Maharishi International University 1971-1995

MAHARISHI UNIVERSITY OF MANAGEMENT

Engaging the Managing Intelligence of Nature

CS522: Big Data

Premchand Nair Ph.D. (Math), Ph.D. (CS)

2020

Maharishi's Eleventh Year of Global Ram Raj

Lesson 1

Introduction to Big Data:

Transcendental consciousness is the simplest form of awareness

Wholeness of the Lesson

The Hadoop and related technology provide shared access to large banks of unstructured data. There are more unstructured data compared to structured data. *The Unified Field is the ultimate unstructured data and it provides access to all knowledge in the simplest state of awareness.*

Measuring data

Bit	1 bit	1/8
Nibble	4 bits	1/2 (rare)
Byte	8 bits	1
Kilobyte	1,024 bytes	1,024
Megabyte	1,024 kilobytes	1,048,576
Gigabyte	1,024 megabytes	1,073,741,824
Terabyte	1,024 gigabytes	1,099,511,627,776
Petabyte	1,024 terrabytes	1,125,899,906,842,624
Exabyte	1,024 petabytes	1,152,921,504,606,846,976
Zettabyte	1,024 exabytes	1,180,591,620,717,411,303,424
Yottabyte	1,024 zettabytes	1,208,925,819,614,629,174,706,176

Byte	A single letter, like "A."
Kilobyte	A 14-line e-mail. A pretty lengthy paragraph of text.
Megabyte	A good sized novel. Shelley's "Frankenstein" is only about four-fifths of a megabyte.
Gigabyte	About 300 MP3s. About 40 minutes of video at DVD quality (this varies, depending on maker). A CD holds about three-fourths of a gigabyte.
Terabyte	About thirty and a half weeks worth of high-quality audio. Statistically, the average person has spoken about this much by age 25.
Petabyte	The amount of data available on the web in the year 2000 is thought to occupy 8 petabytes.
Exabyte	In a world with a population of 3 billion, all information generated annually in any form would occupy a single exabyte. Supposedly, everything ever said by everyone who is or has lived on the planet Earth would take up 5 exabytes.
Zettabyte	Three hundred trillion MP3s; Two hundred billion DVDs. If every person living in the year 2000 had had a 180 gigabyte hard drive filled completely with data, all the data on all those drives would occupy 1 zettabyte.

Growth rate of data

IDC estimates

- 4.4 zettabytes in 2013 (zettabytes = 10^{21})
- 44 zettabytes in 2020

10 fold in 7 years!

Sources of Big Data

- Facebook: 7 petabytes per month
- NYSE: 4 to 5 terabytes per day
- Ancestry.com stores around 10 petabytes
- The Internet Archive stores around 18.5 petabytes of data
- Hadron Collider 30 petabytes per year

PB datasets are rapidly becoming the norm, and the trends are clear: our ability to store and process data is fast overwhelming

Fundamental Data characteristic

- Unstructured Data (vs. Structured Data)
- Volume (Huge amount of data)
- Data is in digital format

- Challenge is to make sense out of it. That is termed as Big Data Analytics

Fundamental Data characteristic

- Volume
- Velocity
- Variety
- Veracity

Also known as 4 V's.

Consider

Challenges and Values

- **Volume-based value:** The more comprehensive your integrated view of the customer and the more historical data you have on them, the more insight you can extract from it. In turn, you are making better decisions when it comes to acquiring, retaining, growing and managing those customer relationships.
- **Velocity-based value:** The more rapidly you can process information into your data and analytics platform, the more flexibility you get to find answers to your questions via queries, reports, dashboards, etc. A rapid data ingestion and rapid analysis capability provides you with the timely and correct decision achieve your customer relationship management objectives.
- **Variety-based value:** The more varied customer data you have – from the Customer relationship management (CRM) system, social media, call-center logs, etc. – the more multifaceted view you develop about your customers, thus enabling you to develop customer journey maps and personalization to engage more with customers.
- **Veracity-based value:** Amassing a lot of data does not mean the data becomes clean and accurate. Data on customers must remain consolidated, cleansed, consistent, and current to make the right decisions.

Google's Idea 1. Scale out, not up

Use a large number of low-cost, low-end servers (i.e., the scaling out approach) is preferred over a small number of high-cost, high-end servers (i.e., the scaling up approach).

Consequence: Failures are unavoidable

Problem 1

Assume that a 10000 server cluster is built from reliable machines with a mean-time between failures (MTBF) of 1000 days

(a) What is the failure rate?

(b) If MTBF of a machine is 10000 days, what is the failure rate?

