大數據分析

How **Big** is big?

1. 什麼是 **Big Data Analytics**?

什麼是 **Analytics**?

What is **Data Scientist**?

What is Big Data?

- IDC estimated the size of the “digital universe” at 0.18 zettabytes in 2006.
  - **1 ZB (zettabytes)** = 10^3 EB (exabytes)
    = 10^6 PB (petabytes)
    = 10^9 TB (terabytes)
    = **10^{12} GB** (gigabytes)
    = **10^{21} Bytes**

What is Big Data?

- Big Data 產業的演進背景：
  - 1990 ~ : **Internet**
  - 2000 ~ : **Ubiquitous Computing**
    - **Context Awareness, Parallel Processing, Grid Computing**
  - 2008 ~ : **Mobile Computing**
    - **Apps on Mobile Devices**
  - 2009 ~ : **Cloud Computing**
    - **IaaS, PaaS, SaaS, DCaaS**
  - 2012 ~ : **IoT (Internet of Things) : IPv6 128 bits**
    - → **2^{128} devices**

What is Big Data? (cont'd)

- **Data Structures — Large and complex datasets**
  - **structured**
  - **semi-structured**
  - **unstructured**

- **3Vs Model of Big Data**
  - **Velocity**
  - **Volume**
  - **Variety**

What is Big Data? Data Structures (cont'd)

Structured data

— data that is organized into entities that have a defined format, such as XML documents or database tables that *conform to a particular predefined schema*.

— This is the realm of the RDBMS (關聯式資料庫管理系統).

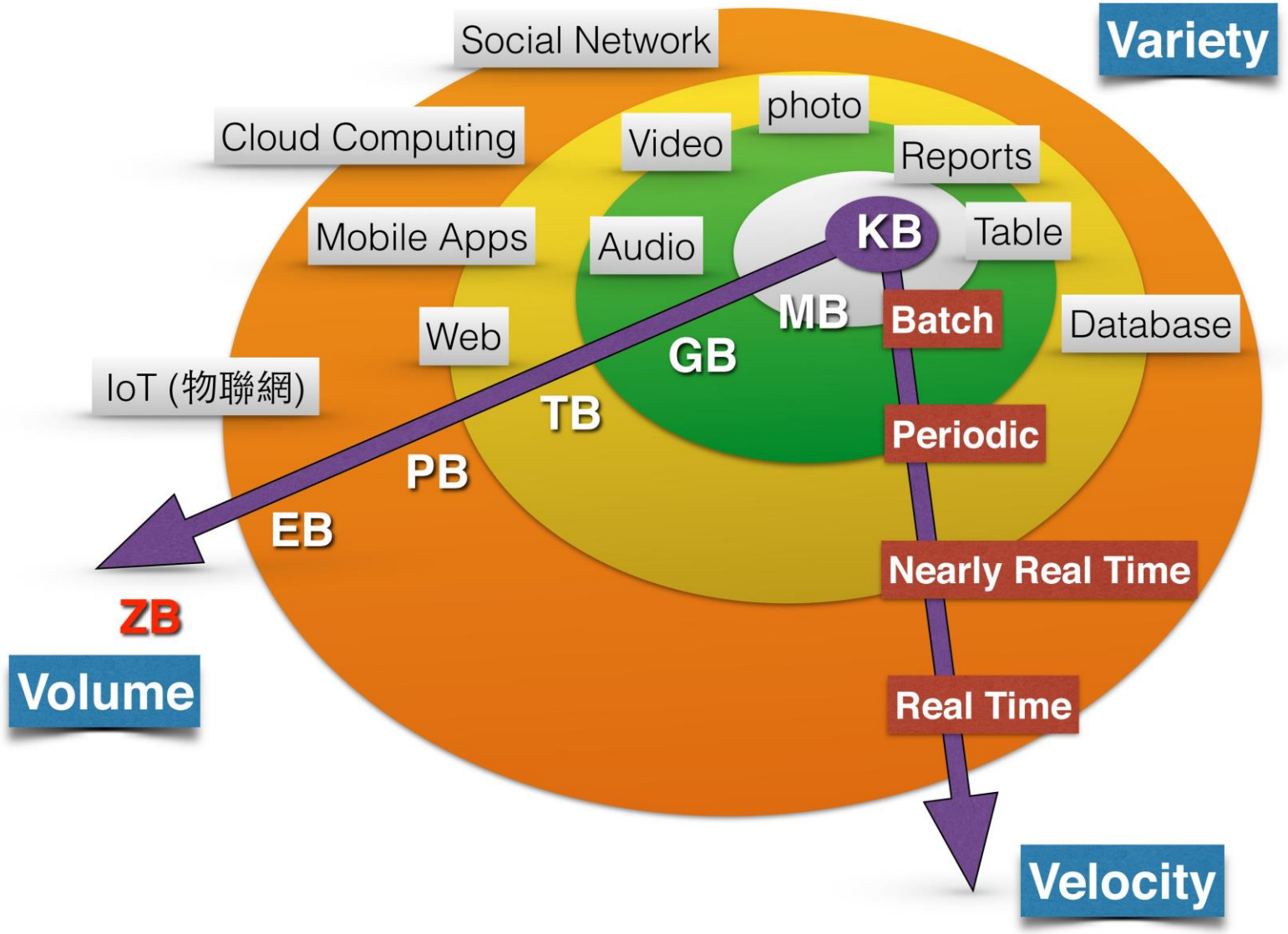
Semi-structured data

— data is looser, and though there may be a schema, it is often ignored for example, a spreadsheet.

Unstructured data

— such data does not have any particular internal structure: for example, plain text or image data.

3V's Model of Big Data



What is Data Scientist (資料科學家) ?

- 統計學家 (Statistician) vs. **資料科學家 (Data Scientist)**
  - ❖ 統計學家通常利用三門數學學科的領域知識：
    1. 數據分析 (Data Analytics) — (SPSS, SAS, Excel…)
    2. 機率 (Probability)
    3. 統計 (Statistics)
  - ❖ 資料科學家需要的領域知識：
    — 數據分析、機率、統計、**程式設計(例：R, Python, Java…)**、**平行運算、資料庫、機器學習 (Machine Learning)**
    + 至少一門專業的領域知識 (製造業、銀行業、股票市場…)

☞ **資料分析基礎 (PDF)**

“ In pioneer days they used oxen for heavy pulling, and when one ox couldn’t budge a log, they didn’t try to grow a larger ox.

We shouldn’t be trying for bigger computers, but for more systems of computers. *”*

“早期社會，人們利用牛來拉重物。當一頭牛無法拉動巨大的圓木時，他們不會想要飼養出更強壯的牛。**我們也不應該使用更強大的電腦，而是使用更多的電腦來解決問題。**”

—Grace Hopper

電腦叢集(CLUSTERS) for Big Data Analytics

Introducing Hadoop …

- **Apache Hadoop** is **an open source Java framework for processing and querying vast amounts of data on large clusters of commodity hardware**.
- **Hadoop** is a top level Apache project, initiated and led by **Yahoo!** and **Doug Cutting**. (http://hadoop.apache.org/)
- It relies on an active community of contributors from all over the world for its success.

Hadoop Features

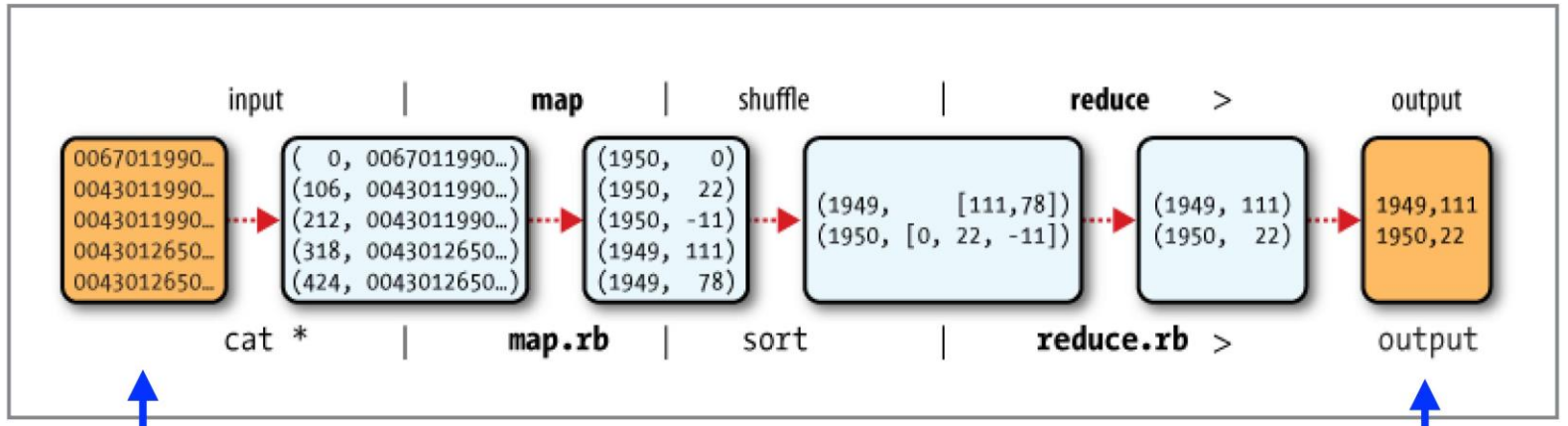
Apache Hadoop has two main features:

- **HDFS (Hadoop Distributed File System)**
  - Reading 1TB data on 1 Hard Disk (100MB/s) → about 2.5 hrs

    Reading 0.01 TB data from 100 Distributed Hard Disks → < 2 min.
  - **100 users' data on these 100 Distributed Hard Disks**

