Big Data Engineering and PySpark Training

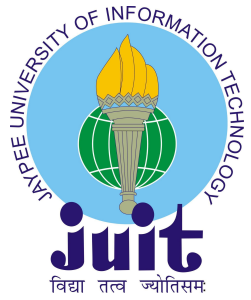
Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Submitted by:
Amolik Vivian Paul (181214)



Jaypee University of Information Technology, Waknaghat
Solan, Himachal Pradesh - 173234

Candidate's Declaration

I hereby declare that the work presented in the report "Big Data Engineering and PySpark Training" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science and Engineering. Jaypee University of Information Technology, Waknaghat is an authentic record of my work from March 2022 to May 2022, under the supervision of Dr. Amol Vasudeva, Assistant Professor, Senior Grade, Computer Science & Engineering

The matter in the report has not been submitted for the award of any other degree or diploma.

Amolik Vivian Paul (181214)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Amol Vasudeva
Assistant Professor (Senior Grade)
Computer Science and Engineering Department

ACKNOWLEDGEMENT

It is a matter of pleasure for me to acknowledge the gratitude towards Jaypee University of Information Technology for giving me an opportunity to explore my abilities during this internship program at Cognizant Technology Solutions. I would like to express my thanks to Mr. Pankaj Kumar, Training and Placement officer, and our faculty Coordinator, Dr. Nafis U Khan.

I also take this opportunity to thank the supervisors at my organization, for valuable guidance and advice during the course of my training. Apart from this the colleagues and mentors at Cognizant Technology Solutions have my deepest regard and gratitude for being supportive all throughout the process of my training, always willing to lend a helping hand whenever needed.

Amolik Vivian Paul
181214
Bachelor of Technology, Computer Science

Table of Contents

1. Introduction

1.1. Cognizant Technology Solutions

Cognizant helps companies modernize **technology**, reimagine **processes** and transform **experiences** so they stay ahead in a fast-changing world. An American multinational technical solutions company, Cognizant has its headquarters in the United States of America with offices in over 20 countries.

Domain	: Information technology consulting company
Company website	: https://www.cognizant.com/
CEO	: Brian Humphries (1 Apr 2019–)
Founded	: 26 January 1994
Headquarters	: Teaneck, New Jersey, United States
Founders	: Kumar Mahadeva, Francisco D'Souza

1.2 Company's Market Presence

Over the years, Cognizant has constantly spread its services in almost every imaginable sector that can accommodate technology as a solution. Few choice sectors are -

1. Automotive
2. Insurance
3. Banking
4. Life Sciences
5. Capital Markets
6. Manufacturing
7. Communications, Media & Technology
8. Oil & Gas
9. Consumer Goods
10. Retail
11. Education
12. Transportation & Logistics
13. Healthcare
14. Travel & Hospitality
15. Information Services
16. Utilities

1.3 Services provided by the Company

1. Application Services
2. Enterprise Application Services
3. Artificial Intelligence
4. Internet of Things
5. Cognizant Infrastructure Services
6. Digital Strategy
7. Cognizant Security
8. Business Process Services
9. Digital Engineering
10. Industry & Platform Solutions

1.4 Cognizant in India

Cognizant India is one of Cognizant Technology Solutions' most notable worldwide distribution centers, and it plays a key role in market outsourcing services, as well as consultation and IT-related solutions.

2. Introduction - Training Programme

Being selected as an Advanced Python Developer as part of the Cognizant GenC Next Hiring Programme, I am being trained in the field of Data Engineering - primarily with Hadoop and PySpark as the main tools and technologies. The Training Programme is a 3 month long programme which aims to let us get familiar with the absolute basics of what is needed to eventually become a data engineer, all the way to learning more advanced technologies with hands-on projects and interactions with experts from the industry on the topics. The programme consists mainly of self-paced and self-learning courses including regular assessments to be taken up which would be evaluated for completion of the programme. The last step of the training programme would be a complete Case Study based on a real life problem to be solved using the tools learnt during the course of the three months - a Big Data Case Study to be solved using the Hadoop Ecosystem and Python - PySpark,

The duration of the programme is 3 months. We were initially divided into cohorts of 30 freshers, each given a Coach from the HR team and a guide who was an SME (Subject Matter Expert). Our primary task for the initial days was to register ourselves for the given courses and get accustomed to the learning portal given by Cognizant.

A detailed description of the week-wise activities and courses is given as follows.

- **Course Summary**

Hadoop Training Summary	
What	Hadoop Training
Why/ Need	To build a pool of Hadoop Developers
Performance Outcome	This Learning plan is to enable participants on Hadoop a by providing the necessary Pre-requisites, Classroom training, Self-paced courses, hands on exercises and assessments. By end of this learning associates should be able to develop with Hadoop.
Performance Objective(High Level)	Introduction To Big Data HDFS MapReduce Apache Hive Architechture Apache Sqoop Impala Nifi Hbase Kafka Spark
Overall Duration	99 hours of training
Pre-requisites	Basics of Big data + Linux+ Java + Python + NoSQL + DW Basics+ Elastic search
S2R Curriculum Duration	58 (self-pcaed , assessment) + ~40 (hands on) hrs

- **Program Plan and Objective**

Program Plan

Detailed Performance Objective	
Apache Hadoop	In this course, you'll learn how to develop Spark applications for your Big Data using Scala and a stable Hadoop distribution, Cloudera CDH and also execute it using a case study. You'll also learn how to use MLLib for Machine Learning and get introduced to Spark Streaming and GraphX.

- **Courses and Learning**

Skill	Activity Code	Activity Name	Activity type	Activities that can be done with Internal trainer (wherever mention Trainer Session)
Hadoop Ecosystem	384458	Hadoop Developer In Real World	Self Paced Learning	Introduction To Big Data HDFS MapReduce Apache Hive Architechture Cluster Setup Hadoop Administrator In Real World (Preview) Troubleshooting and Optimizations Apache Sqoop Kafka
Hbase	1469142	The Complete Apache HBase Developer Course	Self Paced Learning	Module 1 - Introduction to HBase Module 2 - HBase Client API - The Basics Module 3 - Client API: Administrative and Advance Features Module 4 - Available HBase Clients Module 5 - HBase and MapReduce Integration Module 6 - HBase Configuration and Administration
Apache NiFi	1067226	Introduction to Apache NiFi (Cloudera DataFlow - HDF 2.0)	Self Paced Learning	Introduction to NiFi and first concepts Hands On: Getting Started with NiFi Apache NiFi in depth JSON File to MongoDB Integration with Apache Kafka
Spark	892806	Apache Spark with Scala - Hands On with	Self Paced Learning	Module 11 - Scala Crash Course [Optional] Module 12 - Spark Basics and Simple Examples

Skill	Activity Code	Activity Name	Activity type	Activities that can be done with Internal trainer (wherever mention Trainer Session)
Hadoop	ATKDW162	Hadoop KBA [201-Intermediate]	KBA	Assessment
Hadoop	ATHDW098349	Hadoop Case Study S2D [201-Intermediate]	Assessment	Case study based Assessment

2.1 Big Data

Big Data is a set of records that is big in volume, but developing exponentially with time. It is a record with such huge length and complexity that none of conventional records control equipment can save it efficiently. Big records are likewise records however with big length. Put simply, Big Data is larger, greater complicated information units, specifically from new information sources. These information units are so voluminous that conventional information processing software programs simply can't manipulate them. But those huge volumes of information may be used to deal with commercial enterprise troubles you wouldn't be capable of addressing before.