Consequence: Failures are common

Let us suppose that a cluster is built from reliable machines with a mean-time between failures (MTBF) of1000 days (about three years). Even with these reliable servers, a 10,000-server cluster would still experience roughly 10 failures a day.

For the sake of argument, let us suppose that a MTBF of 10,000 days (about thirty years) were achievable at realistic costs (which is unlikely). Even then, a 10,000-server cluster would still experience one failure daily.

Google's Idea 2. Move processing to the data

Assume an architecture where processors and storage (disk) are co-located.

In such a setup, we can take advantage of data locality by running code on the processor directly attached to the block of data we need.

The distributed file system (DFS) is responsible for managing the data.

Limitation: Process data sequentially and avoid random access

Data-intensive processing by definition means that the relevant datasets are too large to fit in memory and must be held on disk.

Seek times for random disk access are fundamentally limited by the mechanical nature of the devices: read heads can only move so fast and platters can only spin so rapidly. As a result, it is desirable to avoid random data access, and instead organize computations so that data is processed sequentially. What are your thoughts in light of advances in storage technology?

Desirable qualities

Seamless scalability

All algorithms must work correctly irrespective of number of nodes in the cluster

Information Hiding

Framework must hide all details that are not necessary from a developers point of view.

DFS

Problem 2

Read 1 TB data

a)1 machine having 4 I/O channels (or 4 hard drives) such that each can read 100 MB/sec.

a)10 machine having each having 4 I/O channels (or 4 hard drives) such that each can read 100 MB/sec.

Why DFS

Read 1 TB data

- 1 machine
  - 4 I/O channels (4 hard drives)
  - Each 100 MB/sec
  - 1000000 MB/400 MB = 41.666 minutes
- 10 machine
  - 4 I/O channels
  - Each 100 MB/sec
  - 1000000 MB/4000 MB = 4.1666 minutes
- I/O speed is the challenge; not the storage capacity. So we need to distribute data among many nodes (machines)

What is DFS

Machines are physically located at different places.

Logically, there is only one file system.

So we can read data in parallel into multiple machines.

What is Hadoop 1.x

Definition

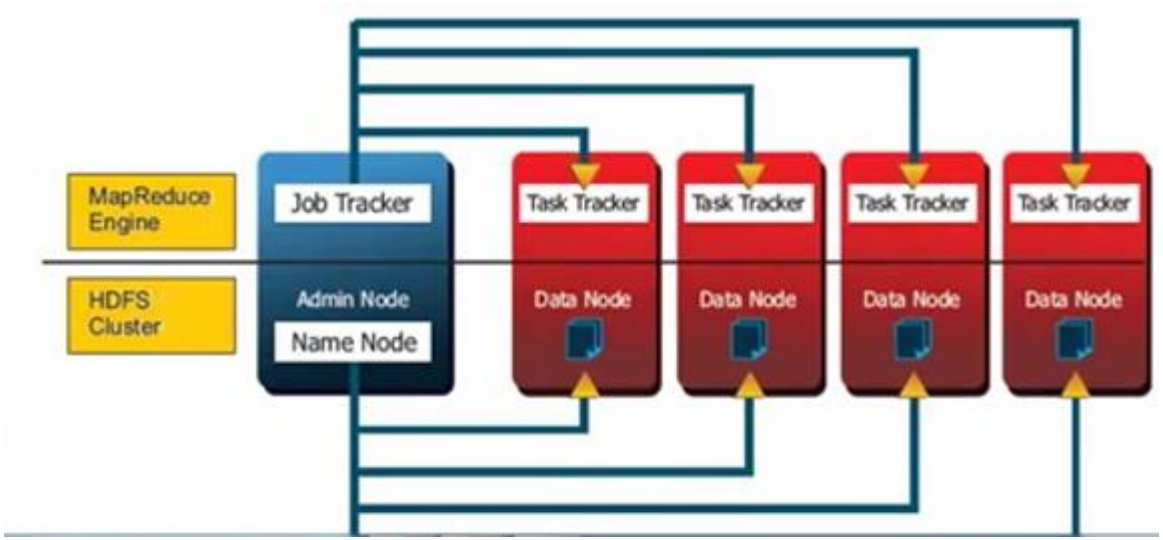
Hadoop is a framework that allows for distributed processing of large data sets across clusters of commodity computers using a simple computing model (called MapReduce to retrieve and analyze data).

