Big Data


BIG
DATA
ANALYTICS
SOFTWARE
SIZE
NEEDED
PETABYTES
INTERNET
TECHNOLOGIES
MANAGEMENT
LARGE
EVERY
STORAGE
RESEARCH
DATABASES
COMPLEX
SETS
EXAMPLES
ABILITY
INFORMATION
NETWORKS
INCLUDE
SYSTEMS
SHARED
TIME
RECORDS
SOCIAL
SAN
PARALLEL
HUNDREDS

What is Big data?

- ‘Big Data’ is similar to ‘small data’, but bigger in size.
- but having data bigger it requires different approaches:
  -Techniques, tools and architecture
- Big data is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them.

Sources of Big Data

- Social Media Data
- Black Box Data
- Stock Exchange Data
- Transport Data
- Power Grid Data
- Search Engine Data

- **Social Media Data:** Social media such as Facebook and Twitter hold information and views posted by millions of people across the globe.

- **Black Box Data:** It is a component of helicopter, airplanes, and jets, etc. It captures voices of the flight crew, recordings of microphones and earphones, and the performance information of the aircraft.

- **Stock Exchange Data:** The stock exchange data holds information about the ‘buy’ and ‘sell’ decisions made on a share of different companies made by the customers.

- **Transport Data:** Transport data includes model, capacity, distance and availability of a vehicle.
- **Search Engine Data:** Search engines retrieve lots of data from different databases.
- **Power Grid Data:** The power grid data holds information consumed by a particular node with respect to a base station.

Three Vs of Big Data

Velocity

- Data speed

Volume

- Data quantity

Variety

- Data Types

Velocity

- high-frequency stock trading algorithms reflect market changes within microseconds
- machine to machine processes exchange data between billions of devices
- on-line gaming systems support millions of concurrent users, each producing multiple inputs per second.

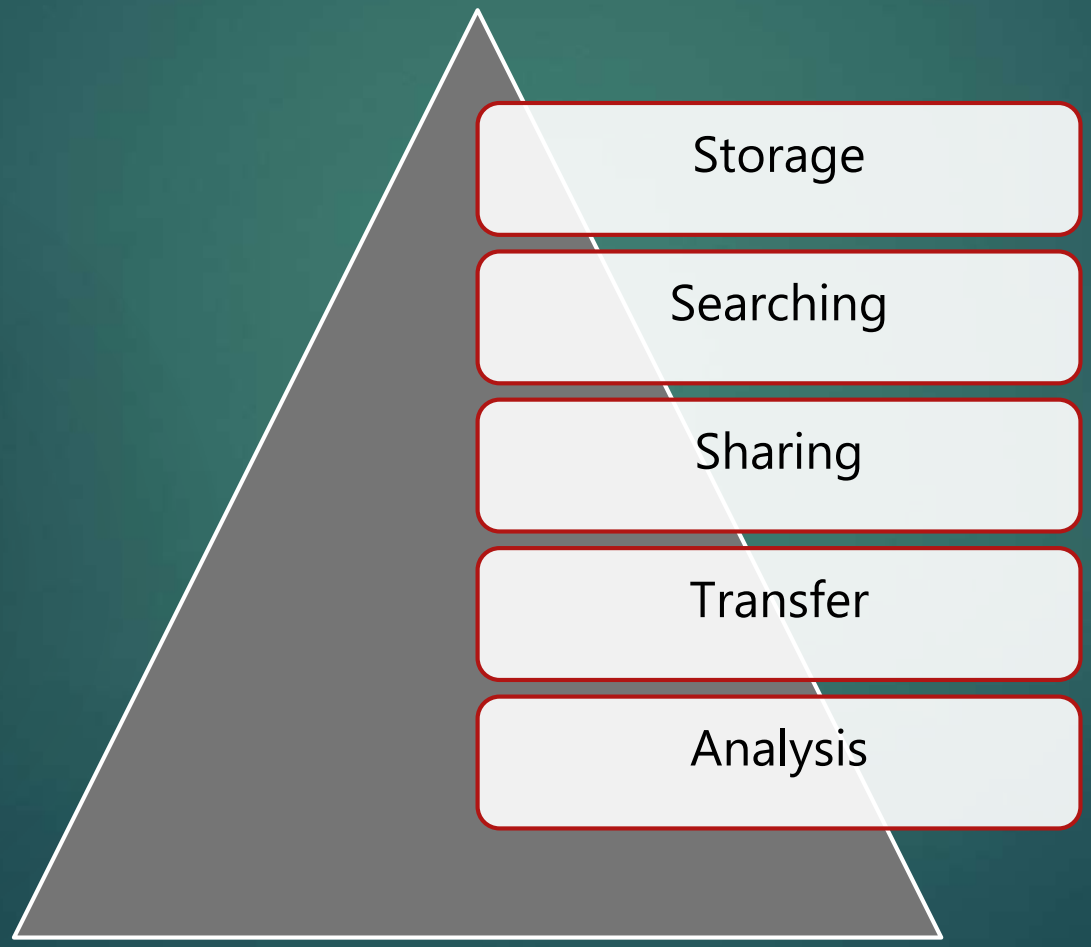
Volume

- A typical PC might have had 10 gigabytes of storage in 2000.
- Today, Facebook ingests 600 terabytes of new data every day.
- The smart phones, the data they create and consume; sensors embedded into everyday objects will soon result in billions of new, constantly-updated data feeds containing environmental, location, and other information, including video.

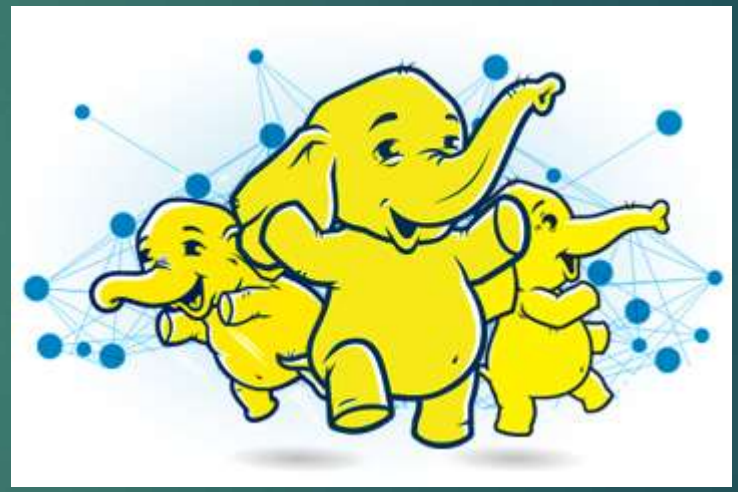
Variety

- Big Data isn't just numbers, dates, and strings. Big Data is also geospatial data, 3D data, audio and video, and unstructured text, including log files and social media.
- Traditional database systems were designed to address smaller volumes of structured data, fewer updates or a predictable, consistent data structure.
- Big Data analysis includes different types of data.