Some examples to help us understand better about what Big Data looks like are,

- The **New York Stock Exchange** is one such example which creates approximately **one terabyte** in trade data every 24 hours.

- **Social Media.** Statistics show that ***500+ terabytes*** of data is added to databases of social media services every day, mainly generated in terms of photo and video uploads, messages, posts.

Since there are so many sources of Big Data collection, it is bound to be classified based on its way of collection and how it looks. Loosely, Big Data can be differentiated into three types of structuring,

1. Structured Data
2. Unstructured Data
3. Semi Structured Data

1. Structured Data

Any data that can be stored in the form of a fixed format is termed as a ‘structured’ data. It has been organized into a formatted structure, usually a database, so that the data can be made more easily accessible for fetching and analysis.

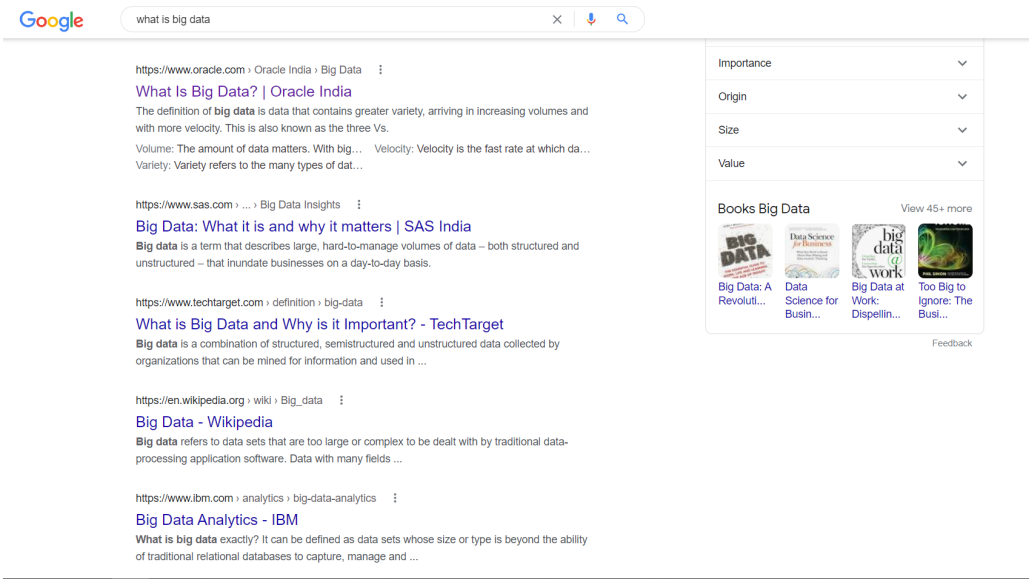
An ‘Employee’ table in a database is an example of Structured Data

Employee_ID	Employee_Name	Gender	Department	Salary_In_lacs
2365	Rajesh Kulkarni	Male	Finance	650000
3398	Pratibha Joshi	Female	Admin	650000
7465	Shushil Roy	Male	Admin	500000
7500	Shubhojit Das	Male	Finance	500000
7699	Priya Sane	Female	Finance	550000

2. Unstructured Data

Any records with unknown shape or the shape is classed as unstructured records. In addition to the scale being huge, un-dependent records pose a couple of demanding situations in phrases of its processing for deriving price out of it. An ordinary instance of unstructured records is a heterogeneous records supply containing an aggregate of textual content files, images, videos.

Google Search, example of unstructured



3. Semi Structured Data

Semi-structured data is a type of record structure that doesn't obey the tabular shape of records related to relational databases or different kinds of records tables, however it consists of tags split semantic factors and put into effect hierarchies of facts and fields in the records.

One of the more explainable examples of Semi Structured Data is an XML file used mainly for config files in softwares and applications.

example.xml

```
<rec><name>Amolik Paul</name><sex>Male</sex><age>22</age></rec>

<rec><name>John Doe</name><sex>Male</sex><age>41</age></rec>

<rec><name>Bill Gates</name><sex>Male</sex><age>58</age></rec>
```

Characteristics of Big Data

Any data can be classified as Big Data based on 3 primary properties. If the data fulfills the threshold values of the 3 properties, it can be called Big Data. They are,

- Volume
- Variety
- Velocity

Volume – The name Big Data itself is associated with a size that is enormous. Size of statistics performs a completely important function in figuring out price out of statistics. Also, whether or not a selected statistics can really be taken into consideration as a Big Data or not, depends upon the quantity of statistics.

Variety – The subsequent element of Big Data is its variety.
Variety refers to heterogeneous assets and the character of information, each based and unstructured. During advanced days, spreadsheets and databases have been the handiest assets of information taken into consideration by means of maximum applications. Nowadays, information withinside the shape of emails, photos, videos, tracking devices, PDFs, audio, etc. also are being taken into consideration withinside the evaluation applications. This form of unstructured information poses positive problems for storage, mining and reading information.

Velocity – The pace at which data is created is referred to as “velocity”.

Applications of Big Data

The current world, as we know it, is majorly being run on information being sourced from Big Data. Applications, Softwares, Social Media are all mostly powered by data stored as Big Data. Some examples of often seen applications of Big Data are -

- Smarter Healthcare: The companies can extract relevant information from the petabytes of patient data and then design programmes that can forecast the patient's worsening state in advance.

- Telecom: Telecom industries gather data, evaluate it, and offer solutions to a variety of issues. Telecom firms have been able to drastically minimise data packet loss, which occurs when networks are overcrowded, and so provide a flawless connection to their clients, by utilising Big Data applications.

- Retail: One of the biggest advantages of big data is retail, which has some of the narrowest margins. Understanding consumer behavior is the beauty of leveraging big data in retail.

