University of East London
Pioneering Futures Since 1898

Big Data Analytics

Dr Amin Karami

a.karami@uel.ac.uk

www.aminkarami.com

"THAT'S your Ark for the Big Data flood? Noah, you will need a lot more storage space!"

CN7022 – Week 1
4 October 2019

Outline

- The Overview of the Module
- What is Big Data?
- Challenges of Big Data
- What is Hadoop and its structure?
- Importance of Hadoop
- Hadoop ecosystem

Module Aim

- This module aims to provide students with the core theoretical and practical background required for big data analytics and developing big data systems. It will provide you with an insight into areas of big data management and advanced analytics. You will develop in-depth practical skills through using tools and techniques from the forefront of the emerging field of data analytics.

Module Team

- **Lectures:**
  - Dr Amin Karami (**module leader**)

    a.karami@uel.ac.uk (office: EB.1.86)

- **Practical sessions:**
  - Dr Sin Wee Lee (s.w.lee@uel.ac.uk, office: EB.1.104)
  - Dr Mustansar Ghazanfar (m.ghazanfar@uel.ac.uk, office: EB.1.104)
  - Two MSc students from last year (Diana and Rajathi)

University of East London

Module Organization

- **Lecture**: Friday (11:00-13:00), weekly, Main Lecture Theatre
- **Practical**: (KD.2.14, KD.2.15, KD.2.28E), find your practical time slot and the lab room at https://ueltt.uel.ac.uk
  - **Friday** (14:00-17:00), begins from **11th Oct**.
    - ✓ Team: Amin, Mustansar, Rajathi
  - **Tuesday** (14:00-17:00), begins from **15th Oct**.
    - ✓ Team: Amin, Sin Wee, Mustansar
  - **Wednesday** (10:00-13:00 and 14:00-17:00), begins from **16th Oct**.
    - ✓ Team: Amin, Diana, Rajathi

Module Assessment

- A **group-based** Coursework (100%): A practical work about Big Data Analytics
- **Presentations**: week 12 (17th, 18th, 20th December 2019)
- **Turnitin Submission**: 27th December 2019
- Groups include 3-4 persons. You should make it up by end of October 2019.
- All the group members must attend in the presentation. If you do not attend, you fail the module.
- Every member of group will be assessed individually.

University of East London

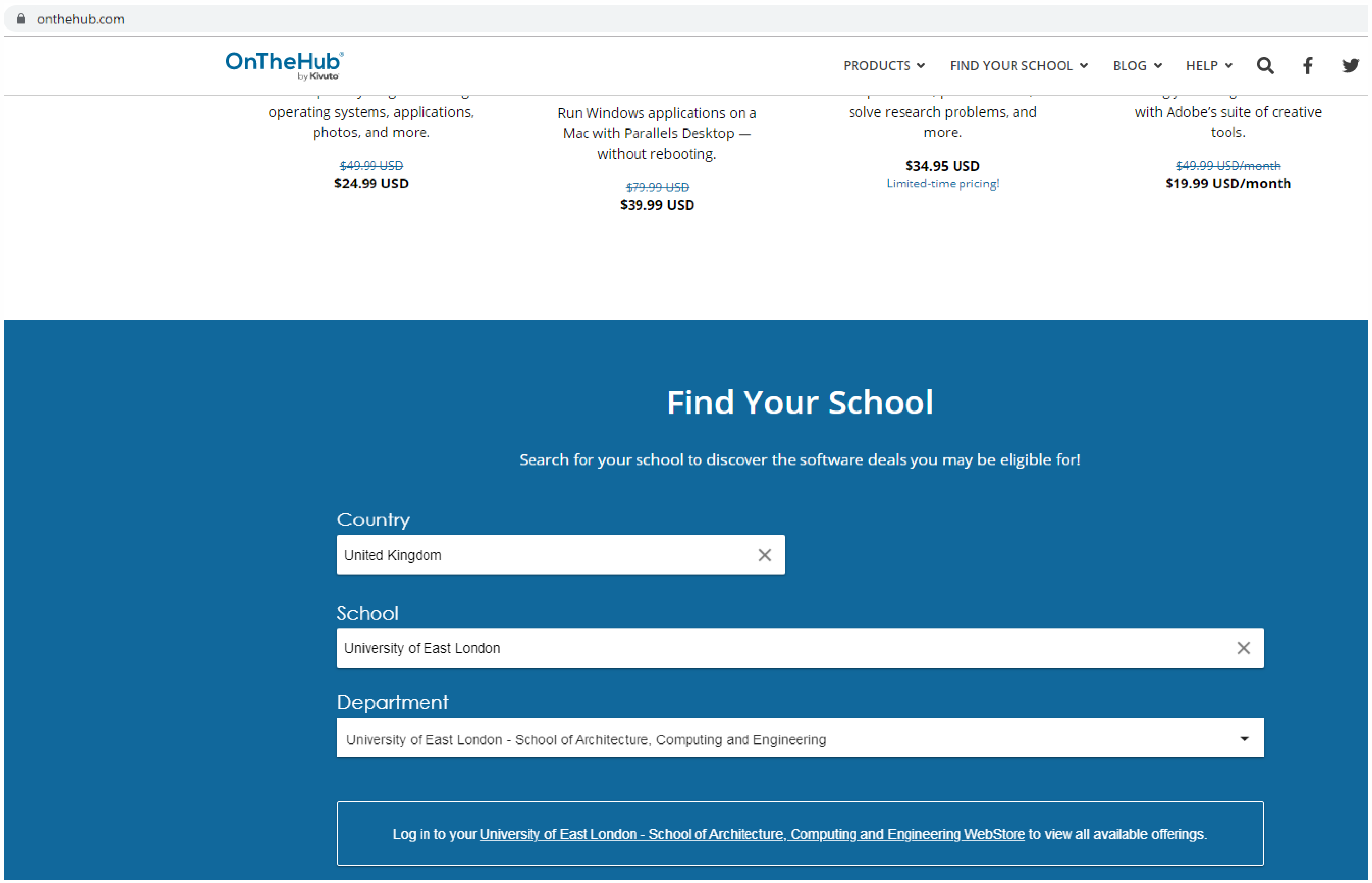
Tentative Module Contents

- **Week 1 (4th Oct.):** Module Introduction + Fundamentals of Big Data
- **Week 2 (11th Oct.):** Hadoop + HDFS
- **Week 3 (18th Oct.):** Hadoop + MapReduce
- **Week 4 (25th Oct.):** Data Acquisition by Sqoop , Oozie (workflow scheduler) + Zookeeper (distributed sync. in clusters)
- **Week 5 (1st Nov.):** Hive for big data query and analysis
- **Week 6 (8th Nov.):** Hive tutorial [CRWK part 1]
- **Week 7 (15th Nov.):** Spark + Python tutorial in Jupyter
- **Week 8 (22nd Nov.):** PySpark tutorial
- **Week 9 (29th Nov.):** Advanced Analytics in PySpark [CRWK part 2]
- **Weeks 10 and 11 (6th & 13th Dec.):** CRWK Preparation (lab)
- **Week 12 (17th, 18th & 20th Dec.):** CRWK Presentation

University of East London

Software Suites for Big Data

- **VMWare Workstation** (v14 or higher)



