CSCI 5408

Dr. Saurabh Dey

saurabh.dey@dal.ca

Content

1. What is BigData and why is BigData needed?

3 Important Statistics About How Much Data Is Created Every Day

FinancesOnline
REVIEWS FOR BUSINESS

1 How much data is generated every minute?

Source: Domo

41,666,667
messages shared by WhatsApp users

1,388,889
video / voice calls made by people worldwide

404,444
hours of video streamed by Netflix users

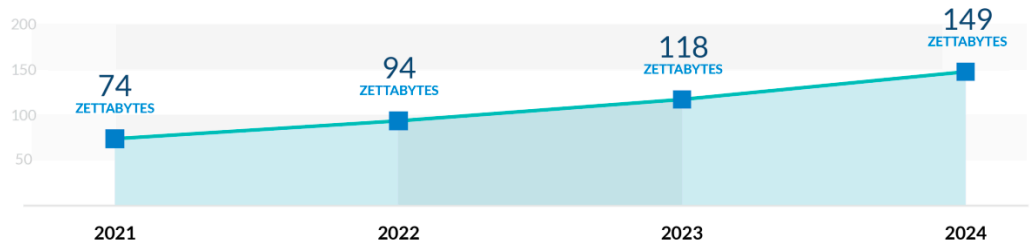
347,222
stories posted by Instagram users

150,000
messages shared by Facebook users

147,000
photos shared by Facebook users

2 Estimated Data Consumption from 2021 to 2024

Source: IDC / Statista



3 Data Growth in 2021

Sources: TechJury, Internet Live Stats, Cisco, PurpleSec

2 TRILLION
searches on Google by the end of 2021

1.134 TRILLION MB
volume of data created every day

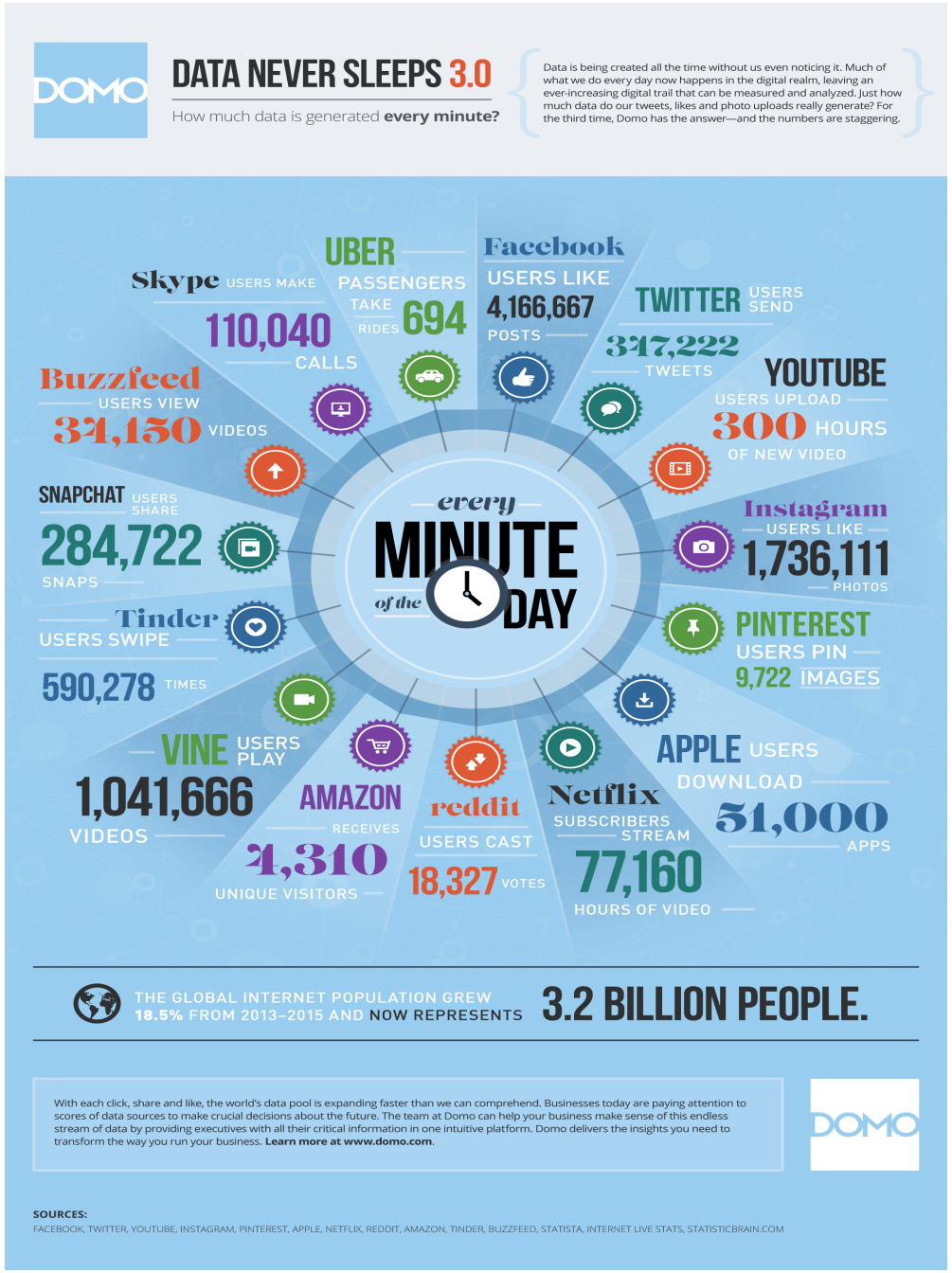
3,026,626
emails sent every second, 67% of which are spam

