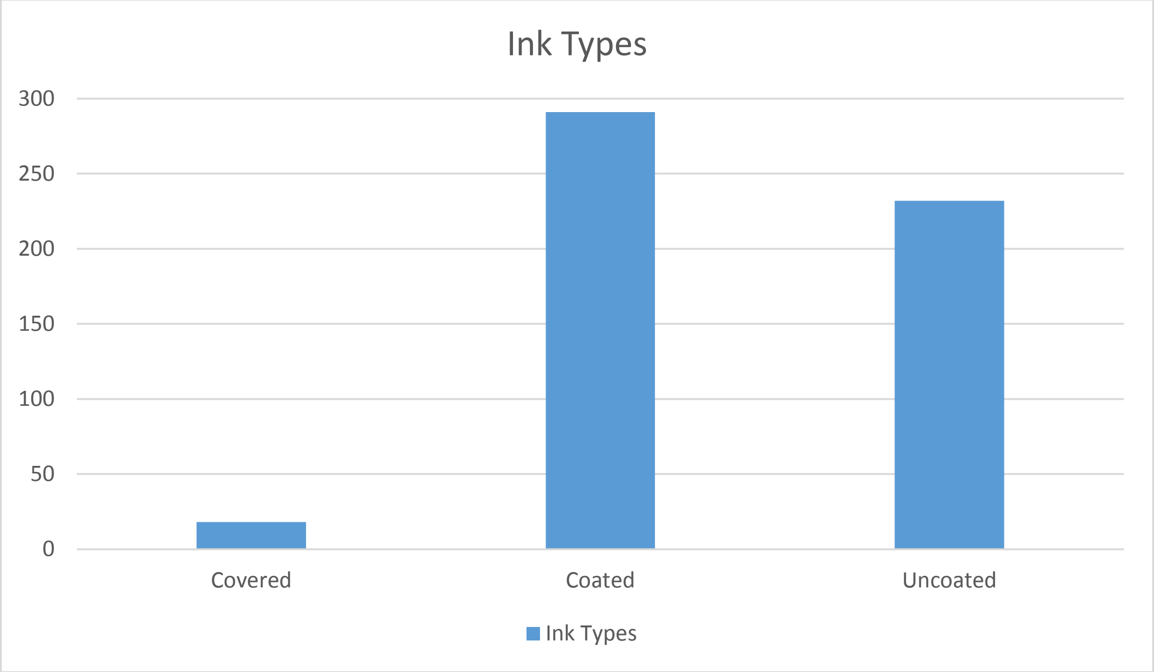


TOP 3 MOST USED PAPER TYPES:



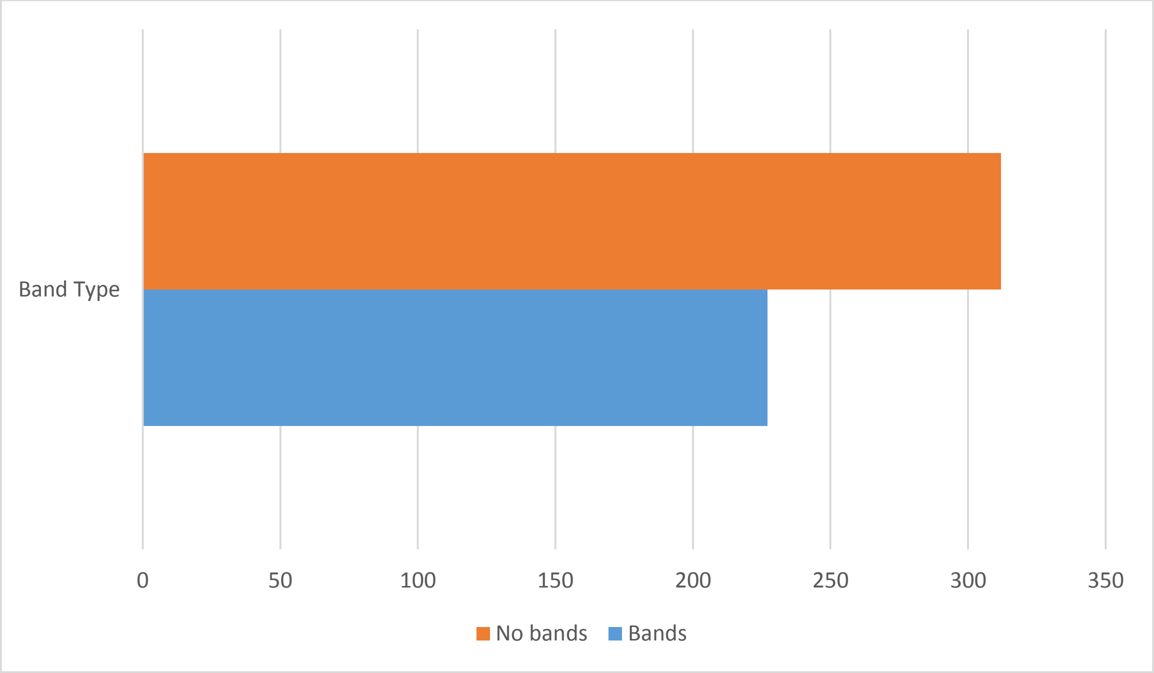
Uncoated: 53.41% Coated: 39.38% Super: 4.26% Others: 2.295%