- Traffic control: Congestion is a big issue in many places throughout the world. As cities grow more densely populated, effective use of data and sensors will be critical to better traffic management.

- Manufacturing: Analyzing big data in the manufacturing industry can reduce component defects, improve product quality, increase efficiency, and save time and money.

- Search Quality: On extracting statistics from google, we're concurrently producing records for it. Google stores this data and makes use of it to enhance its search quality.

Challenges with Big Data

1. Data Quality –

The fourth V, or veracity, is the issue here. The information is disorganized, inconsistent, and incomplete.

2. Discovery –

It's challenging to identify patterns and insights in petabytes of data using incredibly complex and powerful algorithms.

3. Storage –

The more data a company collects, the more difficult it is to manage it. We want a storage solution that can scale up and down on demand.

4. Analytics

In the case of Big Data, most of the time we are unaware of the kind of data we are dealing with, so analyzing that data is even more difficult.

5. Security

Because the data is so large, keeping it safe is also a difficulty. It comprises user identification, user-based access restrictions, data access histories, and effective data encryption, among other things.

3. Tools and Technologies

3.1 Hadoop

Before discussing the need for the Hadoop ecosystem, and why it has become synonymous with Big Data, we need to address the problems that come with traditional approaches of solving the same. From the past we have seen the major issue was how heterogeneous the data is when it comes to Big Data - both in format and in structure. Traditional Relational Database Management Systems rely on structured data, which follow a fixed format. However, Hadoop has the freedom of organizing and managing all kinds of data be it - structured, semi-structured, or data in form of text, audio, vidoe.

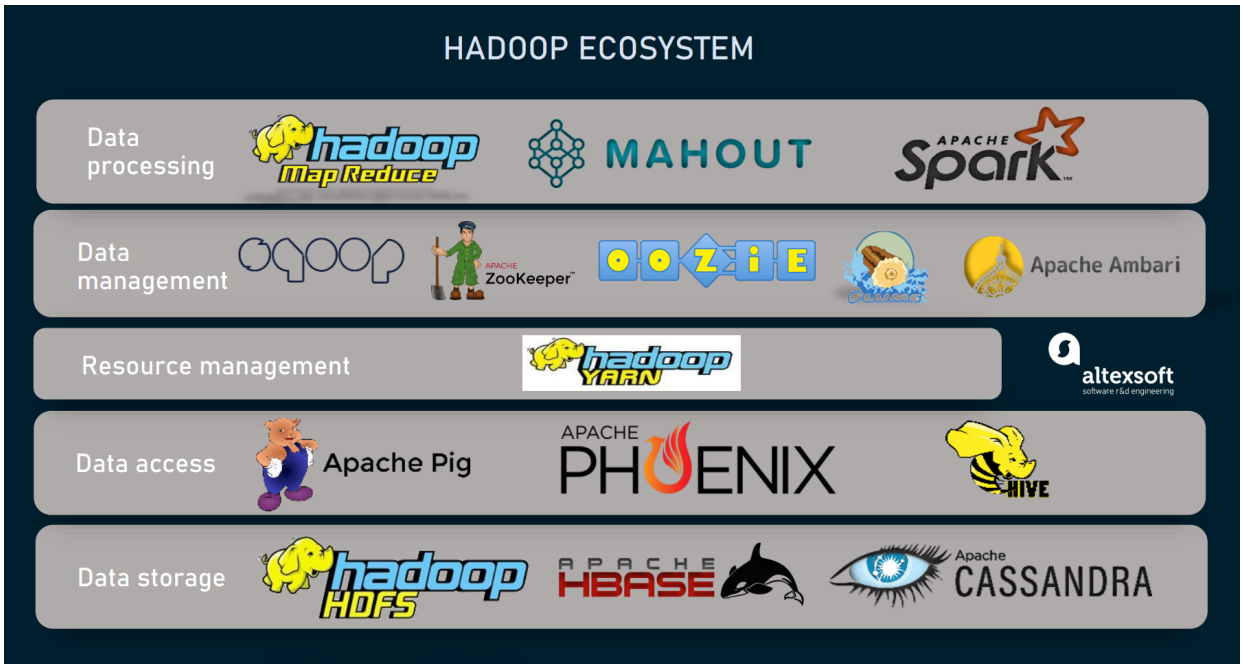
- Storing colossal amounts of data.
- Storing heterogeneous data.
- Accessing and processing speed for large amounts of data.

Having discussed the issues that come with Big Data processing with the more traditional methods, we can take a look at what Hadoop is and what it achieves with Big Data processing.

Hadoop is a framework that allows storing Big Data in a distributed environment, to facilitate parallel processing. The two main components in Hadoop are,

- *HDFS* which stands for “Hadoop Distributed File System”. HDFS allows us to store data across all formats in clusters.
- YARN, manages all resource management and allows parallel processing over the data.

Along with the two main components that we will discuss in detail, the most powerful part of Hadoop is its extensive ecosystem of multiple services performing different tasks for solving big data problems.