278,108 PETABYTES
global IP data per month by the end of 2021

230,000
new malware versions created every day

82%
share of video in total global internet traffic at the end of 2021

https://financesonline.com/how-much-data-is-created-every-day/



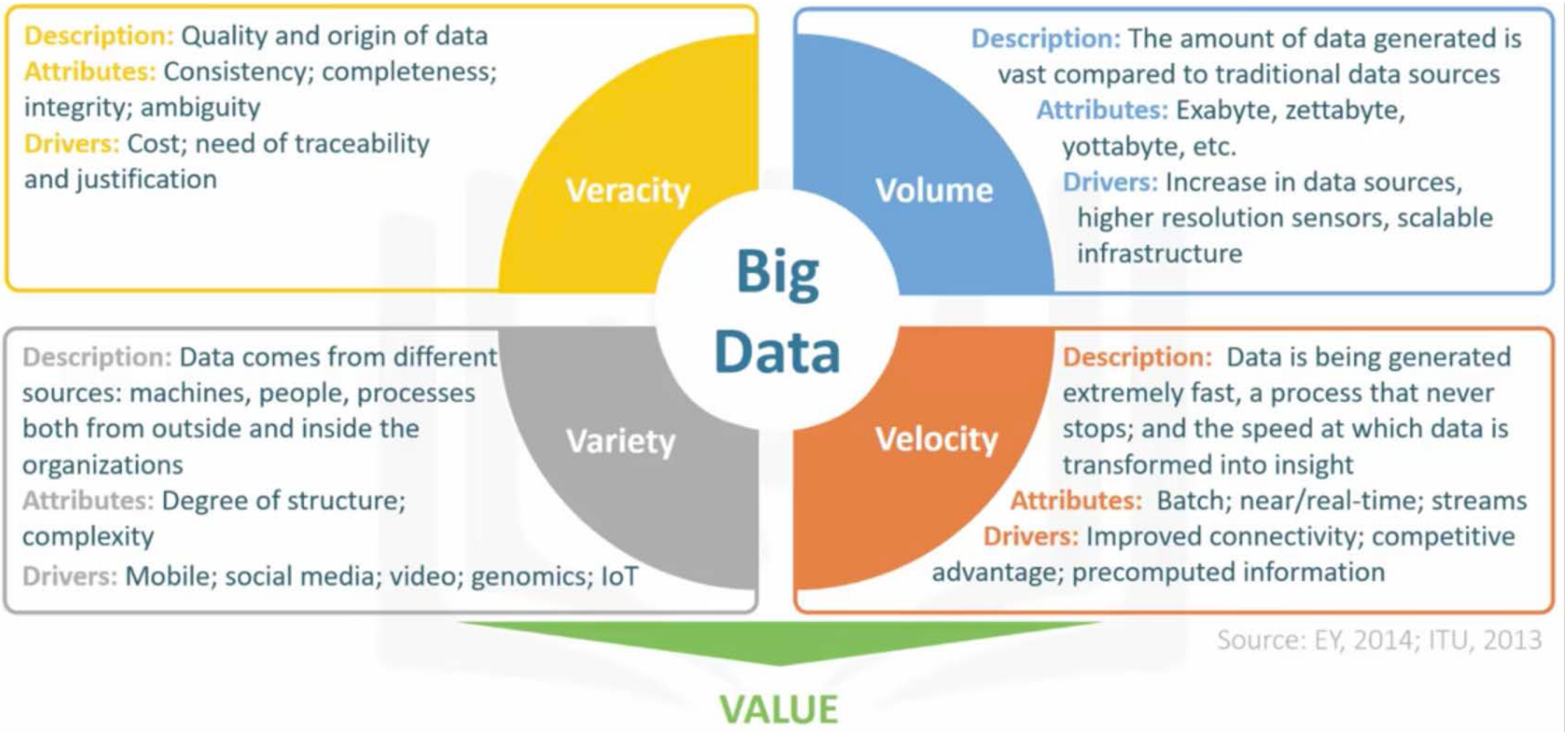
data is growing

- **Are we not generating data everyday?**
- **Are all data useless?**
- **Can we use the data as it is?**
- **Is there any universal system or framework or logic which can give us some insight?**

What is BigData?

- Size in MB/GB … ?
- Speed in Mbps or Gbps?
- Text and Video or something else?
- Data generated within organization only or outside?
- Value = Revenue generation or profits?

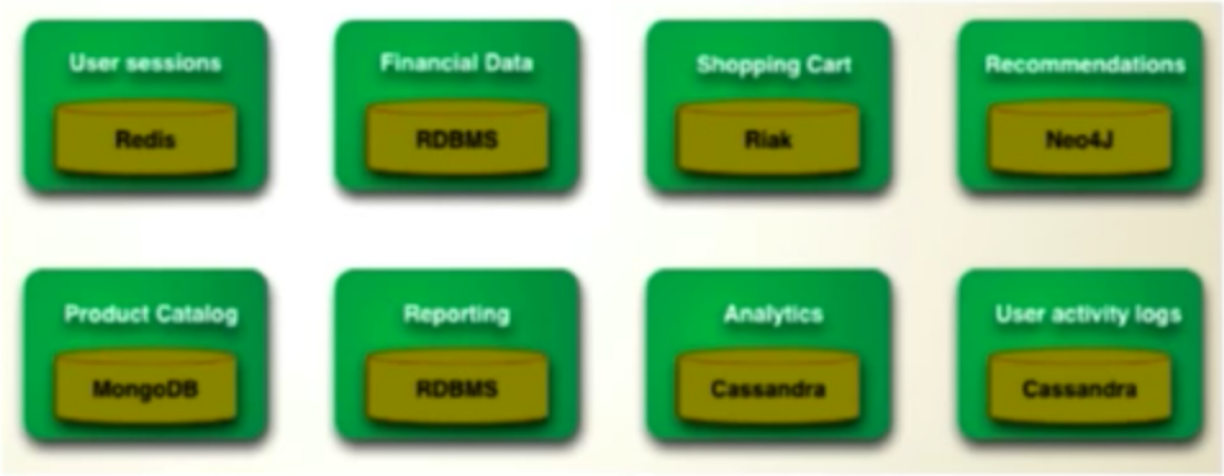
5 Vs of Big Data



Is RDBMS going away?