Google, Yahoo, IBM, LinkedIn, Facebook, eBay, Amazon, …

Hadoop 1.x

HDFS (Hadoop Distributed File System) for storage

MapReduce for processing



Main Point 1

Hadoop is a framework that allows distributed processing of large data sets across clusters of commodity computers using a simple computing model (called MapReduce to retrieve and analyze data). It is always advantageous to find a simple basis for a complex field because it provides a way to manage the complexity of the field. *Vedic Science has discovered that the simplest form of awareness is the basis for the universe.*

HDFS

Definition

HDFS is a file system designed for storing very large files (petabytes or hundreds of terabytes) with streaming data access patterns, running clusters on commodity hardware.

HDFS

- Highly fault-tolerant (replication of data in multiple nodes)
- High throughput (yahoo uses 5000 nodes)
- Suitable for applications with very large data sets
- Streaming access to file system data
  - Write once and read many times.
  - Where getting the entire data faster is more important than getting a specific record.
- Can be built out of commodity hardware

HDFS is not suitable for

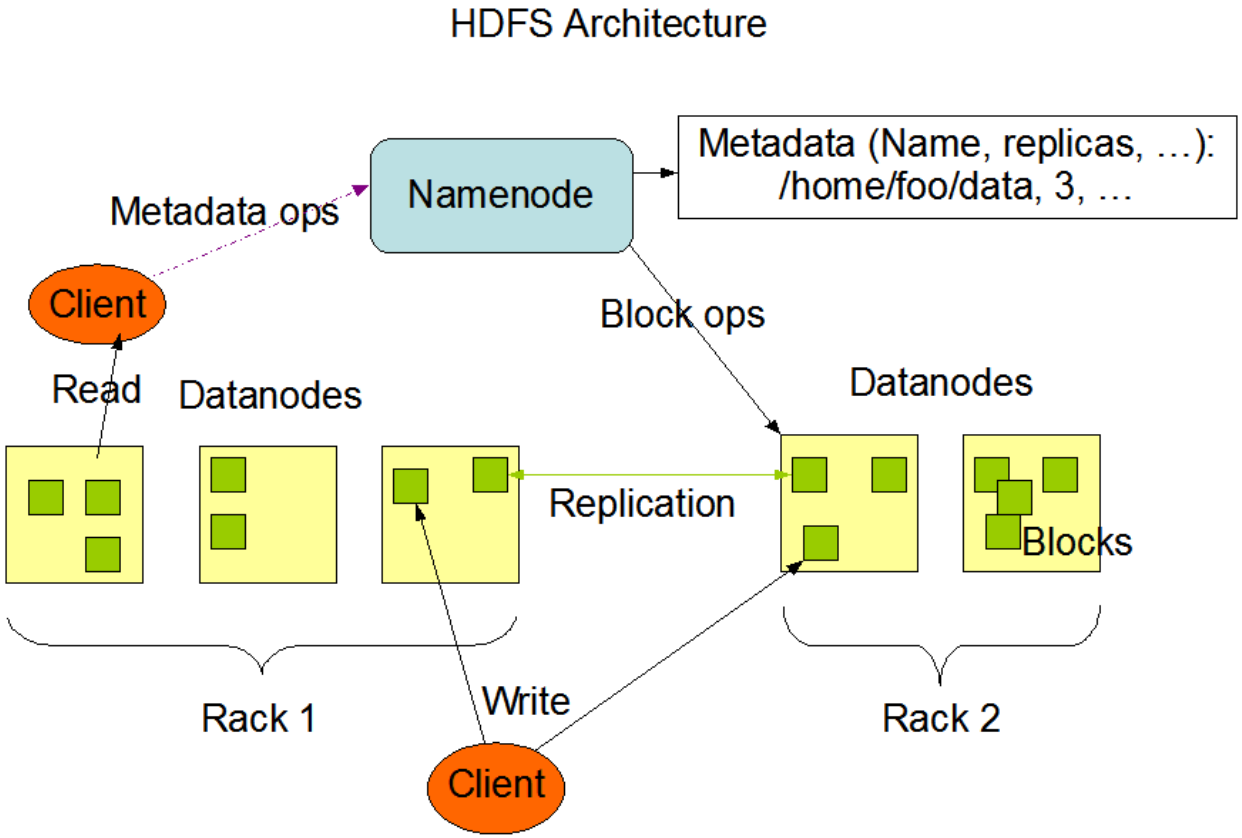
- Low-latency data access
  - latency: time interval between data request and data availability.
  - HDFS is optimized for high throughput of data at the expense of high latency
- Lots of small files
  - Because name node stores filesystem metadata in memory. Each file, directory, block takes about 150 bytes
- Multiple writers or file modification at arbitrary offsets
  - Might be supported in the future.

