Lecture 1 – Introduction

BDAT 1002

Today's Lecture

- Introduction to Big Data
- Understanding Big Data Problem
- Hadoop as a solution
- History of Hadoop

Evaluation

- Evaluation comprised of
  - In-class labs 20%
  - Assignments: 40%
  - Midterm Exam: 40%
- Program and courses should complement your skills/background

Course Topics

- Can give you a list of topics
  - A bit useless right now
- Better to look at an introduction and arrive at the course topic

Ideally it would be nice if you know

- Relational Databases
  - But we will review this later in the course
- Programming language
  - Java, Python
  - Any kind of "procedural" programming language
- Basic Linux commands
  - Homework for next week.

What is Big Data?

- Extremely large volumes of data
  - But what is considers "large"?
  - 10Gb, 100G?
- There is no straight forward answer for two reasons

What is Big Data?

- What is considered "Big" today, is not considered Big in a year
  - It's a moving target
- It's relative
  - What we consider big may not be big to someone else
  - Example Google, Facebook

Factors for Big Data

- Factors to consider to designate something as big Data
- **Volume** → **>** 100TB (at this time)
- But volume is only part of the equation
- Rate of change of data growth or **velocity** is also important
  - example: email server

Factors for Big Data

- Most of the time, volume and velocity are all the we need to determine if we have a "Big" data problem
- Next factor is **variety**
  - Another dimension
  - Traditional databases are very structured with rows and tables
  - Different formats of data can make your "traditional" databases not a good choice
  - If you have pictures, text, comments, "likes"

Factors for Big Data

- So if you want to know if you have big data, take **the three V's** into consideration
- Companies often bring in Big data consultants and hope a "Big" Data solution will help them out
- Most of the time, volume and velocity tests are not met
  - Volume 100's GB
  - Velocity low
  - Really need to optimize what you have

Is there a Big Data Problem?

- You might look around and not really see what the big deal is
- But there are very real cases

Is there an actual use case?

- Science
  - NASA – 2 GB every hour!

- Government
  - NSA
  - One Yottabyte Capacity

- Social Media
  - Facebook → 70 billion photos, 150TB of logs every year
  - Facebook data collection

So what?

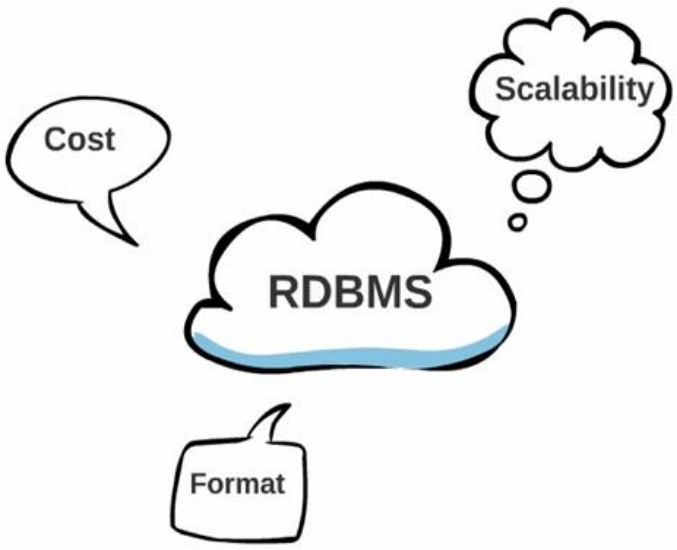
- So hopefully I have convinced you that there is such a thing as "Big" data
- So we have Big Data, but so what?
- Big Data comes with big problems!
- We are going to look at some problems
  - Storage
  - Computational Efficiency
  - Data Loss
  - Cost

Traditional Solutions - RDBMS

- One solution is Relational Database Management Systems (RDBMS)
  - MySQL, PL/SQL, etc
- Scalability issues
  - As the data gets bigger, computational time goes up
- **RDBMS are not horizontally scalable**
  - You can't improve performance by adding more computing power

Traditional Solutions - RDBMS

- RDBMS are designed to hand structured data
  - If the data is unstructured, it is hard for an RDBM to handle



Traditional Solutions – Grid Computing

- Put many computers in parallel
- Good for low volume, intensive computational tasks
  - ex image rendering
  - Not good for large volume
- Also requires good experience with low level programming knowledge
  - Not suitable for mainstream

What we need?