Relational databases are not necessarily the best for storing and managing all organizational data

Polyglot persistence: coexistence of a variety of data storage and management technologies within an organization's infrastructure



- Different types of data in different ways
- Take advantage of strengths of different databases

Source: https://neo4j.com/blog/nosql-polyglot-persistence-tools-integrations/

Relax and watch a short video ☺

https://www.youtube.com/watch?v=UQeyU0YcPKY

Hadoop

De facto standard for most Big Data storage and processing

Java-based framework for distributing and processing very large data sets across clusters of computers

Install Hadoop using the link: https://hadoop.apache.org/

Hadoop

Important components

Hadoop Distributed File System (HDFS): low-level distributed file processing system that can be used directly for data storage

MapReduce: programming model that supports processing large data sets

HDFS (Hadoop Distributed File System)

- The Hadoop Distributed File System (HDFS) is a low-level distributed file system designed to run on commodity hardware.

- It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant.

- HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware.

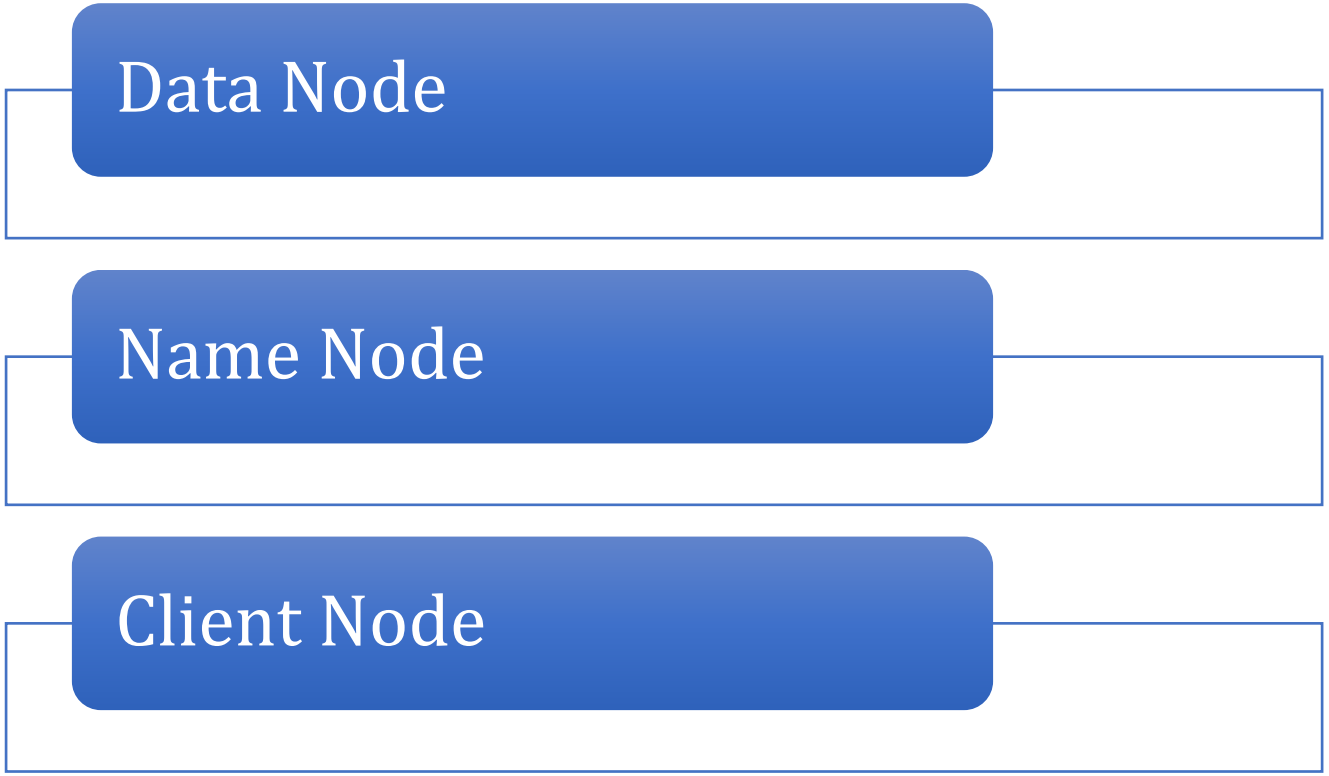
HDFS

The Hadoop Distributed File System (HDFS) approach to distributing data is based on several key assumptions:

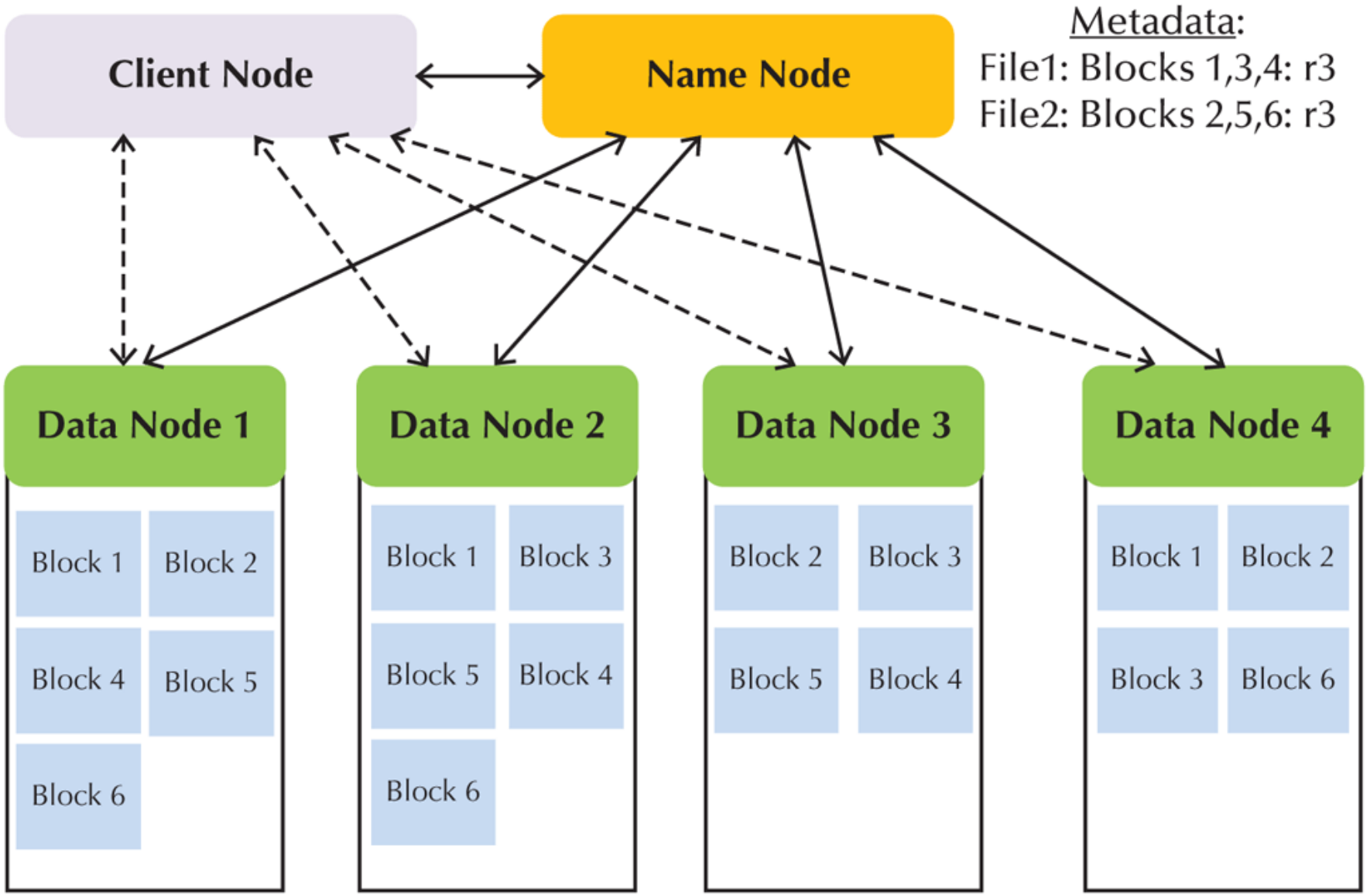
- **High volume:** default block sizes is 64 MB and can be configured to even larger values
- **Write-once, read-many:** model simplifies concurrency issues and improves data throughput
- **Streaming access:** optimized for batch processing of entire files as a continuous stream of data
- **Fault tolerance:** designed to replicate data across many different devices so that when one fails, data is still available from another device

HDFS Nodes

Hadoop uses several types of nodes; computers that perform one or more types of tasks within the system

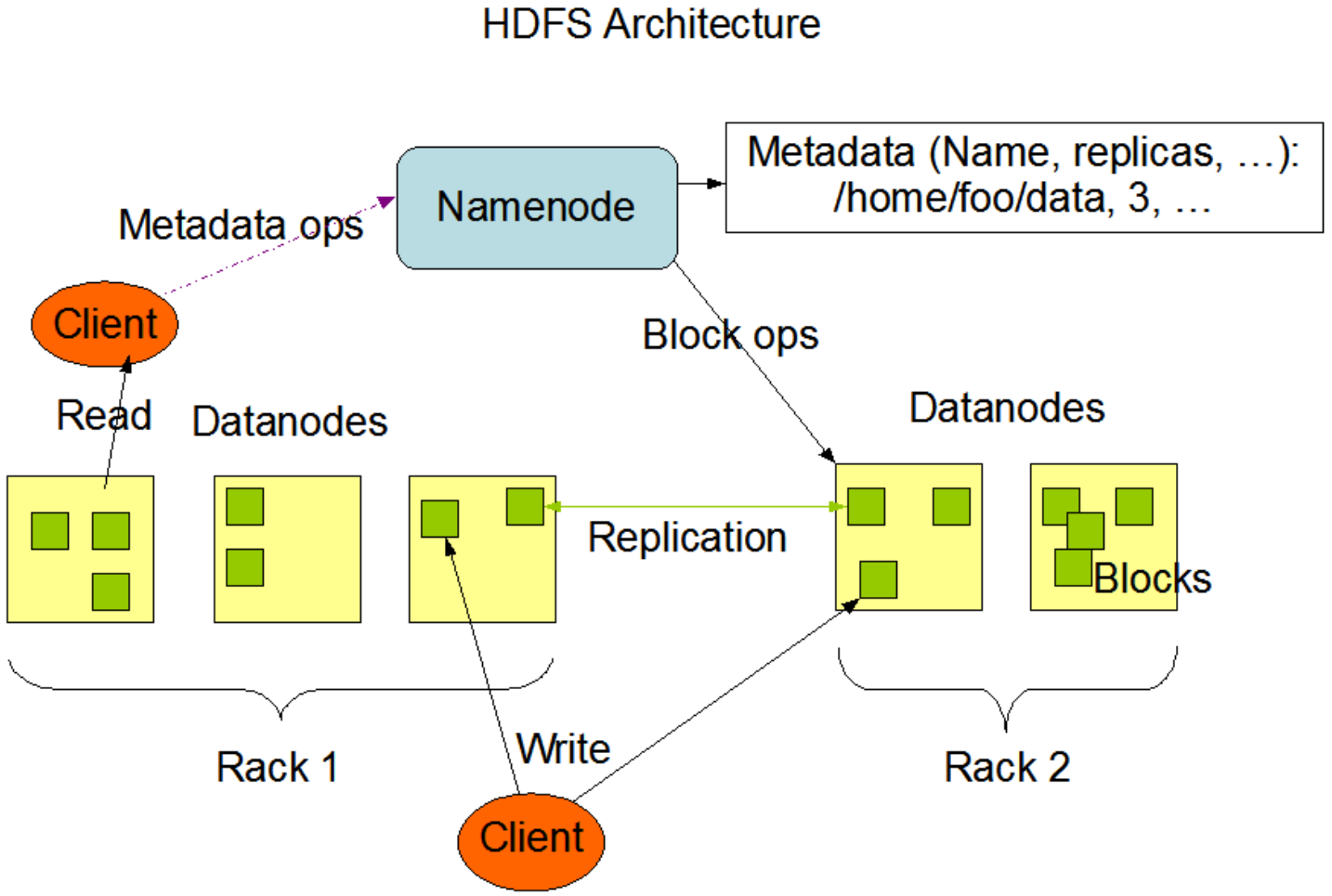


HDFS Nodes


Client Node
Name Node
Metadata:
File1: Blocks 1,3,4: r3
File2: Blocks 2,5,6: r3
Data Node 1
Block 1
Block 2
Block 4
Block 5
Block 6
Data Node 2
Block 1
Block 3
Block 5
Block 4
Block 6
Data Node 3
Block 2
Block 3
Block 5
Block 4
Data Node 4
Block 1
Block 2
Block 3
Block 6