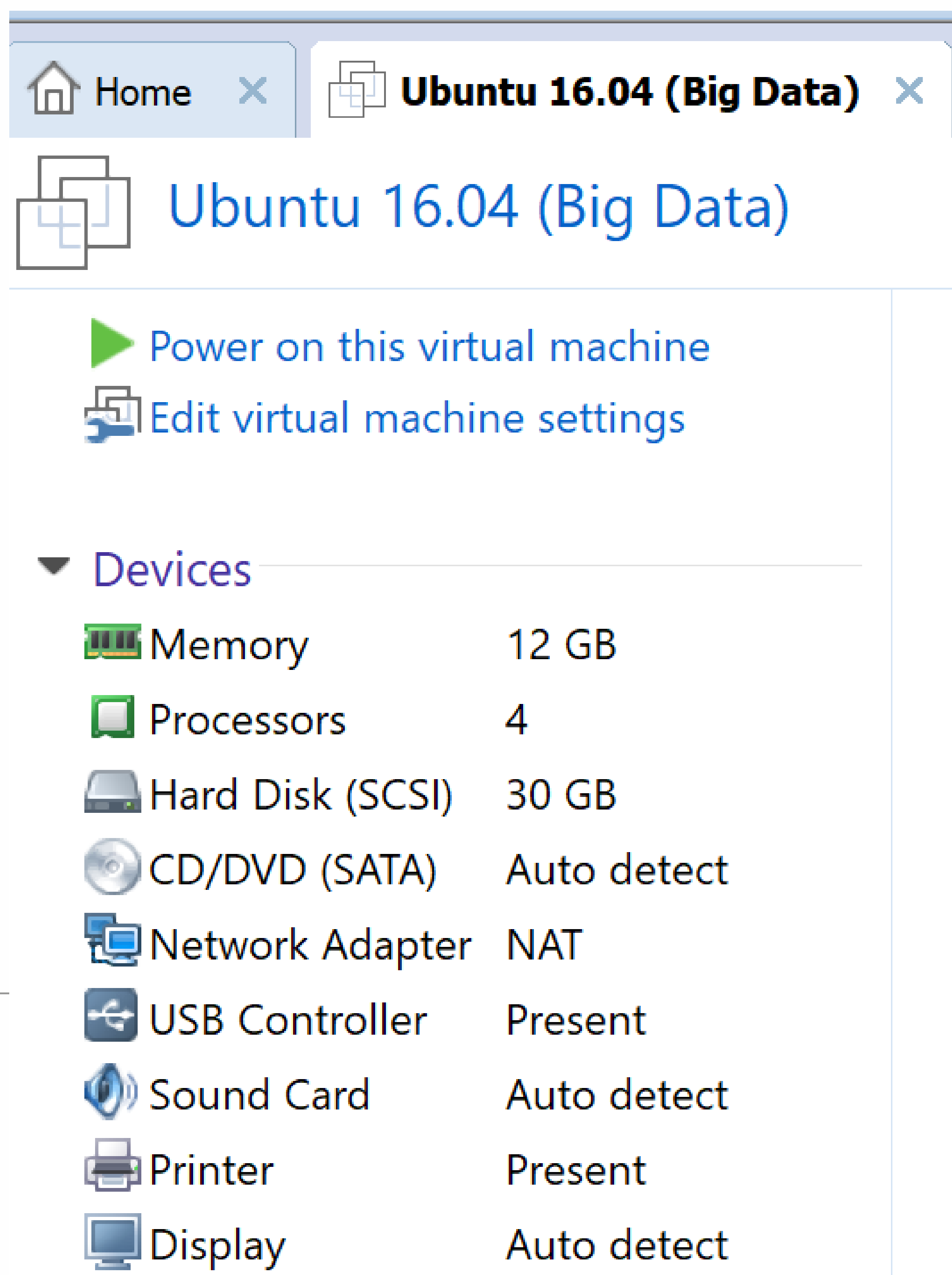
University of East London

Software Suites for Big Data

- **Ubuntu 16.04**

✓ Download, unzip and then open it in VMWare:

https://www.dropbox.com/s/7ibvubx25e16b7j/BigDataClass.zip?dl=1



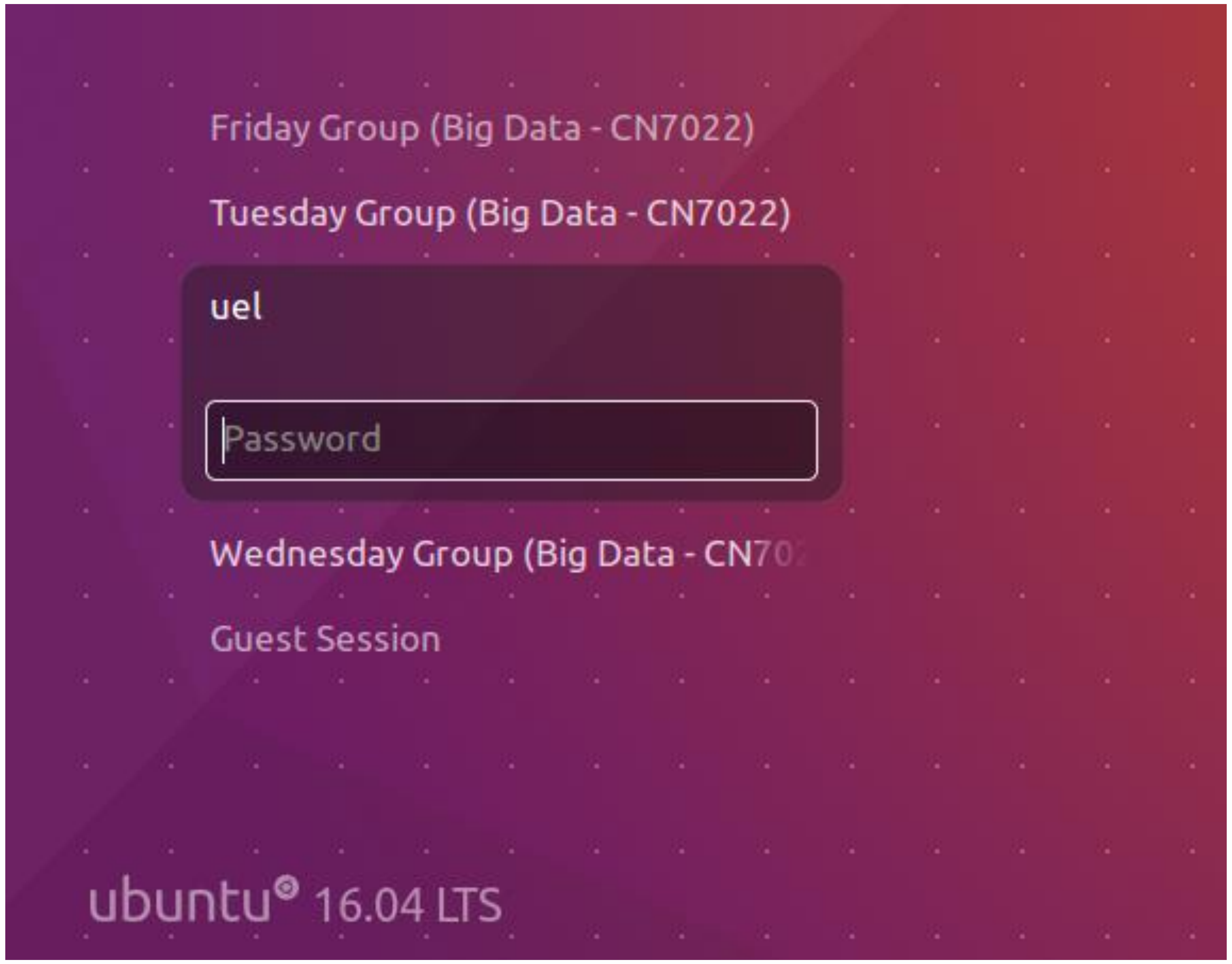
✓ **Minimum requirements:**

- ❑ CPU: 4 processors
- ❑ RAM: 8GB-12GB
- ❑ HDD: 30 GB

University of East London

Software Suites for Big Data

- Once you power on the Ubuntu machine, you need to log in to your own practical week:



Passwords:

- Friday Group: **Friday**
- Tuesday Group: **Tuesday**
- Wednesday Group: **Wednesday**