- Support large volume of data
- Storage efficiency
- Good Data Recovery
- Horizontally scalable
- Cost effective
- Easy for programmers and non-programmers

Understanding Big Data Problems

Introduction

- We like to intuitively arrive at the solution for Big Data
- Analyze a Big Data Problem
- See if we can come up with a solution

"Big" Data Problem

- Scenario: You are given the day to day stock price information for several years
- File size: 1TB

```
ABCSE,KJT,2010-02-08,10.95,11.06,10.70,10.76,115900,10.76
ABCSE,KJT,2010-02-05,10.87,11.00,10.68,10.98,103600,10.98
ABCSE,KJT,2010-02-04,11.07,11.07,10.73,10.82,171000,10.82
ABCSE,KJT,2010-02-03,11.00,11.16,10.61,11.13,164500,11.13
ABCSE,KJT,2010-02-02,10.83,11.15,10.74,10.98,161900,10.98
ABCSE,KJT,2010-02-01,10.52,10.89,10.52,10.77,245800,10.77
ABCSE,KJT,2010-01-29,10.43,10.51,10.26,10.26,373700,10.26
ABCSE,KJT,2010-01-28,10.93,10.96,10.29,10.42,158000,10.42
ABCSE,KJT,2010-01-27,10.67,10.93,10.41,10.91,100300,10.91
ABCSE,KJT,2010-01-26,10.55,11.05,10.46,10.76,229500,10.76
ABCSE,KJT,2010-01-25,10.67,10.86,10.50,10.56,102300,10.56
ABCSE,KJT,2010-01-22,10.92,11.02,10.54,10.59,206100,10.59
ABCSE,KJT,2010-01-21,11.14,11.40,10.90,10.97,165300,10.97
ABCSE,KJT,2010-01-20,11.24,11.35,11.03,11.16,66800,11.16
ABCSE,KJT,2010-01-19,10.79,11.38,10.72,11.35,133300,11.35
ABCSE,KJT,2010-01-15,11.01,11.07,10.75,10.80,115000,10.80
ABCSE,KJT,2010-01-14,10.80,11.14,10.76,11.06,87500,11.06
```