Data Node

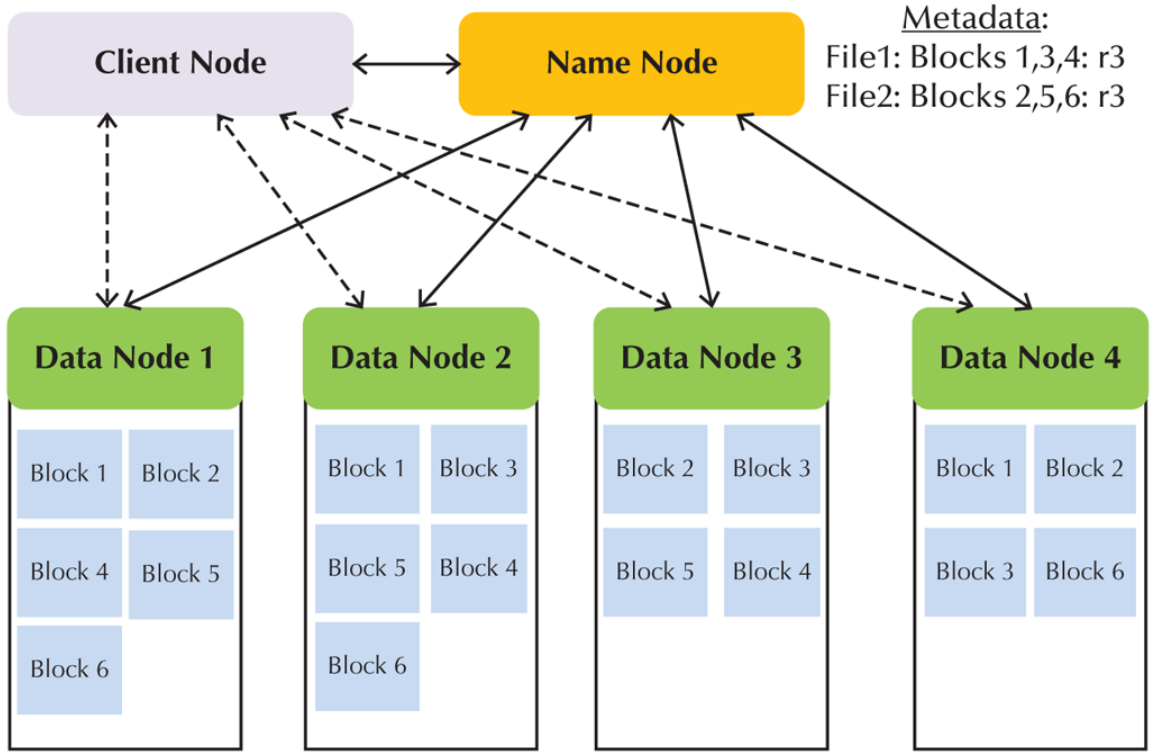
- Data nodes store the actual file data
- Perform block creation
- Performs block deletion and replication
- Data node communicates with name node and send back block reports and heartbeats



Source: http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html#NameNode_and_DataNodes

