DLE COURSE-4

INTRODUCTION TO BIG DATA
&
HADOOP

CLASS-I

Instructor: Palash Gupta

Introduction

Hello Everyone!

This is Palash Gupta, Engineer in background and studied Computer Science & Engineering.

12+ Experience in ICT industry in different roles

Currently working as Solution Architect in leading Telecom Software Company

Conducted multiple training on Big Data & consultant to couple of Startups

Our Overall Plan & Objective

Week	Lesson	Topics	Hours	Type
1	Introduction to Big Data & Hadoop	• What is Big Data? • How Big is Big Data? • What are we trying to solve in Big Data? • Types of Data Structure • Hadoop System Principle • History of Hadoop • Comparison with RDBMS • Hadoop Eco System • Hadoop Distribution • Supported Operating System, Hardware and Resources	2 Hours	Theory
2	Understanding Hadoop HDFS & Map Reduce	• HDFS Concept • HDFS Architecture • Introduction to Map Reduce • Working Methodology of Map Reduce	2 Hours	Theory
3	Basic Hadoop Configuration, Setup, Administration and Command Reference	• Design a Hadoop Cluster • Procedure setting up a basic Hadoop Cluster • Setting Up a Hadoop Cluster • Basic Administration of Hadoop • Basic Command Reference	2 Hours	Theory & Practical
4	Understanding Spark Essential, Architecture	• Introduction to Apache Spark • Sparck Architecture • Introduction to Spark RDD, Dataset, DataFrame and DAG • Introduction to Spark Component • Understanding Spark Execution Model • Why Spark where we have Hadoop?	2 Hours	Theory
5	Spark Configuration, Administration & Setup	• Design a Spark Cluster • Procedure setting up a basic Spark Cluster • Setting Up a Spark Cluster • Basic Administration of Spark • Understanding of how Spark runs our job	2 Hours	Theory & Practical
6	Fundamental of Python and Shell Scripting	• Introduction to Python • Installing & utilizing New Python Package • Introduciton to Shell Scripting • Running a Python Script with a Sheel Script in Shedule	2 Hours	Practical
7	Programming with HDFS & Spark	• Accessing HDFS Files using Python • Loading HDFS Files in Spark • Running basic transformations and actions in Spark • Understanding of how Spark runs our job with an example	2 Hours	Theory & Practical
8	Do a Practical Experiment with a real life problem	• Understand the problem • Desing the solution • Implement the solution • Q&A	2 Hours	Practical

What we are going to Cover today?

- Introduction
- What is Big Data?
- How big is Big Data?
- What are we trying to solve?
- Types of Data Structure
- Hadoop System Principle
- History of Hadoop
- Comparison with RDBMS
- Hadoop Eco System
- Hadoop Distribution
- Supported OS and HW

What is Big Data?

A Simple Answer can be > If the data is large then this is Big Data

Still not convincing, isn't it?

Do you know? > More than **90% of the world data is generated within last few years**

Who are generating all these data?> **Human and Machine**

Why in last few years so much data & why not before? > We all know the answer right? Now-a-days many people are connected in this global village & digital world. Hence machines and humans are generating a lot of data e.g. Social media, Web content, Sensor Data, Black Box etc.

OH Still we don't get the first answer – What is Big data?

Big data means really a big data, it is a collection of large datasets that cannot be processed using traditional computing techniques & may not be feasible to store in traditional storage. Big data is not only the size of data, rather it has become a complete subject, which involves various tools, techniques and frameworks to handle large data set.

How Big is Big Data?

This is really a tough question to answer.

Why? > The answer is really relative. The size of Big Data for Facebook can be in thousand petabytes. However in any bank or telecom, it could be in hundred terabytes.

Most importantly it depends on:

A. The types of data the companies are handling
B. The latency of data processing that they can allow
C. The cost of storage that they can afford
D. The velocity of data that they are dealing with

Companies continue to generate large amounts of data, here are some recent stats:

– More than 3.7 billion people are using internet
– On Average Google is doing 40,000 searches per second
– Facebook ~ 4 petabytes data per day & 1.5 billion active users daily
– Users watch 5 billion YouTube videos per day
– 500 million tweets are sent on twitter per day

What are we trying to solve?

- The fundamental problems that triggers Big Data Evaluation are:

1. Large Data Set Storage
2. Large Data Set Processing

Initially it has been triggered for Web Data handling & Google first introduce the concept to index large web sites data and to reduce latency of google search.

Large Data Storage:

- 1 TB of external hard drive vs 1 TB of Enterprise level Disk Array – Significant Cost Difference!

- Is it feasible to store petabytes of data in large enterprise disk array?

What are we trying to solve? Contd.