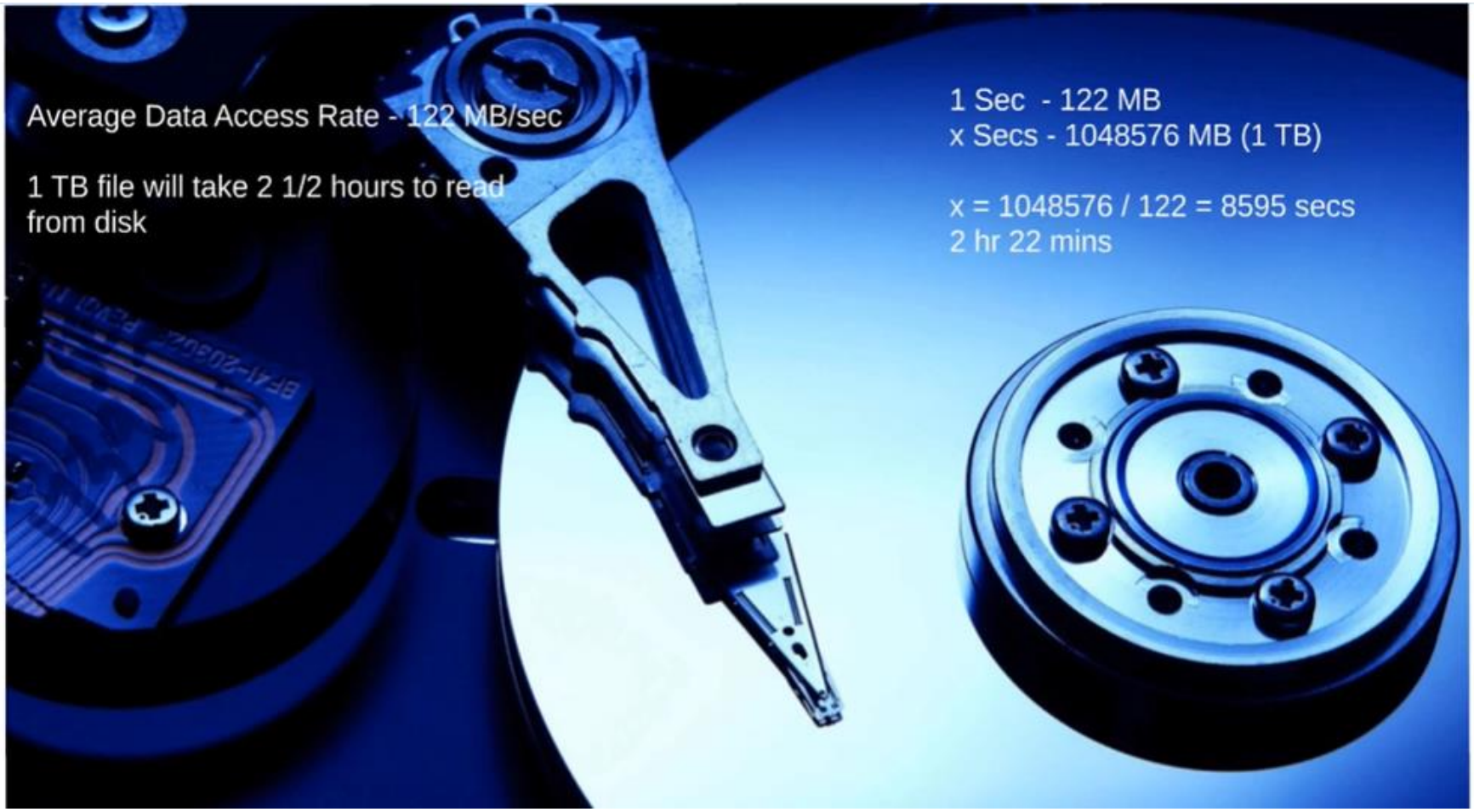
Problem

- Question: Find the maximum closing price for each stock symbol
- Two problems:
- Storage
  - your desktop has only 200 GB
  - Ask your network administrator to put the data on NAS – Network attached storage
  - Anyone with access to the network can get the data
- So storage problem is solved

Solution

- Solving the problem:
  - Write a Java program to parse the dataset
  - Do the computation
- What is the ETA for this problem?
  - We need to access the data
  - We need to do some computation
  - Also have some network latency

Traditional Hard Disk Drive (HDD)

