Tools and Techniques for Data Science

Joko Triloka
17 Oktober 2020
MTI IIB Darmajaya

Agenda

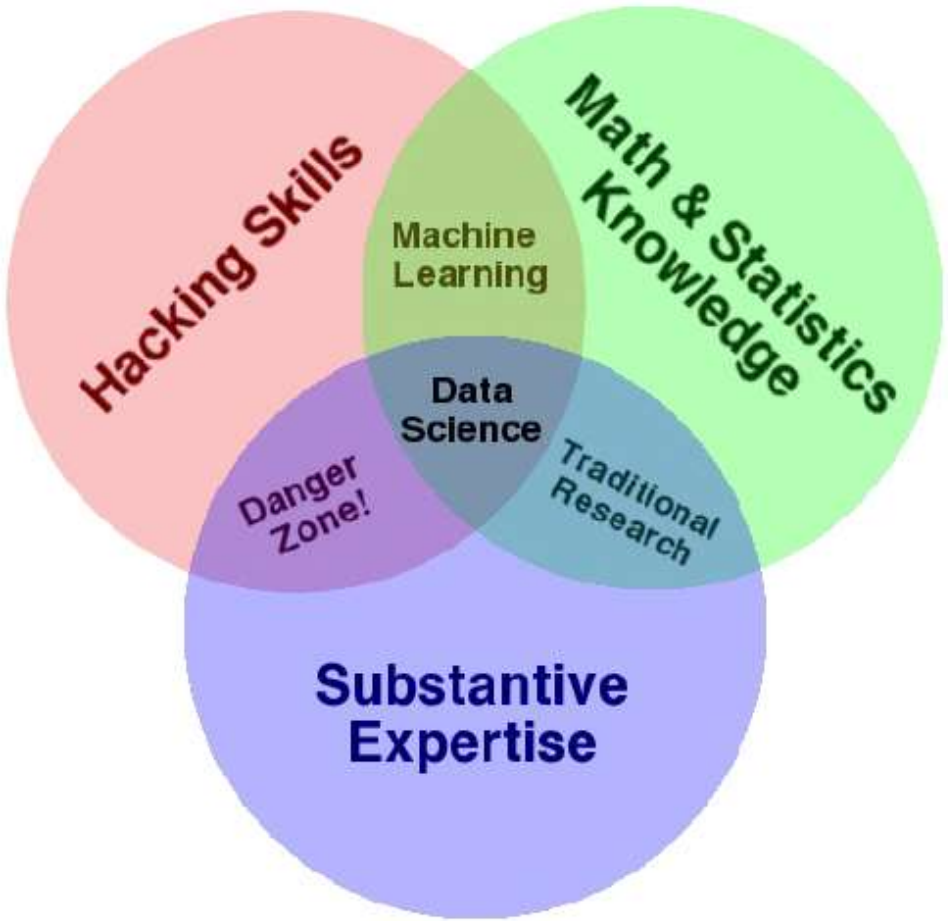
Definition of Data Science

Cloud Computing and Big Data

Tools in Data Science

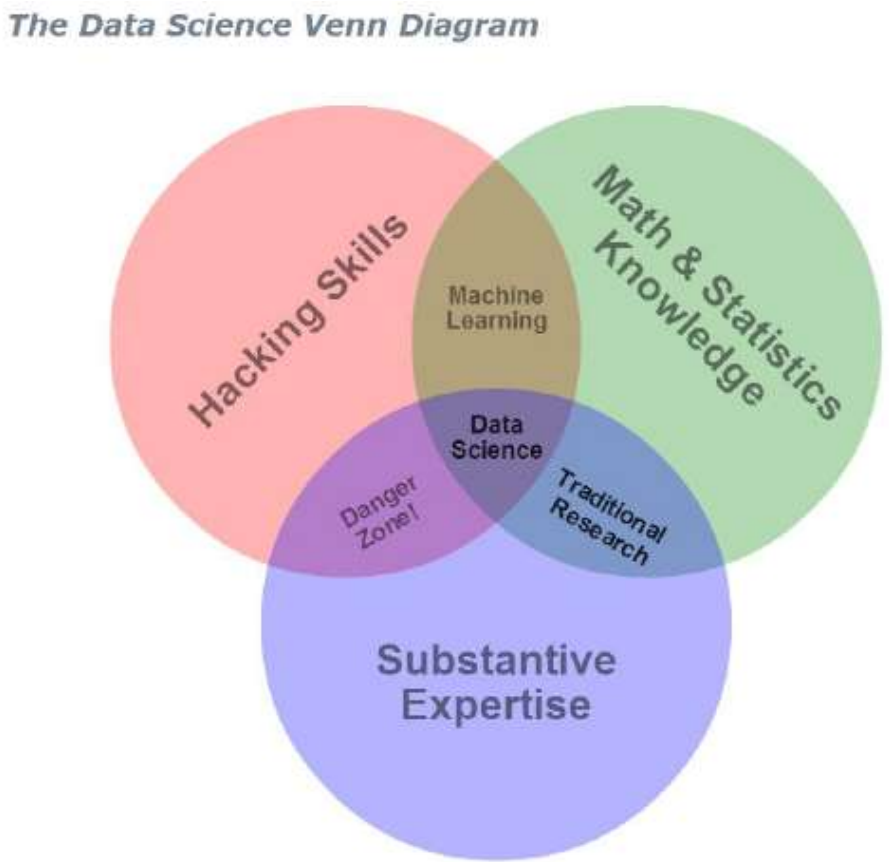
Techniques in Data Science

Definition of Data Science



What is data science ?

Hacking (Programming) + Maths/Statistics + Domain Knowledge = Data Science



What is a Data Scientist ?

a data scientist is simply a person who can

write code

understand statistics

derive insights from data

Oh really, is this a Data Scientist ?

a data scientist is simply a person who can

write code = in R,Python,Java, SQL, Hadoop (Pig,HQL,MR) etc

= for data storage, querying, summarization, visualization

= how efficiently, and in time (fast results?)

= where on databases, on cloud, servers

and understand **enough** statistics