- Large Data Set Processing:

Storage Capacity has grown exponentially but read speed has not kept up:

Let us see an analysis

– 1990:
• Store 1,400 MB
• Transfer speed of 4.5MB/s
• Read the entire drive in ~ 5 minutes

– 2010:
• Store 1 TB
• Transfer speed of 100MB/s
• Read the entire drive in ~ 3 hours

Searching a records over petabytes dataset in traditional technique might take more than hours to fetch.

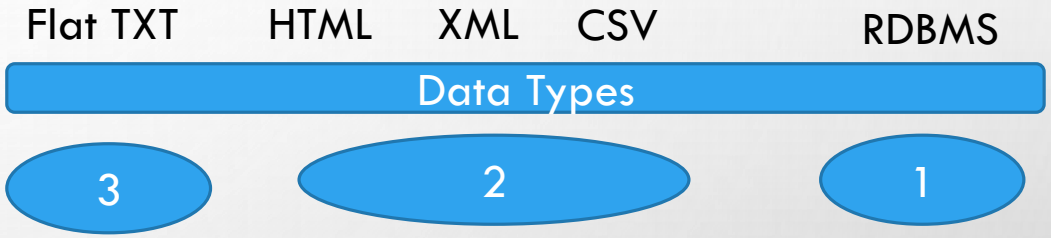
Types of Data Structure

There are three types of data set:

1. Structured
2. Semi-Structured
3. Un-Structured

Structured Data:

- Can be represented in a relational model (ROW & COLUMN FORMAT)
- Schema and Data Types are defined



Semi-Structured Data:

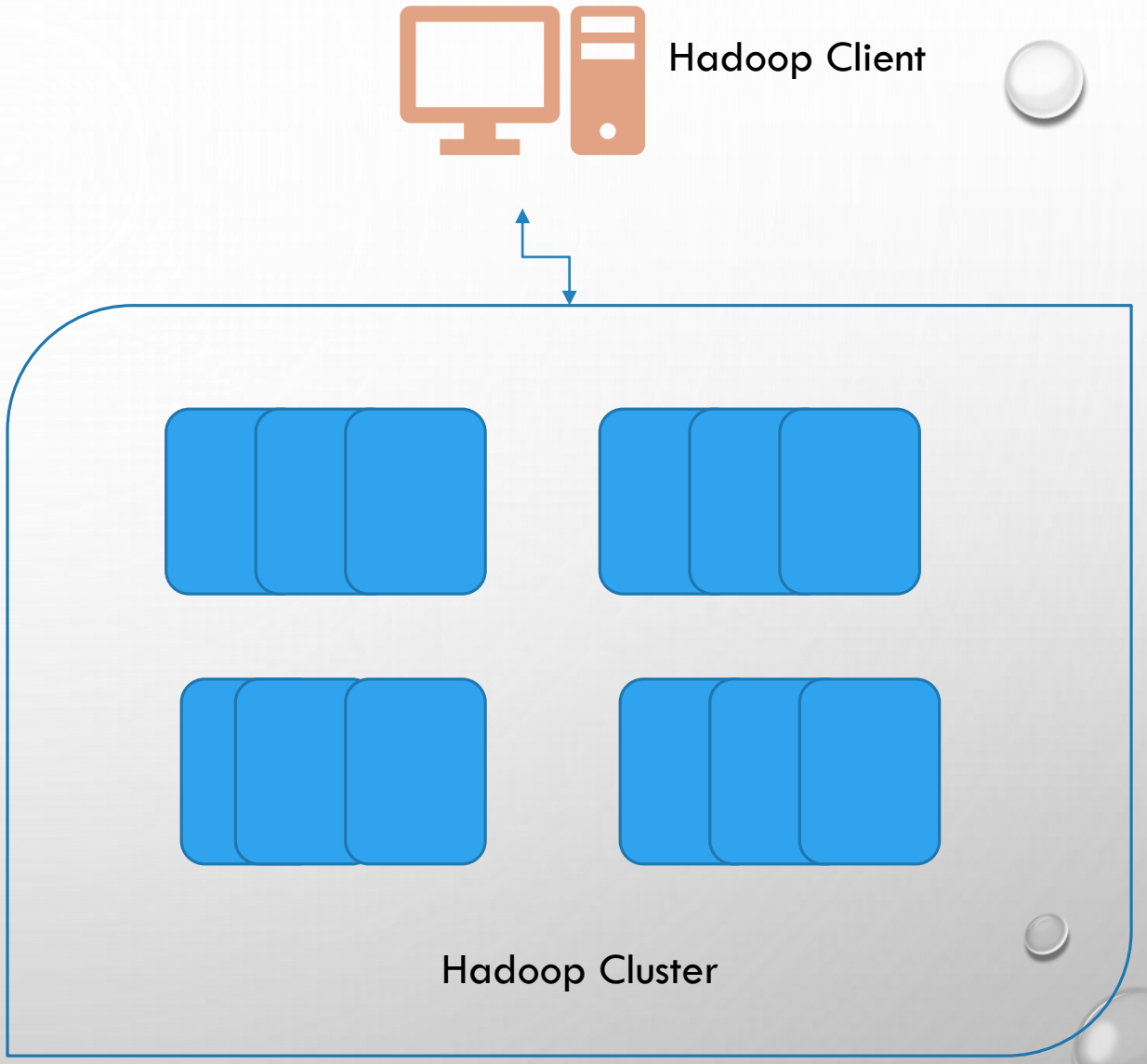
- Lack of fixed, rigid schema
- Unknown data types
- Self describing structure