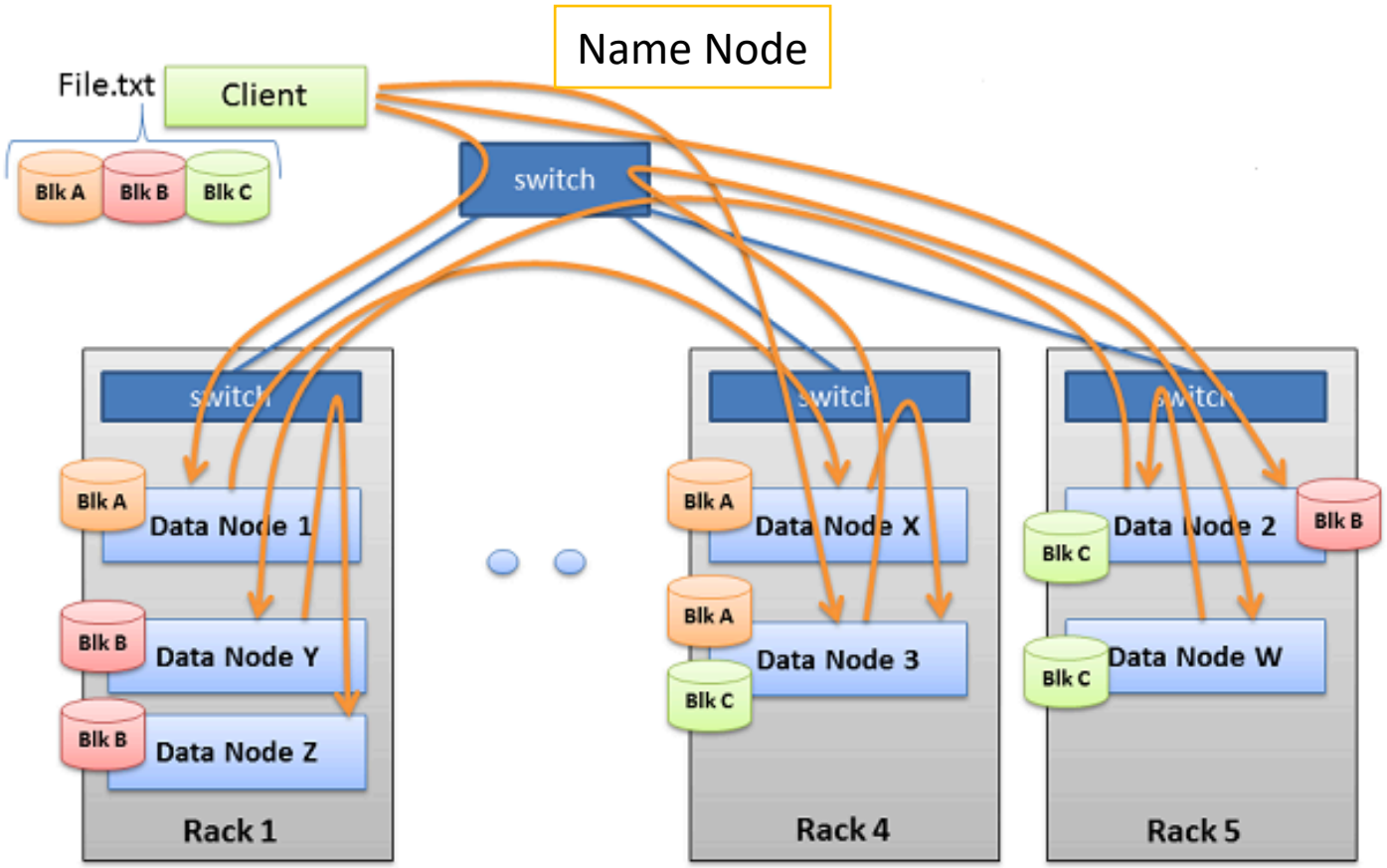
Name Node

- There is typically only one name node within a HDFS cluster
- Name node contains file system metadata
- Metadata is designed to be small, simple, and easily recoverable
- Metadata contains name of each file, the block numbers that comprise each file, and replication factor



Client Node

- Client node makes requests to the file system as needed to support user applications
- If a client node needs to read a file, it contacts Name node to request the list of blocks associated with the file and the data nodes that hold them



Source: https://hadoopabcd.wordpress.com/2015/03/17/hdfs-file-blocks-distribution-in-datanodes/

MapReduce

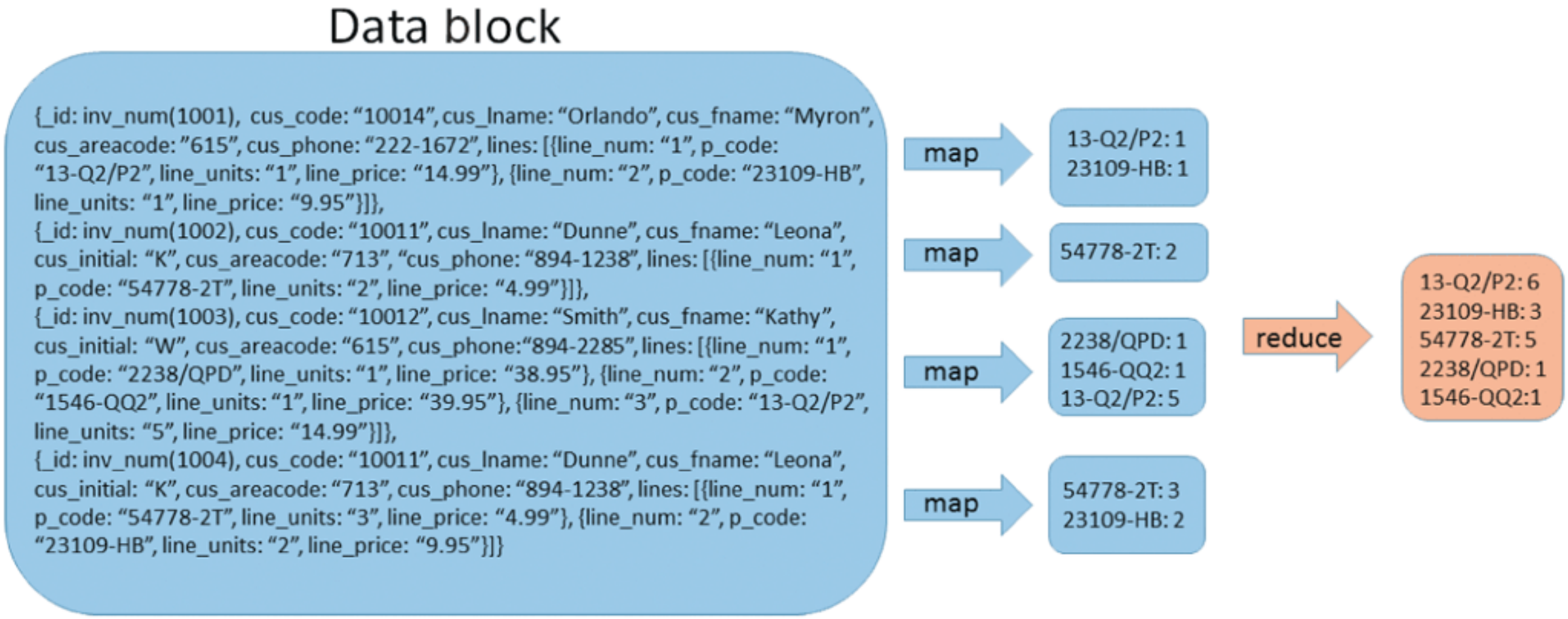
MapReduce framework is used to process large data sets across clusters

- Breaks down complex tasks into smaller subtasks, performing the subtasks, and producing a final result

- Map function takes a collection of data and sorts and filters it into a set of key-value pairs
    - Mapper program performs the map function

- Reduce summaries results of map function produce a single result
    - Reducer program performs the reduce function