Challenges



Hadoop



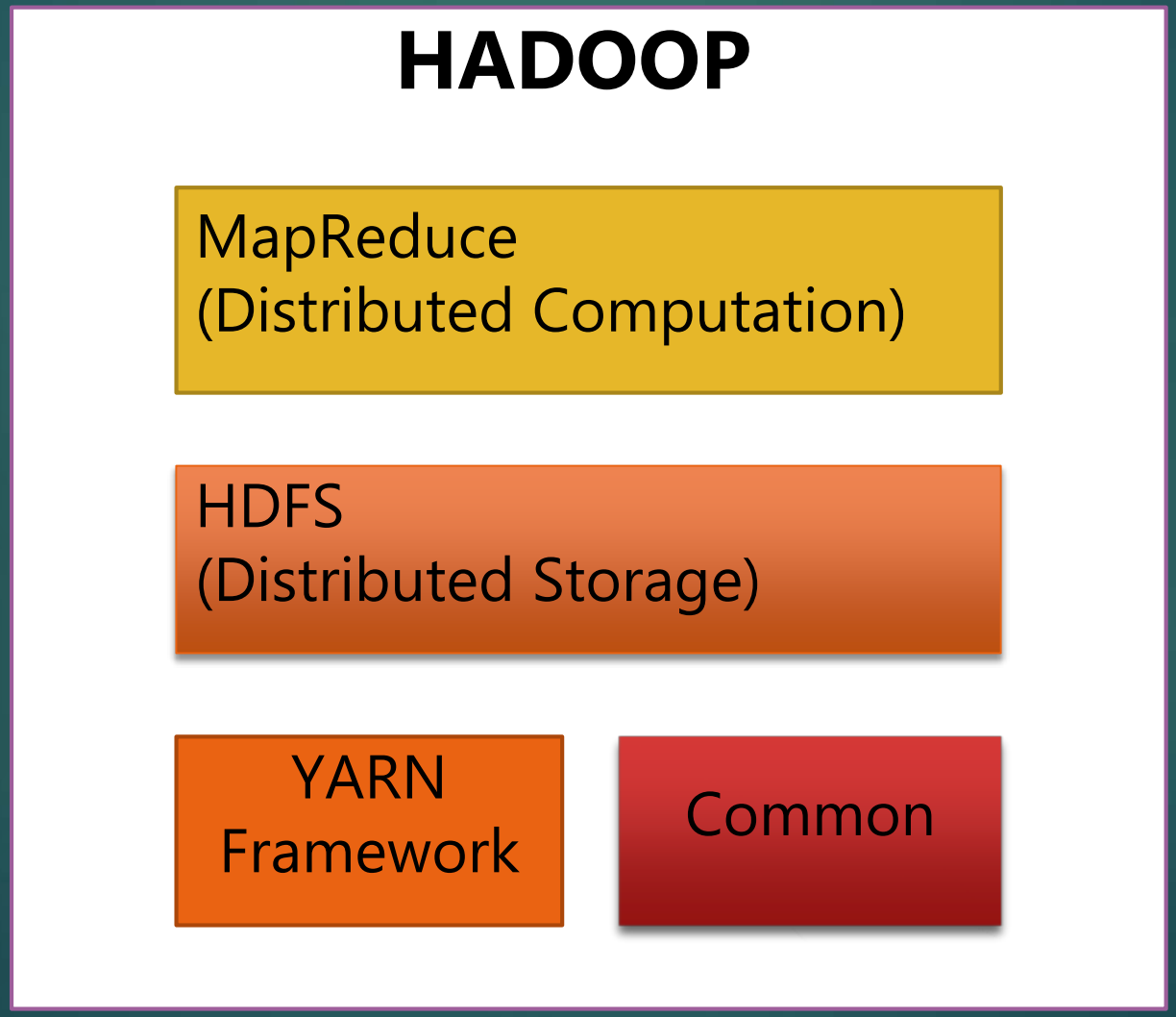
History of Hadoop

- Hadoop was created by computer scientists Doug Cutting and Mike Cafarella in 2005.
- It was inspired by Google's MapReduce, a software framework in which an application is broken down into numerous small parts.
- Doug named it after his son’s toy elephant.
- In November 2016 Apache Hadoop became a registered trademark of the Apache Software Foundation.

What is Hadoop?

- Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment.
- Hadoop runs applications using the mapreduce algorithm, where the data is processed in parallel on different CPU nodes.
- Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating in case of a node failure.
- Hadoop can perform complete statistical analysis for a huge amount of data.

Hadoop Architecture



HADOOP COMMON:

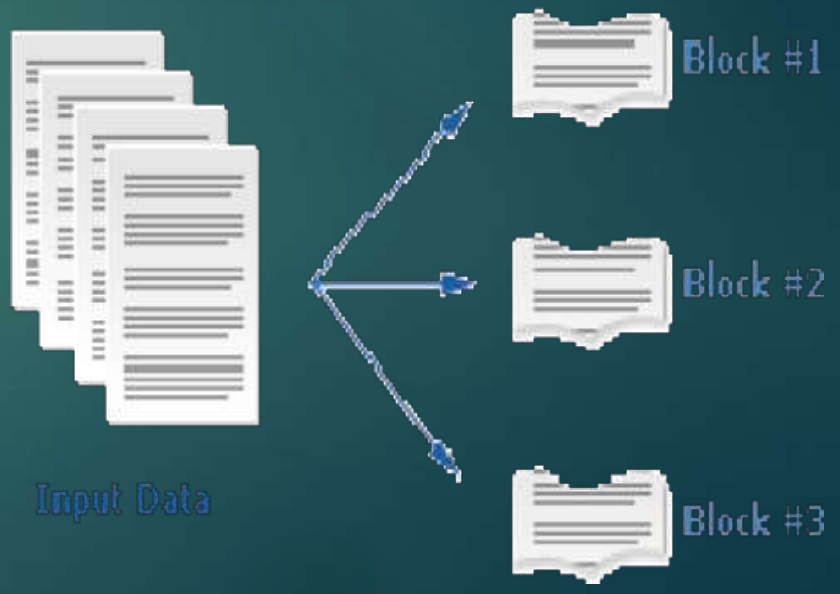
- Common refers to the collection of common utilities and libraries that support other Hadoop modules.
- These libraries provides file system and OS level abstraction and contains the necessary Java files and scripts required to start Hadoop.

HADOOP YARN:

- Yet Another Resource Negotiator
- a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications

HDFS

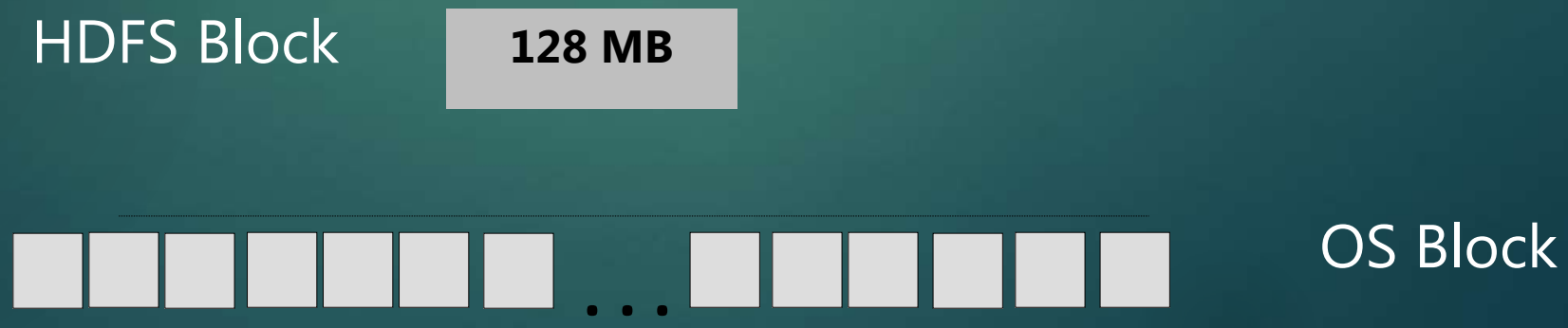
- Hadoop Distributed File System.
- Hadoop file system that runs on top of existing file system
- Designed to handle very large files with streaming data access patterns
- Uses blocks to store a file or parts of a file.