    → **Data Replication** → Similar to **RAID**
- **MapReduce**
  - written in **Java**, Ruby, Python, and C++.
  - **map phase → key-value pairs as input and output**
  - **reduce phase → key-value pairs as input and output**

Hadoop — MapReduce



the raw NCDC data

the final output:
the maximum global temperature
recorded in each year

Hadoop Components

- **Mahout** — This is an extensive library of machine learning algorithms.
- **Pig** — Pig is a high-level language (such as PERL) to analyze large datasets with its own language syntax for expressing data analysis programs, coupled with infrastructure for evaluating these programs.
- **Hive** — Hive is a data warehouse system for Hadoop that facilitates easy data summarization, ad hoc queries, and the analysis of large datasets stored in HDFS. It has its own SQL-like query language called **Hive Query Language (HQL)**, which is used to issue query commands to Hadoop.
- **HBase (Hadoop Database)**

Hadoop Components (cont'd)

- **Sqoop** — Apache Sqoop is a tool designed for efficiently transferring bulk data between Hadoop and Structured Relational Databases. Sqoop is an abbreviation for (SQ)L to Had(oop).
- **ZooKeeper** — ZooKeeper is a centralized service to maintain configuration information, naming, providing distributed synchronization, and group services, which are very useful for a variety of distributed systems.
- **Ambari** — A web-based tool for provisioning, managing, and monitoring Apache Hadoop clusters, which includes support for Hadoop HDFS, Hadoop MapReduce, Hive, HCatalog, HBase, ZooKeeper, Oozie, Pig, and Sqoop.

採用 R & Hadoop 解決 Big Data 問題的原因

- **R** 的優點是具有強大的數據分析程式庫，但是無法處理非常大的資料集(datasets)。
- **Hadoop** 叢集系統能夠儲存和處理巨量的資料 (TB，乃至於 PB 範圍；一般而言，如此的巨量資料集是無法放入單一電腦的主記憶體中，進行處理)。
- 因此，處理大數據的最佳解決方案：結合 ***R*** 與 ***Hadoop*** 的優點，來處理巨量資料。

2. Hadoop 功能介紹
— HDFS 與 MapReduce

Regarding HDFS

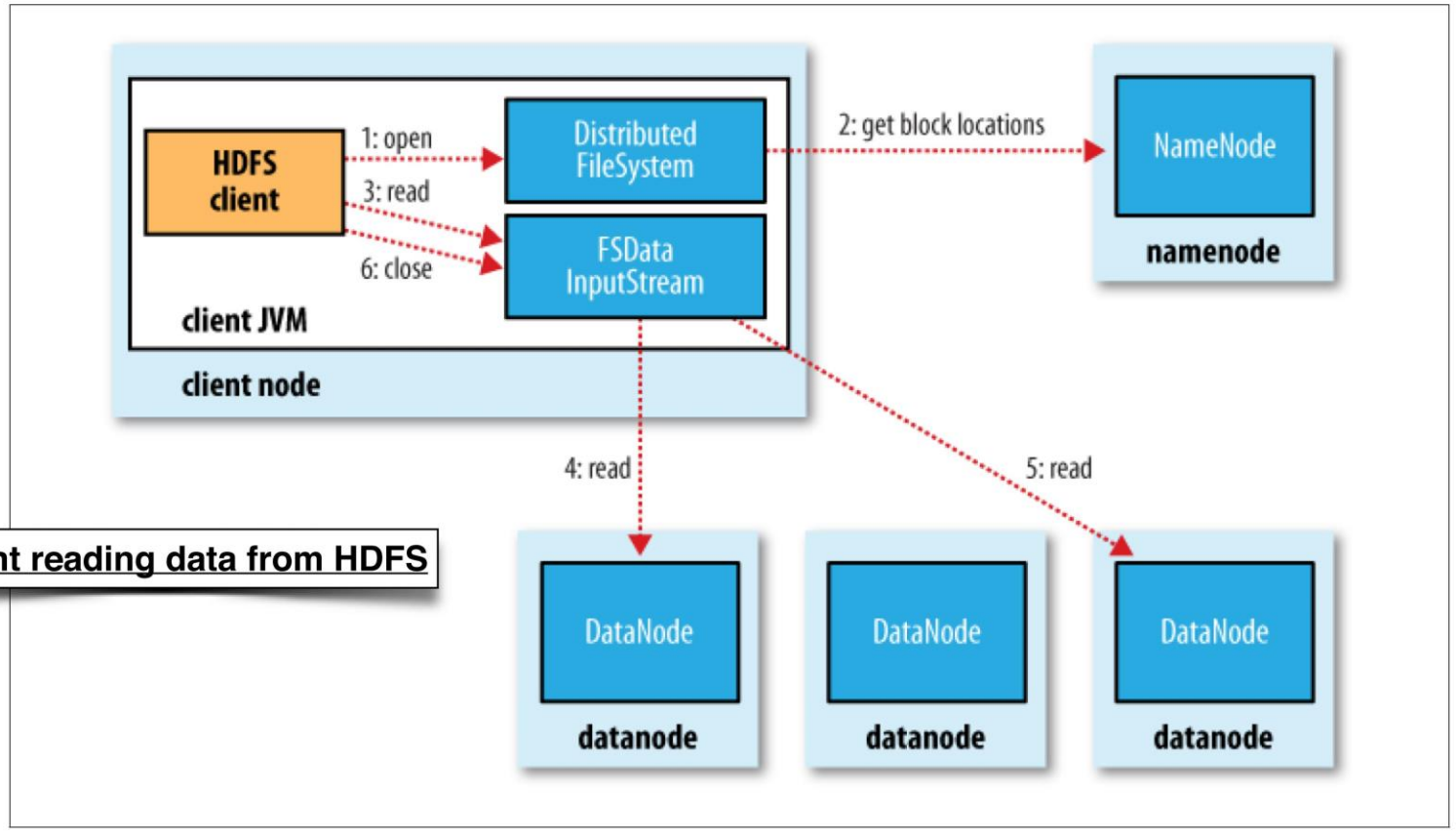
HDFS Concepts

- Blocks
- NameNodes (the masters)
- DataNodes (the workers)
- HDFS Federation
- HDFS High-Availability

☞ **HDFS Explanation (PDF)**

Regarding HDFS (cont'd)

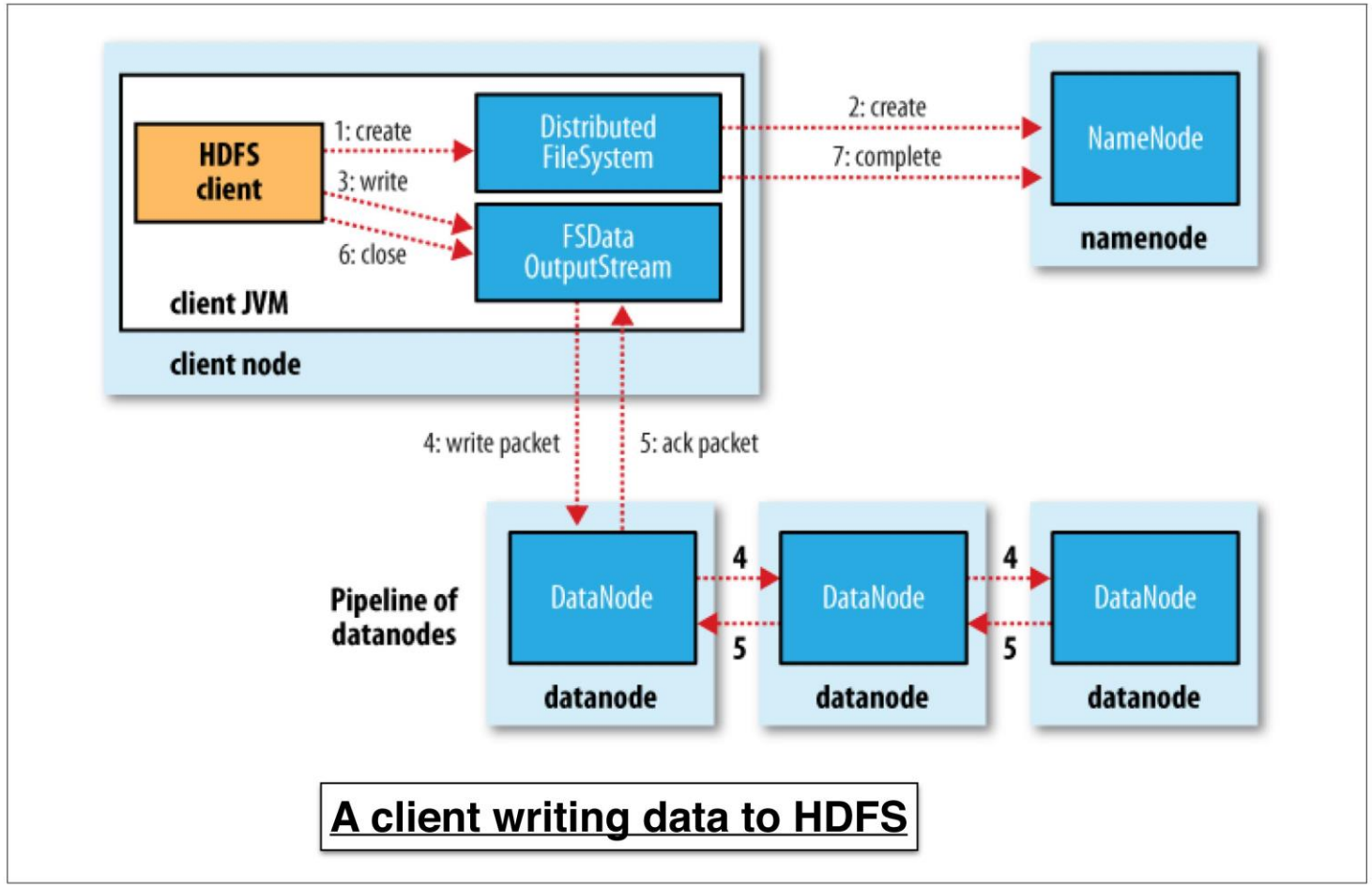
Data Flow — Anatomy of a File Read



A client reading data from HDFS

Regarding HDFS (cont'd)