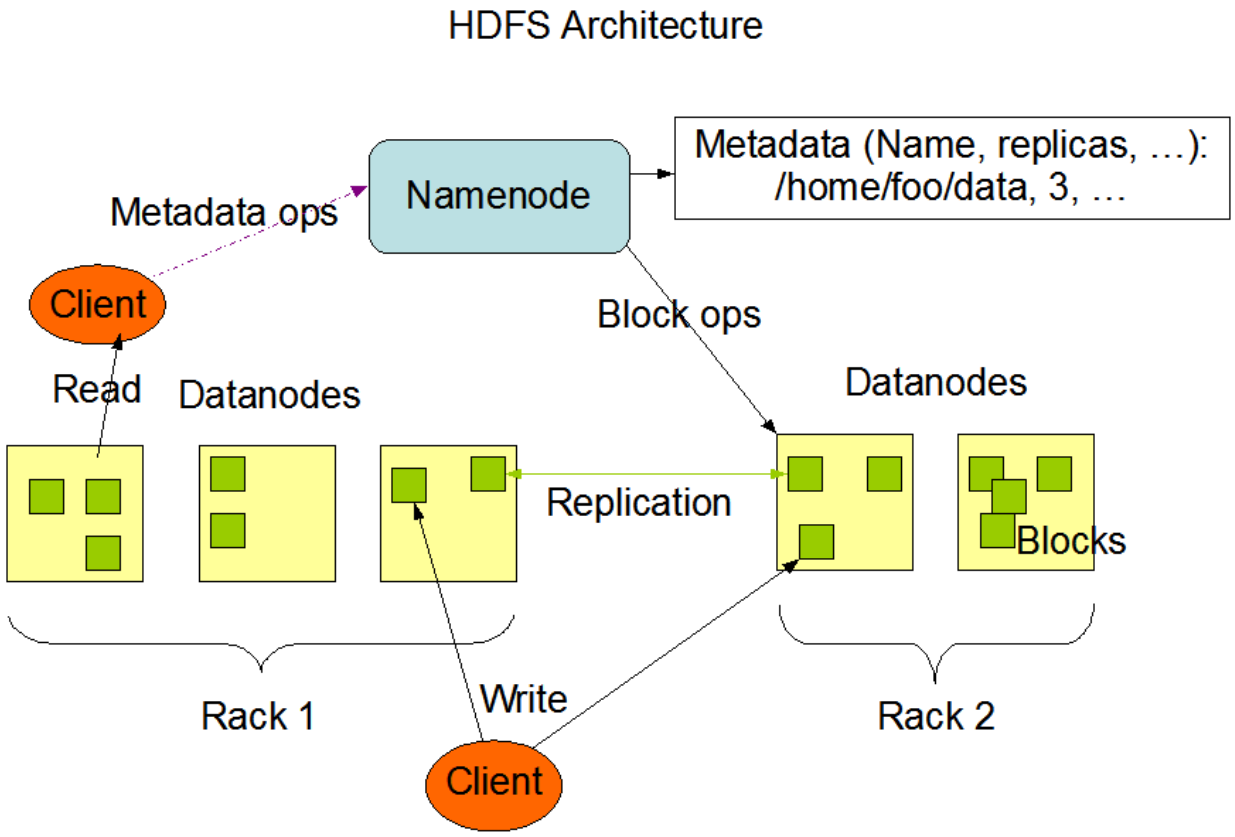
Distributed File System

Distributed File System is a method of managing and storing data across multiple servers, as a distributed network of storage and processing devices. DFS lets us store data over multiple nodes. This also allows multiple access points, to process and use the data parallely.

Despite the fact that the files are stored over a network, any Distributed File System manages data in such a way that a single user feels as though all of the data is being accessed from a single system.

What is HDFS?

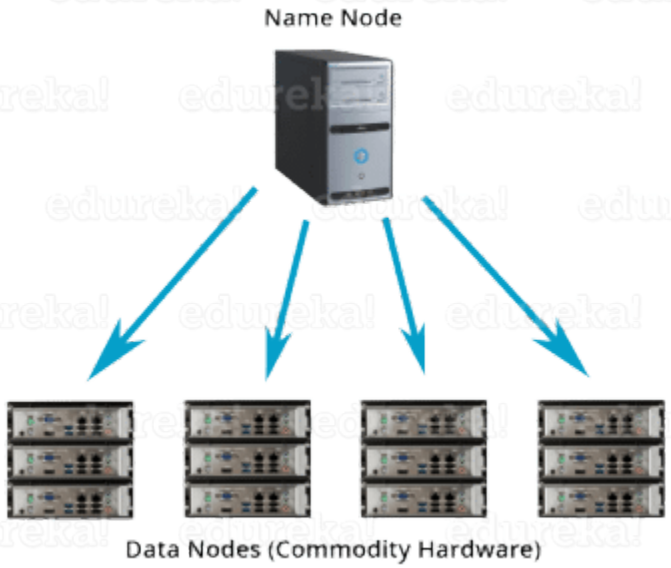
It is a Java based DFS (Distributed File System) that allows us to store large amounts of data across multiple servers, called nodes in a collective environment called a Hadoop cluster.



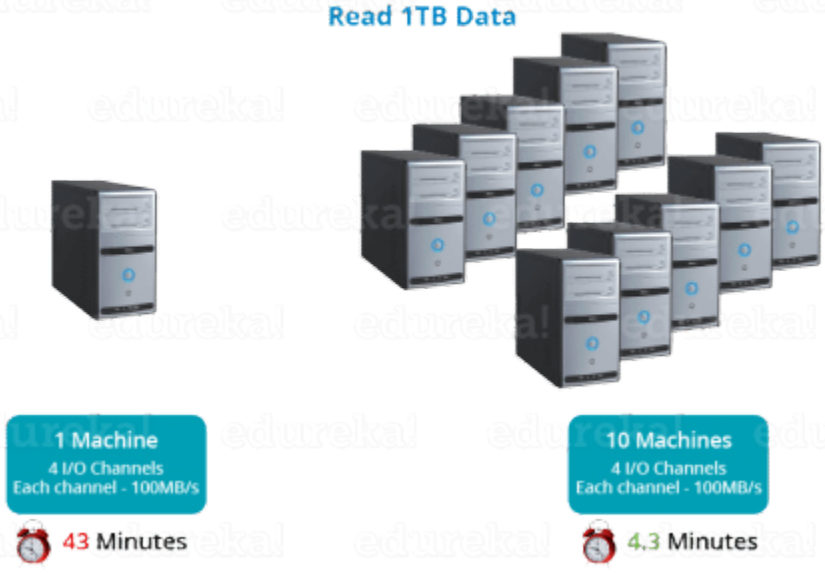
Advantages of HDFS

1. Distributed Storage

HDFS allows to store large files across multiple nodes or systems; however if you access the said large file, even if its data is spread across multiple systems, it organizes and displays in a way as if it is on one system.Therefore, it is not limited to any physical system.