Un-Structured Data:

- No schema and data types
- Programmers have to define how to deal

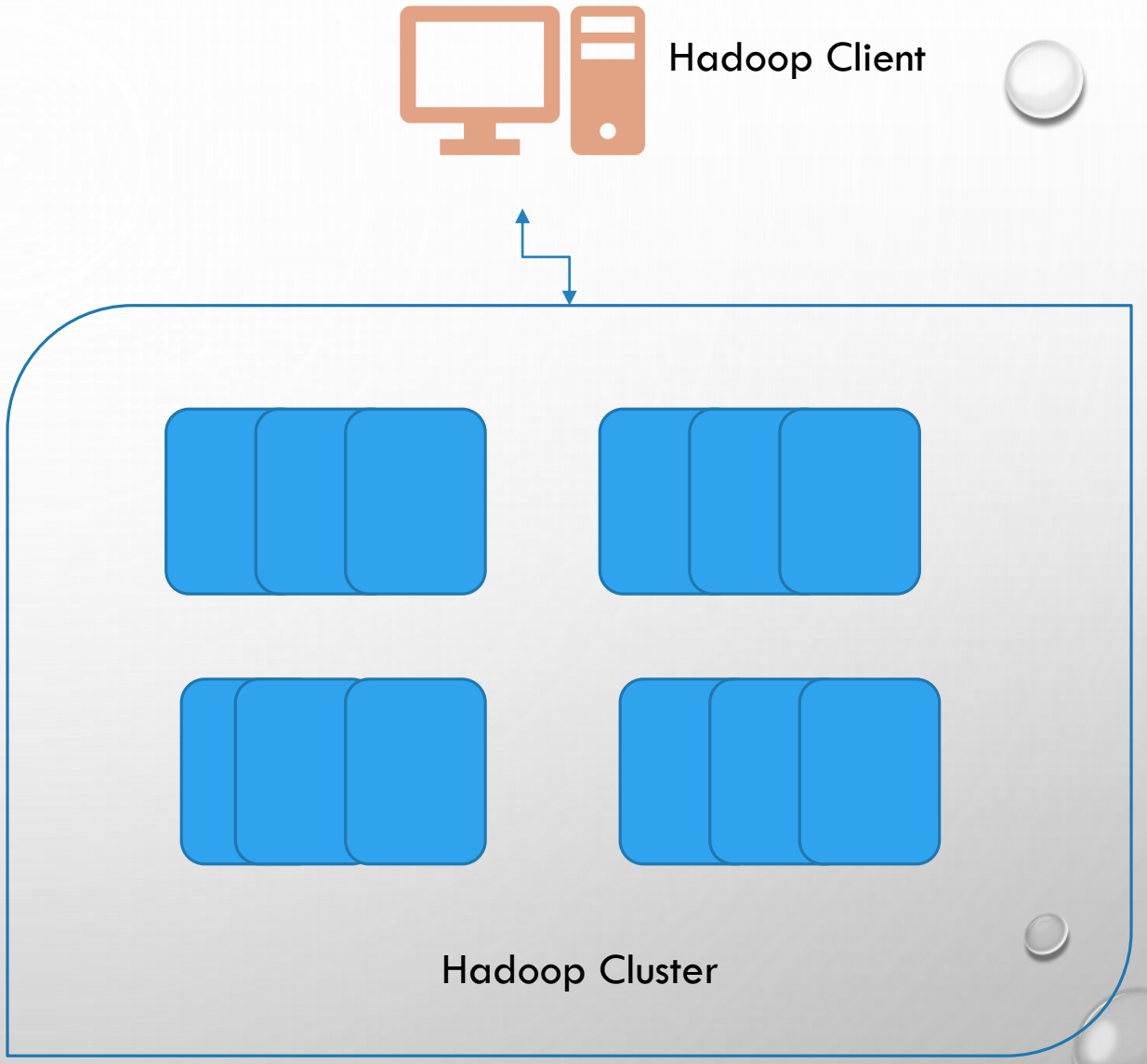
Hadoop System Principle

- A set of "cheap" commodity server hardware Networked together
- Generally Resides in the same location
- Commodity Hardware
  - They are not super computer or expensive servers
  - They are not even desktop



Hadoop System Principle Contd.