HDFS components

- NameNode
- DataNode

Note 1. NameNode, DataNode, Job Tracker and Task Tracker are daemons*

Note 2. Job Tracker and Task Tracker are in Hadoop 1.x and are NOT IN Hadoop 2.0 and up.



Img Ref: https://hadoop.apache.org/doc

*daemon: a service or process that runs in the background in any environment

NameNode

- Master of the system
- Single Point of Failure (SPoF)
- Very expensive hardware with double/triple redundancy (RAID*)
- Responsibility: manages the filesystem namespace
  - maintains the filesystem tree and the metadata for all the files
  - maintains and manages the blocks that are present on DataNodes

*RAID (Redundant Array of Independent Disks) is a storage technology that combines multiple disk drive components into a logical unit for the purposes of data redundancy and performance improvement.

Secondary NameNode

- NameNode keeps all the data in the RAM.
- Secondary NameNode Reads data from the RAM of the NameNode and writes into hard drive.
- In Hadoop 1.x Secondary NameNode will not become (Primary) NameNode if the NameNode fails.

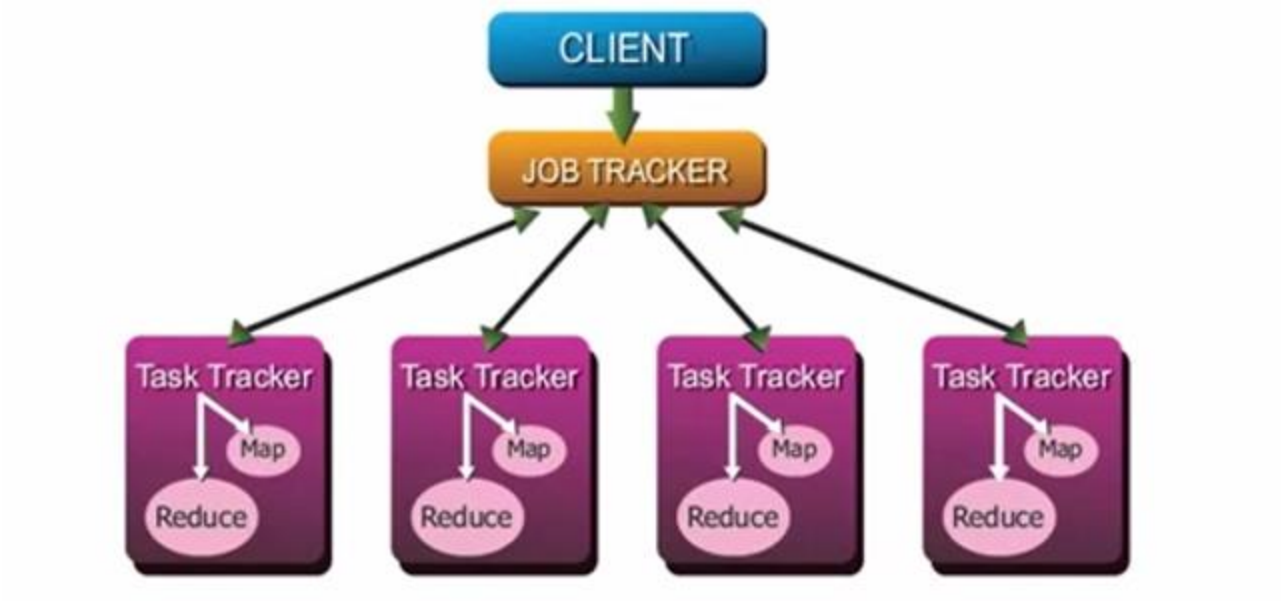
DataNode

- Slaves which are deployed on each machine to provide the actual storage
- Responsible for serving the read and write requests from the clients or namenode
- Datanodes report back to the namenode periodically with lists of blocks that they are storing

Main Point 2

HDFS is a file system designed for storing very large files (in terabytes) with streaming data access patterns, running on clusters of commodity hardware. *All information in nature is ultimately in the Unified Field.*

Job Tracker and Task Tracker (Hadoop 1.x) (Hadoop 2.0: Appendix Lesson 2)



HDFS Architecture

- A file in HDFS is broken into block-sized chunks, which are stored as independent units.
- Default Block size is 128MB (default in linux is 8KB).
- A file in HDFS that is smaller than a single block does not occupy a full block's storage space.
  - Example: A 1 MB file stored with block size of 128 MB uses 1 MB of disk space, not 128 MB.
  - Example: https://data-flair.training/blogs/data-block/

Large block size: pros and cons

Minimizes the cost of seeks. If the block is large enough, the time to transfer the data from the disk can be significantly larger than the time to seek to the start of the block.