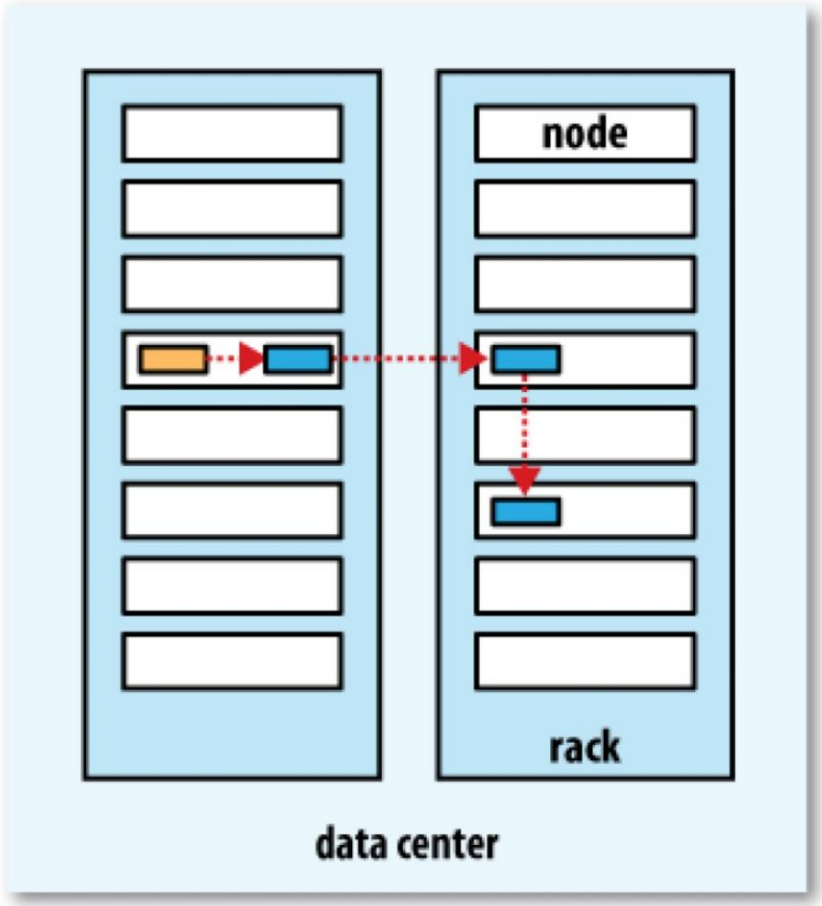
Data Flow — Anatomy of a File Write



A client writing data to HDFS

Regarding HDFS (cont'd)

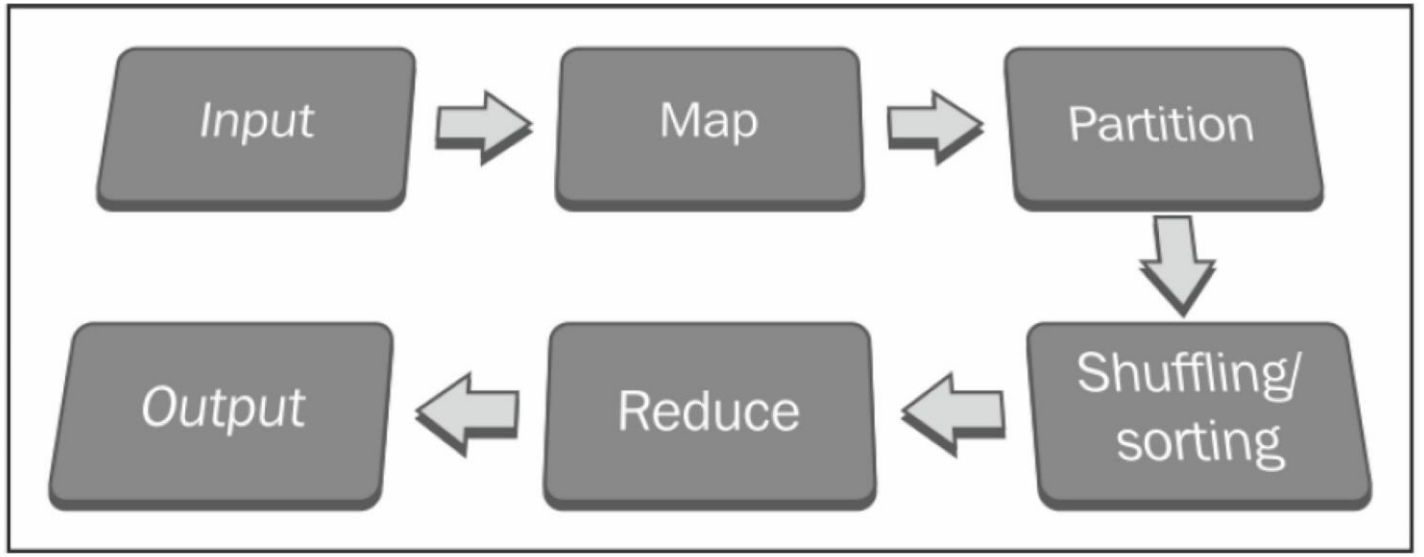
A typical replica pipeline



Regarding MapReduce

- **Map phase**
  - — Once divided, datasets are assigned to the task tracker to perform the Map phase.
  - — The data functional operation will be performed over the data, emitting the mapped key and value pairs as the output of the Map phase.
- **Reduce phase**
  - — The master node then collects the answers to all the subproblems and combines them in some way to form the output; the answer to the problem it was originally trying to solve.

Regarding MapReduce (cont'd)



Execution Process of MapReduce

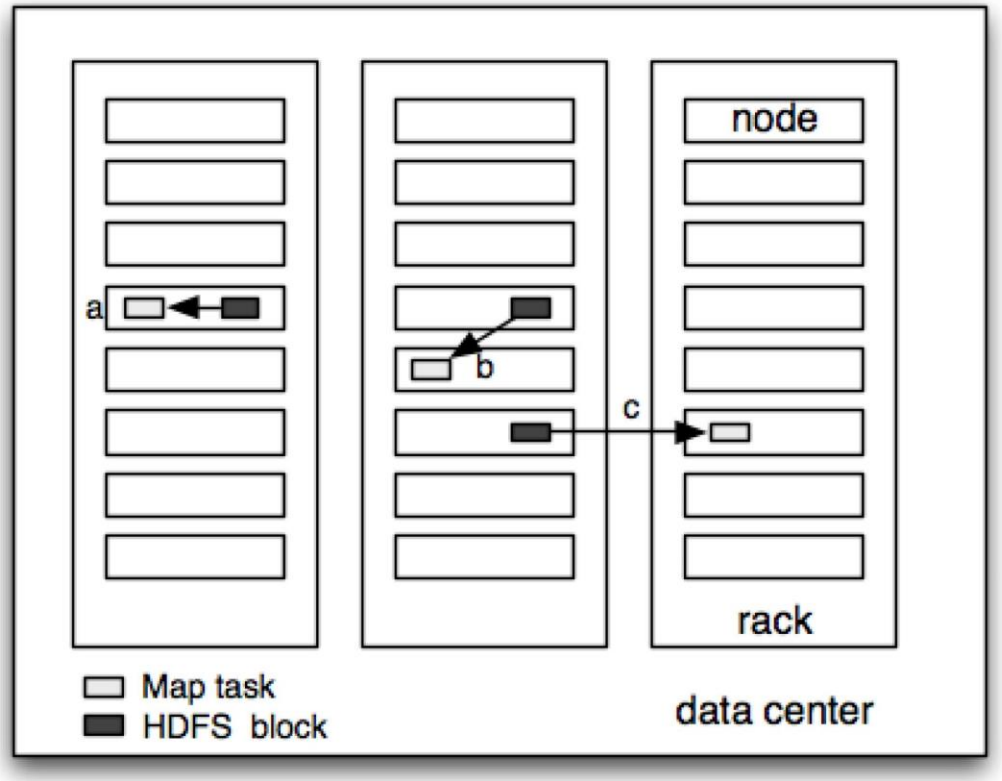
Regarding MapReduce (cont'd)

The five common steps of parallel computing are as follows:

1. Preparing the `Map()` input: This will take the input data row wise and emit key value pairs per rows, or we can explicitly change as per the requirement.
    - Map input: list (k1, v1)
2. Run the user-provided `Map()` code
    - Map output: list (k2, v2)
3. Shuffle the Map output to the Reduce processors. Also, shuffle the similar keys (grouping them) and input them to the same reducer.
4. Run the user-provided `Reduce()` code: This phase will run the custom reducer code designed by developer to run on shuffled data and emit key and value.
    - Reduce input: (k2, list(v2))
    - Reduce output: (k3, v3)
5. Produce the final output: Finally, the master node collects all reducer output and combines and writes them in a text file.