to derive **insights** from data

so **business** can make **decisions**

Cheat Sheets for Data Scientists

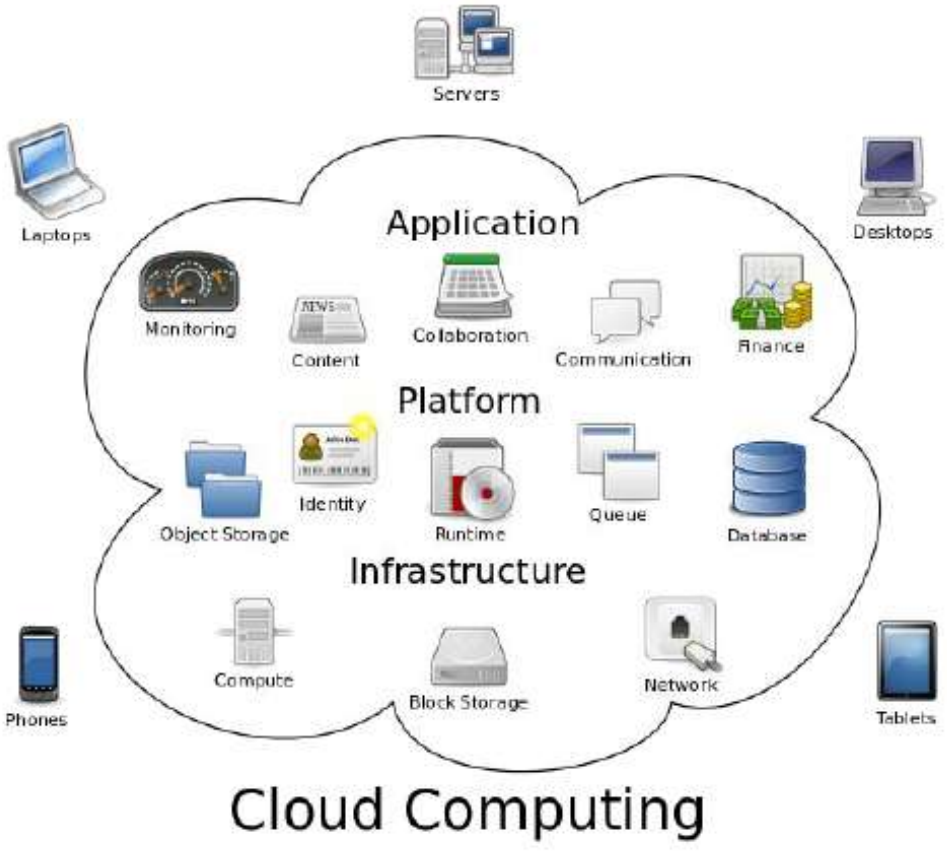
http://www.kdnuggets.com/2014/05/guide-to-data-science-cheat-sheets.html

a data scientist is simply a person who can write code in a few languages (primarily R, Python and SQL) for data querying, manipulation , aggregation, and visualization using enough statistical knowledge to give back actionable insights to the business for making decisions.

Since this rather practical definition of a data scientist is reinforced by the accompanying words on a job website for “data scientists” , ergo, here are some tools for learning the primary languages in data science- Python, R and SQL. A cheat sheet or reference card is a compilation of mostly used commands to help you learn that language’s syntax at a faster rate.