HDFS - Blocks

File Blocks

- 64MB (default), 128MB (recommended) – compare to 4 KB in UNIX
- Behind the scenes, 1 HDFS block is supported by multiple operating system (OS) blocks
- Fits well with replication to provide fault tolerance and availability



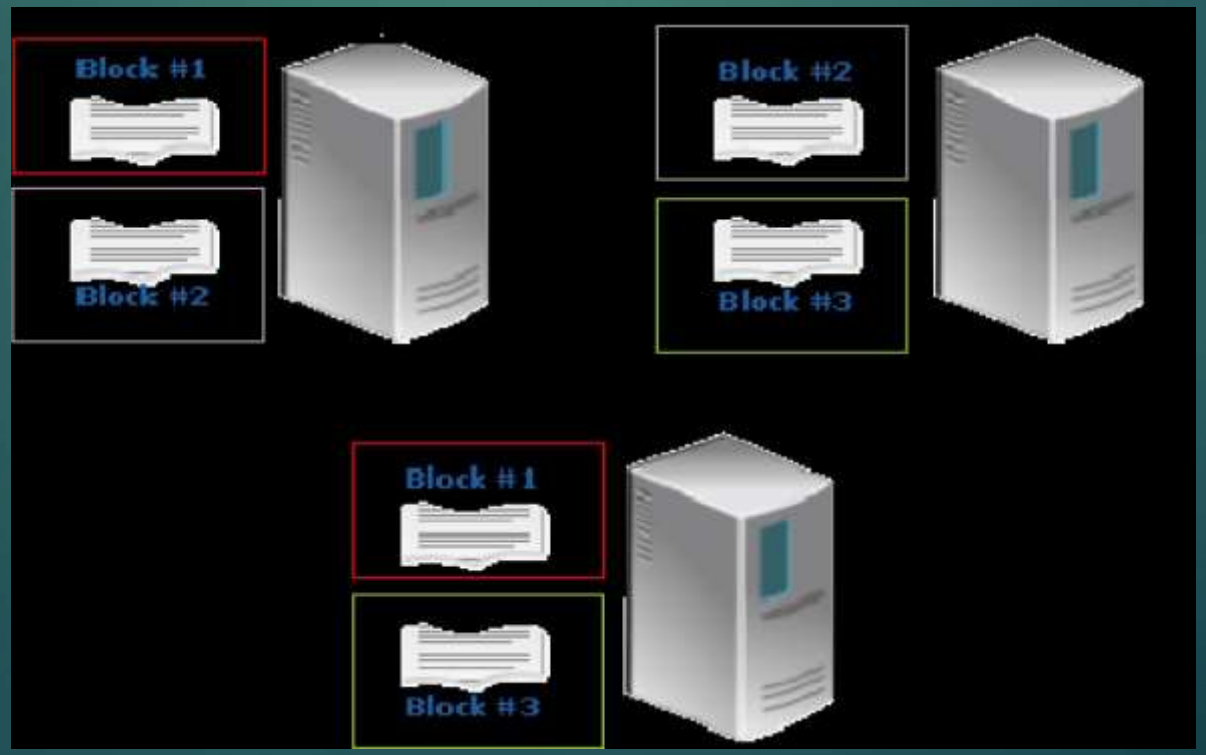
Advantages of blocks

- Fixed size – easy to calculate how many fit on a disk
- file can be larger than any single disk in the network
- If a file or a chunk of the file is smaller than the block size, only needed space is used. Eg: 420MB file is split as:

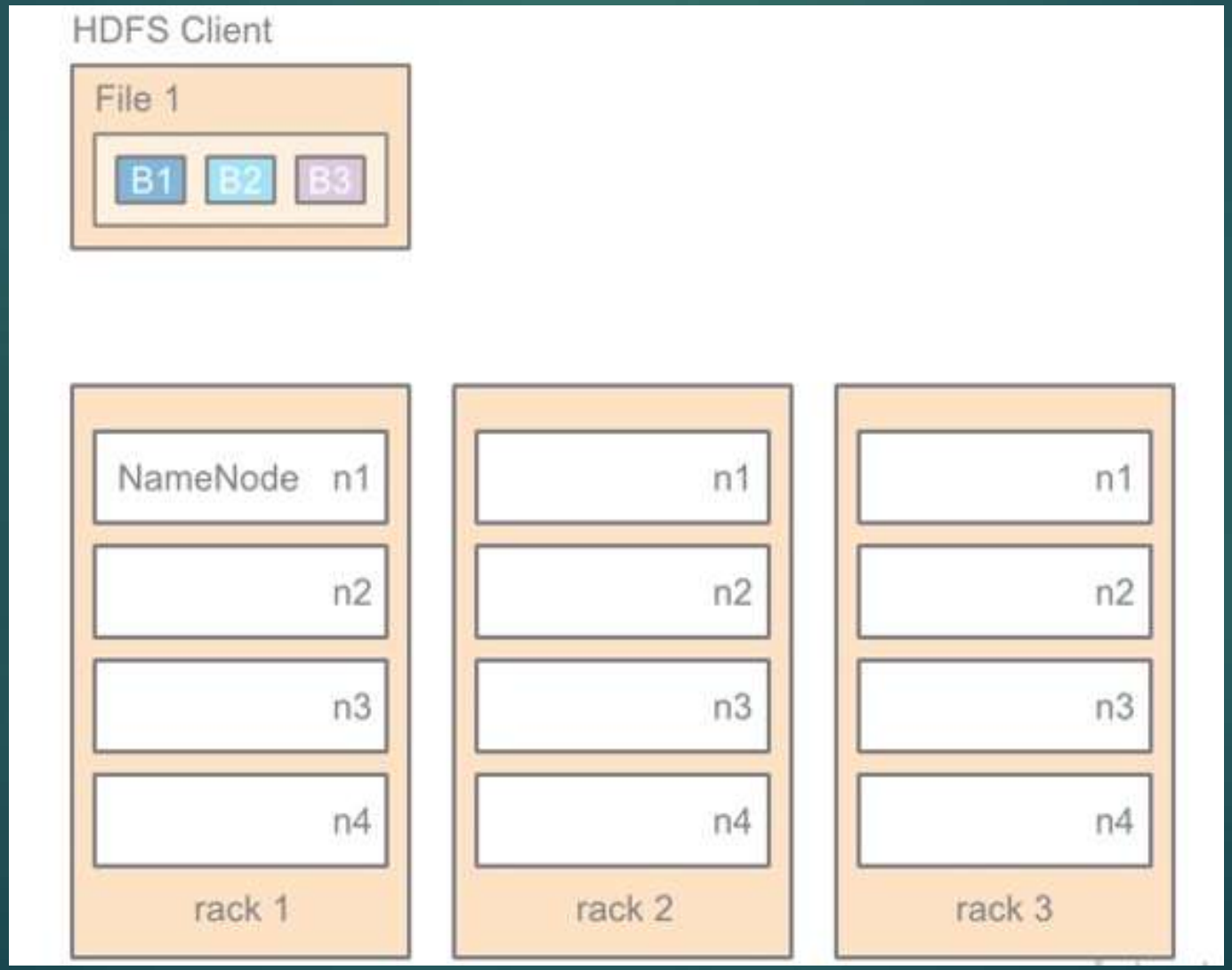
128 MB	128 MB	128 MB	36 MB

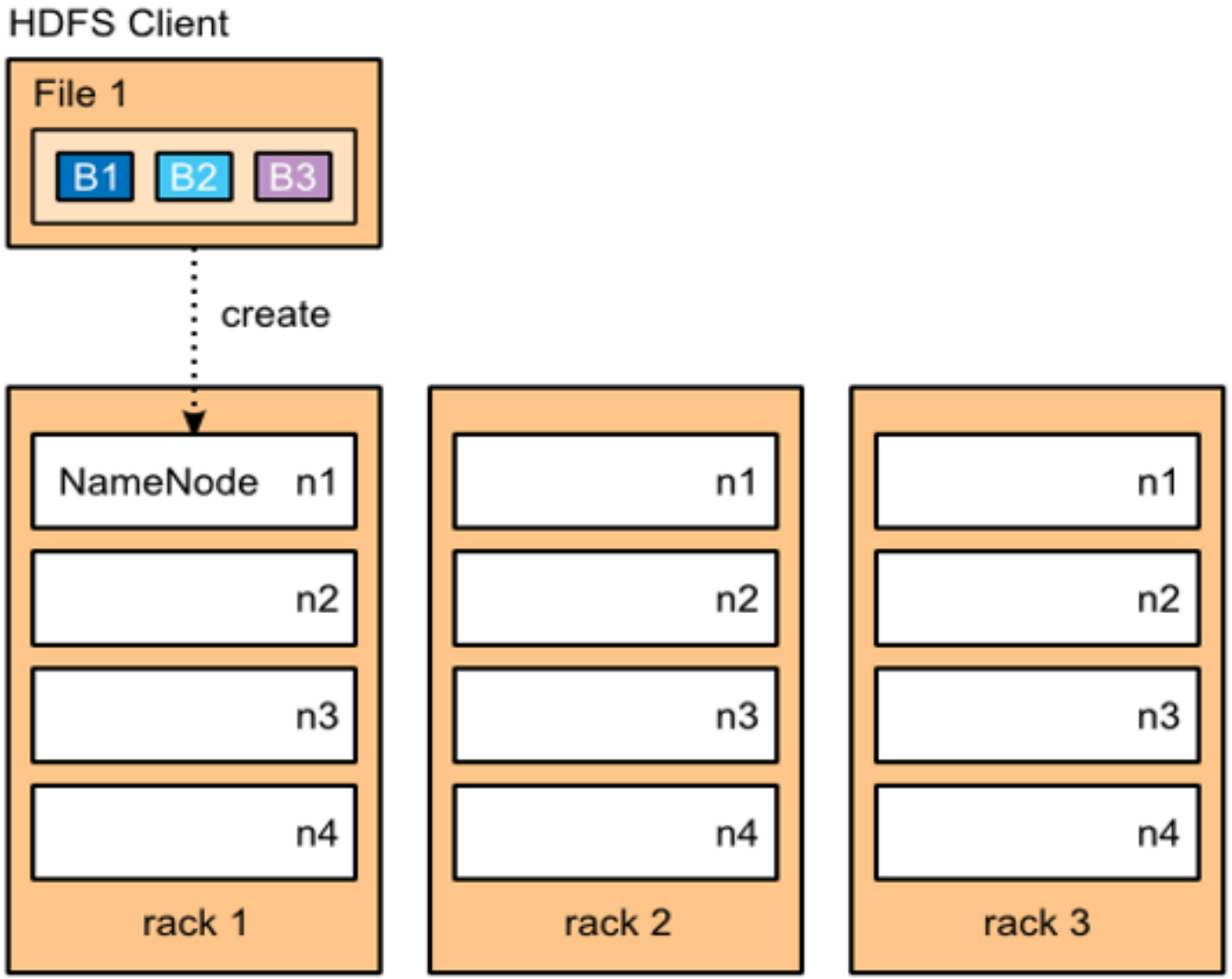
HDFS -Replication

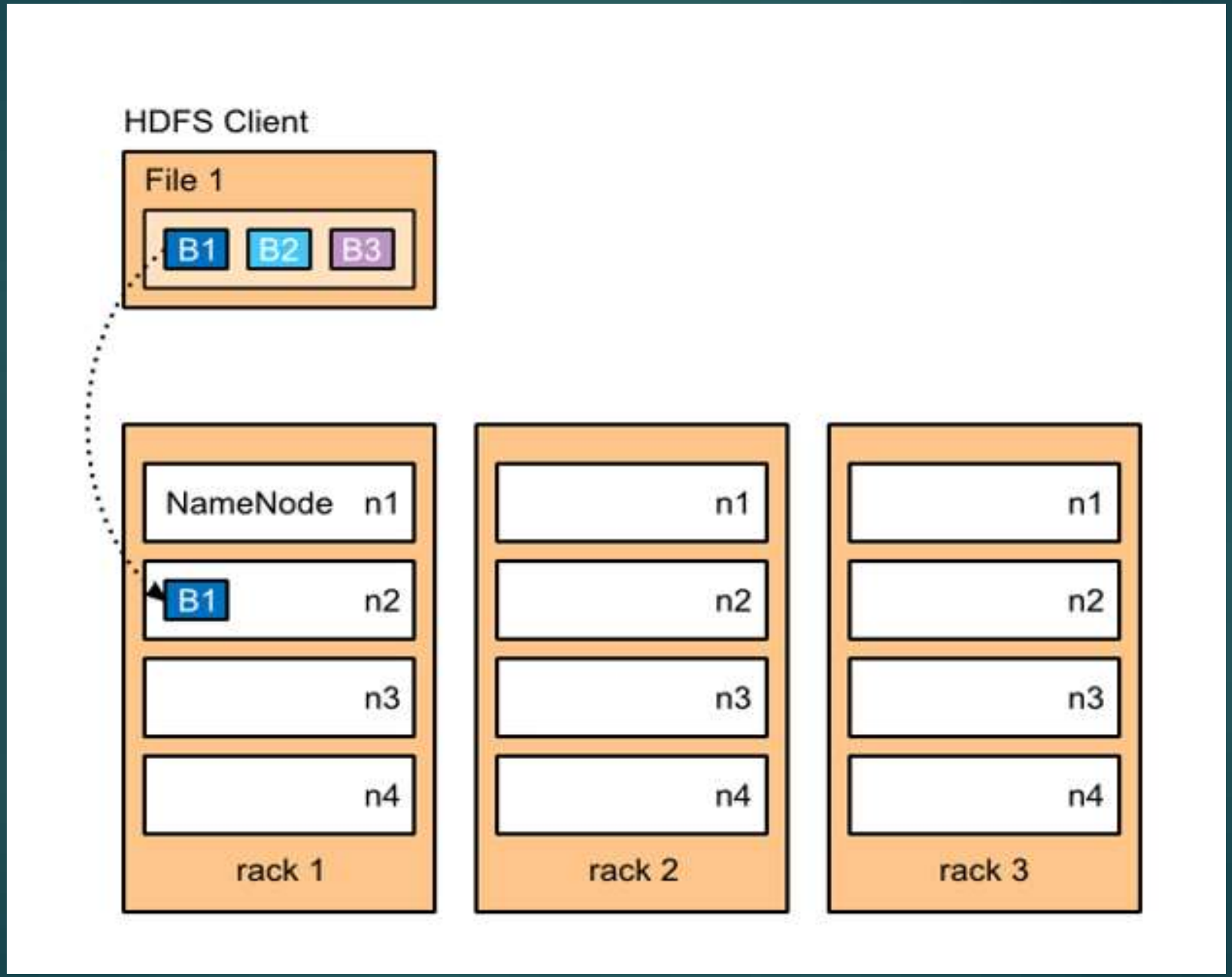
- Blocks with data are replicated to multiple nodes
- Allows for node failure without data loss