- ✓ Scale-Out rather than Scale-Up
- ✓ Keep code and data in same place
- ✓ Deal with failures – taking complexity in software level
- ✓ Abstract complexity of distributed and concurrent applications



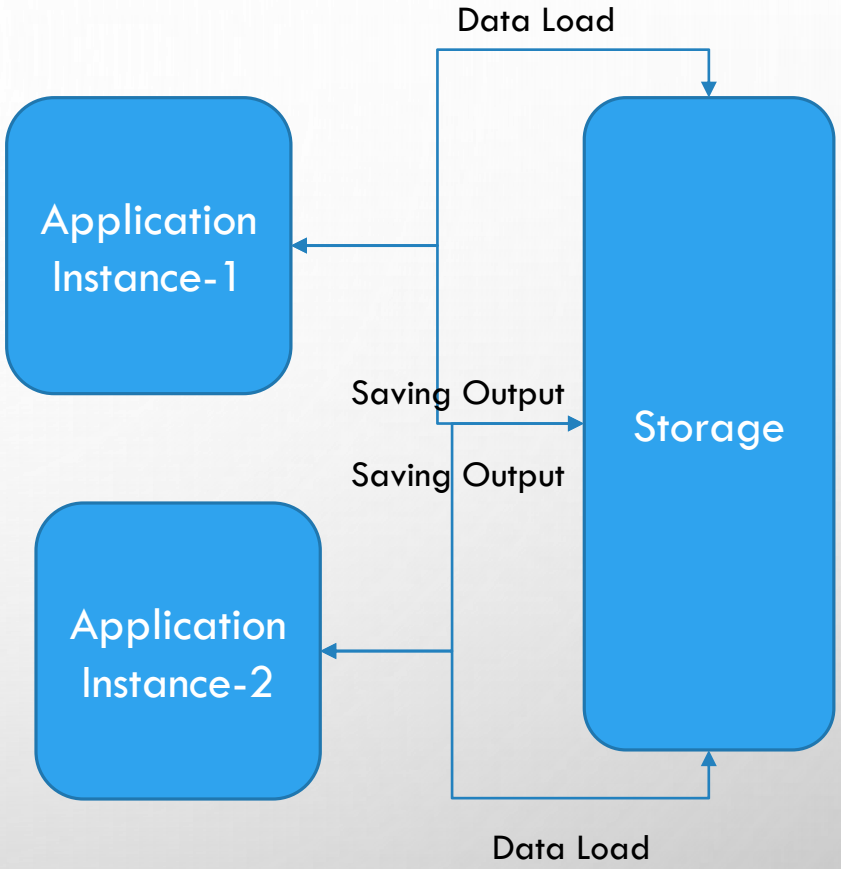
Hadoop System Principle Contd.

- Scale-Up:
  - ✓ It is harder and more costly to scale-up
  - ✓ Add additional resources to an existing node (CPU, RAM, Disk etc.)
  - ✓ New units must be purchased if required resources can not be added
  - ✓ This is also known as scale vertically

- Scale-Out
  - ✓ Add more nodes/machines to an existing distributed application
  - ✓ Software Layer is designed for node additions or removal
  - ✓ Hadoop takes this approach - A set of nodes are bonded together as a single distributed system
  - ✓ Very easy to scale down as well