Cloud Computing

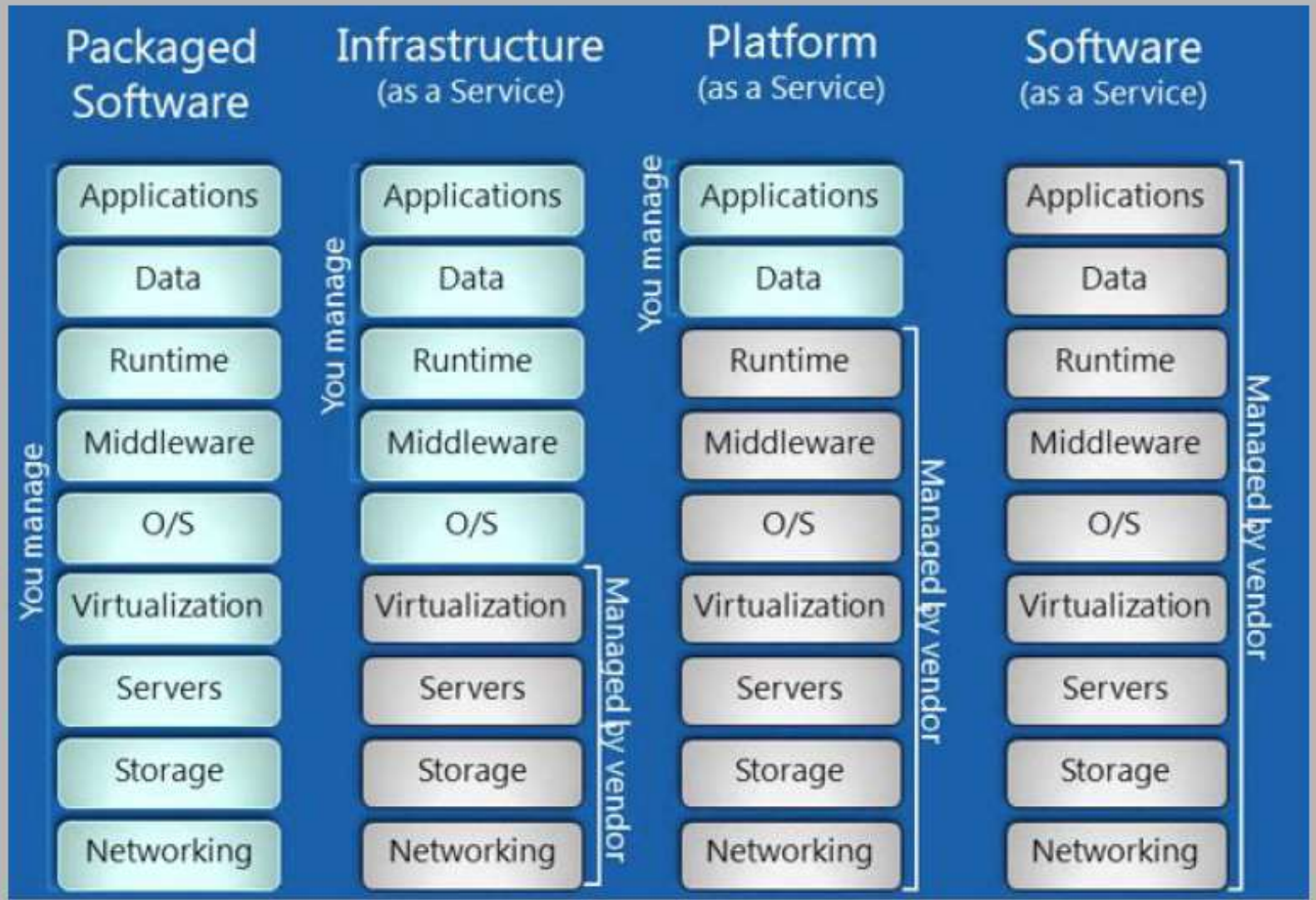
1. the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.



Cloud Computing

1. the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.

Cloud Computing



Introduction to Big Data

extremely large data sets that may be analysed computationally to reveal patterns, trends, and associations, especially relating to human behaviour and interactions.

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.