University of East London

Software Suites for Big Data

- **Cloudera (CDH 5.13)**

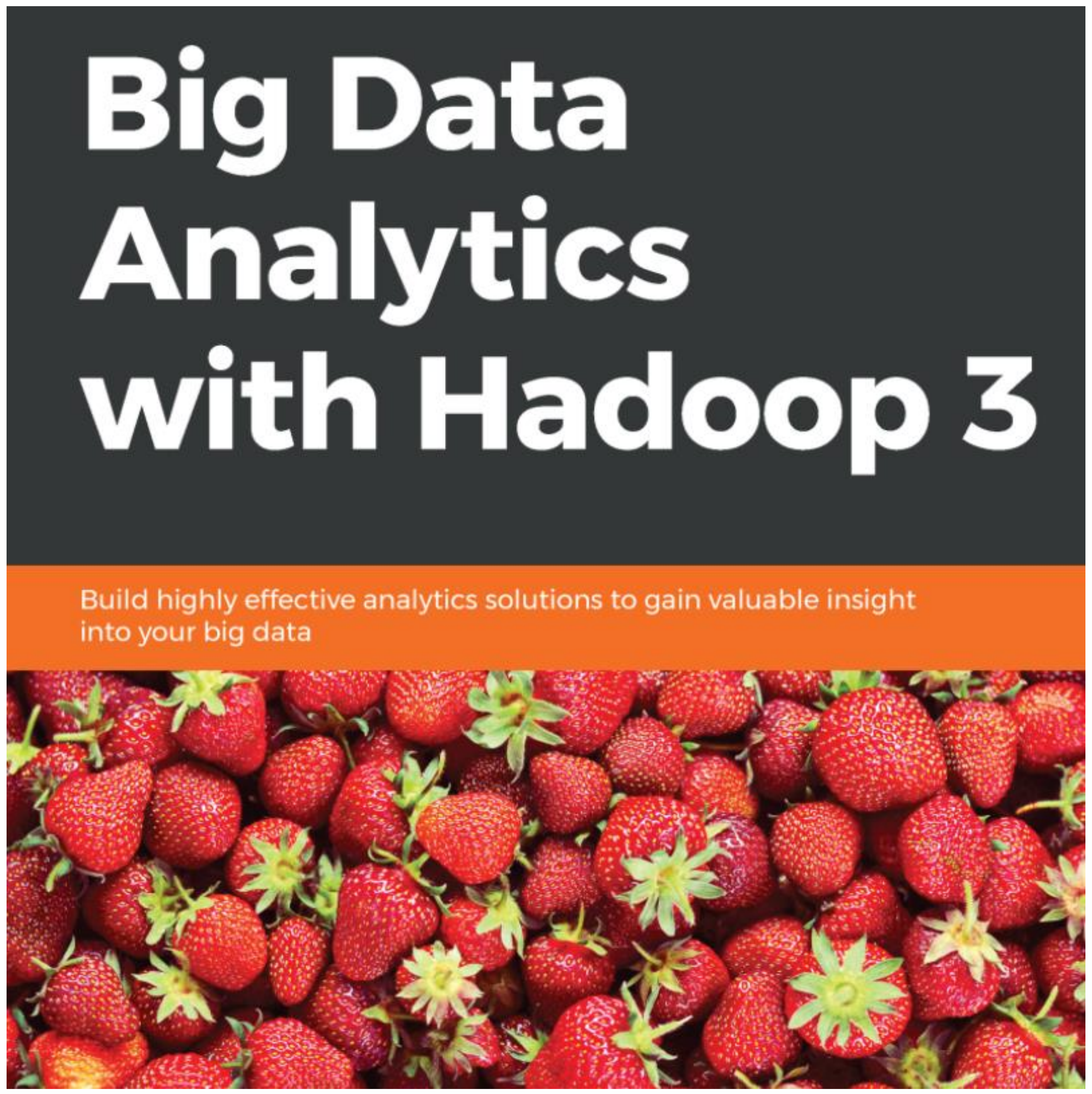
  ✓ CDH (Cloudera Distribution Hadoop) is the open-source Apache Hadoop distribution project provided by Cloudera Inc.

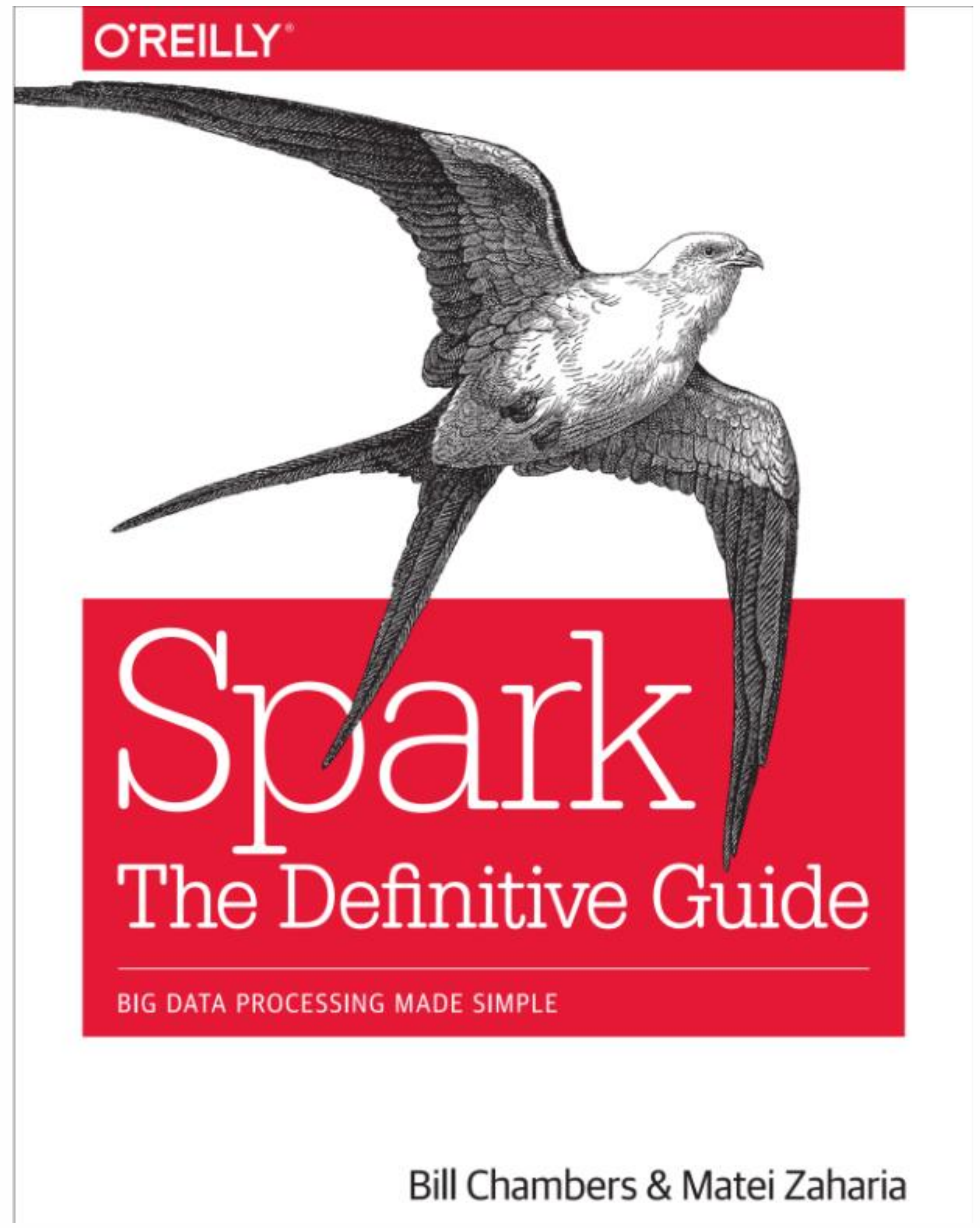
  ✓ Download, unzip and then open it in VMWare:

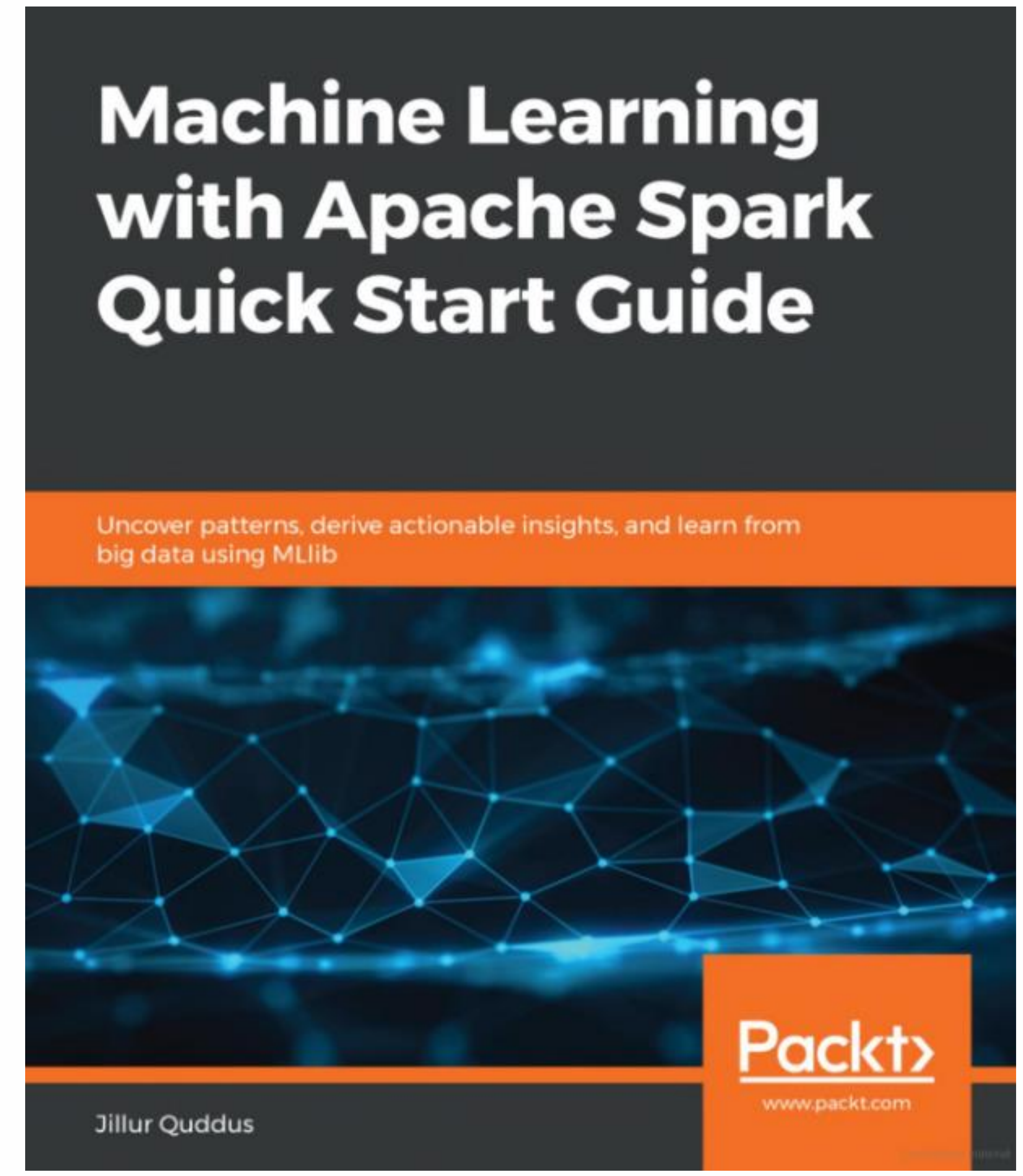
  https://www.dropbox.com/s/0wm0yfv9mhxczey/cloudera-quickstart-vm-5.13.0-0-vmware.zip?dl=1