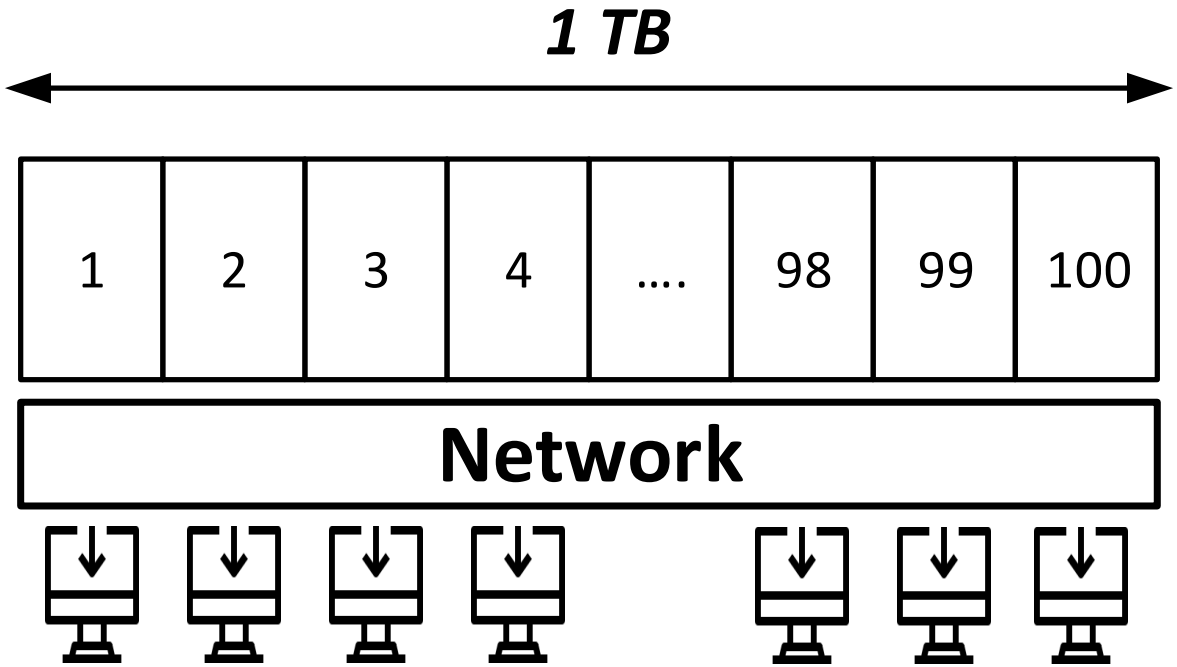


Reducing ETA

- So your ETA is > 3 hours with computation time added
- Many businesses cannot wait 3 hours, especially finance
- So how can we calculate the result in less than 3 hours?
- Replace HDD with SSD
  - No magnetic disk or head
  - Based on flash memory, very fast
- **Problem?**

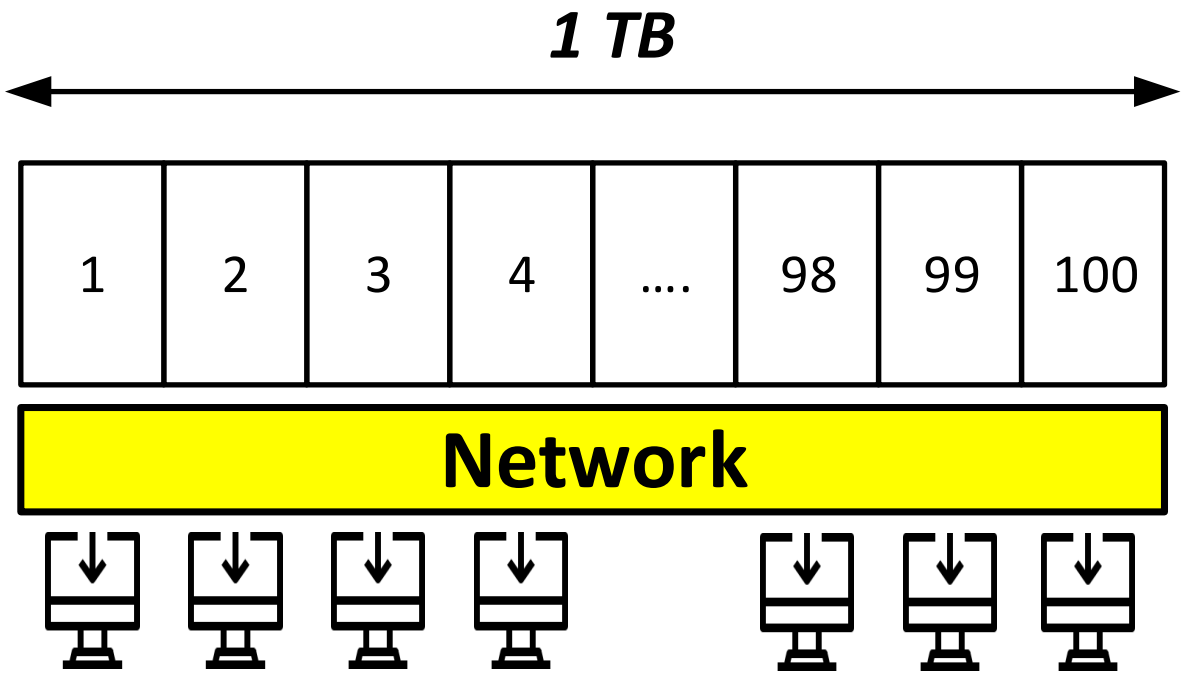
Reducing ETA

- Any other solutions to speed up time?
- Chop up the 1TB into 100 pieces
  - Have 100 parallel computers
  - Read simultaneously
  - Compute simultaneously