Regarding MapReduce (cont'd)

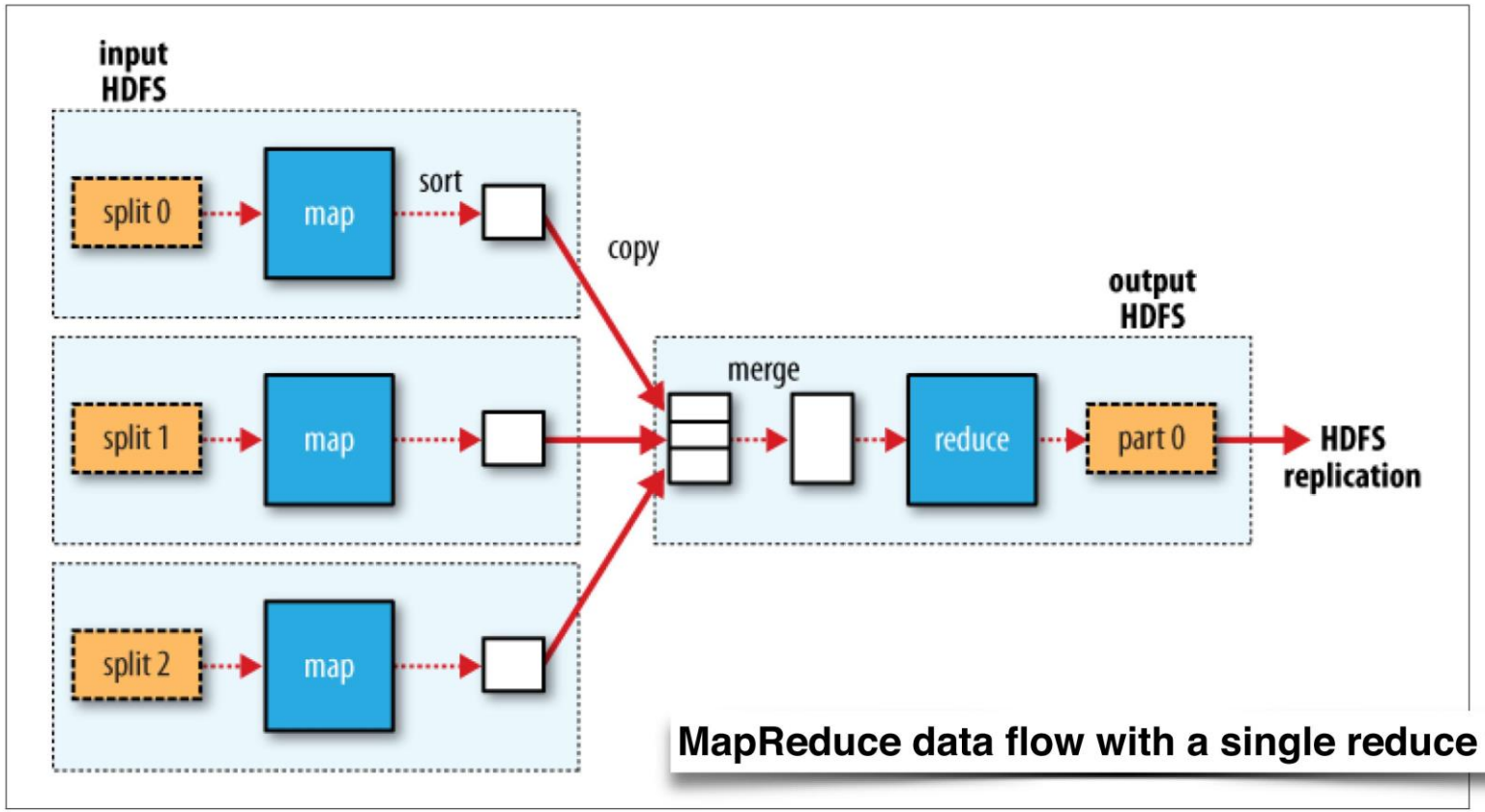
Data Flow



Data-local (a), rack-local (b), and off-rack (c) map tasks

Regarding MapReduce (cont'd)

Data Flow

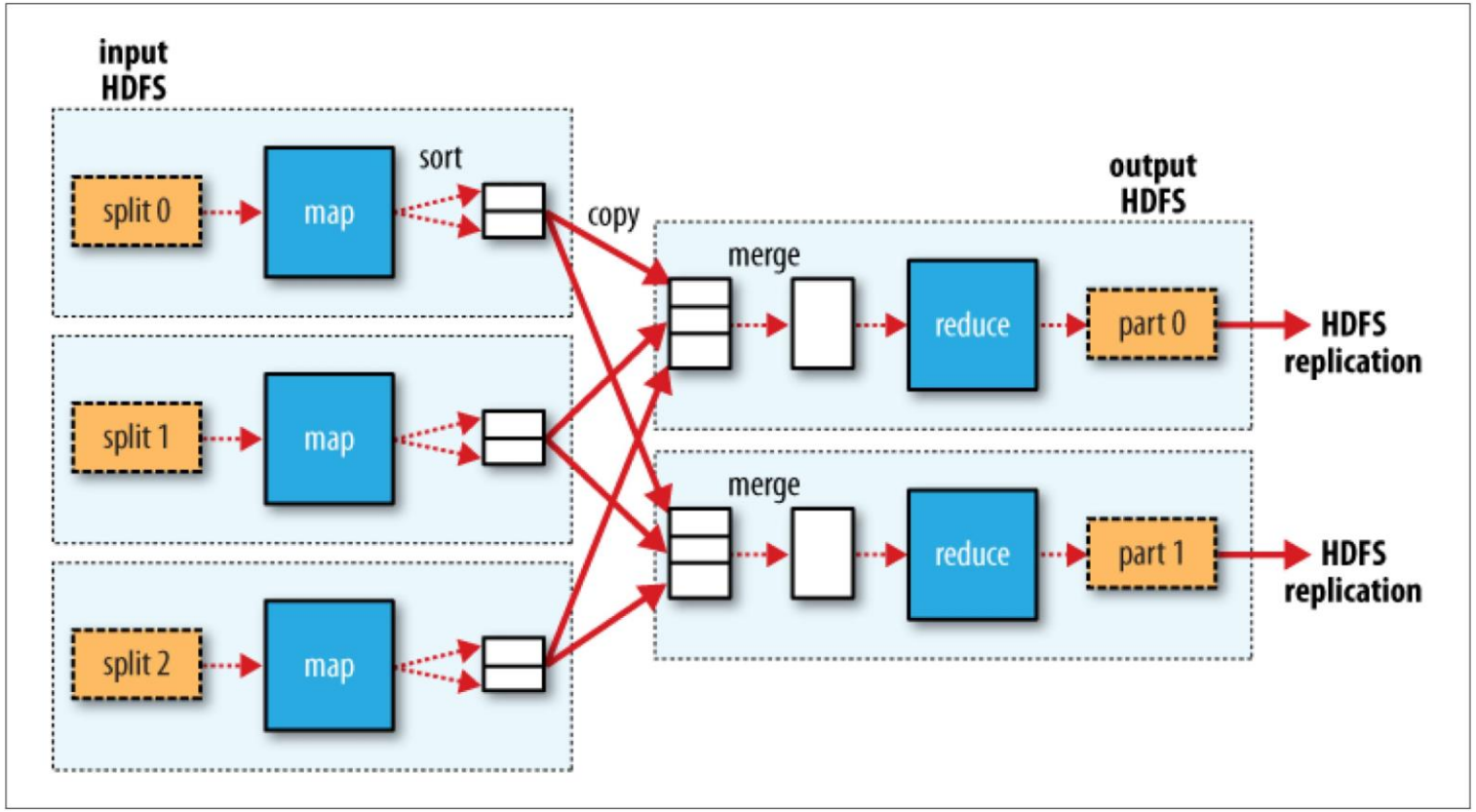


MapReduce data flow with a single reduce task

Regarding MapReduce (cont'd)