  ✓ We will use it for Sqoop, Hive and CRWK.

Resources


Big Data Analytics with Hadoop 3
Build highly effective analytics solutions to gain valuable insight into your big data


O'REILLY
Spark
The Definitive Guide
BIG DATA PROCESSING MADE SIMPLE
Bill Chambers & Matei Zaharia


Machine Learning with Apache Spark Quick Start Guide
Uncover patterns, derive actionable insights, and learn from big data using MLlib
Packt
www.packt.com
Jillur Quddus

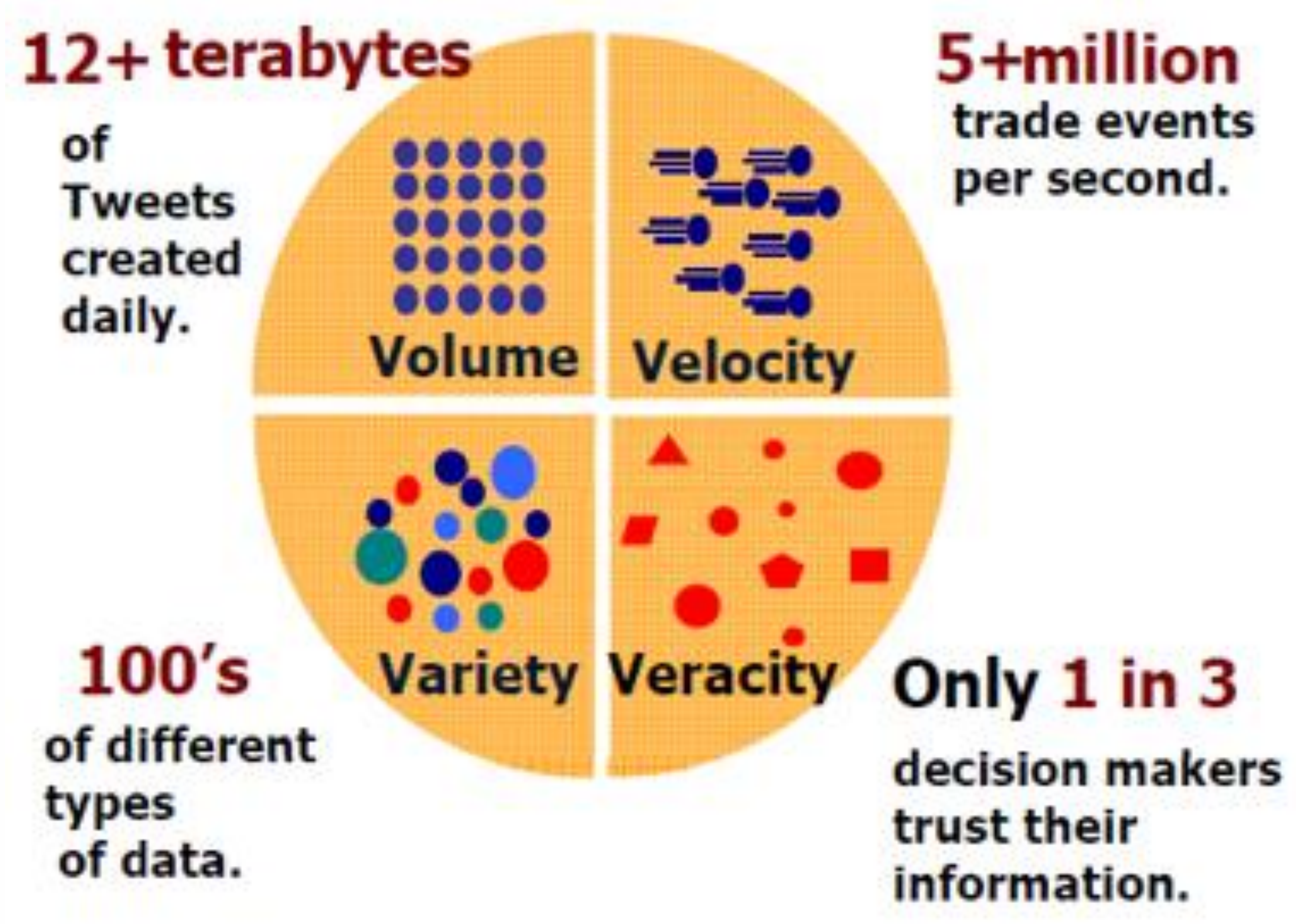
Research Background

- Big Data Technologies and Methodologies
- Computational Intelligence
- Intrusion Detection Systems
- Next Generation Internet: NDN and IoT

Please visit my Google Scholar page for more information.

If you are interested in research and have already an idea, drop me an email to discuss it: a.karami@uel.ac.uk

Where does Big Data come from?



- The number of web pages indexed by Google: one million in 1998, have exceeded one trillion in 2008, and its expansion is accelerated by appearance of the social networks.
- The amount of mobile data traffic is expected to grow to 10.8 Exabyte per month by 2016.
- More than 65 billion devices were connected to the Internet by 2010, and this number will go up to 230 billion by 2020.
- Many companies are moving towards using Cloud services to access Big Data analytical tools.
- Open Source communities are growing up.

Memory Size Chart

Symbol	Abbr.	Value (byte)	Example
Bit	b	10^0	Single binary digit (0 or 1)
Byte	B	10^1	8 bits = 2^3
Kilobyte	KB	10^3	1,024 B = 2^{10}
Megabyte	MB	10^6 (Million)	1,024 KB = 2^{20}
Gigabyte	GB	10^9 (Billion)	1,024 MB = 2^{30}
Terabyte	TB	10^{12} (Trillion)	1,024 GB = 2^{40}
Petabyte	PB	10^{15} (Quadrillion)	1,024 TB = 2^{50}
Exabyte	EB	10^{18} (Quintillion)	1,024 PB = 2^{60}
Zettabyte	ZB	10^{21} (Sextillion)	1,024 EB = 2^{70}
Yottabyte	YB	10^{24} (Septillion)	1,024 ZB = 2^{80}