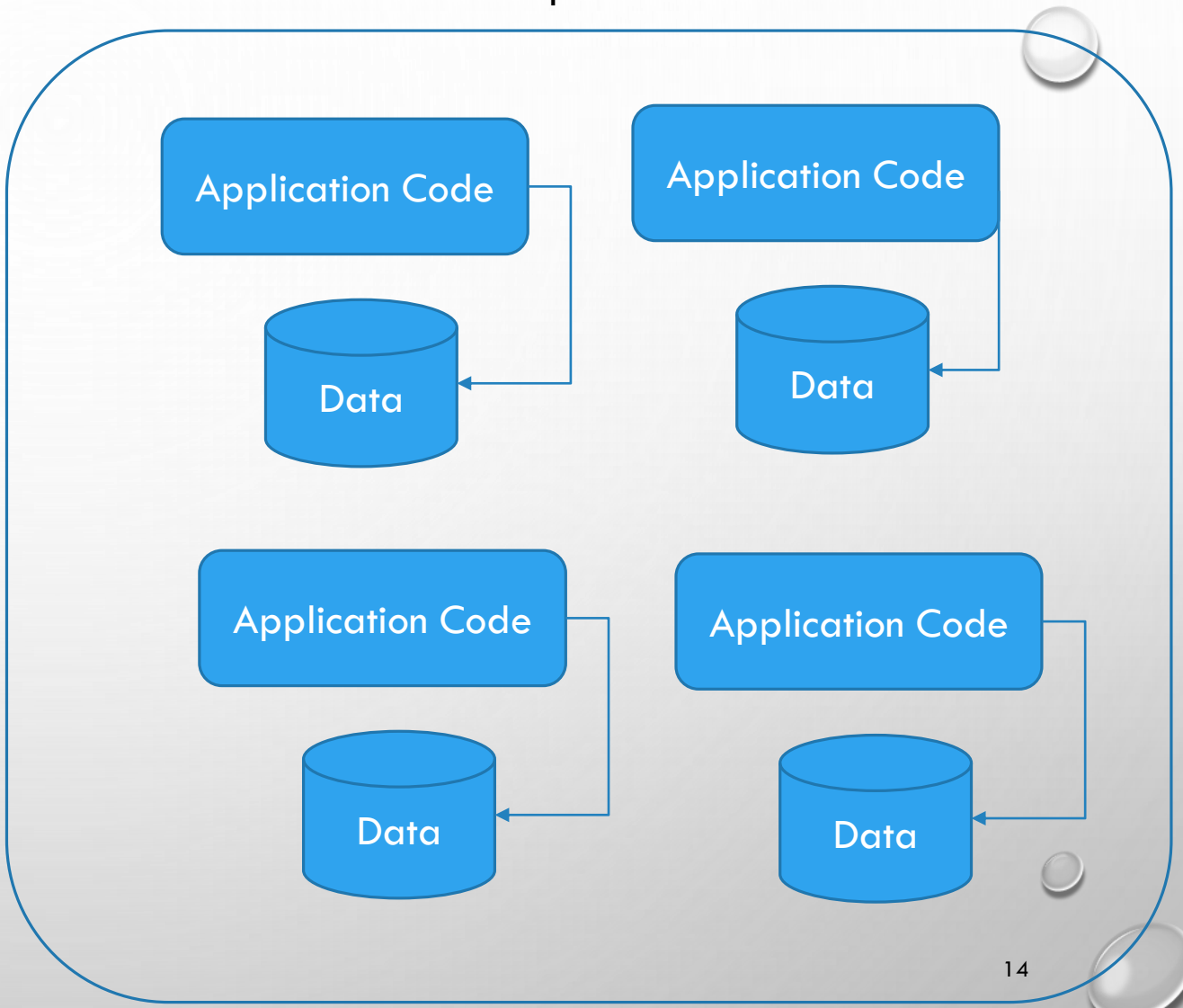
Hadoop System Principle Contd.

- Traditional Data Processing:
  - ✓ Keep program in one location
  - ✓ Maintain data in separate storage
  - ✓ Many data-intensive applications are not CPU intensive and hence causing bottleneck in network



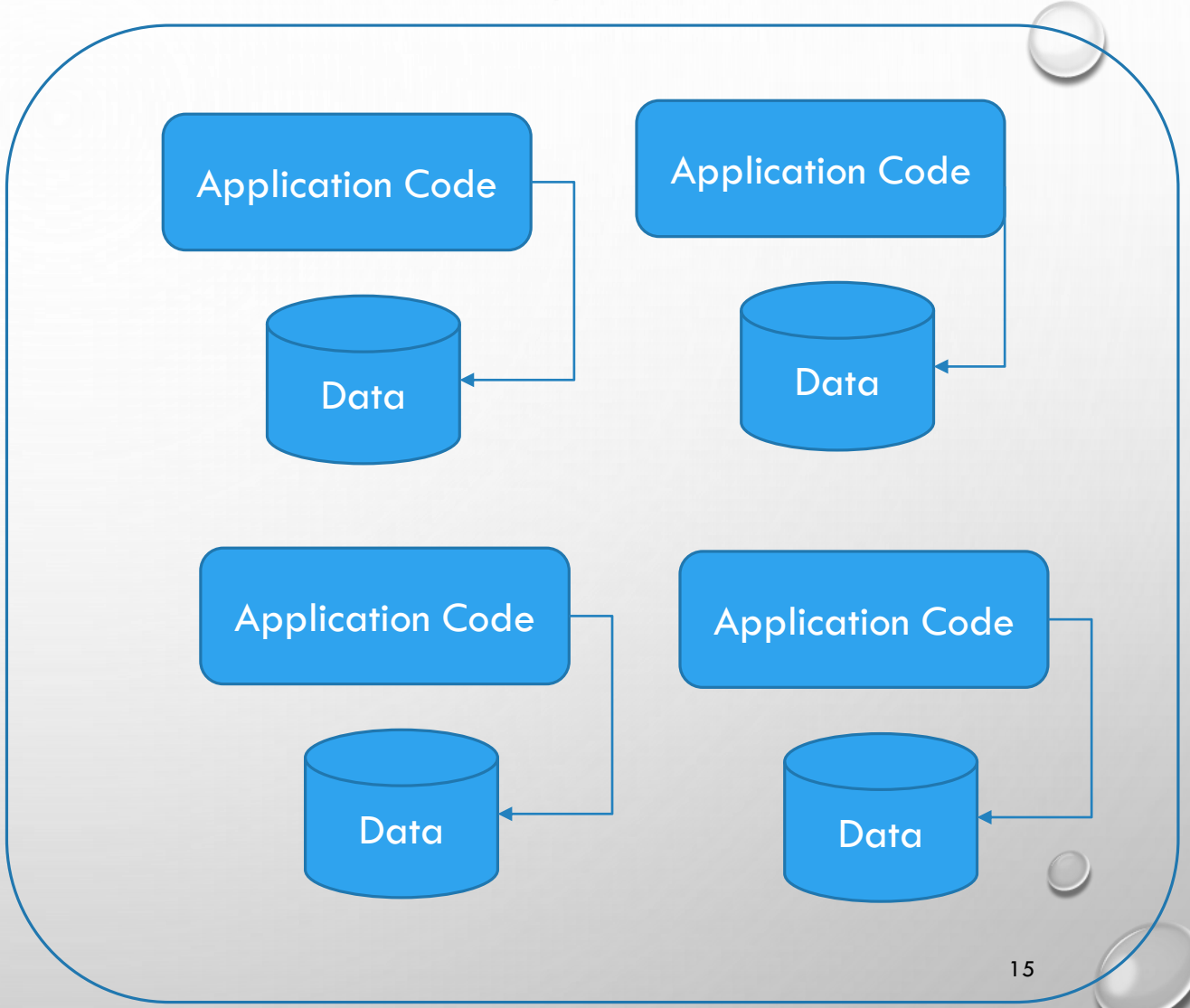
Hadoop System Principle Contd.