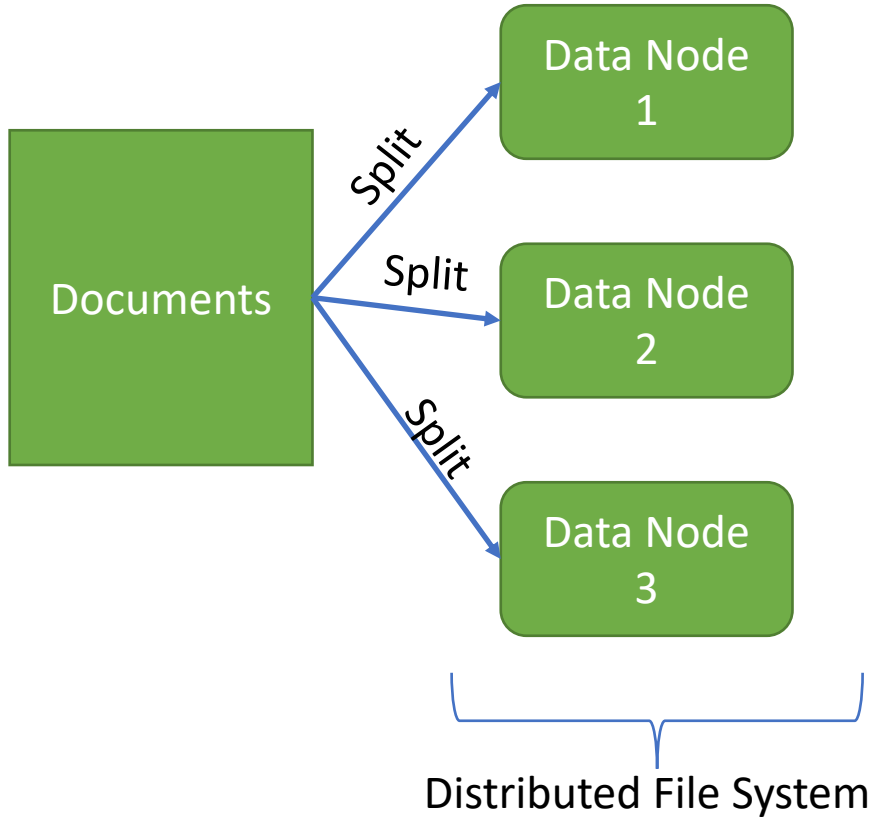
MapReduce


Data block
{_id: inv_num(1001), cus_code: "10014", cus_lname: "Orlando", cus_fname: "Myron", cus_areacode: "615", cus_phone: "222-1672", lines: [{line_num: "1", p_code: "13-Q2/P2", line_units: "1", line_price: "14.99"}, {line_num: "2", p_code: "23109-HB", line_units: "1", line_price: "9.95"}]},
{_id: inv_num(1002), cus_code: "10011", cus_lname: "Dunne", cus_fname: "Leona", cus_initial: "K", cus_areacode: "713", "cus_phone: "894-1238", lines: [{line_num: "1", p_code: "54778-2T", line_units: "2", line_price: "4.99"}]},
{_id: inv_num(1003), cus_code: "10012", cus_lname: "Smith", cus_fname: "Kathy", cus_initial: "W", cus_areacode: "615", cus_phone:"894-2285", lines: [{line_num: "1", p_code: "2238/QPD", line_units: "1", line_price: "38.95"}, {line_num: "2", p_code: "1546-QQ2", line_units: "1", line_price: "39.95"}, {line_num: "3", p_code: "13-Q2/P2", line_units: "5", line_price: "14.99"}]},
{_id: inv_num(1004), cus_code: "10011", cus_lname: "Dunne", cus_fname: "Leona", cus_initial: "K", cus_areacode: "713", cus_phone: "894-1238", lines: [{line_num: "1", p_code: "54778-2T", line_units: "3", line_price: "4.99"}, {line_num: "2", p_code: "23109-HB", line_units: "2", line_price: "9.95"}]}
map
13-Q2/P2: 1
23109-HB: 1
map
54778-2T: 2
map
2238/QPD: 1
1546-QQ2: 1
13-Q2/P2: 5
map
54778-2T: 3
23109-HB: 2
reduce
13-Q2/P2: 6
23109-HB: 3
54778-2T: 5
2238/QPD: 1
1546-QQ2:1

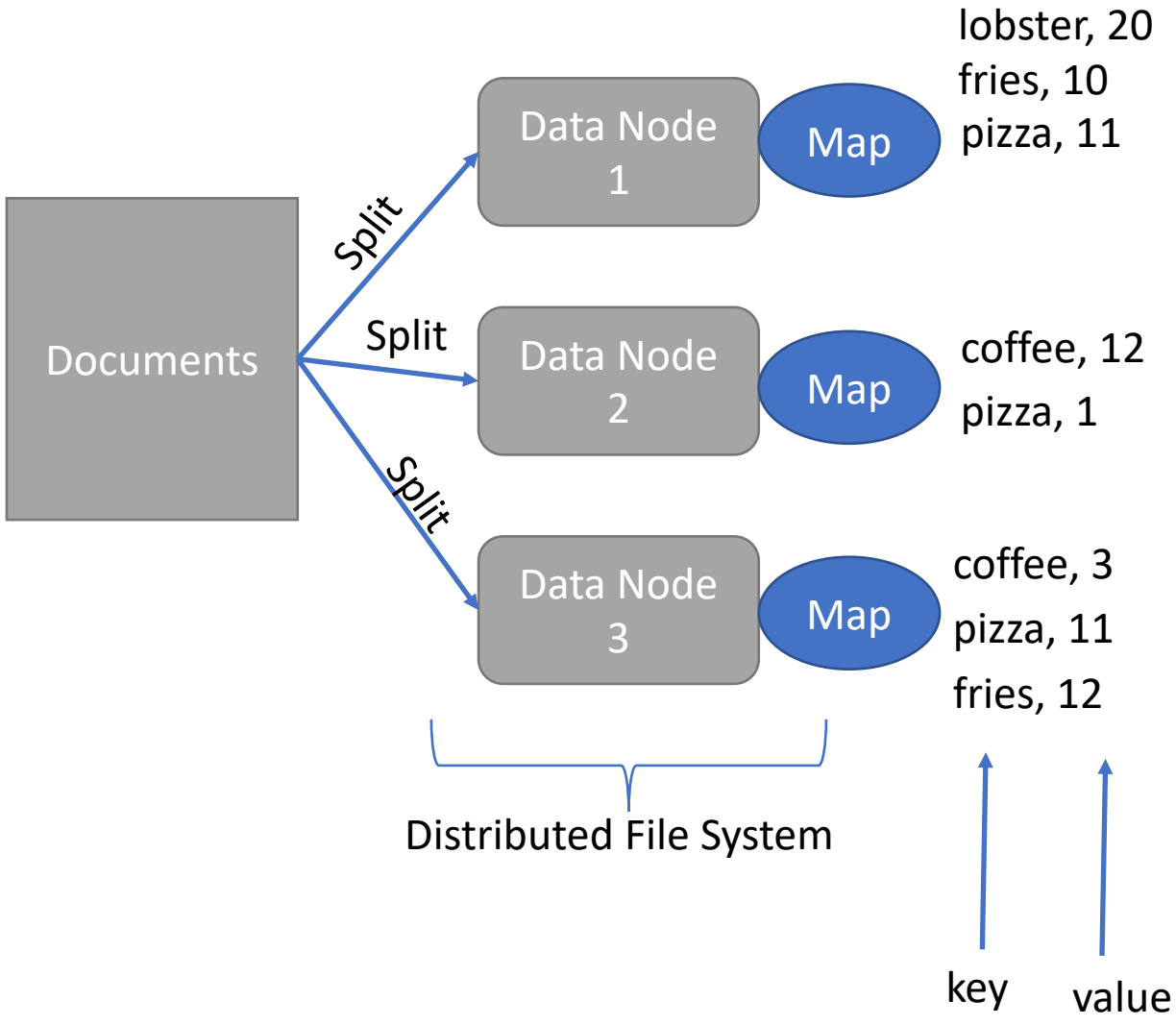
MapReduce – Example (Step 1 of 4)