NUMBER OF INK TYPES:



Covered: 18 Coated: 291 Uncoated: 232

CYLINDERS WITH BANDS AND NO BANDS:



Bands: 227 No Bands: 312

GANTT CHART

The following Chart will tell us how I used this six weeks in my training to learn this technology and develop a proof of concept. There were seven topics according to the syllabus to be covered in this training. First it was the introductory session to big data then familiarizing with Linux platform Cloudera and tools like Sqoop MySQL. Then the tools which are used in big data like HIVE and PIG were covered. After this the project started where Analysis.

SIX WEEK SUMMER TRAINING GANTT CHART

Activities	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
Introduction to Big data and HADOOP.	■					
Linux, Sqoop & MySQL		■	■			
HIVE			■	■		
PIG			■	■	■	■
HBASE				■	■	
Project And analysis					■	
Report & Presentation						■

PROJECT LEGACY

Finally we have come to the end of our report. In this six weeks I have learned a new technology like HADOOP. Explored the ways to tame the big data and minimise the resource hogging. I have successfully completed the proof of concept which I have been working on regarding the Rotogravure cylinders. The data which I produced after processing and analysing will definitely be beneficial for future references to industry or any survey purposes.

Further doing this project I was able to learn tools like HIVE, SQOOP, PIG and HBASE. Each plays roles while taming big data. I’ve learned that the traditional system will not be effective while exploring the big data. Constant hogging and crashed are witnessed when we try to process large datasets with technologies like MySQL or ORACLE. It is then when the framework HADOOP comes into play. The distributed structure of storing file allows it to store the large file in many nodes by breaking down into small segments. Also maximum availability could be ensured by replicating the data in the nodes. Since the storage is distributed therefore the technology which is map reduce, efficient in bringing back the distributed data into one node in an instant. Therefore less time and max performance.

Further, this project made me realise the true meaning of documentation. It is well said any proof of concept or any project without documentation is useless. Since nature of project of a proof of concept should be such that anyone who is new and joins it should be able to understand it. In this proof of concept also I’ve tried to keep up that margin or standard which if any other individual tries to indulge and expand the datasets by analysing it. He or she can easily understand it and carry it further.

The most important lesson which I’ve learned of all is the time management. There is no greater success if you could manage time in an orderly fashion. These six weeks allowed me to push me limits of time management, and I’m really happy to say that yes I was successful in completing the project on time and has enough time to revise it.

BIBLOGRAPHY

- https://archive.ics.uci.edu/ml/datasets/Cylinder+Bands
- https://www.tutorialspoint.com/hadoop/
- https://www.tutorialspoint.com/hive/
- https://www.tutorialspoint.com/apache_pig/
- https://www.tutorialspoint.com/hbase/
- Chuck Lam, Hadoop in Action, Manning Publications, 2010