The Three Vs of Big Data

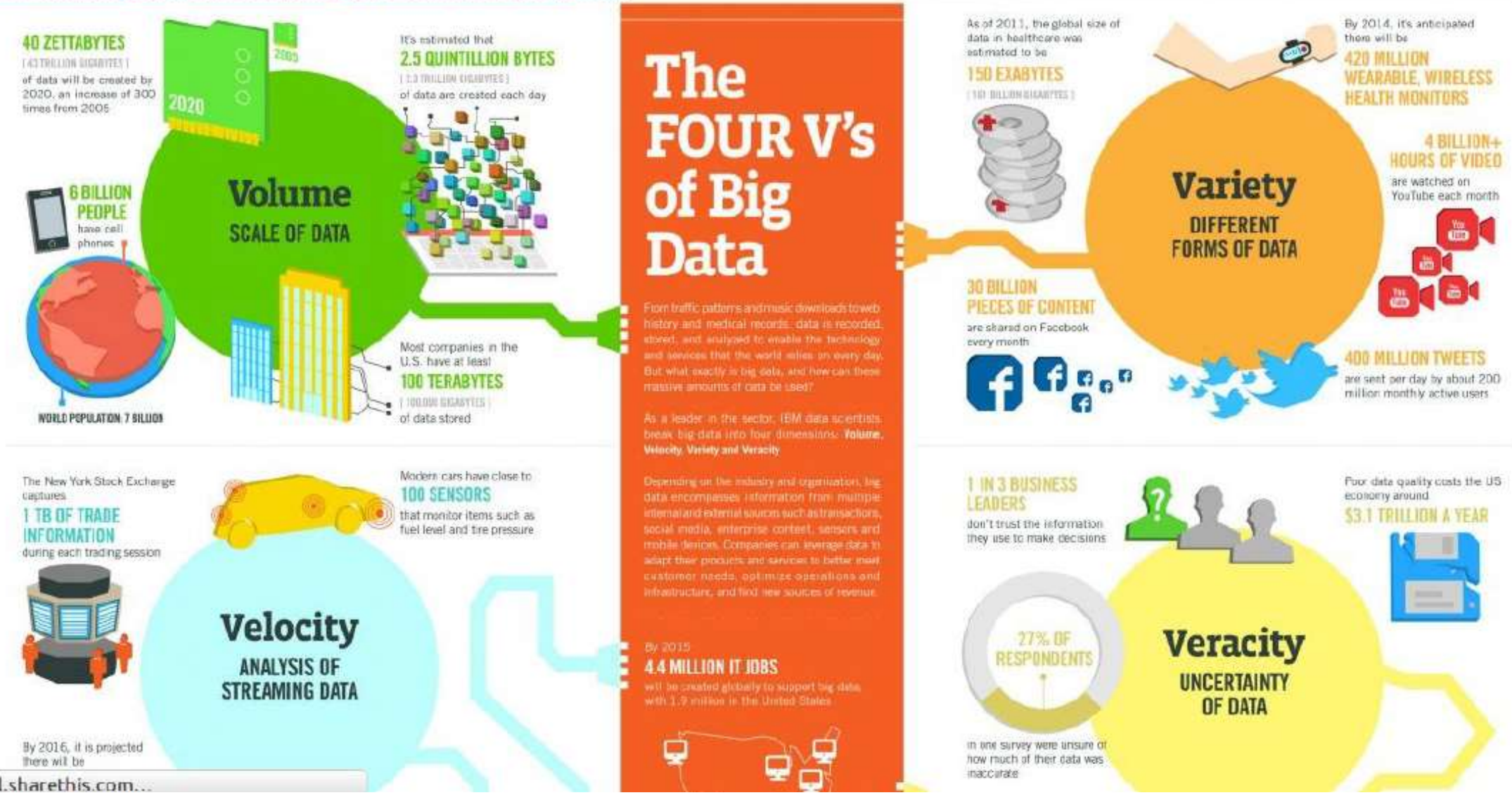
industry analyst Doug Laney articulated the now-mainstream definition of big data as the three Vs:

Volume. Organizations collect data from a variety of sources, including business transactions, social media and information from sensor or machine-to-machine data. In the past, storing it would've been a problem – but new technologies (such as Hadoop) have eased the burden.

Velocity. Data streams in at an unprecedented speed and must be dealt with in a timely manner. RFID tags, sensors and smart metering are driving the need to deal with torrents of data in near-real time.

Variety. Data comes in all types of formats – from structured, numeric data in traditional databases to unstructured text documents, email, video, audio, stock ticker data and financial transactions.

http://www.ibmbigdatahub.com/infographic/four-vs-big-data



Understanding Hadoop

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The project includes these modules:

- **Hadoop Common**: The common utilities that support the other Hadoop modules.
- **Hadoop Distributed File System (HDFS™)**: A distributed file system that provides high-throughput access to application data.
- **Hadoop YARN**: A framework for job scheduling and cluster resource management.
- **Hadoop MapReduce**: A YARN-based system for parallel processing of large data sets.

http://hadoop.apache.org/

Understanding Hadoop

- Other Hadoop-related projects at Apache include:
- 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. Ambari also provides a dashboard for viewing cluster health such as heatmaps and ability to view MapReduce, Pig and Hive applications visually alongwith features to diagnose their performance characteristics in a user-friendly manner.
- Avro™: A data serialization system.
- Cassandra™: A scalable multi-master database with no single points of failure.
- Chukwa™: A data collection system for managing large distributed systems.
- HBase™: A scalable, distributed database that supports structured data storage for large tables.
- Hive™: A data warehouse infrastructure that provides data summarization and ad hoc querying.
- Mahout™: A Scalable machine learning and data mining library.
- Pig™: A high-level data-flow language and execution framework for parallel computation.
- Spark™: A fast and general compute engine for Hadoop data. Spark provides a simple and expressive programming model that supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation.
- Tez™: A generalized data-flow programming framework, built on Hadoop YARN, which provides a powerful and flexible engine to execute an arbitrary DAG of tasks to process data for both batch and interactive use-cases. Tez is being adopted by Hive™, Pig™ and other frameworks in the Hadoop ecosystem, and also by other commercial software (e.g. ETL tools), to replace Hadoop™ MapReduce as the underlying execution engine.
- ZooKeeper™: A high-performance coordination service for distributed applications.