MapReduce tasks operate on one input-split (see next slide to see how input-split is related to a block) at a time. So if data is not distributed into many datanodes, we are not taking advantage of the parallelism possible to its fullest extend.

Input-Splits

Motivating example

Suppose I wrote a book with 24 pages. I tore the book and gave one page to each of you. Now I am asking you to do some simple calculation for me as shown below.

Sample page you got:

is beautiful. Then all on a sudden
rain started. It went on for three
days. As anything in life, nothing
lasts forever! The rain ended, Sun
came out bringing smile in all. What
a wonderful world! Yes, the song by

Expected output is number of words in each sentence. So for the above page, the output is 7, 6, and so on. When it comes to the last sentence, "Yes, the song by" you cannot compute the number of words since the last sentence is not complete. You need that missing data from next page.

Input-Splits

Block is similar to a page (the unit of storage)
Record is similar to a sentence (the unit of processing).

Since a file is stored as blocks, records gets broken, However, to process "the program" needs entire record.
So during processing the missing part of the last record is made available by the Hadoop.

In our example, the input-split is

Then all on a sudden rain started. It went on for three days. As anything in life, nothing lasts forever! The rain ended, Sun came out bringing smile in all. What a wonderful world! Yes, the song by Nat King Cole!

Note that **is beautiful**. in the beginning is part of previous input-split. While **Nat King Cole!** is part of this input-split which is in another page (or Block).

Input-Splits

Example. Assume the following:

data block size is 64 KB.

file size is 150 KB. 15 records; each of size 10 KB.

Now the file is stored in three data blocks:

0 – 63 KB in the first block, 64 – 127 KB in the second block, 128 – 149KB in the third block.

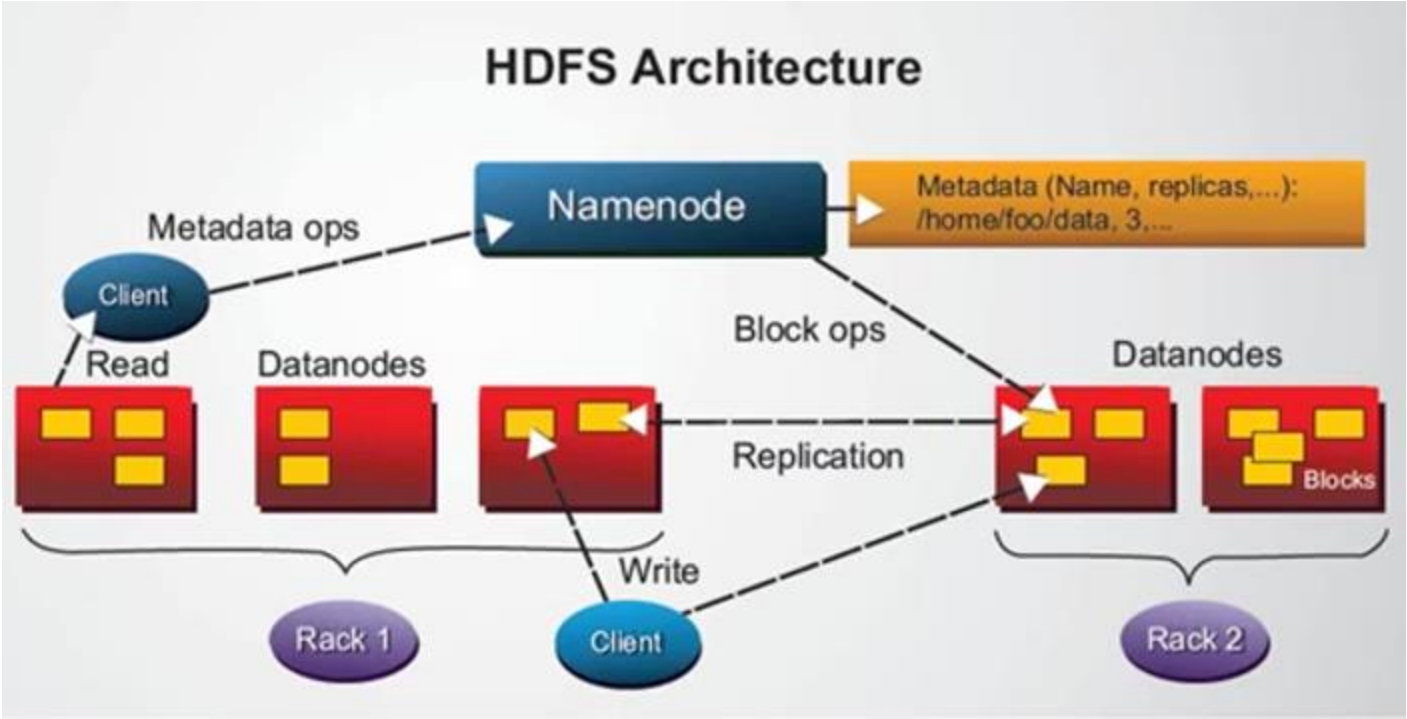
0-6 records are in the 1st input-split **(0 – 69 KB)** (the 1st input-split actually extends beyond the 1st block and gets the “missing part” of the last record from block 2)