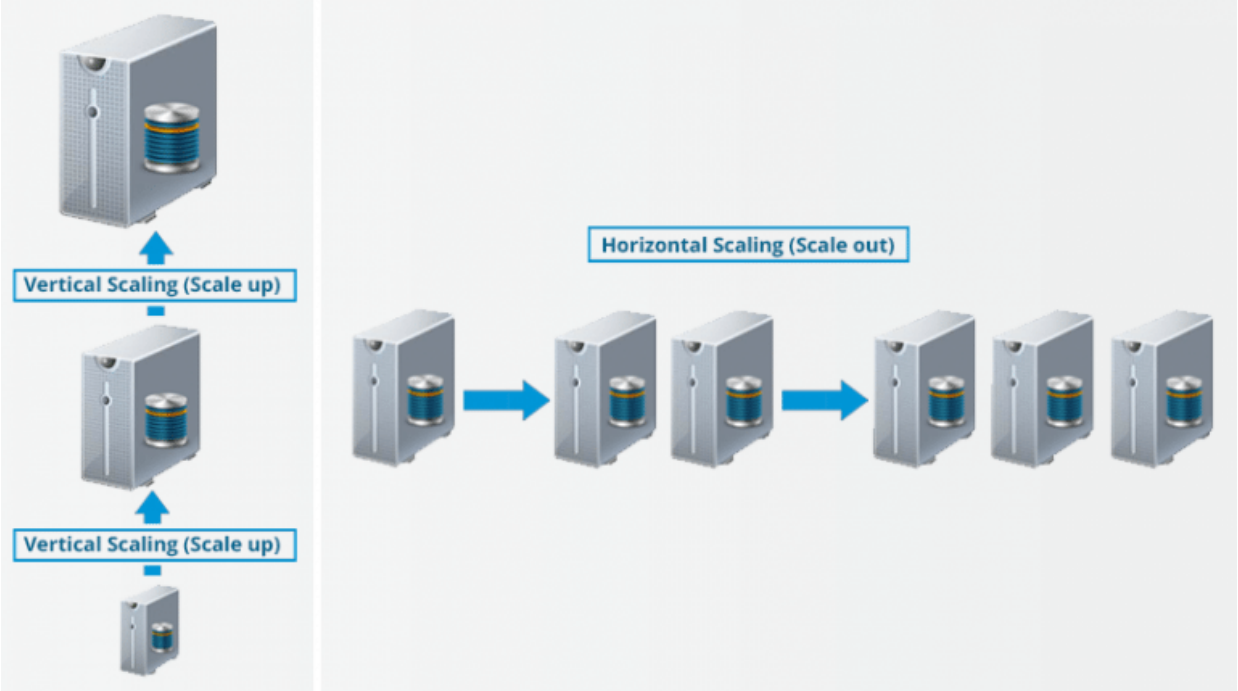
2. Distributed & Parallel Computation



We can use Distributed and Parallel Computation since the data is distributed among the computers. Since each system works independently, we can perform similar or different tasks on the same data parallely without waiting.

3. Horizontal Scaling

Vertical and horizontal scaling are the two forms of scaling. You enhance the hardware capacity of your system through vertical scaling (scale up). To put it another way, you buy additional RAM or a faster CPU to make it more stable and powerful.



There are also problems that occur during scaling up, which are discussed further in this report.

- The limit to how much hardware capacity you can add to systems. As a result, you won't be able to keep upgrading the machine's RAM or CPU.

- When vertical scaling, you must first stop your machine. The RAM or CPU is then increased to make the hardware stack more resilient. You restart the system after increasing the hardware capacity. This downtime while your system is shutting down becomes an issue.

Instead than expanding the hardware capacity of individual computers, horizontal scaling involves adding more nodes to existing clusters. Most crucially, you can add new machines while the system is running, i.e. without interrupting it. As a result, there is no downtime, no green zone, nothing like that while scaling out.

Features of Hadoop Distributed File System (HDFS)

- Cost: In general, HDFS is installed on common hardware. As a result, it is highly cost-effective in terms of project ownership. You won't have to spend a lot of money to scale out your Hadoop cluster because we're employing low-cost commodity hardware.

- Variety and Volume: HDFS has storing massive amounts of data, such as terabytes and petabytes, as well as many types of data. As a result, HDFS can store any structured, unstructured, or semistructured type of data.

- Reliability and Fault Tolerance: HDFS separates the provided data into “blocks” and stores it throughout your Hadoop cluster in a distributed form. The metadata contains information about the location of the block. The NameNode takes care of all meta types of data, while the DataNodes are in charge of the storage task. The data is also replicated by the name node, which keeps numerous copies of the data. HDFS is incredibly dependable and fault tolerant because of the data replication. As a result, even if a node fails, the data may be retrieved via copies on other data nodes.

- Data Integrity: Data Integrity refers to whether or not the data in the HDFS is accurate. The integrity of data saved in HDFS is regularly checked for its "checksum". On discovering a problem, information is passed to the name node. The name node then produces more duplicates and deletes copies which have been corrupted.

- High Throughput: The amount of work completed in a given length of time is known as throughput. It refers to the speed with which data from the file system may be accessed. Essentially, it provides information regarding the system's performance. By processing data in parallel, we were able to drastically reduce processing time and achieve high throughput.

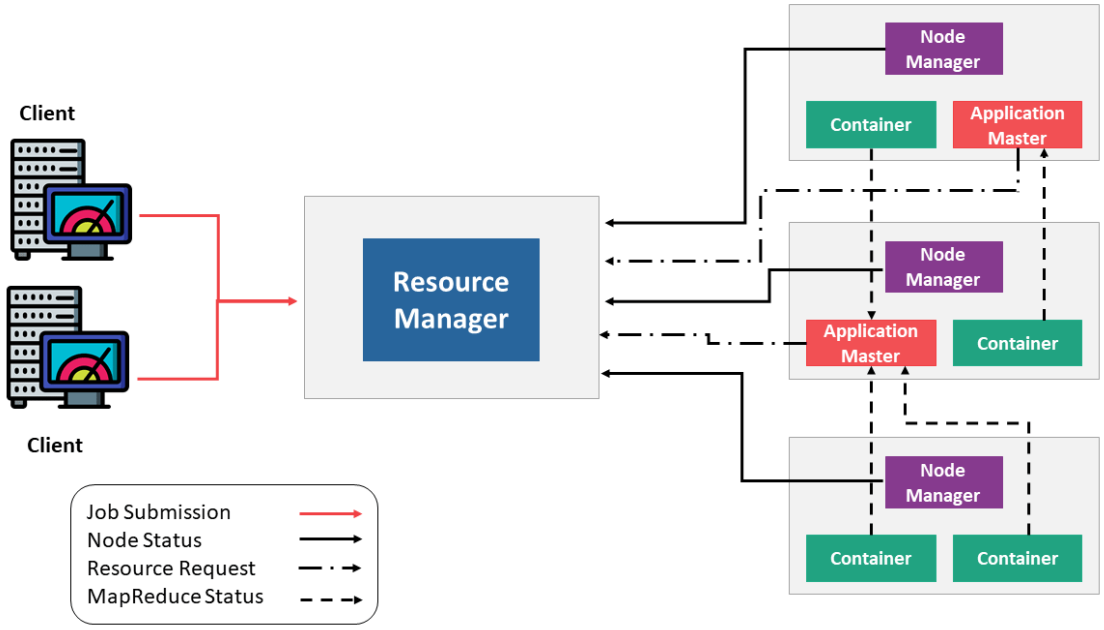
- Data Locality: Data locality or localization refers to the movement of processing units toward data rather than data toward processing units. With HDFS, we move the processes to the location of the data node. As a result, instead of moving the data to the points where it is being processed, Hadoop enables us to bring the programme or processing element to the data.

Yarn - Hadoop's Resource Management

YARN was created to allocate resources and task scheduling functions into daemons. For each Application, YARN consists of a global Resource Manager (RM) and an Application Master. An application is a single job or a collection of jobs.