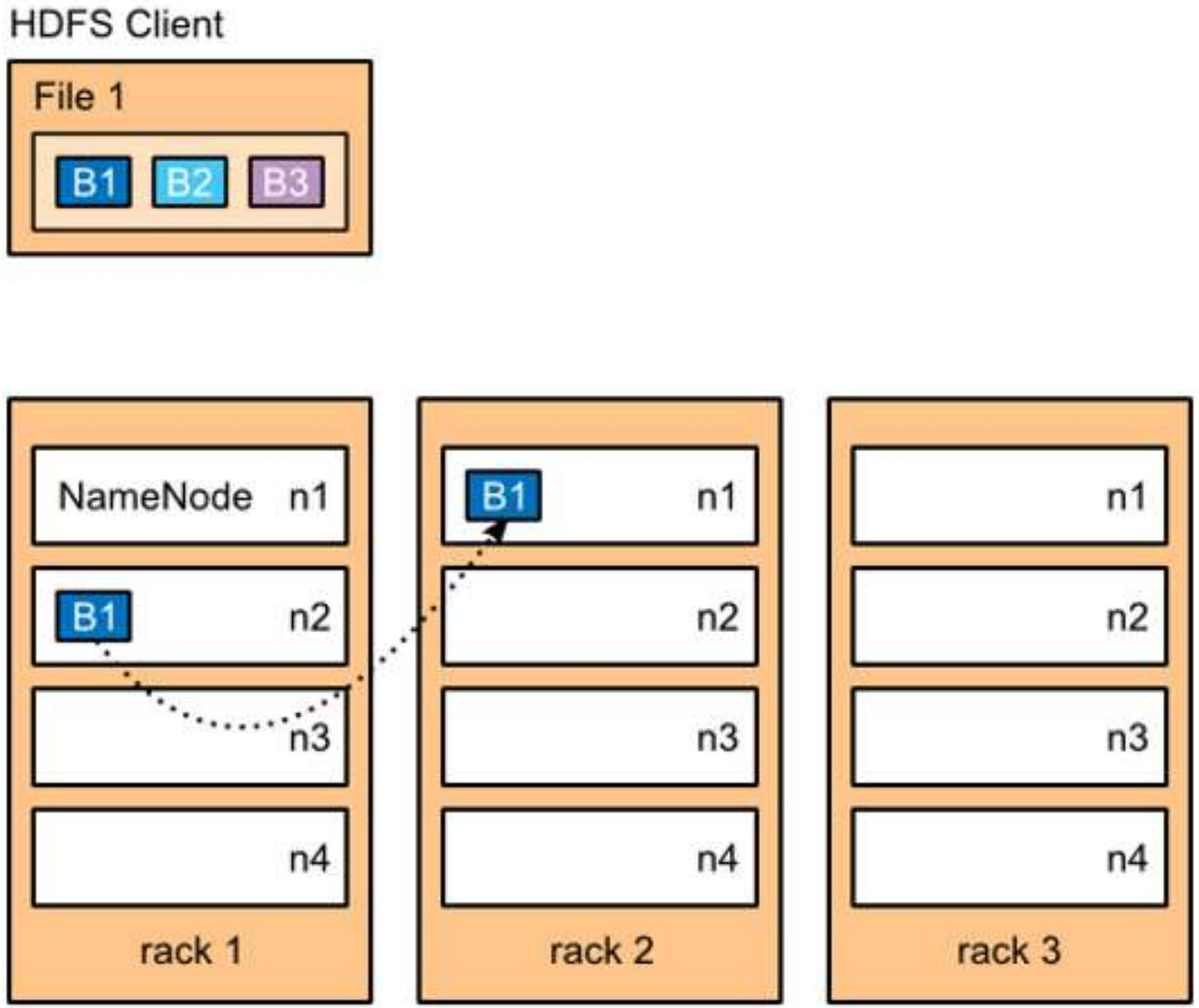


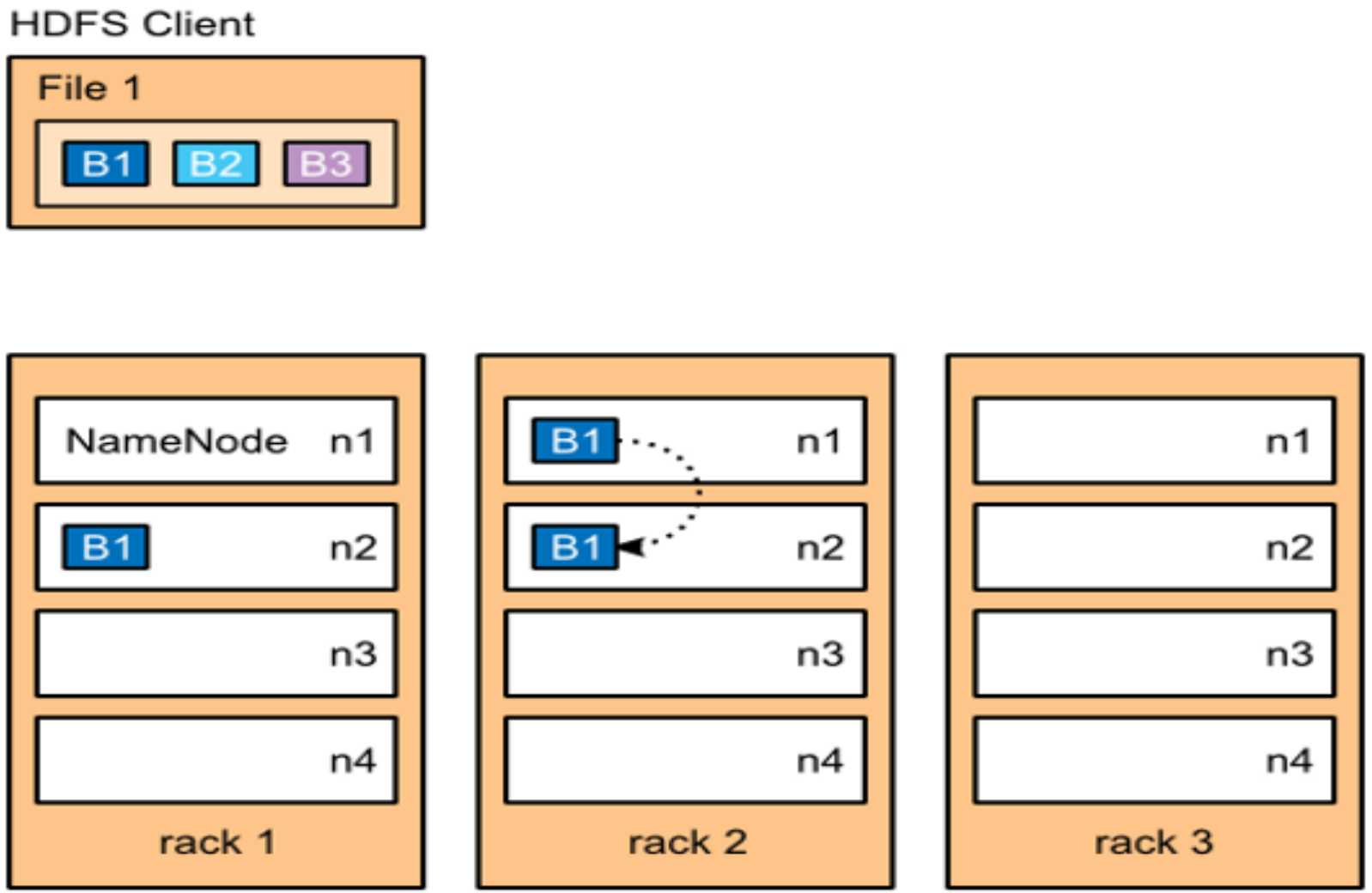
Writing a file to HDFS

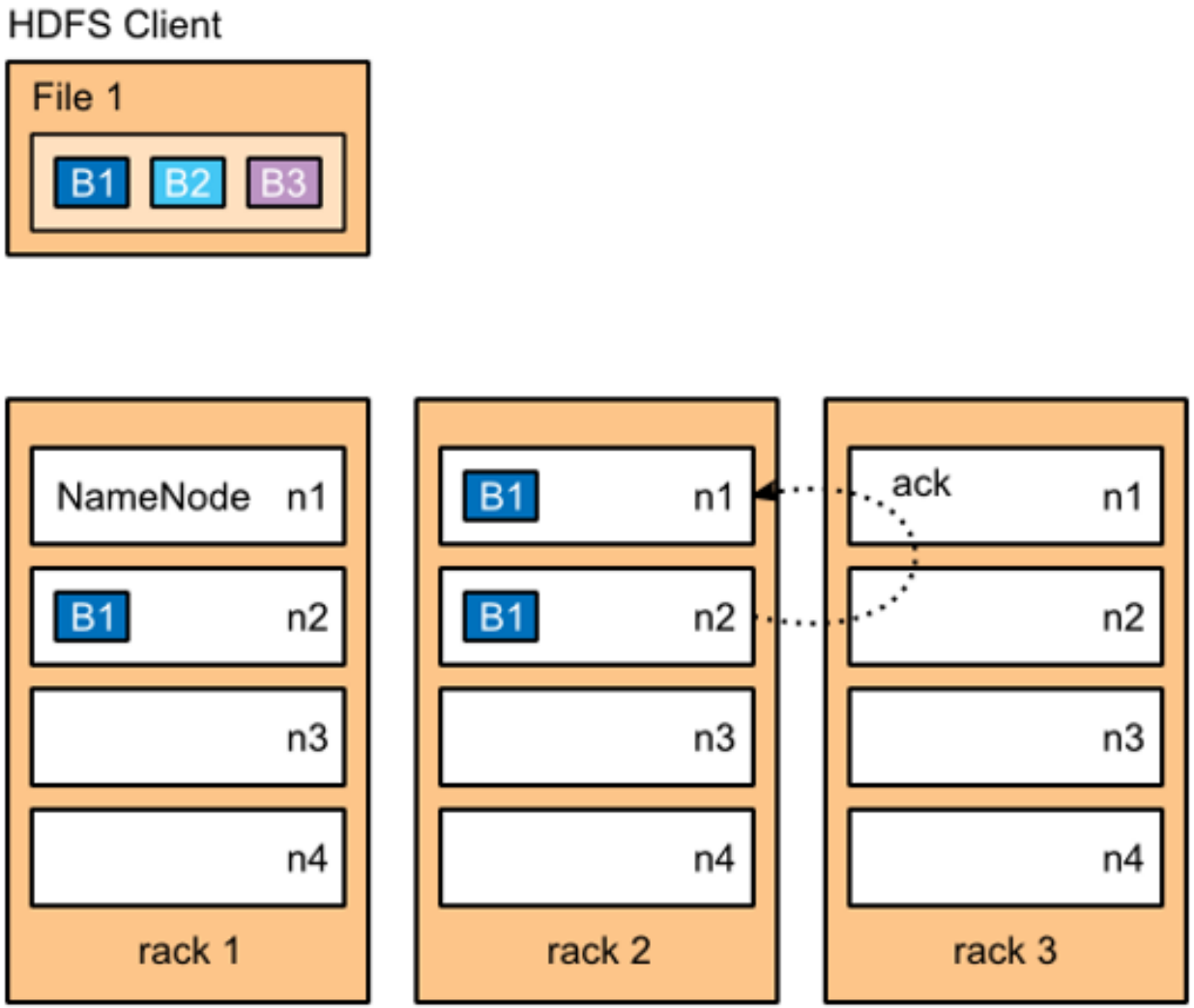



HDFS Client
File 1
B1
B2
B3
create
NameNode
n1
n2
n3
n4
rack 1
n1
n2
n3
n4
rack 2
n1
n2
n3
n4
rack 3


HDFS Client
File 1
B1
B2
B3
NameNode
n1
B1
n2
n3
n4
rack 1
n1
n2
n3
n4
rack 2
n1
n2
n3
n4
rack 3


HDFS Client
File 1
B1
B2
B3
NameNode
n1
B1
n2