University of East London

Reading big data: ordinary machines

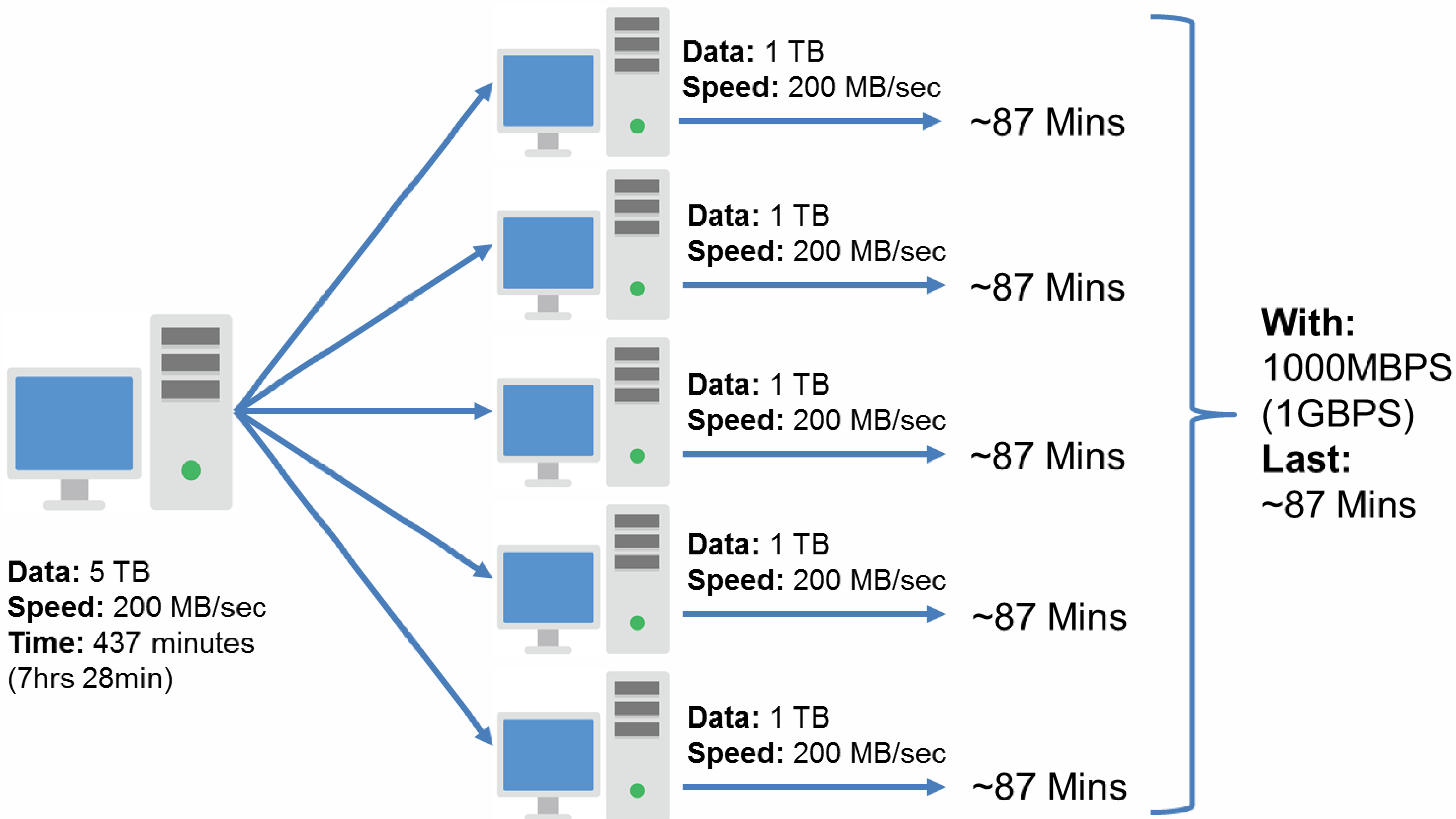
Data: 5 TB
Speed of machine: 200 MB/sec

How much time will take to process 5 TB data?

$$Processing = \frac{5 \ * 1024 \ * 1024}{200 \ * 60} = approx.\ 437 \ \ Mins$$

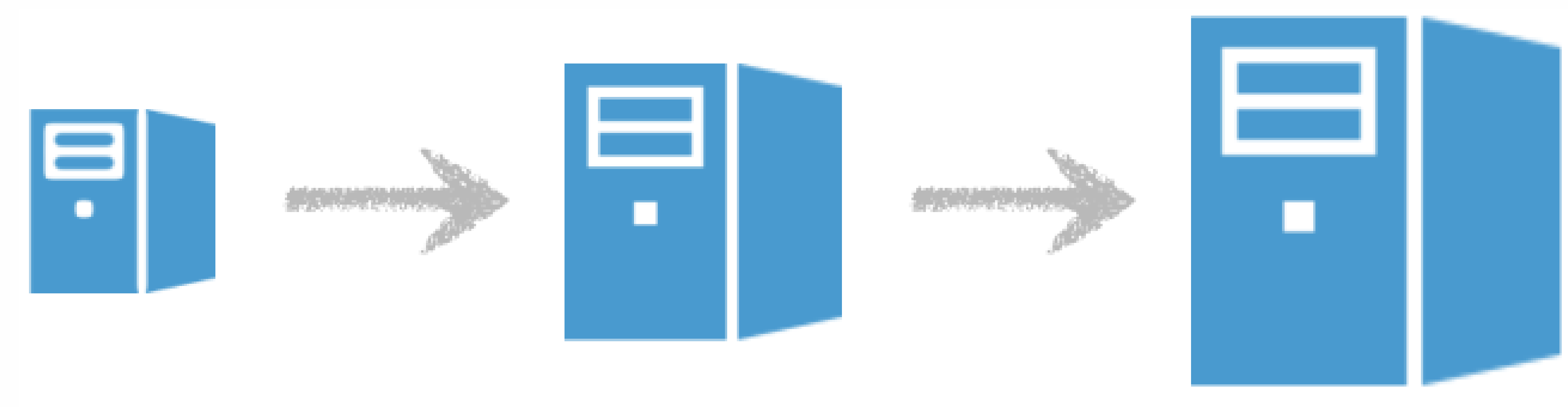
Do you think does it reasonable? Could we process our data?
Certainly, NO

Reading big data: parallel processing



How to process Big Data?

- **Scale Up** (scale vertically): adding resources to a single node in a system. [expensive, not possible for massive data.]



- **Scale Out** (scale horizontally): adding more nodes to a system. [expensive, fault tolerance, development problems]

University of East London

Parallel Processing Challenges

1. **Dividing and Distributing** Divide 5TB data into 1TB and send them into workstations
2. **Parallel Processing** Run all the workstations in parallel without delay and fault
3. **Combining Results** We need combine all the results from each workstation
4. **Costly Servers** more workstations take more cost

How to overcome these challenges?

What is Hadoop?

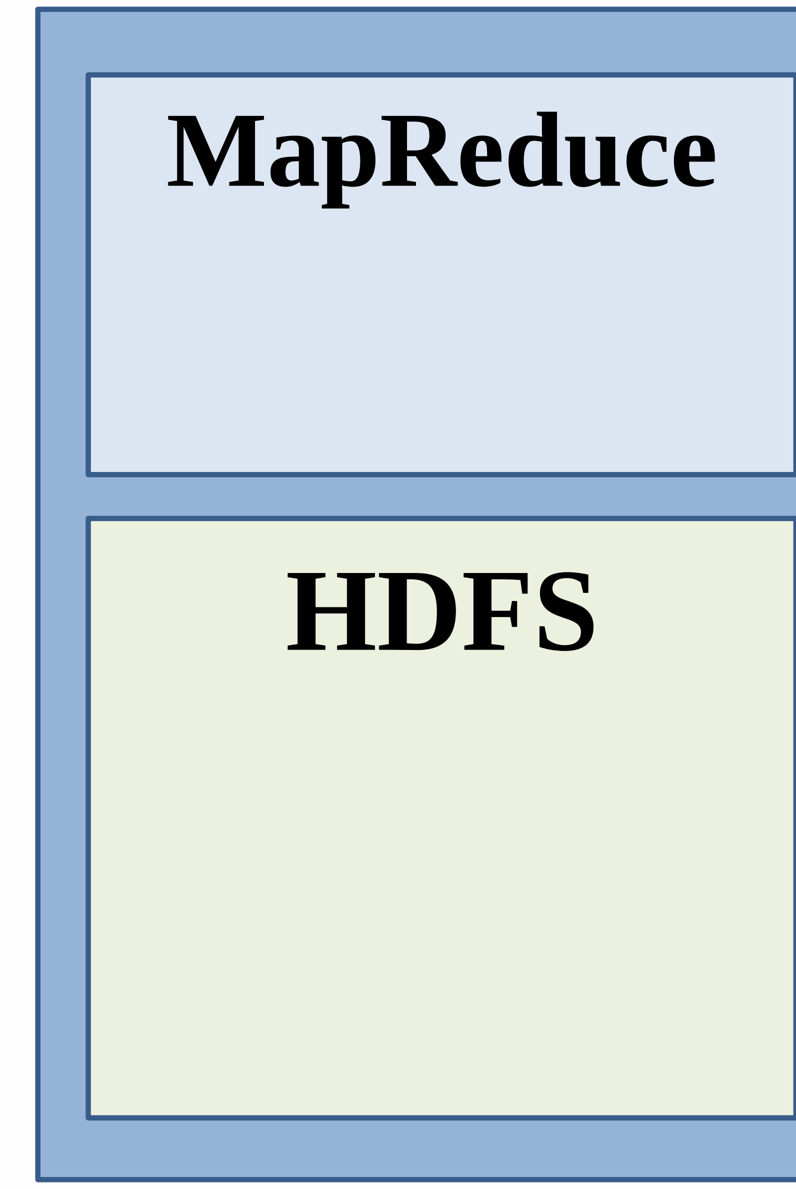
- **Apache Hadoop** is an open-source framework written in Java, that supports the processing and storing of massive amount of data (referred as Big Data) on clusters of commodity hardware.