- Hadoop keeps both in same place:
  - ✓ Keep program & data in same place in clustered environment
  - ✓ Code can access data locally



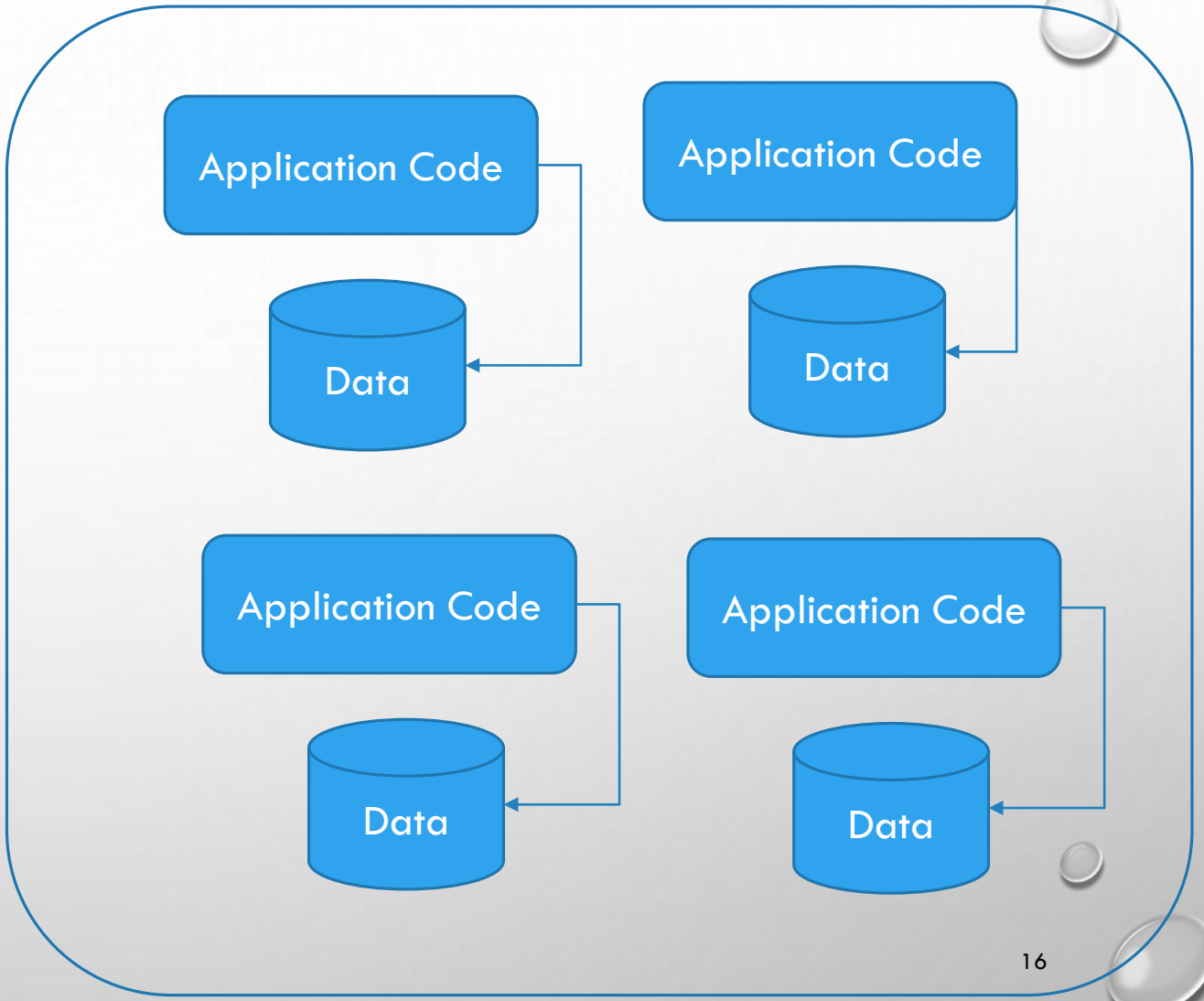
Hadoop System Principle Contd.