n3
n4
rack 1
B1
n1
n2
n3
n4
rack 2
n1
n2
n3
n4
rack 3


HDFS Client
File 1
B1
B2
B3
NameNode
n1
B1
n2
n3
n4
rack 1
B1
n1
B1
n2
n3
n4
rack 2
n1
n2
n3
n4
rack 3


HDFS Client
File 1
B1
B2
B3
NameNode
n1
B1
n2
n3
n4
rack 1
B1
n1
B1
n2
n3
n4
rack 2
ack
n1
n2
n3
n4
rack 3

HADOOP MapReduce

COMPONENTS OF HADOOP

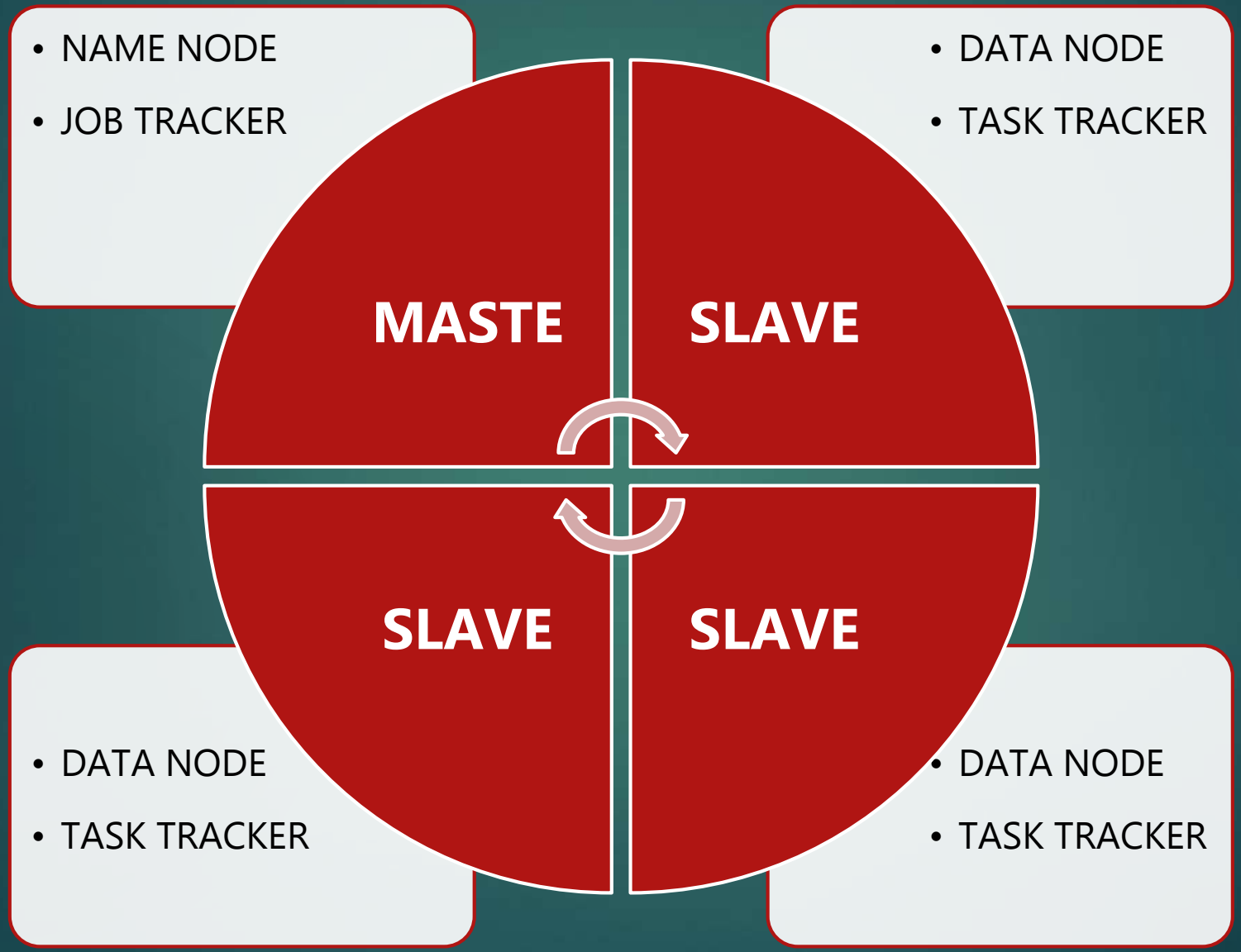
- HDFS
- MapReduce
- YARN Framework
- Libraries