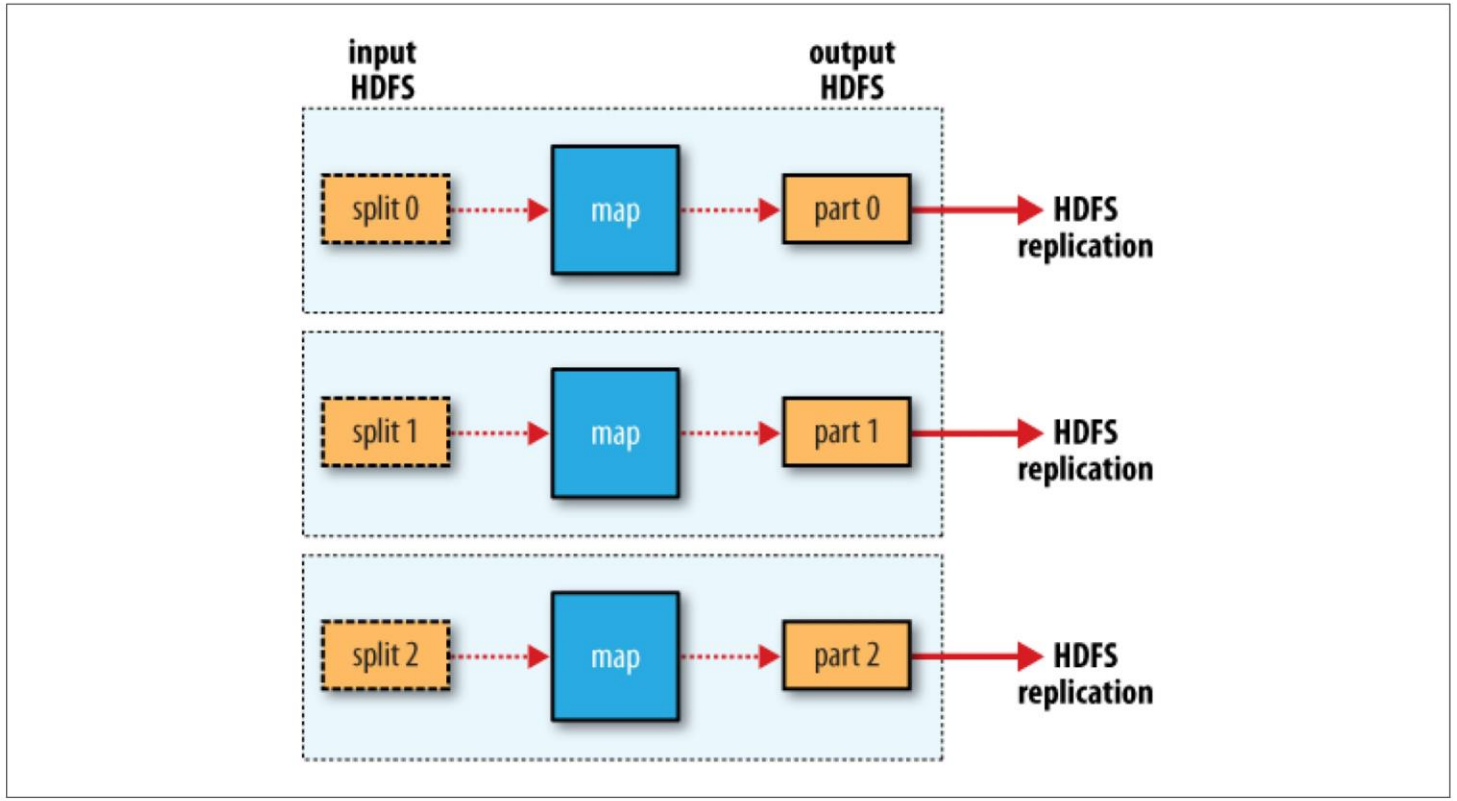
Data Flow

MapReduce data flow with multiple reduce tasks



Regarding MapReduce (cont'd)

Combiner Functions



MapReduce data flow with no reduce tasks

Regarding MapReduce (cont'd)

Companies using MapReduce include:

- **Amazon**: This is an online e-commerce and cloud web service provider for Big Data analytics
- **eBay**: This is an e-commerce portal for finding articles by its description
- **Google**: This is a web search engine for finding relevant pages relating to a particular topic
- **LinkedIn**: This is a professional networking site for Big Data storage and generating personalized recommendations
- **Trovit**: This is a vertical search engine for finding jobs that match a given description
- **Twitter**: This is a social networking site for finding messages

Hadoop :

HDFS and MapReduce

Architecture

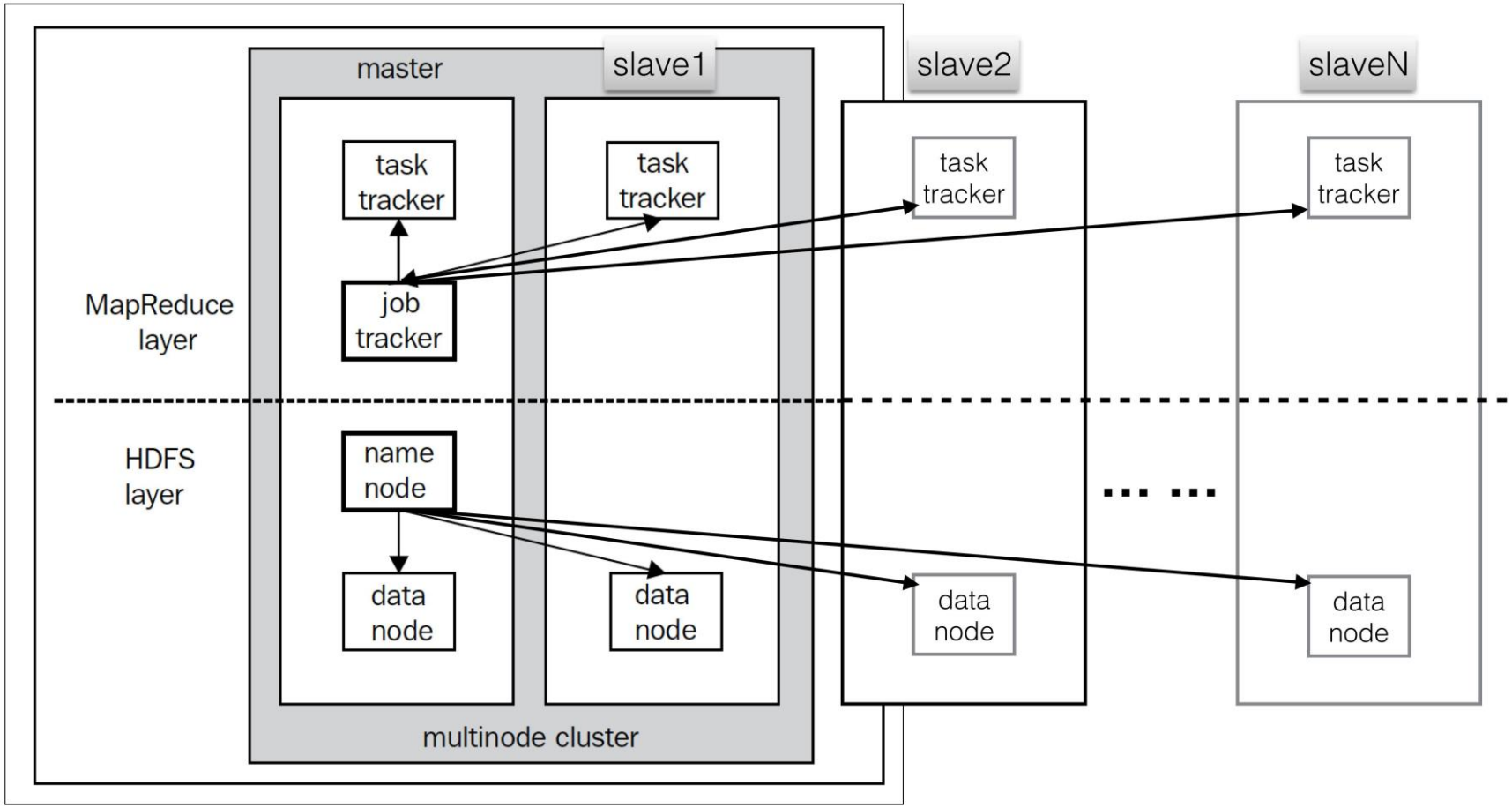
HDFS Components

- NameNode
- DataNode
- Secondary NameNode

MapReduce Components

- JobTracker
- TaskTracker

HDFS and MapReduce Architecture



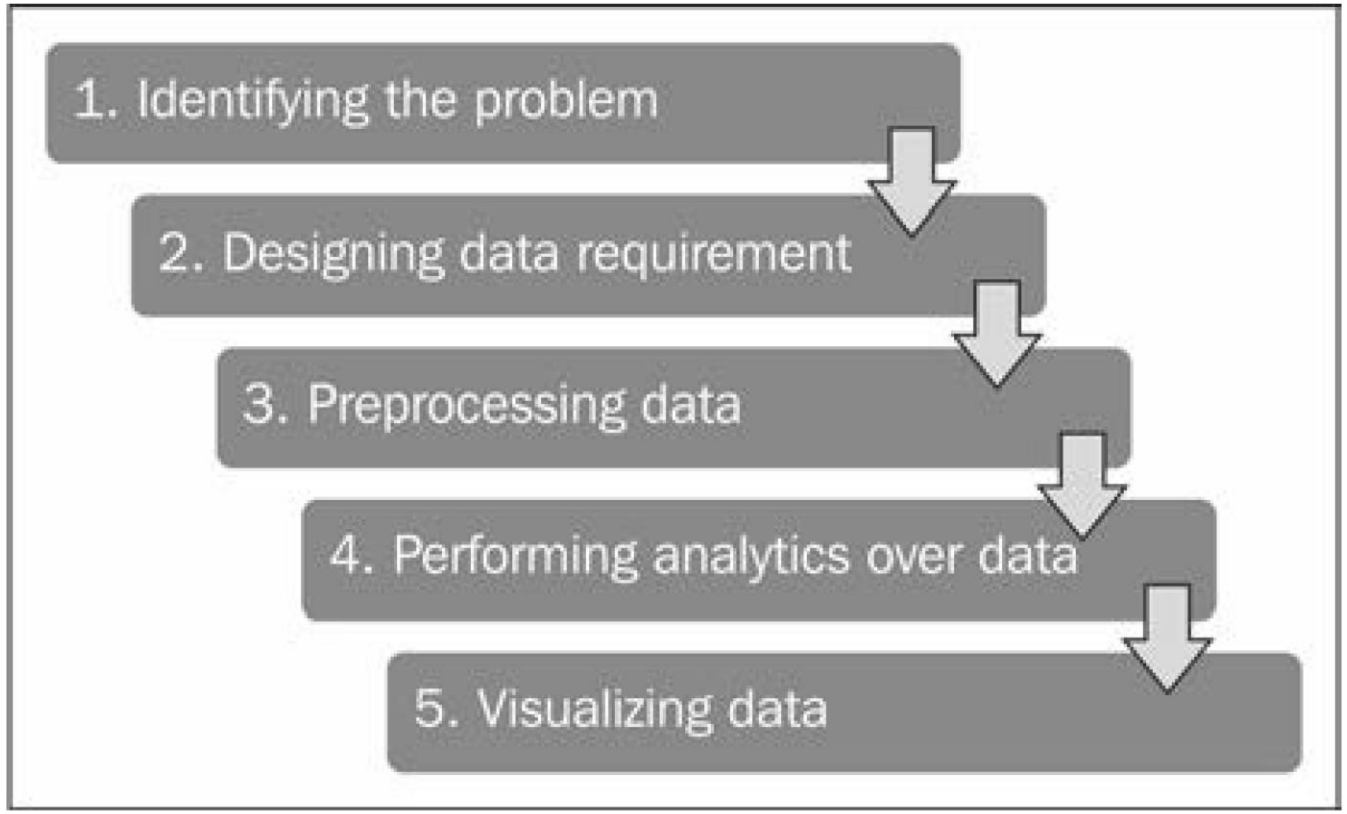
Hadoop 實作

Hadoop is used with three different modes:

- **The standalone mode** (單機模式)
  - In this mode, you do not need to start any Hadoop daemons.
  - All daemons, such as NameNode, DataNode, JobTracker, and TaskTracker run in a single Java process.
- **The pseudo mode** (虛擬分散模式，或稱 偽分散模式)
  - In this mode, you configure Hadoop for all the nodes.
  - A separate Java Virtual Machine (JVM) is spawned for each of the Hadoop components or daemons like mini cluster on a single host.
- **The full distributed mode** (完整分散模式)
  - In this mode, Hadoop is distributed across multiple machines.
  - Dedicated hosts are configured for Hadoop components. Therefore, separate JVM processes are present for all daemons.