- **Hadoop** was created by computer scientists Doug Cutting and Mike Cafarella in 2006 to support distribution for the **Nutch search engine**. It was inspired by **Google File System** (Oct. 2003) **Google MapReduce** (Dec. 2004), a software framework in which an application is broken down into numerous small part.

What is Hadoop?

- **Apache Hadoop 1.0** became publically available in **November 2012** as part of the Apache project sponsored by the Apache Software Foundation.
- The latest stable release is Apache Hadoop 3.2.0 which is available (Sept. 2019) in https://hadoop.apache.org

Hadoop Pieces



1. **HDFS** (Hadoop Distributed File System) is the data part of Hadoop which is the primary storage system used by Hadoop applications.

2. **MapReduce** is the processing part of Hadoop.

HDFS components:

- **NameNode (Master)** is a centrepiece of HDFS that stores the metadata of HDFS. It keeps a list of **blocks and its location** for files stored in HDFS. It is a single point of failure. It needs more memory (RAM) to be executed.
- **DataNode (Slave)** is responsible for storing the actual data in HDFS. It is usually configured with a lot of hard disk.

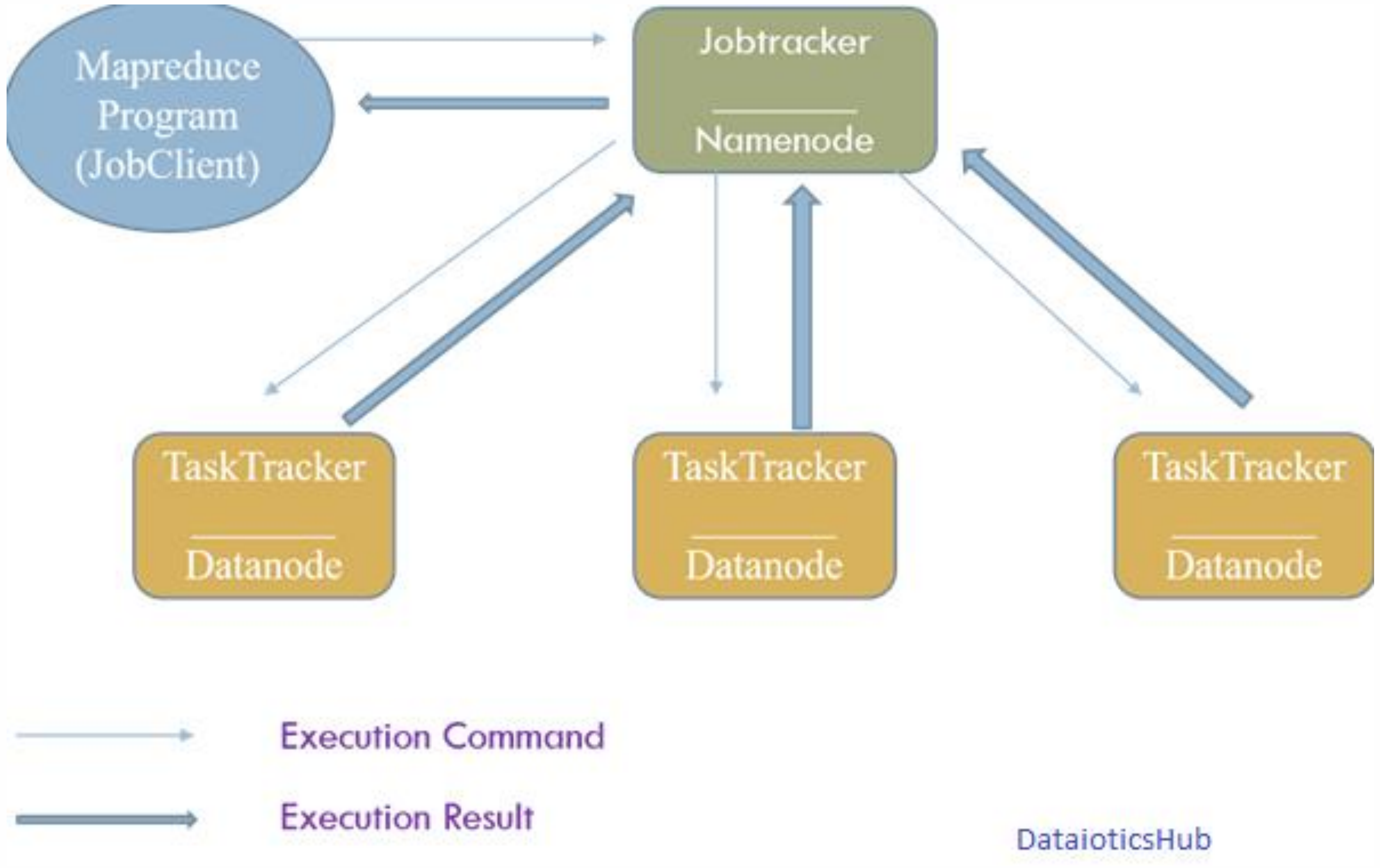
HDFS components

- **NameNode**
  - Keep all namespaces in memory
  - Maintains metadata
  - Monitors data node health
  - Replicates missing blocks
  - Maintains mapping of list of blocks and locations
  - Maintains authorization and authentication data
  - Manages checkpoints
- **DataNode**
  - Serves data blocks directly to clients
  - Handles block storage on multiple volumes, block integrity
  - Periodically sends heartbeats and block reports to NameNode
  - Stores blocks as underlying OS's file

MapReduce components

- **JobTracker**: A coordinator for tasks. It keeps track of jobs being run on clusters. Each cluster has a single job tracker. It settles in the Memory.

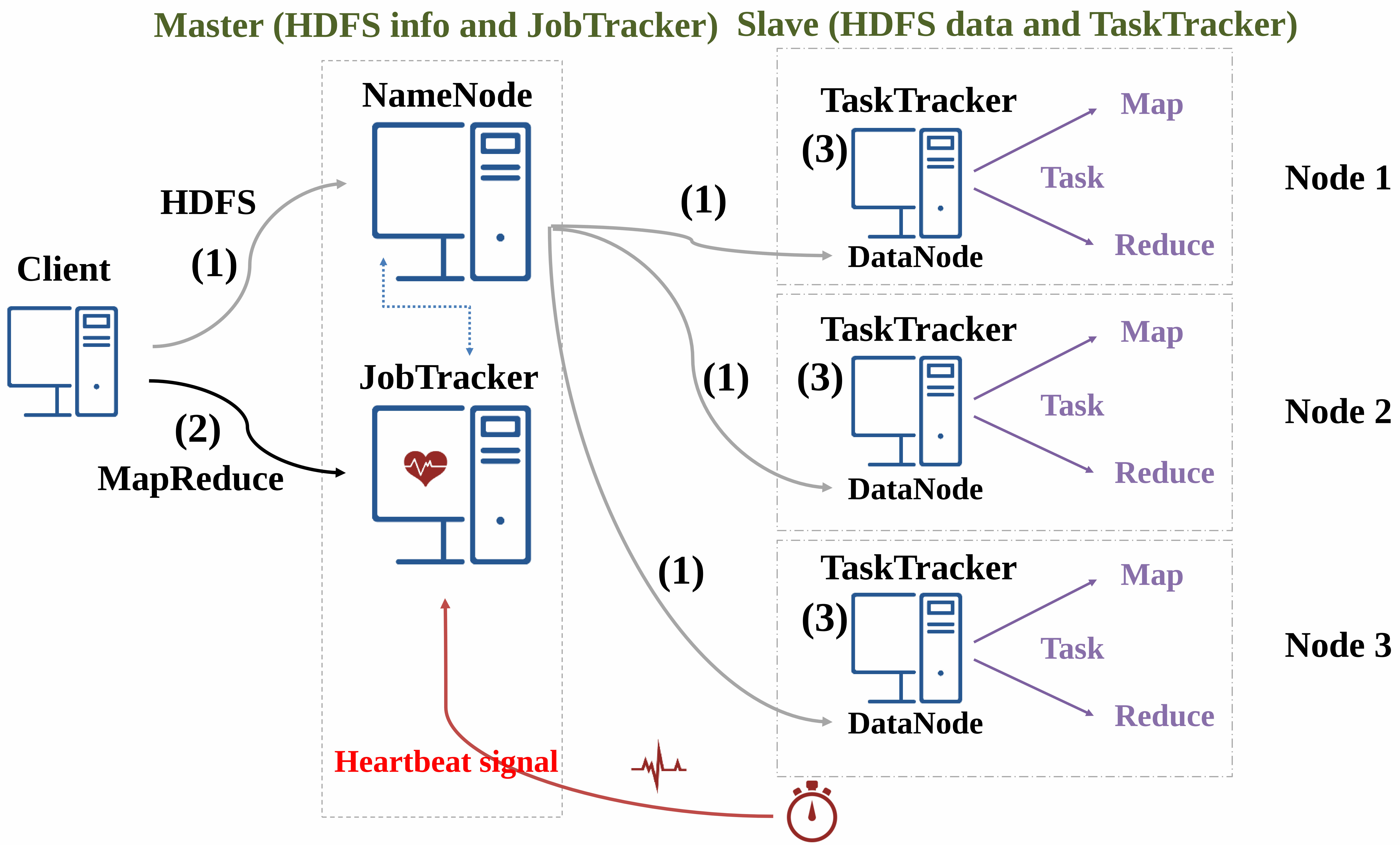
- **TaskTracker**: A node that accepts tasks (e.g., **Map, Reduce, Shuffle**). Each cluster can have multiple task trackers.