A DEFINITION

- MapReduce is the heart of Hadoop. It is this programming paradigm that allows for massive scalability across hundreds or thousands of servers in a Hadoop cluster.
- MapReduce is the original framework for writing applications that process large amounts of structured and unstructured data stored in the Hadoop Distributed File System (HDFS).
- MapReduce is a patented framework by GOOGLE to support distributed computing on large data sets.

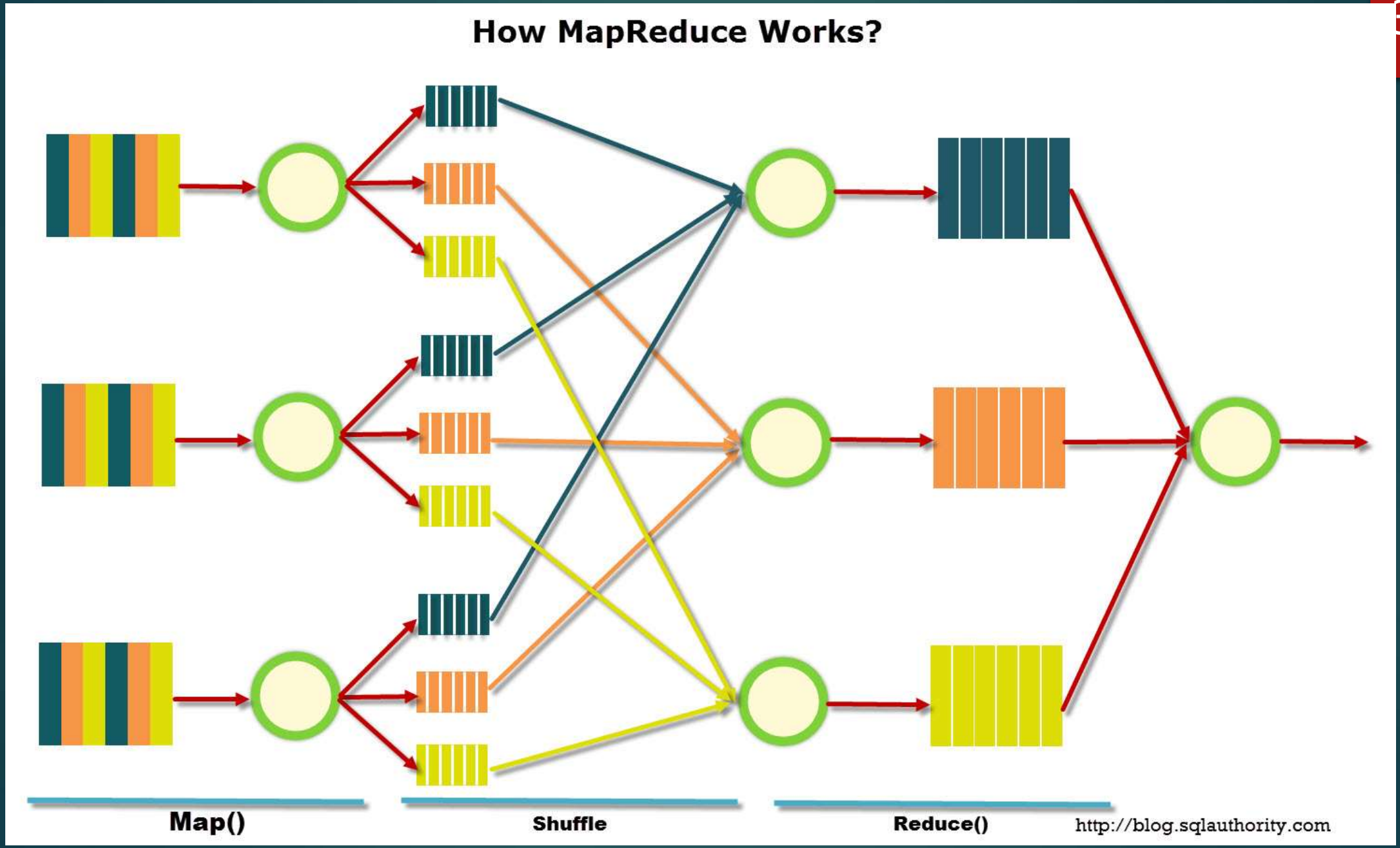
INSPIRATION

- The name MapReduce comes from functional programming
- **map** is the name of a higher-order function that applies a given function to each element of a list.
- **reduce** is the name of a higher-order function that analyze a recursive data structure and recombine through use of a given combining operation the results of recursively processing its constituent parts, building up a return value.


• NAME NODE
• JOB TRACKER
• DATA NODE
• TASK TRACKER
MASTE
SLAVE
SLAVE
SLAVE
• DATA NODE
• TASK TRACKER
• DATA NODE
• TASK TRACKER

HOW MapReduce WORKS?

- **Init** - Hadoop divides the input file stored on HDFS into splits (typically of the size of an HDFS block) and assigns every split to a different mapper, trying to assign every split to the mapper where the split physically resides
- **Mapper** - Hadoop reads the split of the mapper line by line. Hadoop calls the method map() of the mapper for every line passing it as the key/value parameters - the mapper computes its application logic and emits other key/value pairs
- **Shuffle and sort** -Hadoop's partitioner divides the emitted output of the mapper into partitions, each of those is sent to a different reducer. Hadoop collects all the different partitions received from the mappers and sort them by key
- **Reduce**r -Hadoop reads the aggregated partitions line by line. Hadoop calls the reduce() method on the reducer for every line of the input - the reducer computes its application logic and emits other key/value pairs - locally, Hadoop writes the emitted pairs output (the emitted pairs) to HDFS