7-12 records are in the 2nd input-split **(70 -129 KB)**

13 and 14 records are in the 3rd input-split **(130 - 149 KB)**

HDFS Architecture

Read and Write



Anatomy of file read

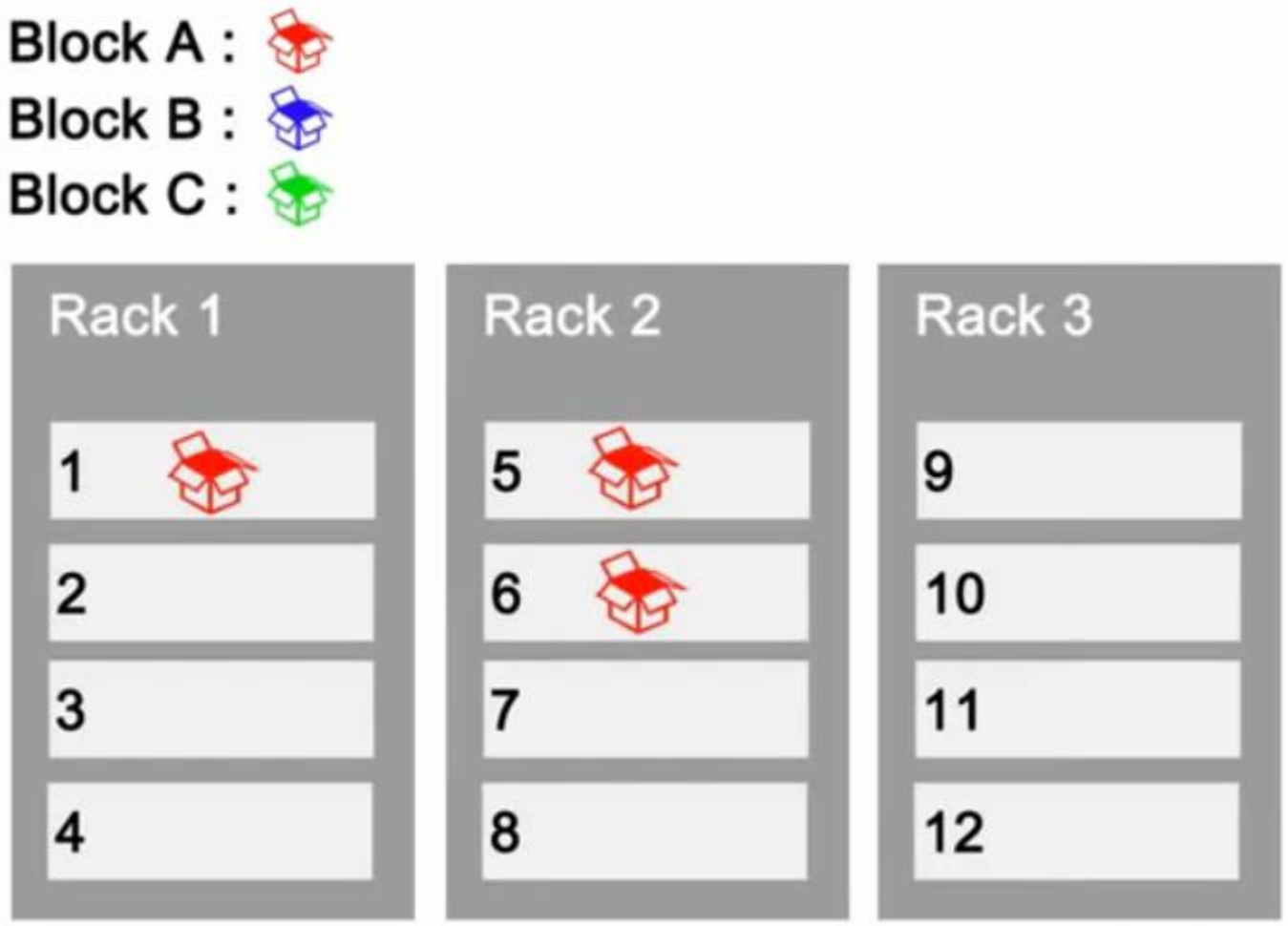
Important Points:

1. Datanode directly perform the I/O.
2. Data never move through Namenode.

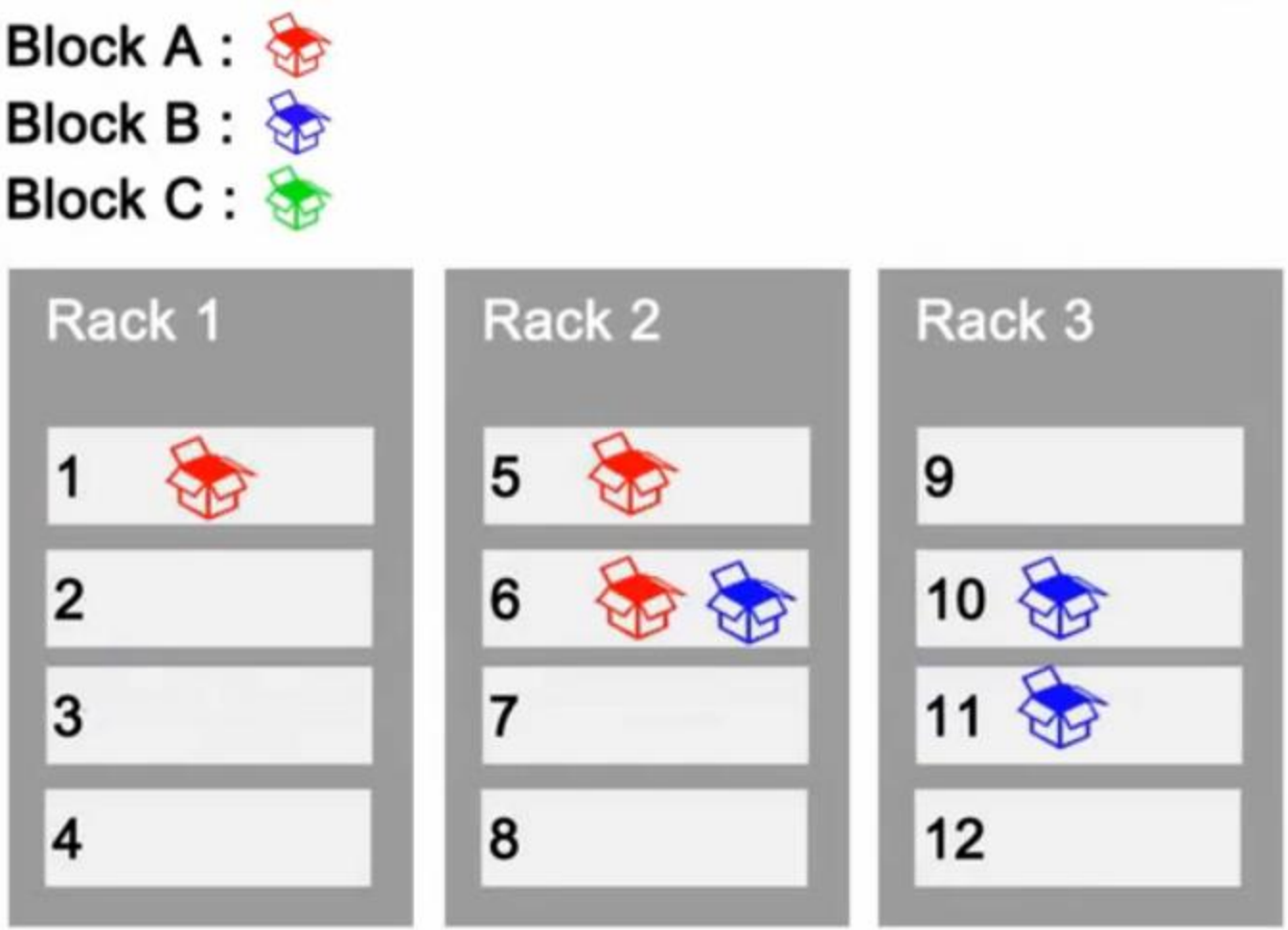
Read Tom White's Book for details.