Why Learn Hadoop

Hadoop, however, is specifically designed to have a very flat scalability curve. After a Hadoop program is written and functioning on ten nodes, very little--if any--work is required for that same program to run on a much larger amount of hardware.

MapReduce: Isolated Processes

Hadoop limits the amount of communication which can be performed by the processes, as each individual record is processed by a task in isolation from one another. While this sounds like a major limitation at first, it makes the whole framework much more reliable

In a Hadoop cluster, data is distributed to all the nodes of the cluster as it is being loaded in. The Hadoop Distributed File System (HDFS) will split large data files into chunks which are managed by different nodes in the cluster. In addition to this each chunk is replicated across several machines, so that a single machine failure does not result in any data being unavailable.

https://developer.yahoo.com/hadoop/tutorial/module1.html

HDFS

HDFS, the Hadoop Distributed File System, is a distributed file system designed to hold very large amounts of data (terabytes or even petabytes), and provide high-throughput access to this information. Files are stored in a redundant fashion across multiple machines to ensure their durability to failure and high availability to very parallel applications.

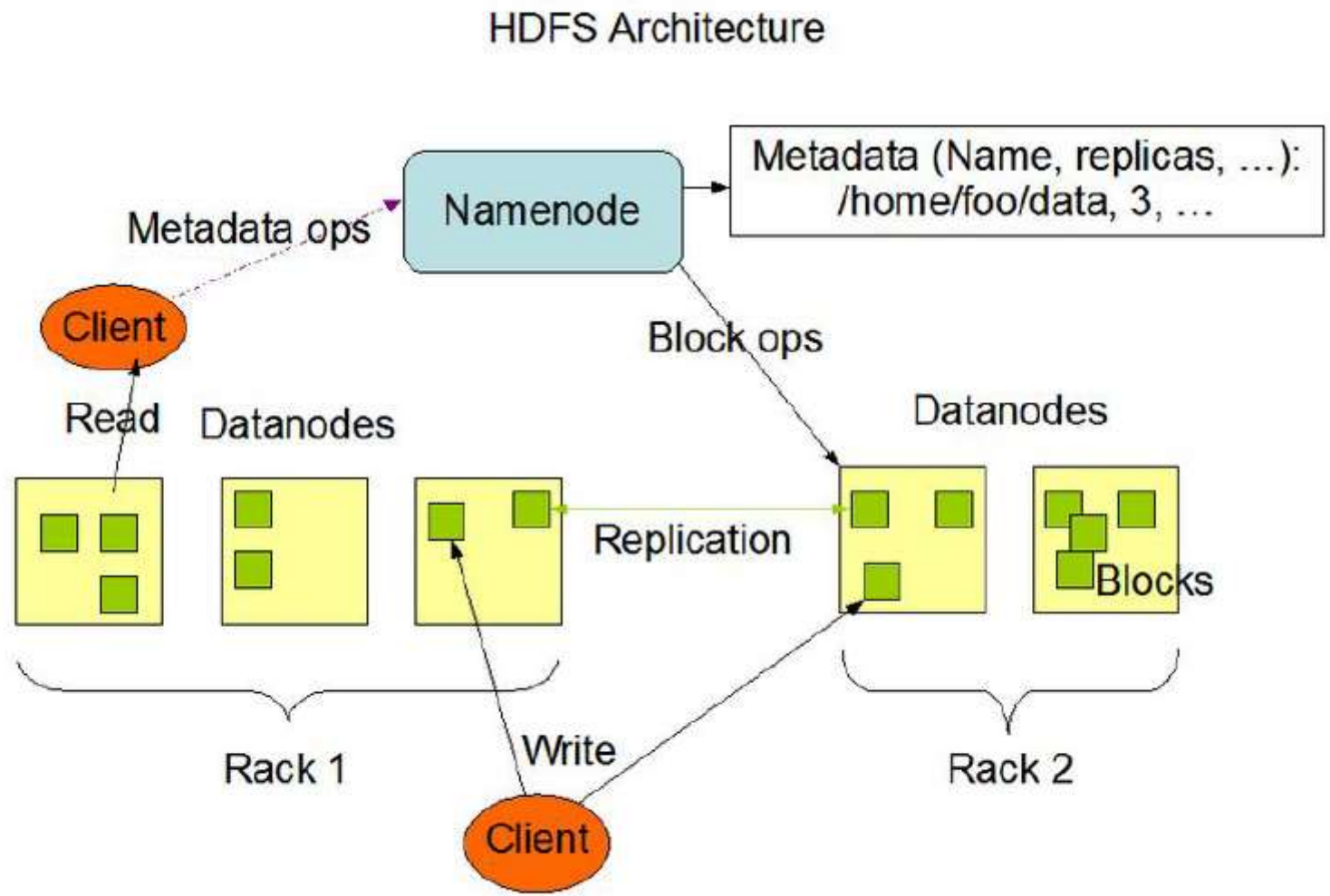
HDFS is a block-structured file system: individual files are broken into blocks of a fixed size. These blocks are stored across a cluster of one or more machines with data storage capacity. Individual machines in the cluster are referred to as **DataNodes**. A file can be made of several blocks, and they are not necessarily stored on the same machine; the target machines which hold each block are chosen randomly on a block-by-block basis. Thus access to a file may require the cooperation of multiple machines, but supports file sizes far larger than a single-machine DFS; individual files can require more space than a single hard drive could hold.