- Hardware/Software Failures:
  - ✓ In large data center, server or component of server failure is common.
  - ✓ Hadoop is designed to cope up with hardware and network failure by implementing:
    - Replicated Data
    - Broken down jobs/tasks are timeout controlled.



Hadoop System Principle Contd.

- Hiding Complexity of Software:
  - ✓ It abstracts complexities in distributed and concurrent applications
  - ✓ Developers don't need to think about race condition, code distribution etc.



History of Hadoop

- In 2004 Google publishes Google File System (GFS) and MapReduce framework papers
- GFS is Google File System, BigTable is Google's large structure data storage, Mapreduce is distributed computing platform.
- Doug Cutting and Nutch team implemented Google's frameworks in Nutch
- In 2006 Yahoo! hires Doug Cutting to work on Hadoop with a dedicated team
- Hadoop: Yellow stuffed elephant named by Doug Cutting
- In 2008 Hadoop became Apache Top Level Project
- – http://hadoop.apache.org

Comparison with RDBMS

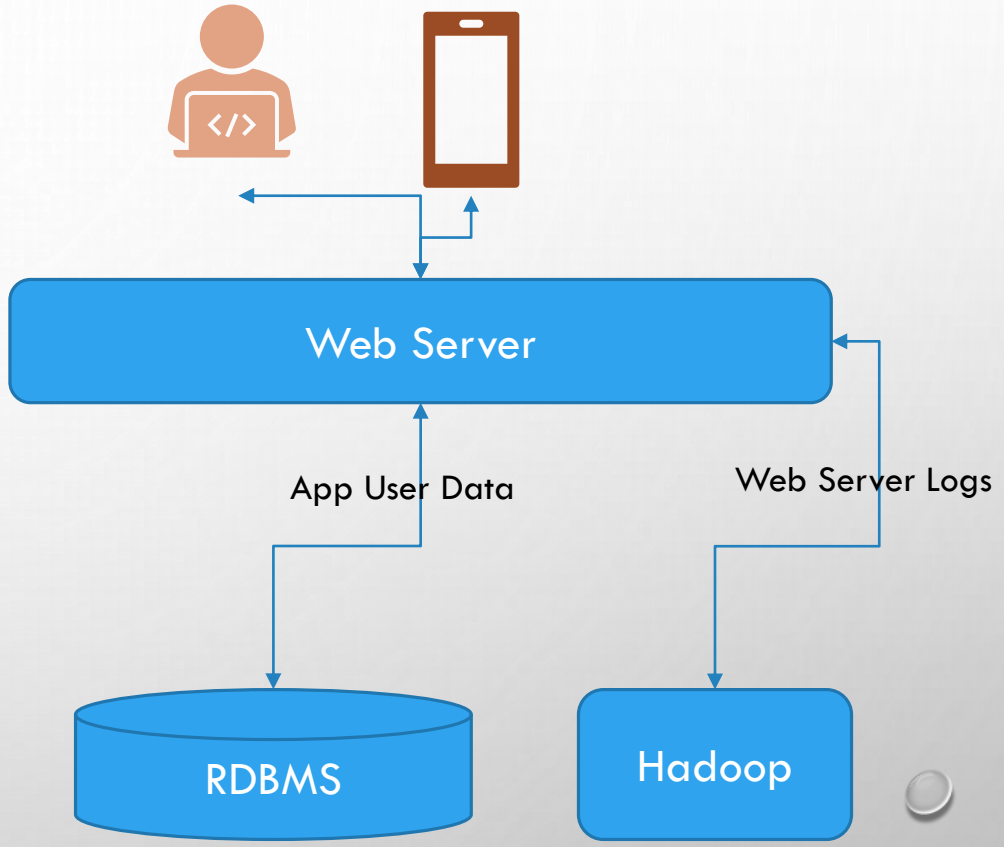
- Now-a-days RDBMS are also being utilized for batch processing e.g. Oracle, Sybase MySQL, MS SQL etc.
- Hadoop is not fully replacing RDBMS as many application yet needs both RDBMS & Hadoop to be efficient.
- Scaling up RDBMS is still a concern when the database is crossing 100 TB, however Hadoop cluster can be even more than 1000 nodes for example to handle petabytes of data
- RDBMS stored relational data meaning structured data set, however Hadoop is best to store unstructured & semi-structure data
- Hadoop performs best for offline batch processing on large amount of data, however certain NOSQL database named HBASE can give low latency query but with limited query features
- Hadoop was not designed for real time and low latency queries
- Hadoop is designed to stream large files and large amount of data
- RDBMS is best for online transaction and low latency query on small records.