MapReduce components

- **Job Tracker**
  - Client applications submit jobs to JobTracker
  - JobTracker talks to NameNode to determine data location
  - JobTracker locates TaskTracker nodes with available slots at or near data.
  - JobTracker submits work to chosen TaskTracker nodes.
  - When work is completed, JobTracker updates status.
- **Task Tracker nodes are monitored:**
  - If they do not submit heartbeat signals periodically, they are deemed to have failed and work is scheduled on different TaskTracker
- **Task Tracker will notify JobTracker when a task fails:**
  - JobTracker may resubmit job elsewhere, mark that specific record as something to avoid, and even blacklist TaskTracker as unreliable.

Hadoop Scheme



JobTracker Workflow

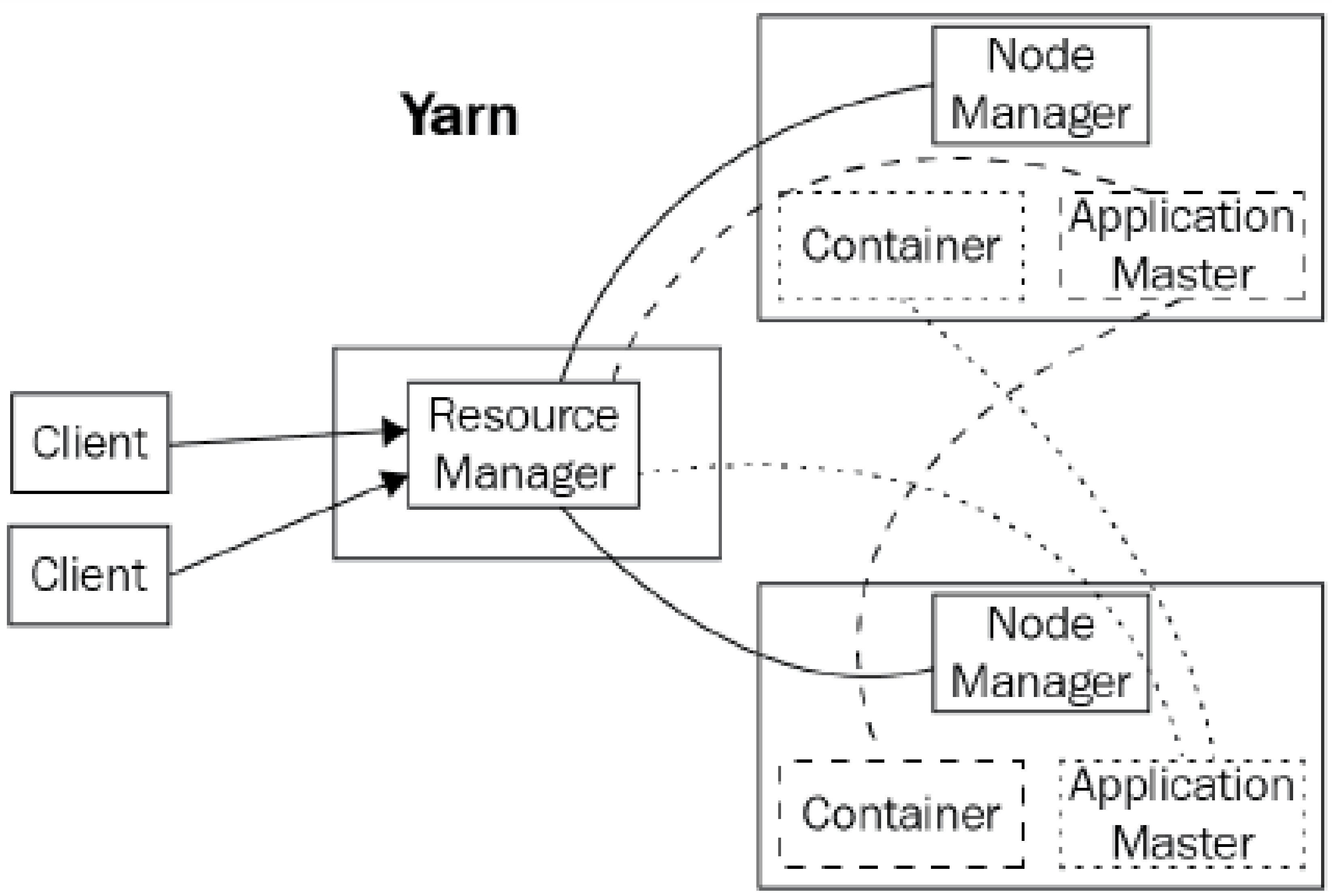
1. User applications submit jobs to the **JobTracker**
2. **JobTracker** talks to the **NameNode** to determine the location of the data
3. **JobTracker** locates **TaskTracker** with available slots in clusters
4. **JobTracker** submits the work to the chosen **TaskTracker** nodes
5. **TaskTracker** nodes are monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different **TaskTracker**
6. A **TaskTracker** will notify the **JobTracker** when a task fails. The **JobTracker** decides what to do then: it may resubmit the job elsewhere, it may mark that specific record as something to avoid, and it may even blacklist the **TaskTracker** as unreliable
7. When the work is completed, the **JobTracker** updates its status
8. The **JobTracker** is a point of failure for the Hadoop **MapReduce** service. If it goes down, all running jobs are halted

TaskTracker Workflow

1. A **TaskTracker** is a node in the cluster that accepts tasks - Map, Reduce and Shuffle operations from a **JobTracker**
2. When the **JobTracker** tries to find somewhere to schedule a task within the MapReduce operations, it first looks for an empty slot on the same server that hosts the **DataNode** containing the data, and if not, it looks for an empty slot on a machine in the same rack
3. The **TaskTracker** spawns a separate JVM processes to do the actual work; this is to ensure that process failure does not take down the task tracker
4. The **TaskTracker** monitors these spawned processes, capturing the output and exit codes. When the process finishes, successfully or not, the tracker notifies the **JobTracker**.
5. The **TaskTrackers** also send out heartbeat messages to the **JobTracker**, usually every few minutes, to reassure the **JobTracker** that it is still alive. These message also inform the **JobTracker** of the number of available slots, so the **JobTracker** can stay up to date with where in the cluster work can be delegated.

Apache Hadoop YARN

- The fundamental idea of YARN (Yet Another Resource Negotiator) is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.
- Hadoop YARN stands for '**next-generation MapReduce**' that help to **Manage resources and schedule tasks.**
- YARN is now available in Hadoop 2.0 and 3.0.