https://developer.yahoo.com/hadoop/tutorial/module2.html

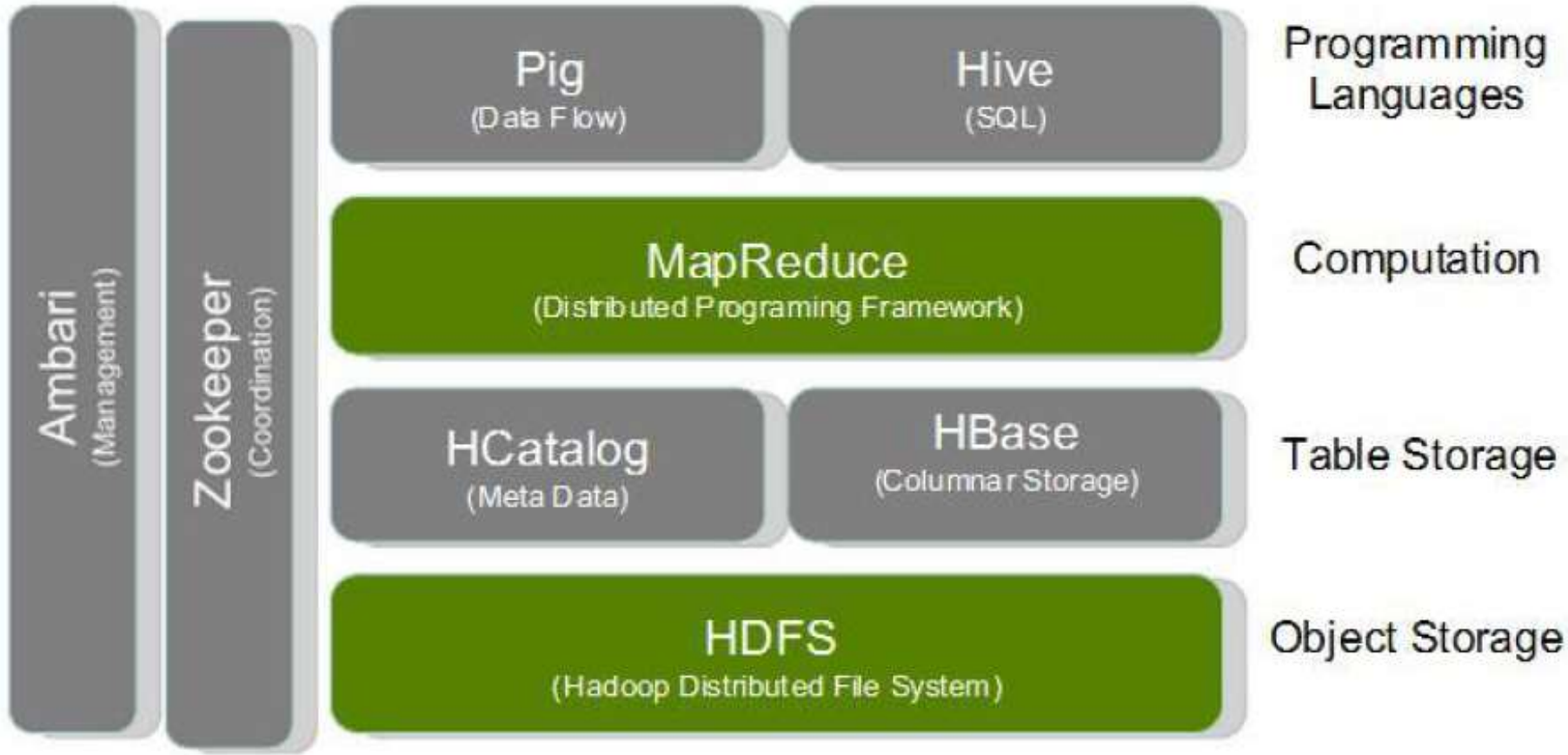
HDFS

HOW HDFS WORKS

An HDFS cluster is comprised of a NameNode, which manages the cluster metadata, and DataNodes that store the data. Files and directories are represented on the NameNode by inodes. Inodes record attributes like permissions, modification and access times, or namespace and disk space quotas.