Data Analytics Project Life Cycle

Data analytics project life cycle stages



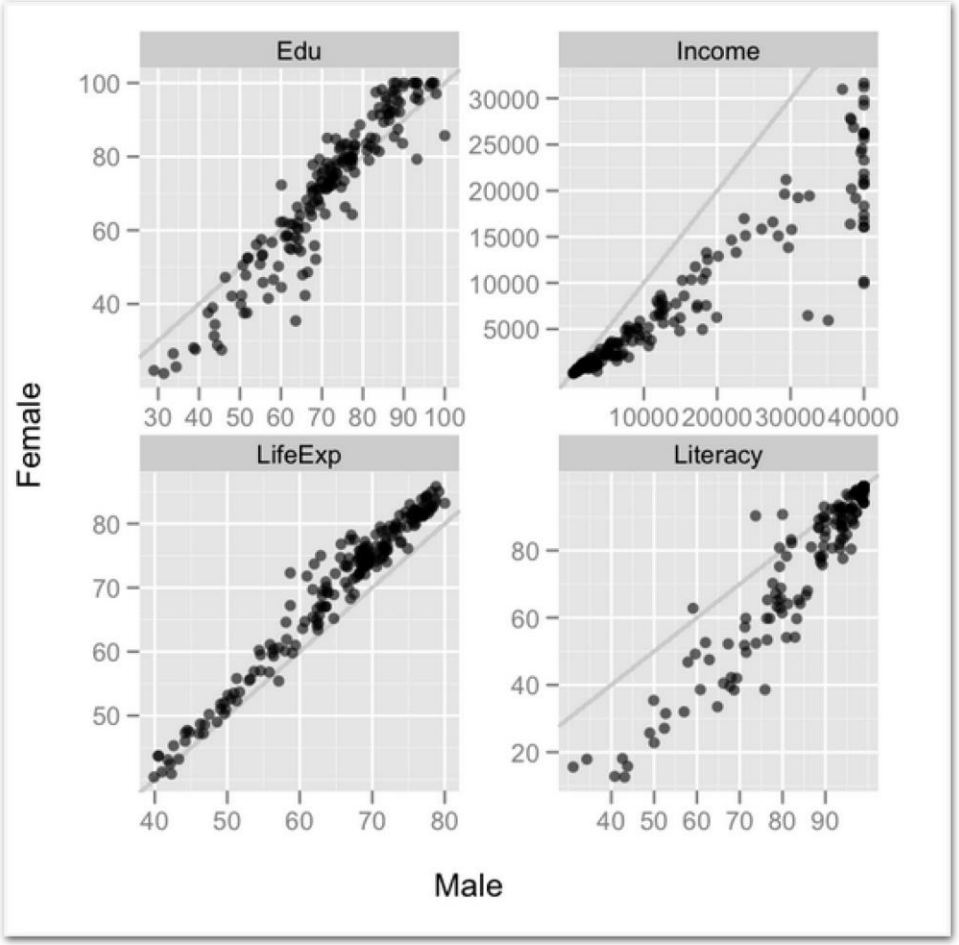
Data Analytics Project Life Cycle (cont'd)

STAGE 5 : Visualizing data

Plots for facet scales (ggplot):

The figure on the right shows the comparison of males and females with different measures; namely, education, income, life expectancy, and literacy, using ggplot.

(ggplot ref. http://cran.r-project.org/web/packages/ggplot2/.)



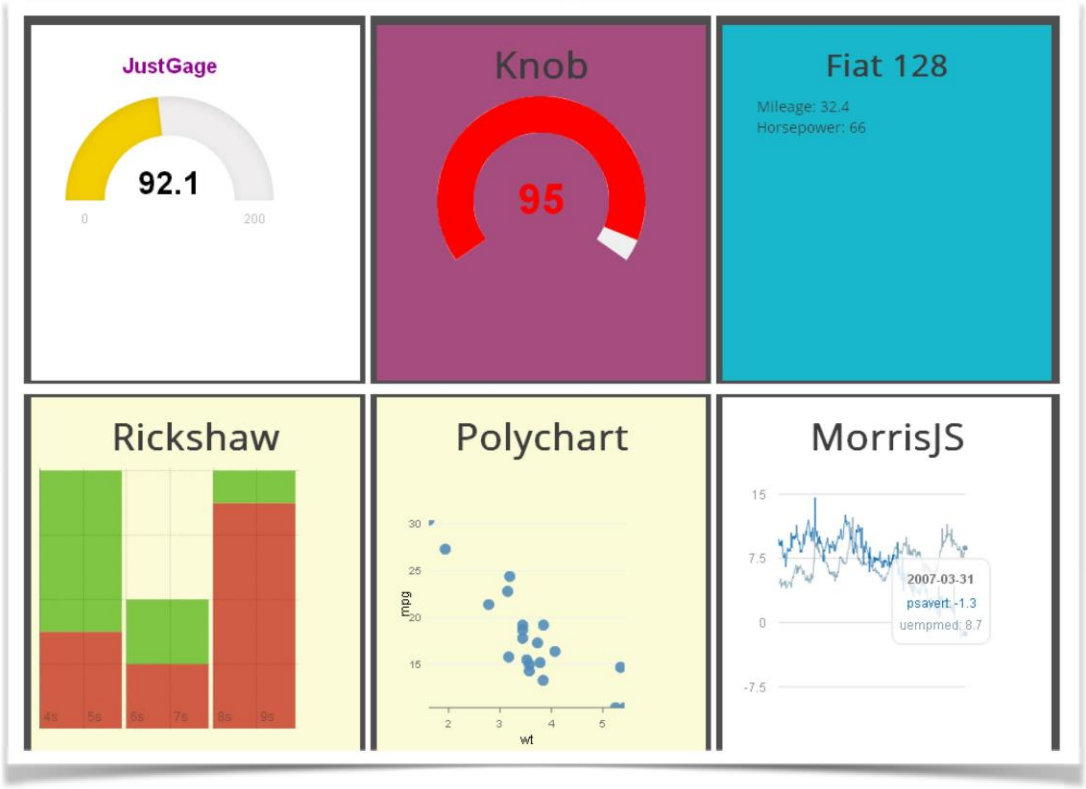
Data Analytics Project Life Cycle (cont'd)

STAGE 5 : Visualizing data (cont'd)

Dashboard charts:

This is an **rCharts** type. *Using this we can build interactive animated dashboards with R.*

(rCharts ref. http://ramnathv.github.io/rCharts/)



Data Analytics Problem

Computing the frequency of stock market change

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (1/7)

This data analytics MapReduce problem is designed for calculating the frequency of stock market changes.

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (2/7)

STAGE 1 : Identifying the problem

- Since this is a typical stock market data analytics problem, it will calculate the frequency of past changes for one particular symbol of the stock market, such as a **Fourier Transformation**.
- **Based on this information, the investor can get more insights on changes for different time periods.**
- So the **goal of this analytics** is **to calculate the frequencies of percentage change**.

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (3/7)

STAGE 1 : Identifying the problem (cont'd)

Yahoo finance data for symbol BP

Date	Open	High	Low	Close	Volume	Adj Close
2013-08-23	41.16	41.54	41.11	41.51	4117400	41.51
2013-08-22	40.82	40.99	40.75	40.91	2808300	40.91
2013-08-21	40.84	40.89	40.51	40.53	4296800	40.53
2013-08-20	41.02	40.90	40.90	4354200	40.90	
2013-08-19	41.29	41.35	41.05	41.10	3633800	41.10

Change frequency calculation for Yahoo Finance data

Change	Frequency
-0.1	20
0.3	2
0.8	1
1.0	22
1.9	12

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (4/7)

STAGE 2 : Designing data requirement

- For this stock market analytics, we will use **Yahoo! Finance** as the **input dataset**.
- We need to retrieve the specific symbol's stock information.
- To retrieve this data, we will use the **Yahoo! API** with the following parameters:
  - From month
  - From day
  - From year
  - To month
  - To day
  - To year
  - Symbol

NOTE : For more information on this API, visit http://developer.yahoo.com/finance/.