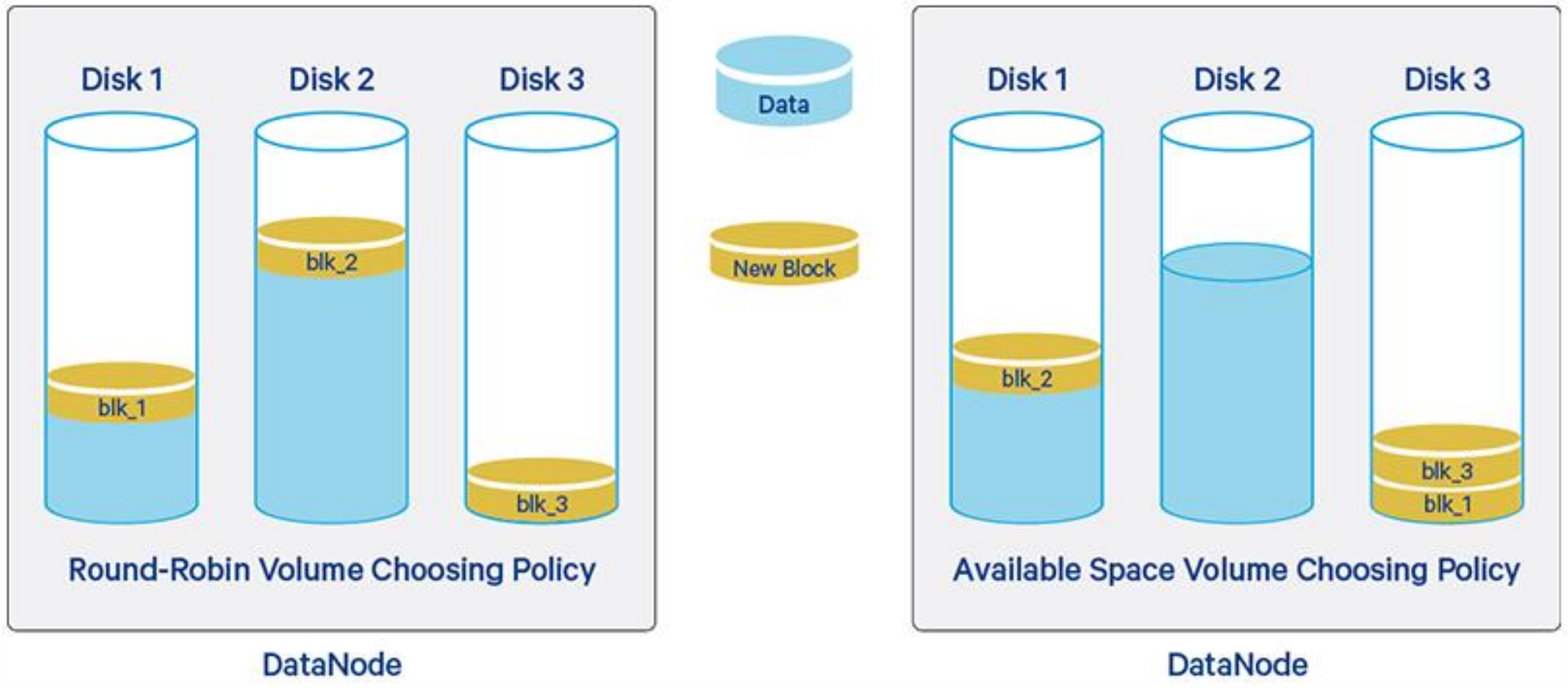
- **ResourceManager** gets resources from the cluster in the form of Container (CPU and RAM).
- **Containers** are supervised by the NodeManager and scheduled by the ResourceManager.

Hadoop 3.x against 1.x and 2.x

- **High Availability for NameNode:**
  - The loss of NameNode can crash the cluster in both Hadoop 1.x and 2.x.
  - Hadoop 2.x: the active-passive setup (one active and one stand-by NameNode) to help recover NameNode failures.
  - Hadoop 3.x: 1 active NameNode + 2 passive NameNodes + five JournalNodes to assist for **catastrophic failures**:
    - **NameNode machines**: run active and passive (standby) nodes. They should have the equivalent hardware to each other.
    - **JournalNode machines**: it is relatively lightweight and collects: NameNode and YARN ResourceManager (JobTracker in the old version).

Intra-DataNode Balancer in Hadoop 3.x

- HDFS 3.x now includes a comprehensive storage capacity-management approach for moving data across nodes.
- When writing new blocks to HDFS, DataNode uses a volume-choosing policy to choose the disk for the block. Two such policy types are currently supported: *round-robin* or *available space*.

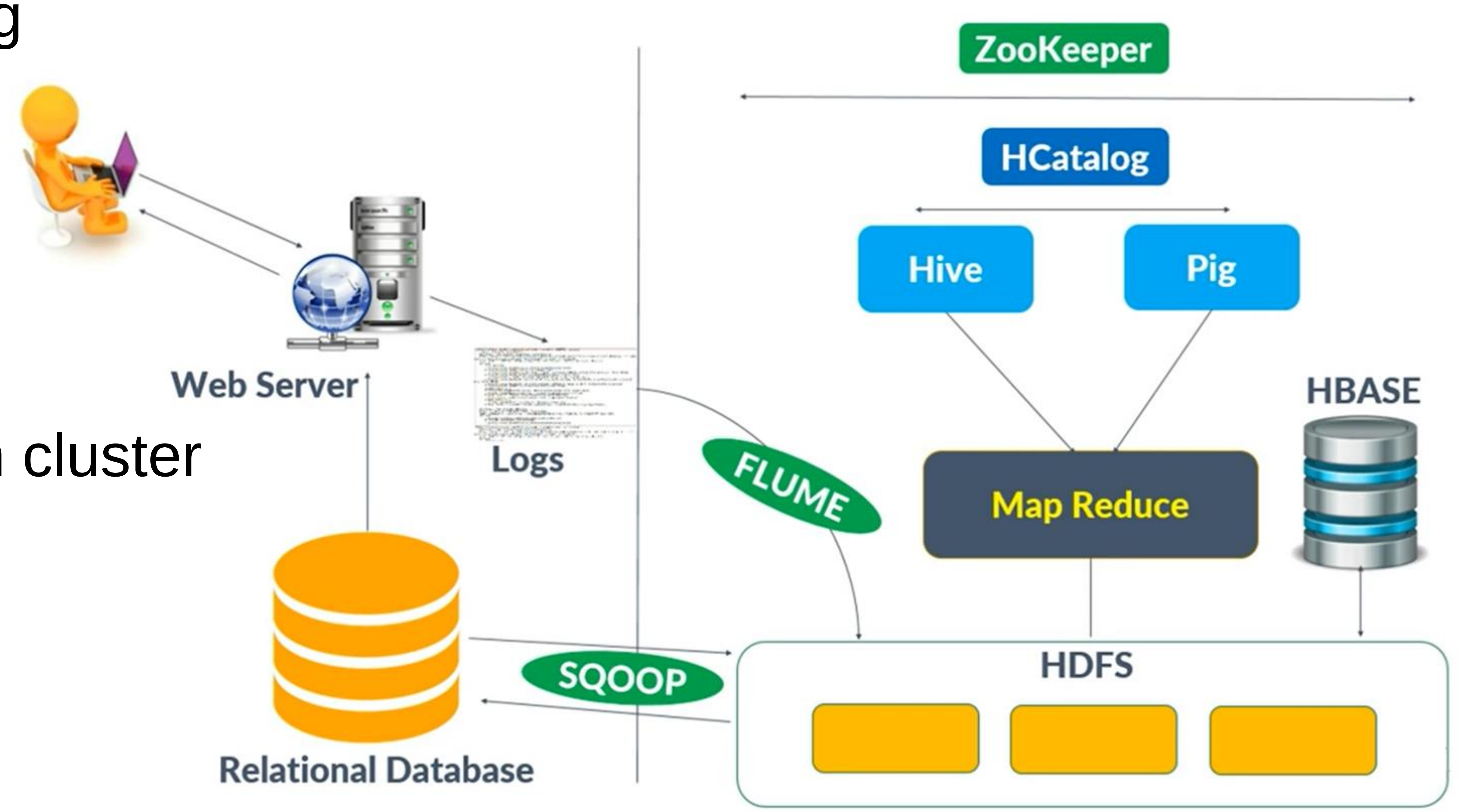


- The *round-robin* policy distributes the new blocks evenly across the available disks, while the *available-space* policy preferentially writes data to the disk that has the most free space by percentage.

University of East London

Main Hadoop Components

- **HDFS:** Hadoop Distributed File System
- **YARN:** Cluster Management Technology (Hadoop V2)
- **MapReduce:** Data processing using programming
- **Spark:** In-memory Data Processing
- **PIG, HIVE:** Data Processing Services using Query (SQL-like)
- **HBase:** NoSQL Database
- **Mahout, Spark MLlib:** Machine Learning
- **Apache Drill:** SQL on Hadoop
- **Zookeeper:** Managing Cluster
- **Oozie:** Job Scheduling
- **Flume, Sqoop:** Data Ingesting Services
- **Solr & Lucene:** Searching & Indexing
- **Ambari:** Provision, Monitor and Maintain cluster



Summary

- Overviewed the course requirements
- Discussed the reasons for needing Big Data
- Introduced Hadoop
- Discussed Hadoop Infrastructure
- Discussed HDFS and MapReduce
- Introduced Hadoop Ecosystem Components

University of East London