YARN consists of the ResourceManager and NodeManager. The Resource Manager is responsible for assigning resources to all applications in a system.

The NodeManager is more of a per-machine framework that is in charge of monitoring and reporting on utilization of resources such as CPU, Memory etc. to the said Resource Manager.



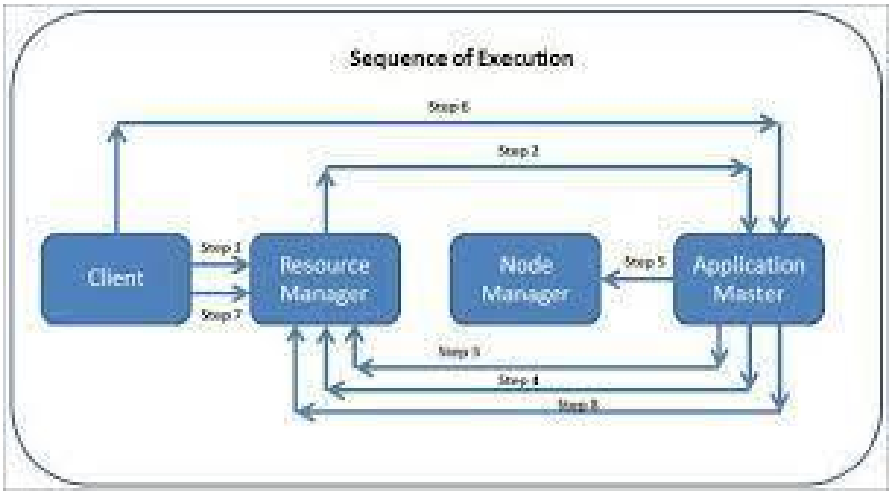
Resource Manager

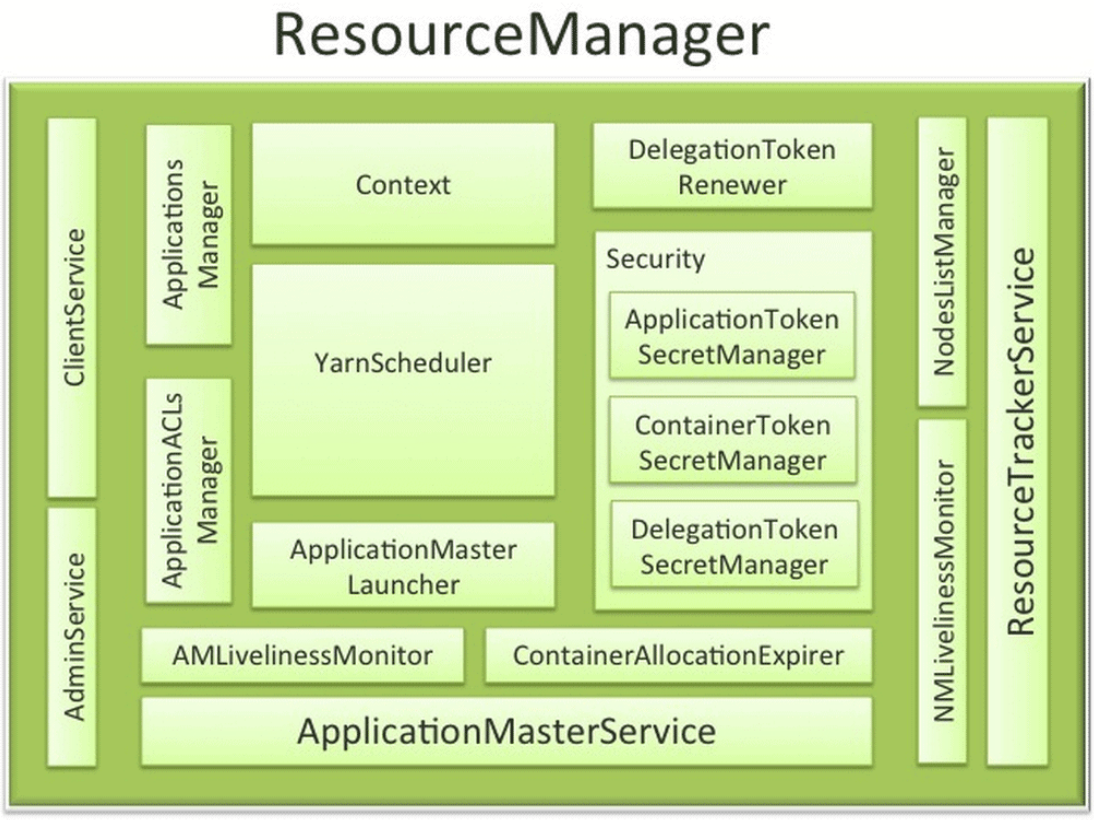
The ResourceManager has two main components:

Scheduler and ApplicationsManager

The Scheduler is in charge of assigning resources to the many operating programmes while keeping in mind the usual restrictions of capacity, queues, and so on. The Scheduler is a pure scheduler that does not monitor application's state.

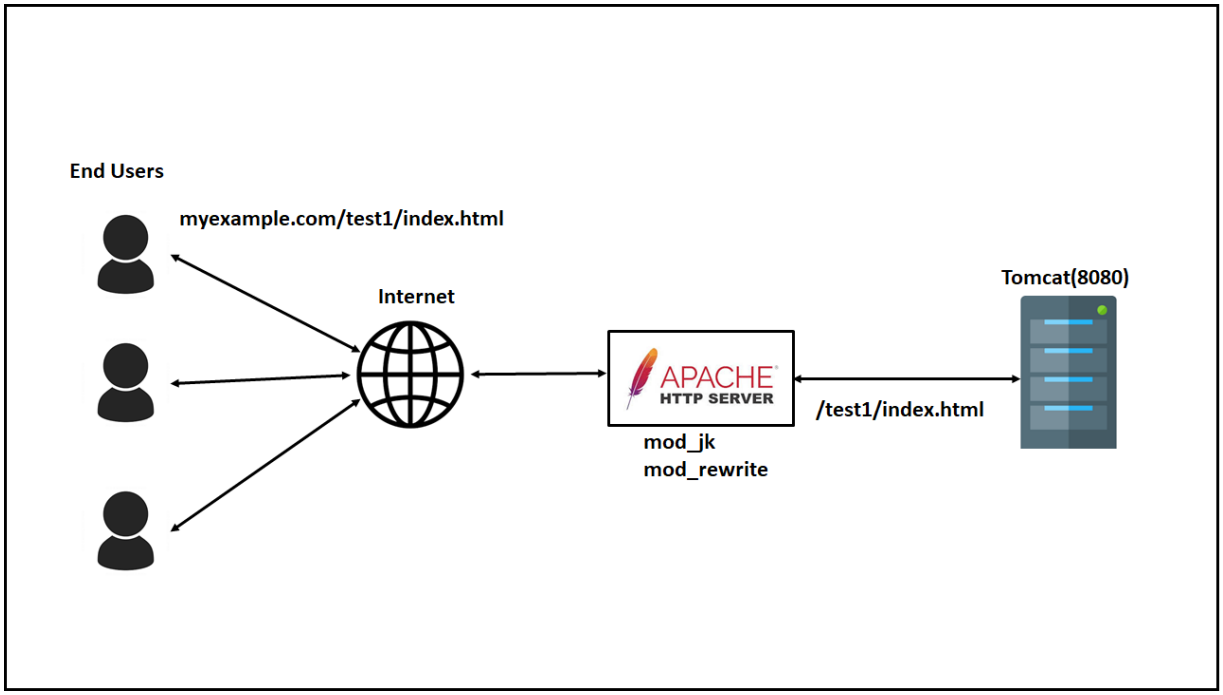
The programme schedules are based on the requirements of the process; it does so using the abstract concept of a resource Container, which includes elements such as memory, CPU, disc, network, and so on.