Big Data: Hadoop Stack



NoSQL

A NoSQL (Not-only-SQL) database is one that has been designed to store, distribute and access data using methods that differ from relational databases (RDBMS's). NoSQL technology was originally created and used by Internet leaders such as Facebook, Google, Amazon, and others who required database management systems that could write and read data anywhere in the world, while scaling and delivering performance across massive data sets and millions of users.

NoSQL

https://www.datastax.com/nosql-databases

Use an RDBMS when you need/have…	Use NoSQL when you need/have…
Centralized applications (e.g. ERP)	Decentralized applications (e.g. Web, mobile and IOT)
Moderate to high availability	Continuous availability; no downtime
Moderate velocity data	High velocity data (devices, sensors, etc.)
Data coming in from one/few locations	Data coming in from many locations
Primarily structured data	Structured, with semi/unstructured
Complex/nested transactions	Simple transactions
Primary concern is scaling reads	Concern is to scale both writes and reads
Philosophy of scaling up for more users/data	Philosophy of scaling out for more users/data
To maintain moderate data volumes with purge	To maintain high data volumes; retain forever

Basic Statistics

Some of the basic statistics that every data scientist should know are given here. This assumes rudimentary basic knowledge of statistics (like measures of central tendency or variation) and basic familiarity with some of the terminology used by statisticians.

- **Random Sampling**- In truly random sampling,the sample should be representative of the entire data. RAndom sampling remains of relevance in the era of Big Data and Cloud Computing

- **Distributions-** A data scientist should know the distributions (normal, Poisson, Chi Square, F) and also how to determine the distribution of data.

- **Hypothesis Testing** - Hypothesis testing is meant for testing assumptions statistically regarding values of central tendency (mean, median) or variation. A good example of an easy to use software for statistical testing is the “test” tab in the Rattle GUI in R.

- **Outliers-** Checking for outliers is a good way for a data scientist to see anomalies as well as identify data quality. The box plot (exploratory data analysis) and the outlierTest function from car package (Bonferroni Outlier Test) is how statistical rigor can be maintained to outlier detection.

Basic Techniques

Some of the basic techniques that a data scientist must know are listed as follows-

- Text Mining - In text mining , text data is analyzed for frequencies, associations and corelation for predictive purposes. The tm package from R greatly helps with text mining.
- Sentiment Analysis- In sentiment analysis the text data is classified based on a sentiment lexicography (eg which says happy is less positive than delighted but more positive than sad) to create sentiment scores of the text data mined.
- Social Network Analysis- In social network analysis, the direction of relationships, the quantum of messages and the study of nodes,edges and graphs is done to give insights..
- Time Series Forecasting- Data is said to be auto regressive with regards to time if a future value is dependent on a current value for a variable. Technqiues such as ARIMA and exponential smoothing and R packages like forecast greatly assist in time series forecasting.
- Web Analytics
- Social Media Analytics
- Data Mining or Machine Learning

Data Science Tools

- R
- Python
- Tableau
- Spark with ML
- Hadoop (Pig and Hive)
- SAS
- SQL

Statistical Software Landscape

SAS

Python (Pandas)

IBM SPSS

R

Julia

Clojure

Octave

Matlab

JMP

E views

What is SAS?

- **SAS** (Statistical Analysis System
- Software suite developed by SAS Institute for advanced analytics, business intelligence, data management, and predictive analytics
- Developed at North Carolina State University from 1966 until 1976, when SAS Institute was incorporated.
- Further developed in the 1980s and 1990s with the additional statistical procedures and

Components of SAS

Currently, SAS has more than 200 components, some of them are –

- Base SAS – Basic procedures and data management
- SAS/STAT – Statistical analysis
- SAS/GRAPH – Graphics and presentation
- SAS/OR – Operations research
- SAS/ETS – Econometrics and Time Series Analysis

Python

What is Python

Python is a widely used general-purpose, high-level programming languagn

Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java.