Comparison with RDBMS Contd.

- An application architecture can keep both Hadoop as well as RDBMS.

Let us see an example:

- Application data are stored and processed with RDBMS
- However large set of web server logs are stored in Hadoop
- Periodically the Hadoop data set summary are Stored again back in RDBMS for user visualization.



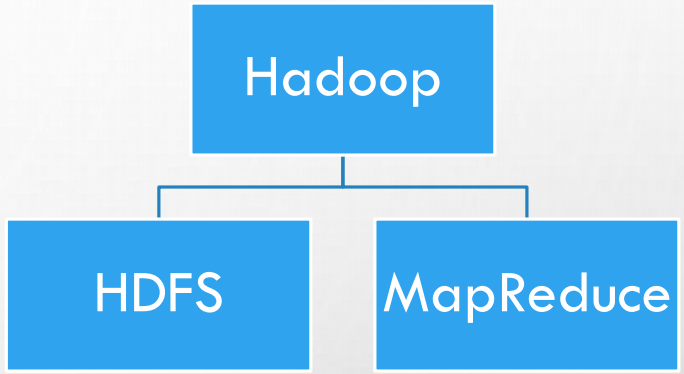
- **Sample Web Server Logs:**

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif HTTP/1.0" 200 2326

Hadoop Eco System

Hadoop includes two main functions:

- ❑ HDFS
- ❑ MapReduce



❖ HDFS:

-Hadoop distributed file system

-It is used to store large data set in a distributed environment using commodity machine

❖ MapReduce:

-This is distributing computing framework

-It is used to compute large data set using a distributed framework to reduce the latency while using cost effective method

Hadoop Eco System Contd.

Today, in addition to HDFS and MapReduce, the term also represents a multitude of products:

– **HBase:** Hadoop column database; supports batch and random reads and limited queries

– **Zookeeper:** Highly-Available Coordination Service

– **Oozie:** Hadoop MapReduce workflow scheduler and manager

– **Pig:** Data processing language and execution environment (Addition of Apache Pig, a high-level data flow scripting language, may be beneficial)

– **Hive:** Data warehouse with SQL interface (Addition of Apache Hive, a data warehouse solution that provides a SQL based interface, may bridge the gap)

- **Spark:** In-memory Distributed Computing Framework (Understanding MapReduce can be complex and hence one can use Spark)

Hadoop Distributions

- Visit and download HDFS and MapReduce from the following URL: https://hadoop.apache.org/

- Compatibility is a known issue between let say Hadoop, HBase and Pig etc.

- Distribution vendors generally combine all these in one pack by testing internal combability & bringing enterprise usable facilities.

- Distribution vendors are for example Cloudera Distribution for Hadoop (CDH), MapR Distribution, Hortonworks Data Platform etc.

- Cloudera is one of the leading distribution in Hadoop (https://www.cloudera.com/hadoop)

- Cloudera employees large percentage of core Hadoop committers

Supported HW, OS

❑ **Hardware**

Hadoop runs on commodity hardware. That doesn't mean it runs on cheapo hardware. Hadoop runs on decent server class machines.

Here are some possibilities of hardware for Hadoop nodes production cluster.

Resource Item	Medium	High End
CPU	8 physical cores	12 physical cores
Memory	16 GB	48 GB
Disk	4 TB	36 TB
Network	1 GB Ethernet	10 GB Ethernet

Yahoo Hadoop Cluster

Hadoop HW and SW

- **Software**
- **Operating System**

-Hadoop runs well on Linux. The operating systems of choice are:
-RedHat Enterprise Linux (RHEL), This is a well tested Linux distro that is geared for Enterprise. Comes with RedHat support
-CentOS Source compatible distro with RHEL. Free. Very popular for running Hadoop. Use a later version (version 6.x).
-Ubuntu The Server edition of Ubuntu is a good fit -- not the Desktop edition. Long Term Support (LTS) releases are recommended, because they continue to be updated for at least 2 years.
- Windows and MAC OS X can be used for development platform only *

- **Java**

Hadoop is written in Java. The recommended Java version is Oracle JDK 1.6 release and the recommended minimum revision is 31 (v 1.6.31).
So what about OpenJDK? At this point the Sun JDK is the 'official' supported JDK. You can still run Hadoop on OpenJDK (it runs reasonably well) but you are on your own for support :-)

- **Hadoop Binaries**

- https://hadoop.apache.org/releases.html

QUESTION & ANSWER

THANKS FOR ATTENDING THE CLASS & YOUR CO-OPERATION

References

- https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/
- http://elephantscale.com/
- https://www.hadoop.apache.org