ResourceManager
ClientService
AdminService
Applications Manager
ApplicationACLs Manager
Context
YarnScheduler
ApplicationMaster Launcher
DelegationToken Renewer
Security
ApplicationToken SecretManager
ContainerToken SecretManager
DelegationToken SecretManager
AMLivelinessMonitor
ContainerAllocationExpirer
ApplicationMasterService
NodesListManager
NMLivelinessMonitor
ResourceTrackerService

3.2 Apache

The Apache HTTP Server is to develop and maintain HTTP servers for UNIX and Windows. Apache provides a “secure” and “extensible” server that provides HTTP services.



Apache web servers are fast, dependable, and secure and easy to configure. Hence, a lot of best-in-class firms frequently use Apache to set up their servers specially for Big Data applications.

3.3 Apache Nifi

The Apache Software Foundation's Apache NiFi is a software that allows automation of flow of data across software systems using a web interface.

The features of Apache Nifi that make it a go to tool are -

- Web based UI
- Configurable
    - Low latency, high throughput
    - Runtime modification
    - Back pressure
- Data Provenance
    - Tracking data from start to end
- Designed for extension
    - Build your own processors
    - Rapid development and testing
- Secure
    - Multi-tenant authorization and internal authorization.

3.4 Python

Python is a scripting language designed to write and execute code across multiple domains be it web, data analysis, data scraping, or even Big Data pipelines with the help of multiple open source third party libraries as a general purpose language.

Python enables easy coding, development and quick prototyping which is why it is growing in popularity among developers even in the Big Data domain. Further, this report discusses the use of PySpark, a Python library built on the API of Spark to make developing Spark applications easier for developers with a background in Python Development. It allows us to use concepts of OOPs and functional programming of Python merged with the API of Spark thus giving us a lot of control and customization.

3.5 PySpark

PySpark is the Python API for Apache Spark. Spark was originally written in Scala, however since Scala was not intuitive enough and with the growing popularity of Python, PySpark was developed as a wrapper for all things Spark to be used with Python.

Data Science and Analysts have a fair knowledge of libraries like Pandas, and the structure and code of PySpark was written to provide ease to them. The functions, methods are very similar and intuitive in their nomenclature which makes for an easy learning curve and efficient development.

3.6 SQL

Structured Query Language or SQL is a query language for working with data - manipulating, retrieving and updating. The Structured part refers to the way data is stored in the database, in RDBMS format.

SQL is the standard language for Relational Database Systems. MySQL, Oracle, MS Access, Postgres utilize it as the standard language to retrieve, manipulate and create data.

Data Definition Language

These commands are used to define the database of a system. They deal with the descriptions of the schemas of the database - creating them and modifying them. They are mostly used on the admin end as they have a lot of control over how your database shapes up.

- **CREATE**: Used to create the database or its objects.
- **DROP**: Used to delete objects from the database.
- **COMMENT**: Add comments to the data.
- **TRUNCATE:** Remove all records from a table
- **RENAME:** Rename an object existing in the database.
- **ALTER:** Alter the structure of the database.

Data Query Language

DQL statements are used to query the data contained in the database. SQL statements that fetch data from a database and put it in order are called DQL statements.

Data Control Language

DCL commands are GRANT and REVOKE which primarily handle rights and permissions of a database system.

Data Manipulation Language

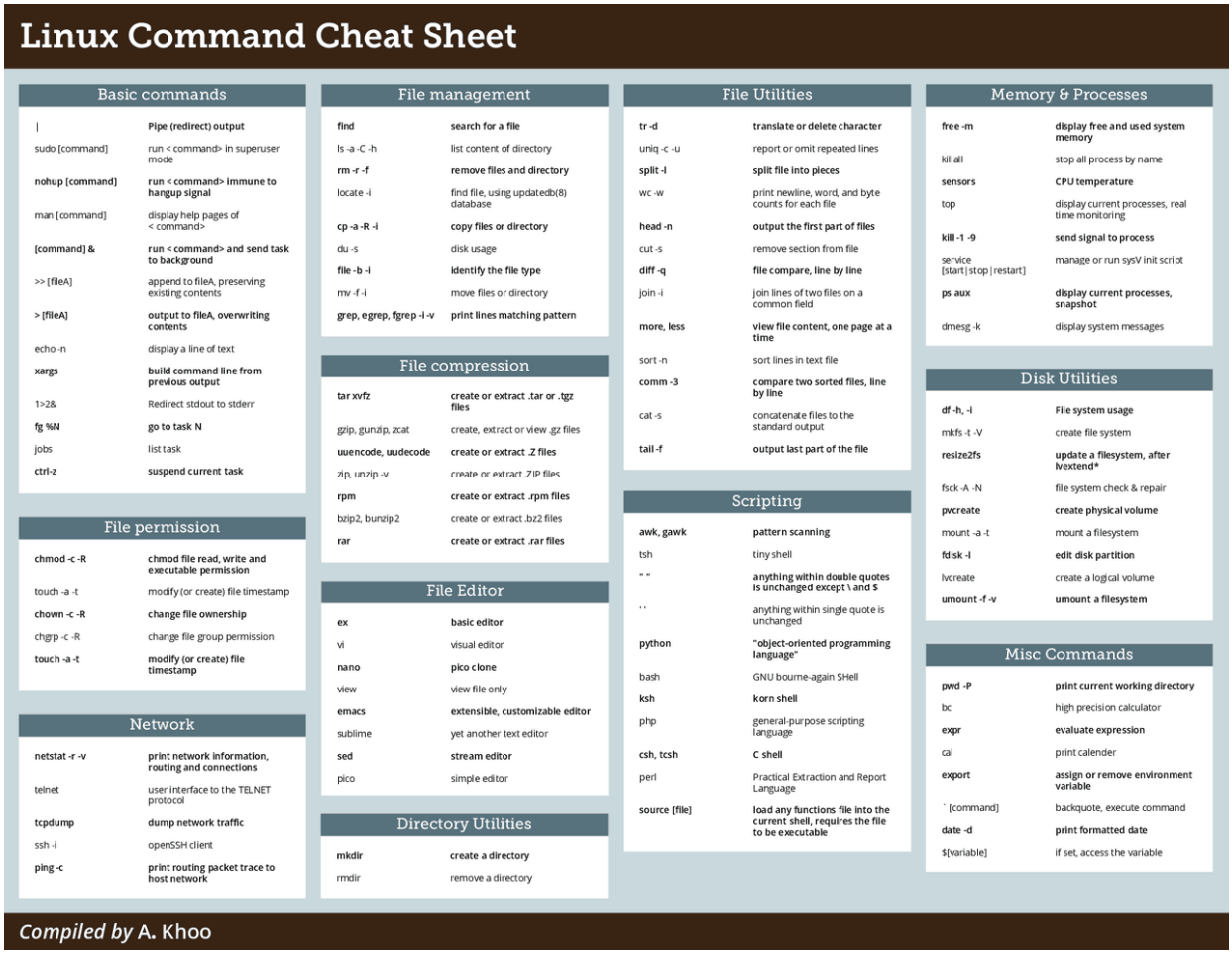
These commands are responsible for manipulating data in the database and monitors and changes the access to the database.