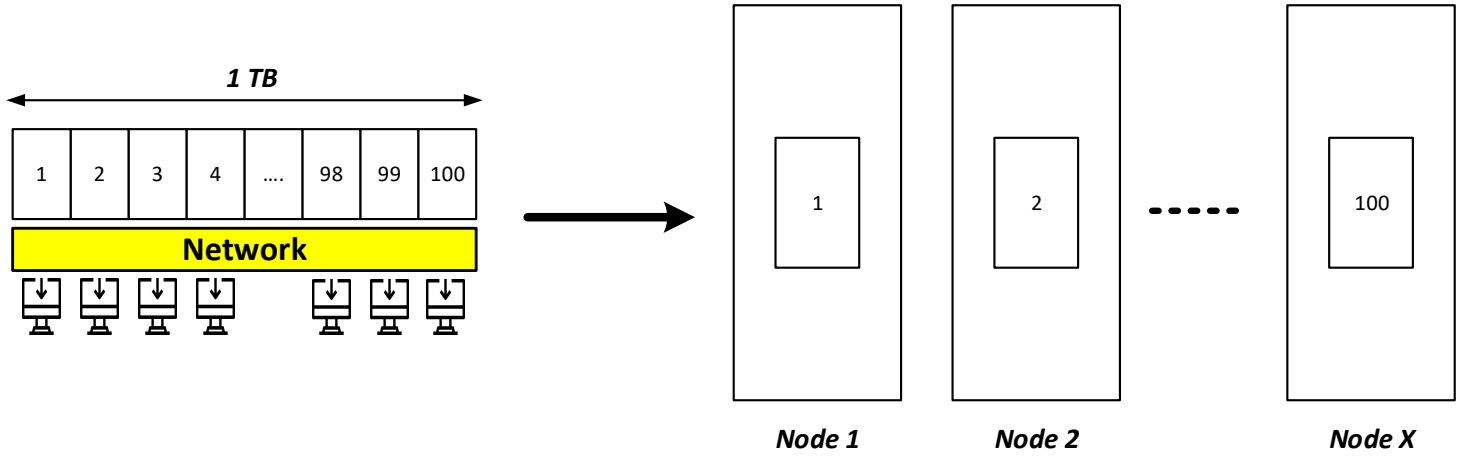
Reducing ETA

- One problem is that the network now can become the bottleneck
- Say you have 10 people in a household all trying to watch Netflix, what happens?



Reducing ETA

- Bring the data closer to the computation
  - Store the data in several different nodes
  - Not restricted with bandwidth
- Seems great …
  - System used to make billions
  - You get a bonus, glory etc
  - Any problems?