How MapReduce Works?
Map()
Shuffle
Reduce()
http://blog.sqlauthority.com

COMMON JOBS FOR MapReduce

TEXT MINING

INDEX BUILDING

GRAPHS

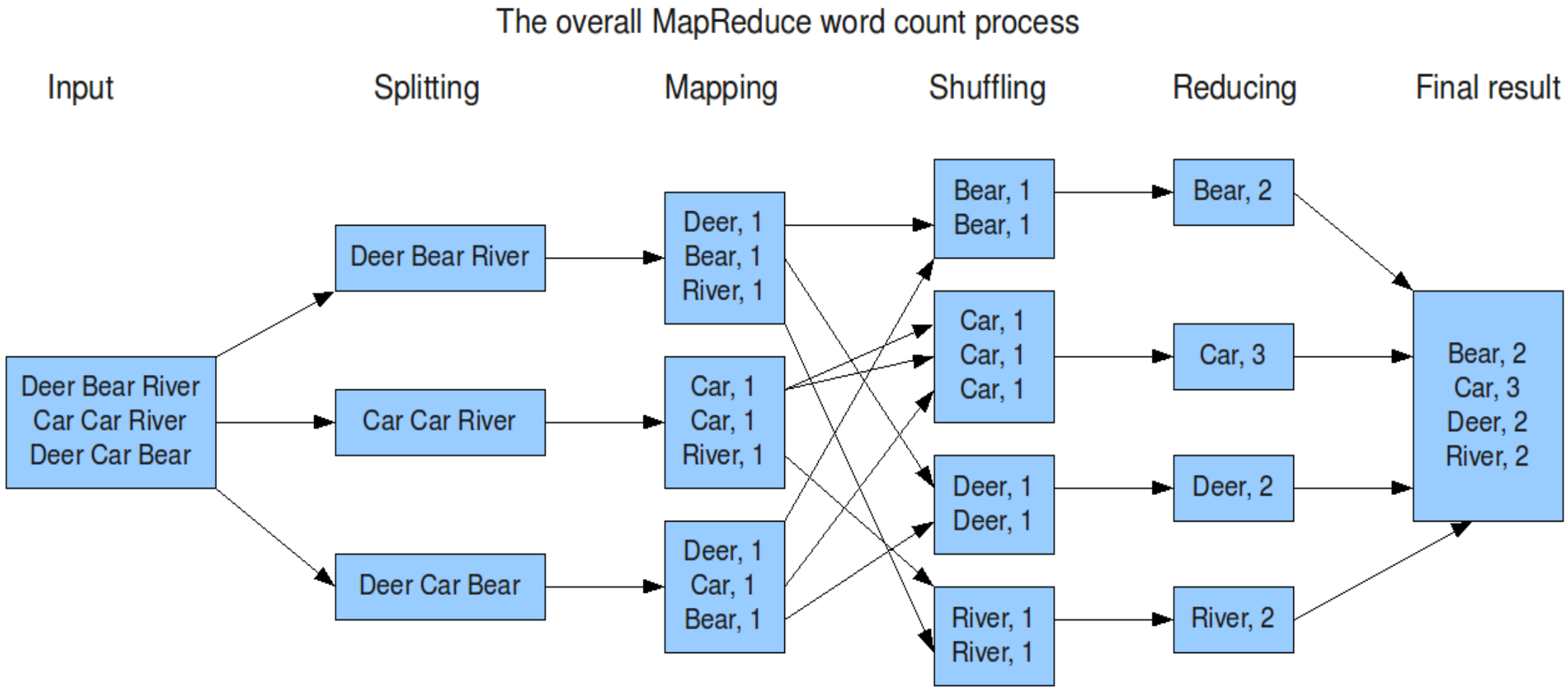
PATTERNS

FILTERING

PREDICTION

RISK ANALYSIS

WORD COUNT USING MapReduce



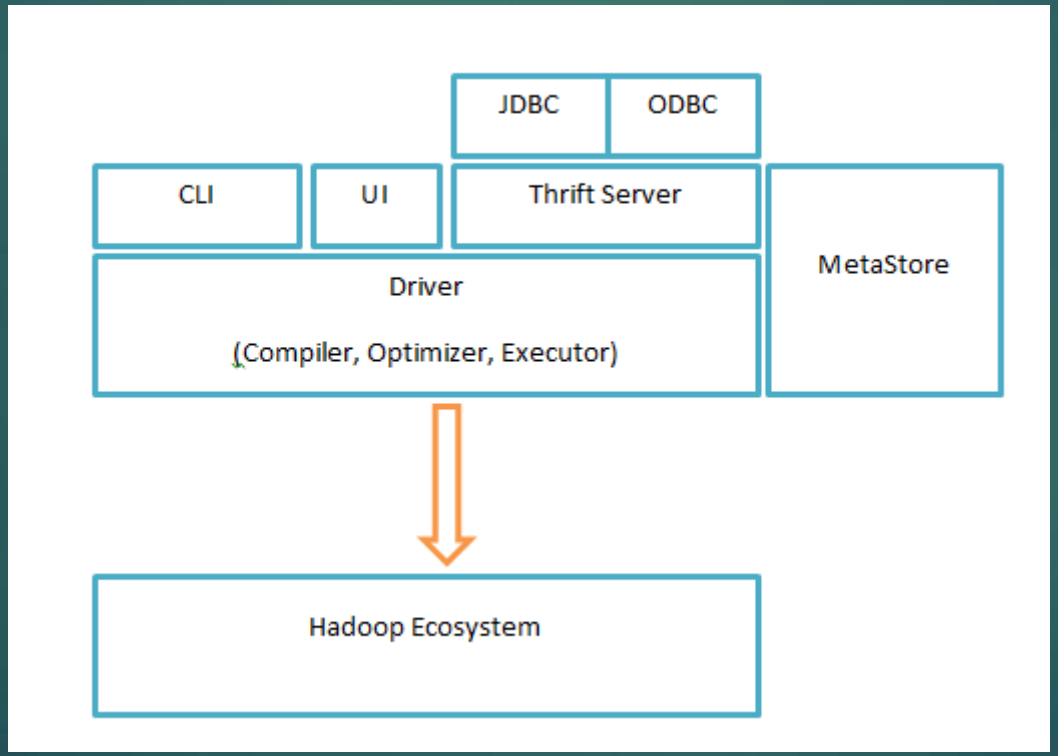
BENEFITS

- Simplicity
- Scalability
- Speed
- Recovery
- Minimal data motion

HIVE

- Originated as an internal project by facebook.
- data warehouse infrastructure built on top of Hadoop.
- SQL-like interface to query called HiveQL.
- Compiles query as Map Reduce jobs and runs them in cluster.
- Structures data into well defined database concept.

HIVE ARCHITECTURE



Apache PIG

What is Pig?

- **Pigs Eat Anything**

  Pig can operate on data whether it has metadata or not. It can operate on data that is relational, nested, or unstructured. And it can easily be extended to operate on data beyond files, including key/value stores, databases, etc.

- **Pigs Live Anywhere**

  Pig is intended to be a language for parallel data processing. It is not tied to one particular parallel framework. It has been implemented first on Hadoop, but we do not intend that to be only on Hadoop.

- **Pigs Are Domestic Animals**

  Pig is designed to be easily controlled and modified by its users.

- Pig Latin was designed to fit in a sweet spot between the declarative style of SQL, and the low-level, procedural style of MapReduce.
- Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs.
- Pig's infrastructure layer consists of
  - a compiler that produces sequences of Map-Reduce programs,
  - Pig's language layer currently consists of a textual language called Pig Latin.

KEY PROPERTIES OF PIG LATIN

- **Ease of programming.** It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.
- **Optimization opportunities.** The way in which tasks are encoded permits the system to optimize their execution automatically, allowing the user to focus on semantics rather than efficiency.
- **Extensibility.** Users can create their own functions to do special-purpose processing.

pig

- Cd $PIG_HOME/bin

grunt

- ./pig –x local