List of DML commands:

- INSERT : Used to insert data into a table.
- UPDATE: Used to update existing data within a table.
- DELETE : Used to delete records from a database table.
- LOCK: Table control concurrency.

3.7 Bash Scripting

Bash Scripting was another important part of our training since all Hadoop operations are accessed through the bash terminal.



Linux Command Cheat Sheet

Basic commands

|	**Pipe (redirect) output**
sudo [command]	run < command> in superuser mode
nohup [command]	**run < command> immune to hangup signal**
man [command]	display help pages of < command>
[command] &	**run < command> and send task to background**
>> [fileA]	append to fileA, preserving existing contents
> [fileA]	**output to fileA, overwriting contents**
echo -n	display a line of text
xargs	**build command line from previous output**
1>2&	Redirect stdout to stderr
fg %N	**go to task N**
jobs	list task
ctrl-z	**suspend current task**

File permission

chmod -c -R	**chmod file read, write and executable permission**
touch -a -t	modify (or create) file timestamp
chown -c -R	**change file ownership**
chgrp -c -R	change file group permission
touch -a -t	**modify (or create) file timestamp**

Network

netstat -r -v	**print network information, routing and connections**
telnet	user interface to the TELNET protocol
tcpdump	**dump network traffic**
ssh -i	openSSH client
ping -c	**print routing packet trace to host network**

File management

find	**search for a file**
ls -a -C -h	list content of directory
rm -r -f	**remove files and directory**
locate -i	find file, using updatedb(8) database
cp -a -R -i	**copy files or directory**
du -s	disk usage
file -b -i	**identify the file type**
mv -f -i	move files or directory
grep, egrep, fgrep -i -v	**print lines matching pattern**

File compression

tar xvfz	**create or extract .tar or .tgz files**
gzip, gunzip, zcat	create, extract or view .gz files
uuencode, uudecode	**create or extract .Z files**
zip, unzip -v	create or extract .ZIP files
rpm	**create or extract .rpm files**
bzip2, bunzip2	create or extract .bz2 files
rar	**create or extract .rar files**

File Editor

ex	**basic editor**
vi	visual editor
nano	**pico clone**
view	view file only
emacs	**extensible, customizable editor**
sublime	yet another text editor
sed	**stream editor**
pico	simple editor

Directory Utilities

mkdir	**create a directory**
rmdir	remove a directory

File Utilities

tr -d	**translate or delete character**
uniq -c -u	report or omit repeated lines
split -l	**split file into pieces**
wc -w	print newline, word, and byte counts for each file
head -n	**output the first part of files**
cut -s	remove section from file
diff -q	**file compare, line by line**
join -i	join lines of two files on a common field
more, less	**view file content, one page at a time**
sort -n	sort lines in text file
comm -3	**compare two sorted files, line by line**
cat -s	concatenate files to the standard output
tail -f	**output last part of the file**

Scripting

awk, gawk	**pattern scanning**
tsh	tiny shell
" "	**anything within double quotes is unchanged except \ and $**
' '	anything within single quote is unchanged
python	**"object-oriented programming language"**
bash	GNU bourne-again SHell
ksh	**korn shell**
php	general-purpose scripting language
csh, tcsh	**C shell**
perl	Practical Extraction and Report Language
source [file]	**load any functions file into the current shell, requires the file to be executable**

Memory & Processes

free -m	**display free and used system memory**
killall	stop all process by name
sensors	**CPU temperature**
top	display current processes, real time monitoring
kill -1 -9	**send signal to process**
service [start|stop|restart]	manage or run sysV init script
ps aux	**display current processes, snapshot**
dmesg -k	display system messages

Disk Utilities

df -h, -i	**File system usage**
mkfs -t -V	create file system
resize2fs	**update a filesystem, after lvextend***
fsck -A -N	file system check & repair
pvcreate	**create physical volume**
mount -a -t	mount a filesystem
fdisk -l	**edit disk partition**
lvcreate	create a logical volume
umount -f -v	**umount a filesystem**

Misc Commands

pwd -P	**print current working directory**
bc	high precision calculator
expr	**evaluate expression**
cal	print calender
export	**assign or remove environment variable**
` [command]	backquote, execute command
date -d	**print formatted date**
$[variable]	if set, access the variable

Compiled by A. Khoo

4. Conclusion

4.1 Conclusion

The training at Cognizant has been an enriching experience, it has exposed me to learn about one of the most increasing trends in the world right now which is Big Data and specific to Data Engineering. Over the course of the last three months I have got comfortable with understanding business problems when it comes to large datasets, and how to prepare them. Creating pipelines with various technologies and tools like Spark, and working with softwares like Apache, Nifi, Hadoop, PySpark, PIG etc.

I look forward to working more with Cognizant and developing Big Data Solutions and contributing to projects as soon as I am done with my training.