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (5/7)

STAGE 3 : Preprocessing data

To perform the analytics over the extracted dataset, we will use R to fire the following command:

```
stock_BP <- read.csv("http://ichart.finance.yahoo.com/table.csv?s=BP")
```

Or you can also download via the terminal:

```
wget http://ichart.finance.yahoo.com/table.csv?s=BP
#exporting to csv file

write.csv(stock_BP,"table.csv", row.names=FALSE)
```

Then upload it to HDFS by creating a specific Hadoop directory for this:

```
# creating /stock directory in hdfs
bin/hadoop dfs -mkdir /stock

# uploading table.csv to hdfs in /stock directory
bin/hadoop dfs -put /home/Vignesh/downloads/table.csv /stock/
```

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (6/7)

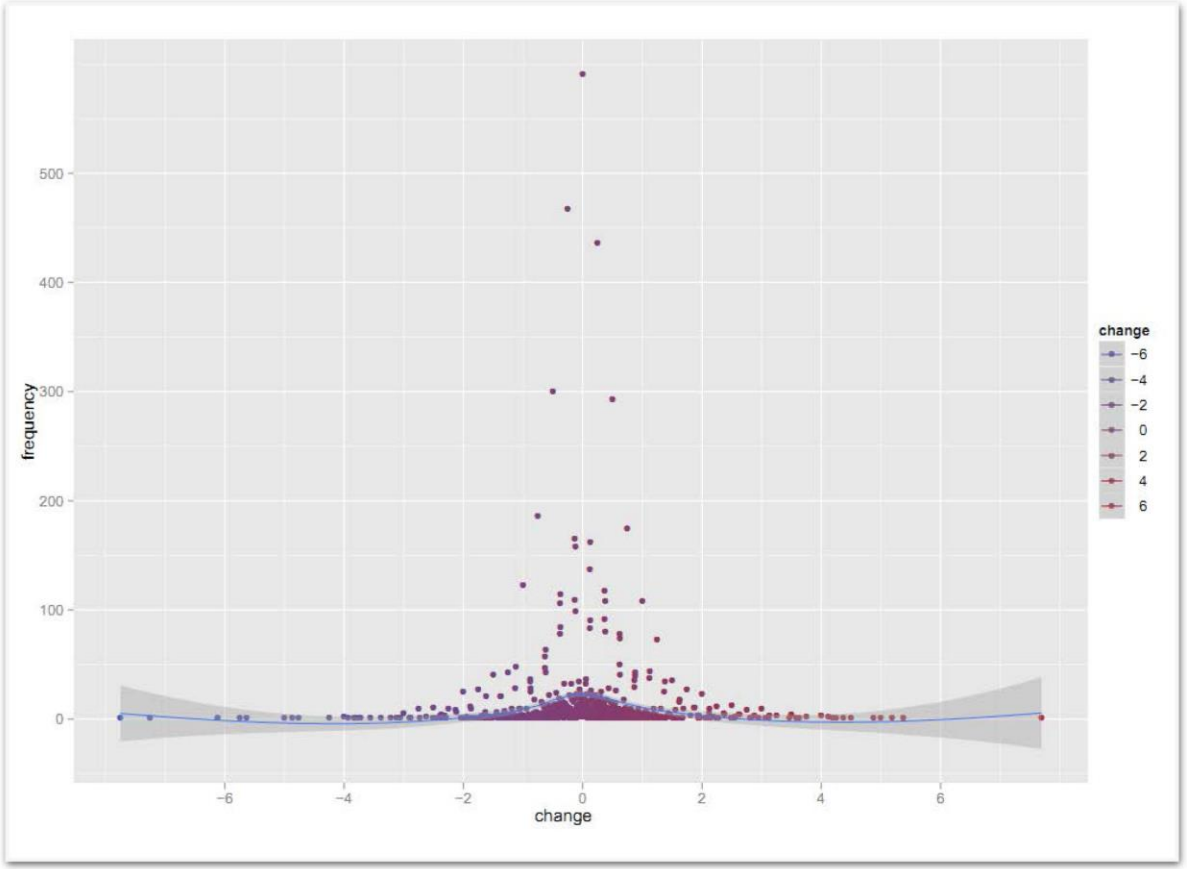
STAGE 4 : Performing analytics over data

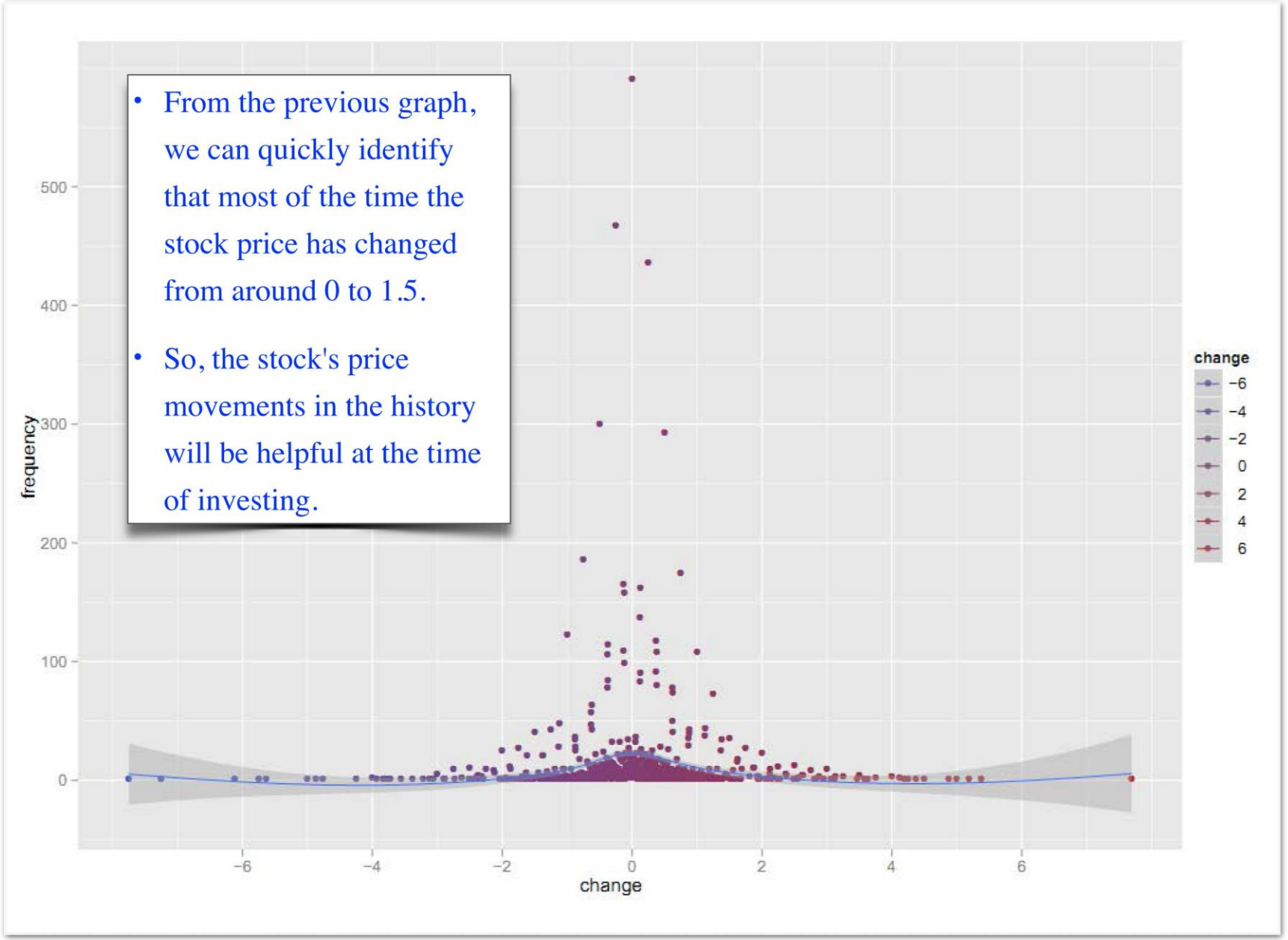
- To perform the data analytics operations, we will use streaming with **R** and **Hadoop** (without the **HadoopStreaming package**).
- So, the development of this **MapReduce** job can be **done without any RHadoop integrated library/package**.

Data Analytics Problems (cont'd)

Problem 2 : Computing the frequency of stock market change (7/7)

STAGE 5 : Visualizing data (by the ggplot2 package)




From the previous graph, we can quickly identify that most of the time the stock price has changed from around 0 to 1.5.
So, the stock's price movements in the history will be helpful at the time of investing.
frequency
500
400
300
200
100
0
−6
−4
−2
0
2
4
6
change
change
−6
−4
−2
0
2
4
6