Replication Example:
Three data blocks (A, B, C), three racks and each rack has 4 datanodes

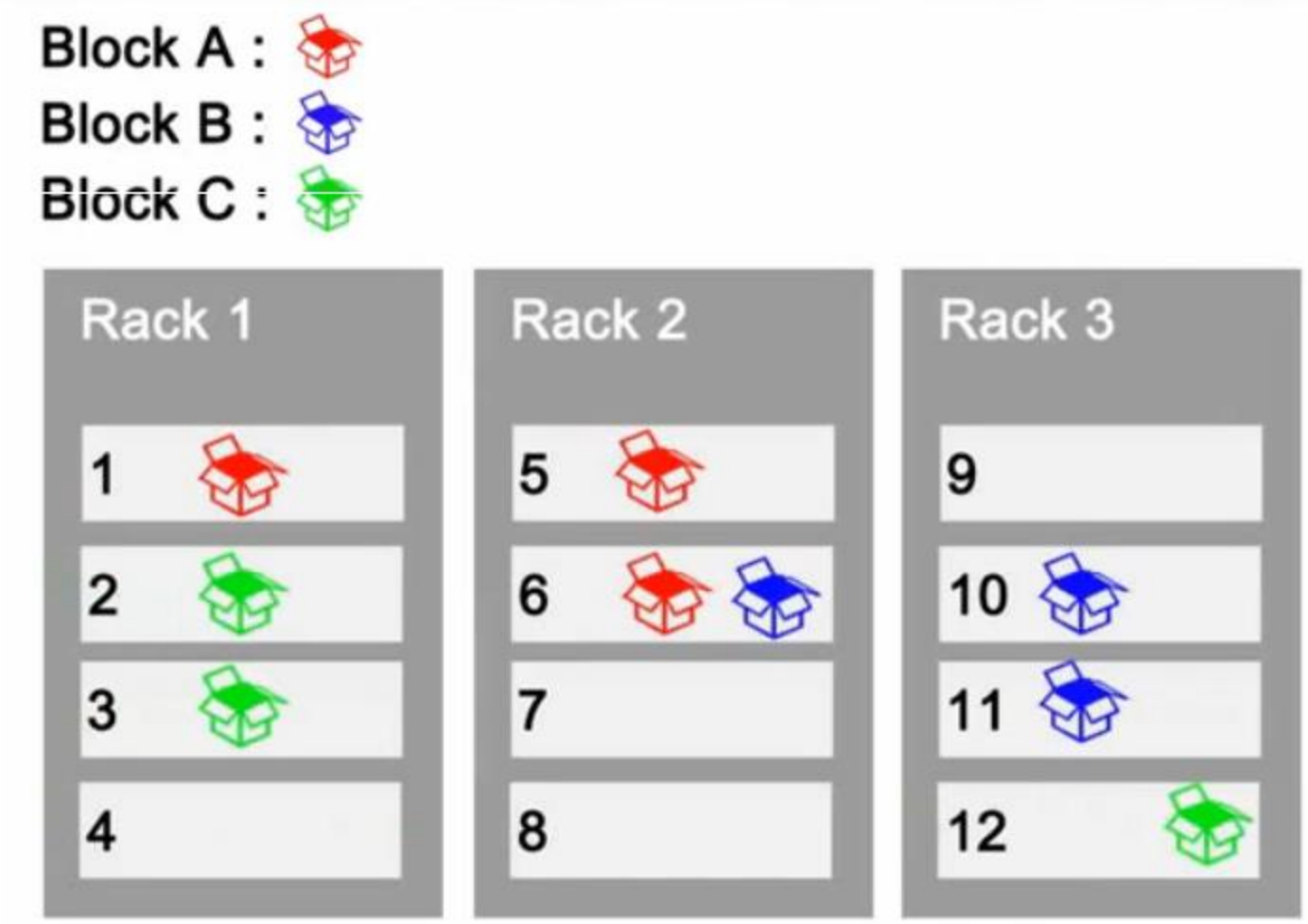
After placing Block A



After placing Block B



After placing Blocks A, B and C



Main Point 3

MapReduce paradigm is used to extract valuable information from big data. The objective means of gaining knowledge attempts to extract knowledge from the ever-changing relative field of existence. *The subjective means of gaining knowledge starts with the wholeness of the non-changing Absolute, which is the basis of the changing relative.*

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

1. HDFS is a simple and abstract form of file system to store and retrieve large data sets.

1. Hadoop is found to be an ideal solution to deal with big data.

---

3. ***Transcendental consciousness:*** *is the experience of the simplest and most abstract state of awareness which underlies all states of greater excitation.*

4. ***Impulses within the Transcendental Field:*** *Nature accomplishes what it needs by having its impulses in the transcendental field be as efficient as possible.*

5. ***Wholeness moving within itself:*** *In unity consciousness one experiences everything as excitations of pure consciousness that underlies and connects all diversity.*