Python is used widely

https://www.python.org/about/success/

Python

http://www.astro.up.pt/~sous

Python 2.7 Quick Reference Sheet

ver 2.01 – 110105 (sjd)

Interactive Help in Python Shell

help()	Invoke interactive help
help(*m*)	Display help for module *m*
help(*f*)	Display help for function *f*
dir(*m*)	Display names in module *m*

Small Operator Precedence Table

func_name(args, ...)	Function call
x[index : index]	Slicing
x[index]	Indexing
x.attribute	Attribute reference
**	Exponentiation
*, /, %	Multiply, divide, mod
+, -	Add, subtract
>, <, <=, >=, !=, ==	Comparison
in, not in	Membership tests
not, and, or	Boolean operators NOT, AND, OR

Module Import

import *module_name*
from *module_name* import *name* , ...
from *module_name* import *

Common Data Types

Type	Description	Literal Ex
int	32-bit Integer	3, -4
long	Integer > 32 bits	101L
float	Floating point number	3.0, -6.55
complex	Complex number	1.2J
bool	Boolean	True, False
str	Character sequence	"Python"
tuple	Immutable sequence	(2, 4, 7)
list	Mutable sequence	[2, x, 3.1]
dict	Mapping	{ x:2, y:5 }

Common Syntax Structures

Assignment Statement
var = exp

Console Input/Output
var = input([*prompt*])
var = raw_input([*prompt*])
print *exp*[,] ...

Selection
if (*boolean_exp*):
  stmt ...
[elif (*boolean_exp*):
  stmt ...] ...
[else:
  stmt ...]

Repetition
while (*boolean_exp*):
  stmt ...

Traversal
for *var* in *traversable_object*:
  stmt ...

Function Definition
def *function_name*(*parmameters*):
  stmt ...

Function Call
function_name(*arguments*)

Class Definition
class *Class_name* [(*super_class*)]:
  [*class variables*]
  def *method_name*(self, *parameters*):
    stmt

Object Instantiation
obj_ref = *Class_name*(*arguments*)

Method Invocation
obj_ref.method_name(*arguments*)

Exception Handling
try:
  stmt ...
except [*exception_type*] [, *var*]:
  stmt ...

Common Built-in Functions

Function	Returns
abs(*x*)	Absolute value of *x*
dict()	Empty dictionary, eg: d = dict()
float(*x*)	int or string *x* as float
id(*obj*)	memory addr of *obj*
int (*x*)	float or string *x* as int
len(*s*)	Number of items in sequence *s*
list()	Empty list, eg: m = list()
max(*s*)	Maximum value of items in *s*
min(*s*)	Minimum value of items in *s*
open(*f*)	Open filename *f* for input
ord(*c*)	ASCII code of *c*
pow(*x,y*)	x ** y
range(*x*)	A list of *x* ints 0 to *x* - 1
round(*x,n*)	float *x* rounded to *n* places
str(*obj*)	str representation of *obj*
sum(*s*)	Sum of numeric sequence *s*
tuple(*items*)	tuple of *items*
type(*obj*)	Data type of *obj*

Common Math Module Functions

Function	Returns (all float)
ceil(*x*)	Smallest whole nbr >= *x*
cos(*x*)	Cosine of *x* radians
degrees(*x*)	*x* radians in degrees
radians(*x*)	*x* degrees in radians
exp(*x*)	e ** *x*
floor(*x*)	Largest whole nbr <= *x*
hypot(*x, y*)	sqrt(*x* * *x* + *y* * *y*)
log(*x* [, *base*])	Log of *x* to *base* or natural log if *base* not given
pow(*x, y*)	x ** y
sin(*x*)	Sine of *x* radians
sqrt(*x*)	Positive square root of *x*
tan(*x*)	Tangent of *x* radians
pi	Math constant pi to 15 sig figs
e	Math constant e to 15 sig figs

Python

https://s3.amazonaws.com/quandl-static-content/Documents/Quandl+-+Par

NumPy / SciPy / Pandas Cheat Sheet

python

NumPy / SciPy

`arr = array([])`	Create numpy array.
`arr.shape`	Shape of an array.
`convolve(a,b)`	Linear convolution of two sequences.
`arr.reshape()`	Reshape array.
`sum(arr)`	Sum all elements of array.
`mean(arr)`	Compute mean of array.
`std(arr)`	Compute standard deviation of array.
`dot(arr1,arr2)`	Compute inner product of two arrays.
`vectorize()`	Turn a scalar function into one which accepts & returns vectors.

Pandas

Create Structures

`s = Series (data, index)`	Create a Series.
`df = DataFrame (data, index, columns)`	Create a Dataframe.
`p = Panel (data, items, major_axis, minor_axis)`	Create a Panel.

DataFrame commands

`df[col]`	Select column.
`df.iloc[label]`	Select row by label.
`df.index`	Return DataFrame index.
`df.drop()`	Delete given row or column. Pass axis=1 for co
`df1 = df1.reindex_like(df1,df2)`	Reindex df1 with index of df2.
`df.reset_index()`	Reset index, putting old index in column name