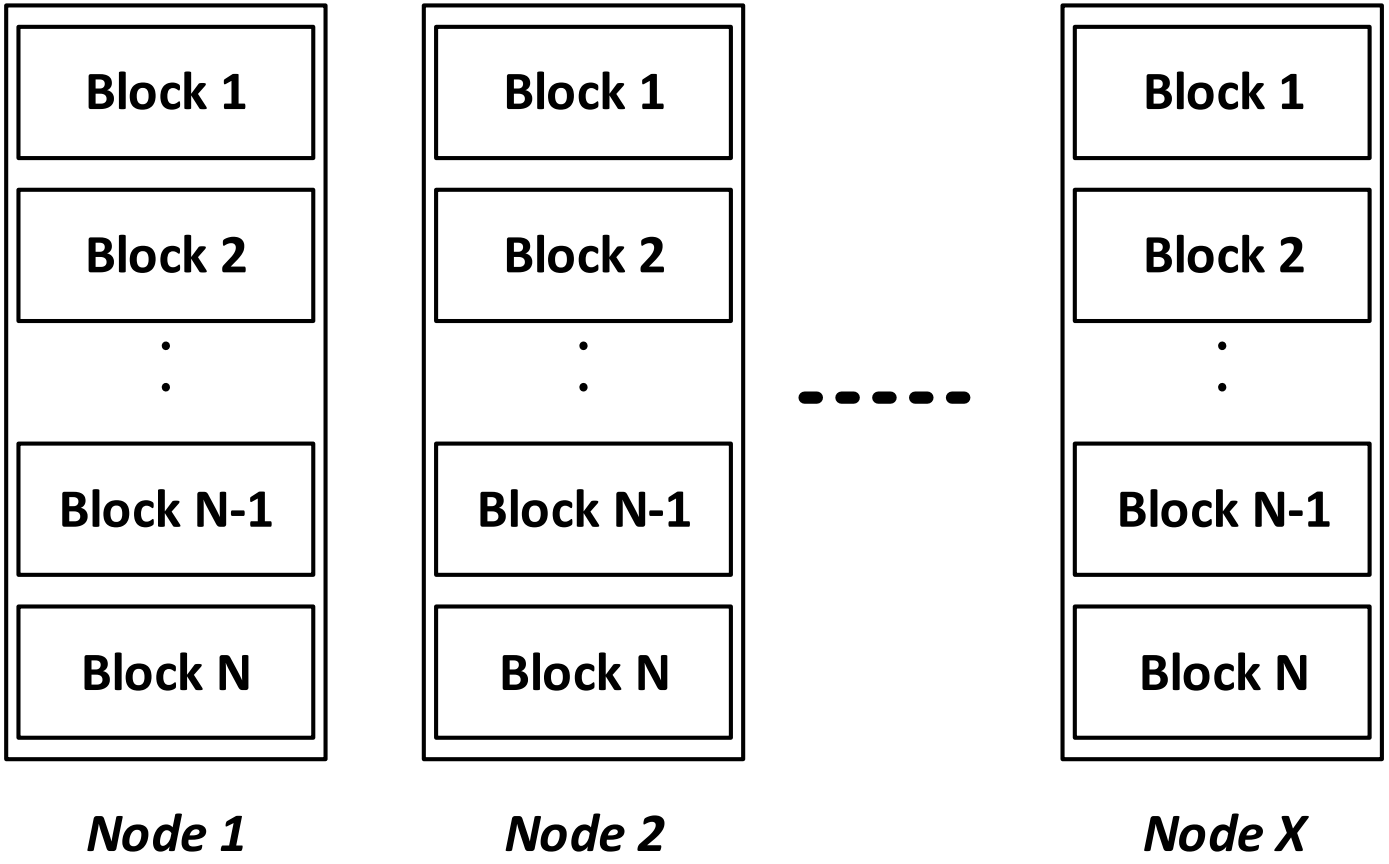


Hard Disk Failure

- How do you protect yourself from Hard Disk failure?
  - Backup!
- We want something similar for our nodes
- How would you do this?

Hard Disk Failure

- Want to the same with our cluster
- Copy each block to different nodes



Challenges - Storage

- Who breaks the 1TB into blocks?
- How does node 1 know that node 3 has block 1?
- Who decides that block 7 should be in nodes 1, 2 and 3.

Challenges - Computation

- There are also computational challenges
- Data for a stock can be in many different blocks
  - Somehow you have to consolidate the results to get a final result

Challenges

- What we have described in the previous slides is really called **distributed computing**
- And it is not easy
- Lots of work must go into it
- And all this came from trying to solve a simple problem
  - But the data was very large

Answer to all these questions and challenges…

Hadoop

- A framework for distributed computing
- Two main components
  - HDFS
  - MapReduce

HDFS

- **H**adoop **D**istributed **F**ile **S**ystem
- Takes care of all your distributed storage complexities
  - Splitting your data into blocks
  - Replicating each block to more than one node
  - Keep track of which block is stored in which node

MapReduce

- A programming model
- Implemented by Hadoop
- Takes care of all the computational complexities
  - Bring all the intermediate results to get a consolidated output