資料匯入與匯出介紹

INTRODUCTION

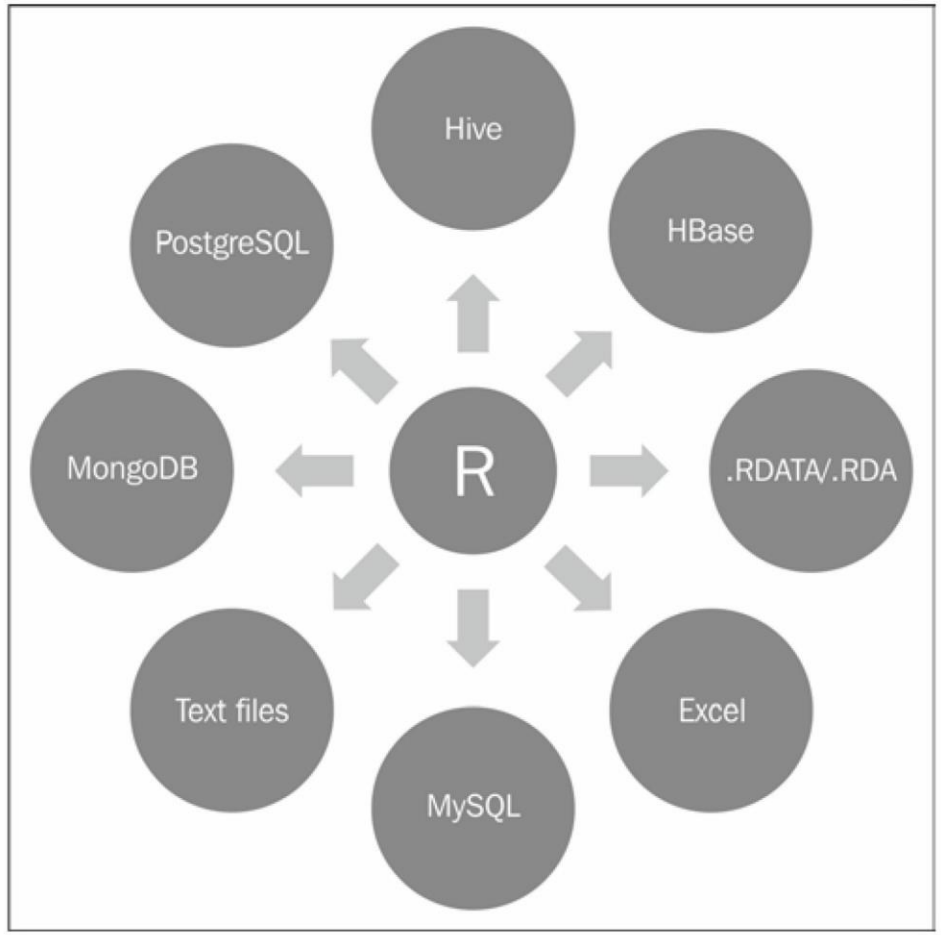
- Since, R has available methods to use customized functions via installing R packages, many database packages are available in **CRAN** to perform database connection with R.
- Therefore, **the R programming language is becoming more and more popular due to database, as well as operating system, independence.**

☞ RHadoop Programming - RStudio (PDF)

☞ R 於資料探勘之應用 (PDF)

Popular Data Sources Used with R

- RData
- MySQL
- Excel
- MongoDB
- SQLite
- PostgreSQL
- Hive
- HBase



Possible Database Systems & Related R Packages

Database system name	Useful R packages / function utilities
Text files	Text data files such as `.csv`, `.txt`, and `.r`
MySQL	RMySQL
Excel	Xlsx
Mongo	RMongo
SQLlite	RSQLlite
PostgreSQL	RPostgreSQL
HDFS	RHDFS
Hive	RHive
HBase	RHBase

Learning about Data Files as Database

Learning about Data Files as Database

- While dealing with the data analytics activities, we need to do data importing, loading, or exporting functionalities all the time.
- Sometimes the same operations need to be iterated with R programming language.
- So, we can use the available R function for performing the same data activities.

Learning about Data Files as Database (cont'd)

Understanding different types of files

There are commonly four different types of data files used with R for data storage operations. They are as follows:

- CSV (Comma Separated Values)
- Txt (with Tab Separated Values)
- .RDATA (R's native data format)
- .rda (R's native data format)

Installing R packages

To use the data file with the format specified earlier, we don't need to install extra R packages. We just need to use the built-in functions available with R.

Learning about Data Files as Database (cont'd)

Importing the data into R

To perform analytics-related activities, we need to use the following functions to get the data into R:

- CSV: `read.csv()` is intended for reading the **comma separated value (CSV)** files, where the decimal point is `","`. The retrieved data will be stored into one R object, which is considered as `Dataframe`.

```
Dataframe <- read.csv("data.csv",sep=",")
```

- TXT: To retrieve the tab separated values, the `read.table()` function will be used with some important parameters and the return type of this function will be `Dataframe type`.

```
Dataframe <- read.table("data.csv", sep="\t")
```

- .RDATA: Here, the .RDATA format is used by R for storing the workspace data for a particular time period. It is considered as image file. This will store/retrieve all of the data available in the workspace.

```
load("history.RDATA")
```

- .rda: This is also R's native data format, which stores the specific data variable as per requirement.

```
load("data_variables_a_and_b.rda")
```

Learning about Data Files as Database (cont'd)

Exporting the data from R

To export the existing data object from R and to support data files as per requirements, we need to use the following functions:

- CSV: Write the dataframe object into the `csv` data file via the following command:

```
write.csv(mydata, "c:/mydata.csv", sep=",", row.names=FALSE)
```

- TXT: Write the data with the tab delimiters via the following command:

```
write.table(mydata, "c:/mydata.txt", sep="\t")
```

- .RDATA: To store the workspace data variables available to R session, use the following command:

```
save.image()
```

- .rda: This function is used to store specific data objects that can be reused later. Use the following code for saving them to the `.rda` files.

```
# column vector
a <- c(1,2,3)

# column vector
b <- c(2,4,6)

# saving it to R (.rda) data format
save(a, b, file=" data_variables_a_and_b.rda")
```

Learning about Data Files as Database (cont'd)

MySQL & R

Understanding MySQL

MySQL is world's most popular open source database. Many of the world's largest and fastest growing organizations including Facebook, Google, Adobe, and Zappos rely on MySQL databases, to save time and money powering high-volume websites, business critical systems, and software packages.

Since both R and MySQL both are open source, they can be used for building the interactive web analytic applications. Also simple data analytics activities can be performed for existing web applications with this unique package.

To install MySQL on your Linux machine, you need to follow the given steps in sequence:

- Install MySQL
- Install RMySQL

Learning about Data Files as Database (cont'd)

MySQL & R

Importing the data into R

We know how to check MySQL tables and their fields. After identification of useful data tables, we can import them in R using the following RMySQL command. To retrieve the custom data from MySQL database as per the provided SQL query, we need to store it in an object:

```
rs = dbSendQuery(mydb, "select * from sample_table")
```

The available data-related information can be retrieved from MySQL to R via the `fetch` command as follows:

```
dataset = fetch(rs, n=-1)
```

Here, the specified parameter `n = -1` is used for retrieving all pending records.

Learning about Data Files as Database (cont'd)

Excel

Understanding Excel

Excel is a spreadsheet application developed by Microsoft to be run on Windows and Mac OS, which has a similar function to R for performing statistical computation, graphical visualization, and data modeling. Excel is provided by Microsoft with the Microsoft Office bundle, which mainly supports `.xls` spreadsheet data file format. In case, we want to read or write to Microsoft Excel spreadsheets from within R, we can use many available R packages. But one of the popular and working R library is xlsx.

This package programmatically provides control of the Excel files using R. The high level API of this allows users to read a spread sheet of the `.xlsx` document into a `data.frame` and writing `data.frame` to a file. This package is basically developed by *Adrian A. Dragulescu.*

Learning about Data Files as Database (cont'd)

Excel

Importing data into R

Suppose we have created one excel file and now we want to perform the data analytics related operations with R, this is the best package to load the excel file to be processed within R.

```
es <- read.xlsx("D:/ga.xlsx",1)
```

The preceding command will store the excel data with sheet 1 into the `es` dataframe format in R.

Exporting the data to Excel

As per the defined name, the processed data with the dataframe format can be stored as a `xls` file to be supported with Excel.

```
ress <- write.xlsx(r, "D:/ga1.xls")
```

Learning about Data Files as Database (cont'd)

HBase

Understanding HBase

Apache HBase is a distributed Big Data store for Hadoop. This allows random, real-time, read/write access to Big Data. This is designed as a column-oriented, data-storage model, innovated after being inspired by Google Big table.

Understanding HBase features

Following are the features for HBase:

- RESTful web service with XML
- Linear and modular scalability
- Strict consistent reads and writes
- Extensible shell
- Block cache and Bloom filters for real-time queries

Learning about Data Files as Database (cont'd)

HBase

Importing the data into R

Once RHBase is installed, we can load the dataset in R from HBase with the help of RHBase:

- To list all tables we use:

```
hb.list.tables ()
```

- To create a new table we use:

```
hb.new.table ("student")
```

- To display the table structure we use:

```
hb.describe.table("student_rhbase")
```

- To read data we use:

```
hb.get ('student_rhbase', 'mary')
```

Learning about Data Files as Database (cont'd)

HBase

Understanding data manipulation

Now, we will see how to operate over the dataset of HBase from within R:

- To create the table we use:

```
hb.new.table ("student_rhbase", "info")
```

- To insert the data we use:

```
hb.insert ("student_rhbase", list (list ("mary", "info: age",
"24")))
```

- To delete a sheet we use:

```
hb.delete.table ('student_rhbase')
```

6. 結論

- 商業週刊 **1438**期 **(2015/6/8 ~ 6/14)**

  標題：“ 大數據的新生意經 ”

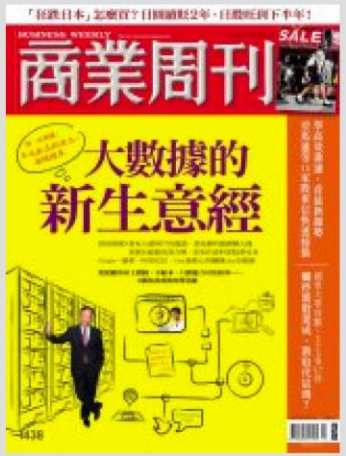
  副標題 — “ 你一定要懂：

  羊毛出在狗身上，由豬買單。”

  （羊：消費者

  狗：擁有大數據的企業

  豬：花錢買大數據的企業）



- **Big Data Analytics** — 新的商業營運模式
- 資料科學家 **(Data Scientist)** 該具備的養成訓練