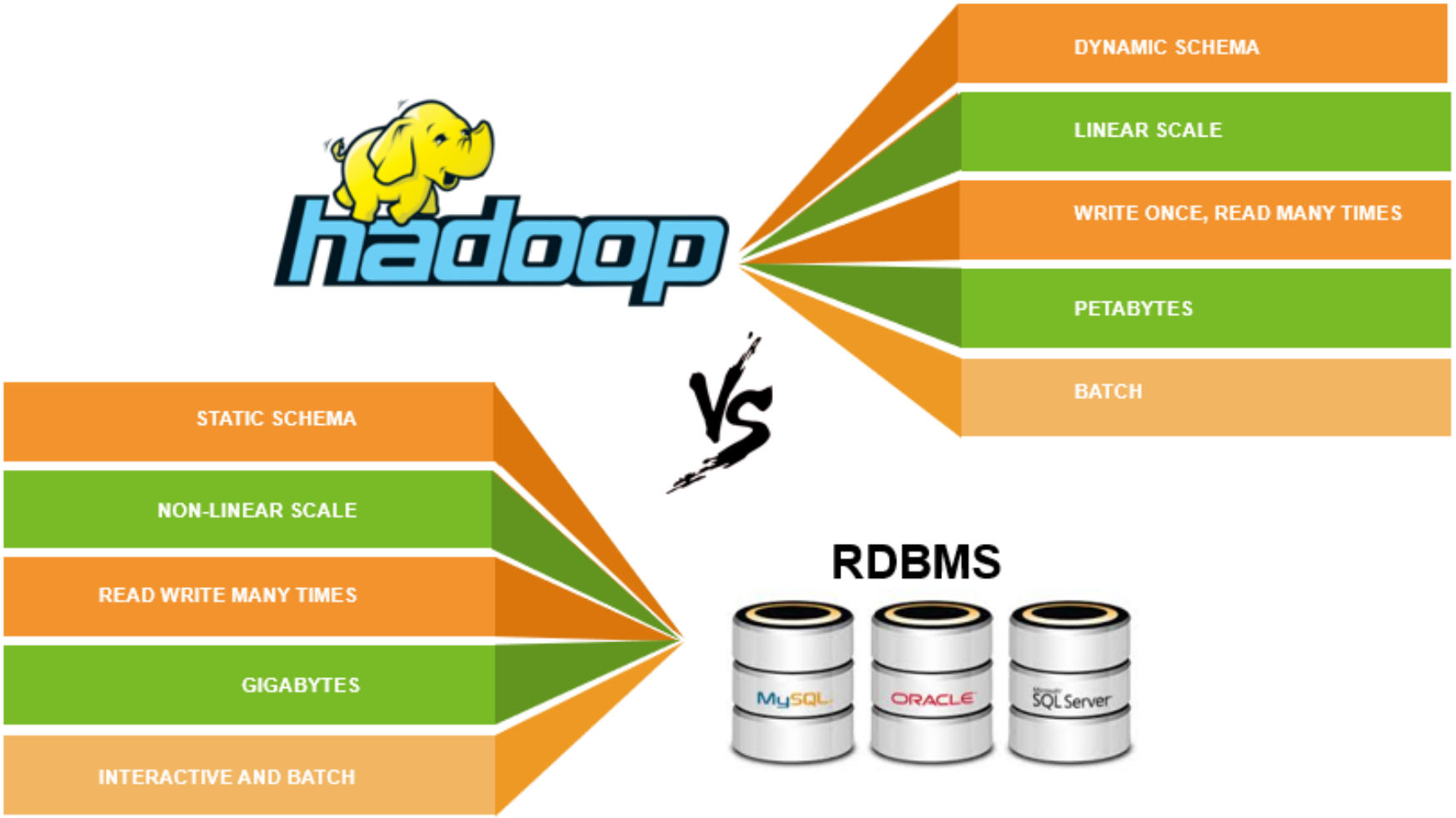
One last thing

- Hadoop was built to work on **commodity** hardware
- We need a machine that has processor, hard disk and RAM
- But it's not cheap hardware!
  - Still need certain amount of memory, CPU power
  - We'll look at specifications later

Is Hadoop a replacement for Databases?

- No
- There are things Hadoop is good at, and there are things databases are good at

Is Hadoop a replacement for Databases?



But…

- Gaps between Hadoop and RDBMS are closing in…
- There is a third set of tools called NoSQL databases
  - HBase, Cassandra
- Sit between Hadoop and RDBMS

History of Hadoop

Doug Cutting

Nutch

2002

GFS

2003

NDFS

MapReduce

2004

MapReduce
NDFS
Nutch

2005

YAHOO!

2006

Google

World Record
Terasort 209 sec.

hadoop

2008

Terasort 68 sec