Scenario: You have 30 GB documents with names of food items and restaurants, where a food item is served. You want to find, which food item is popular.

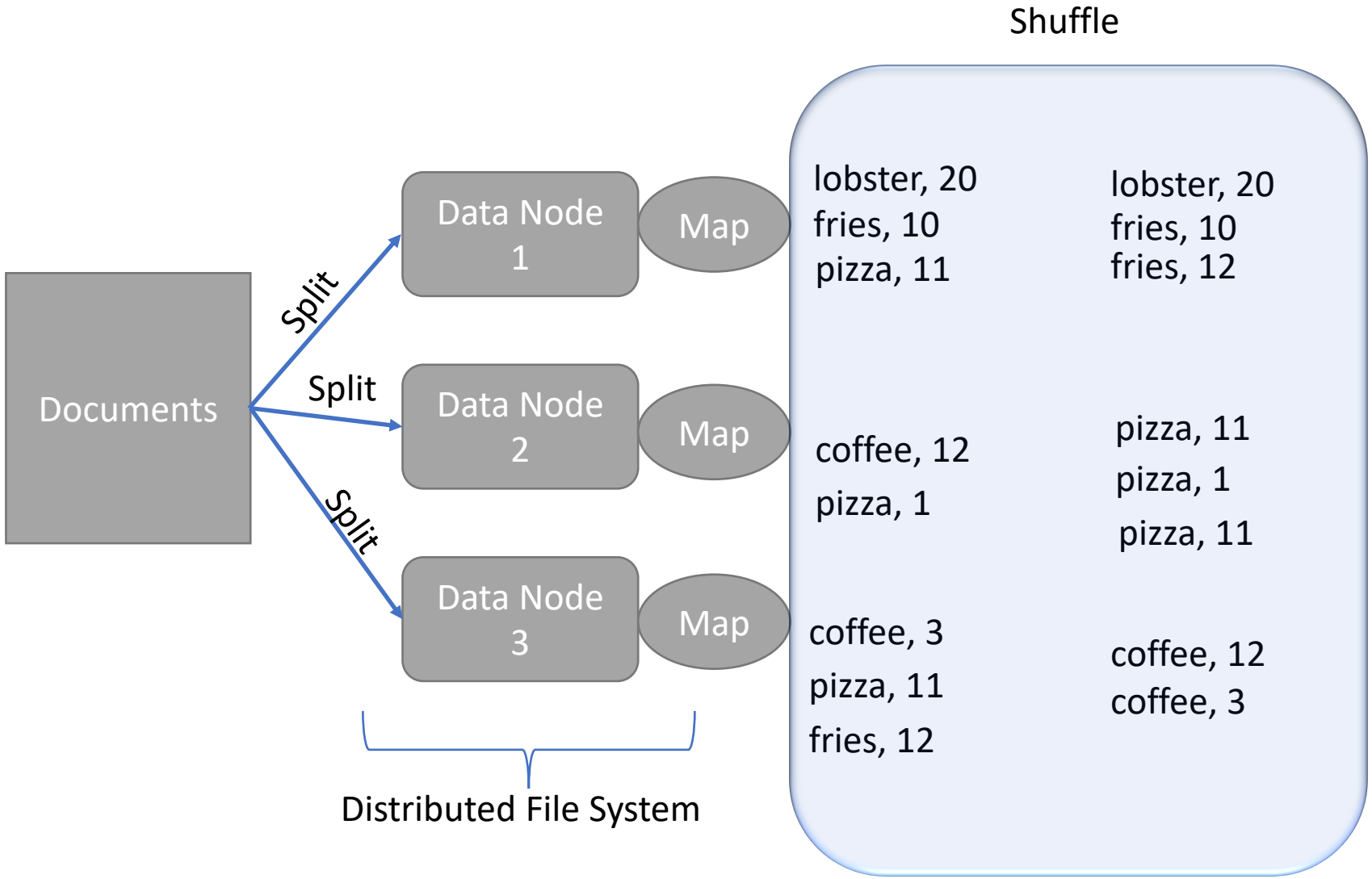
Step 1: Split and distribute the documents for processing



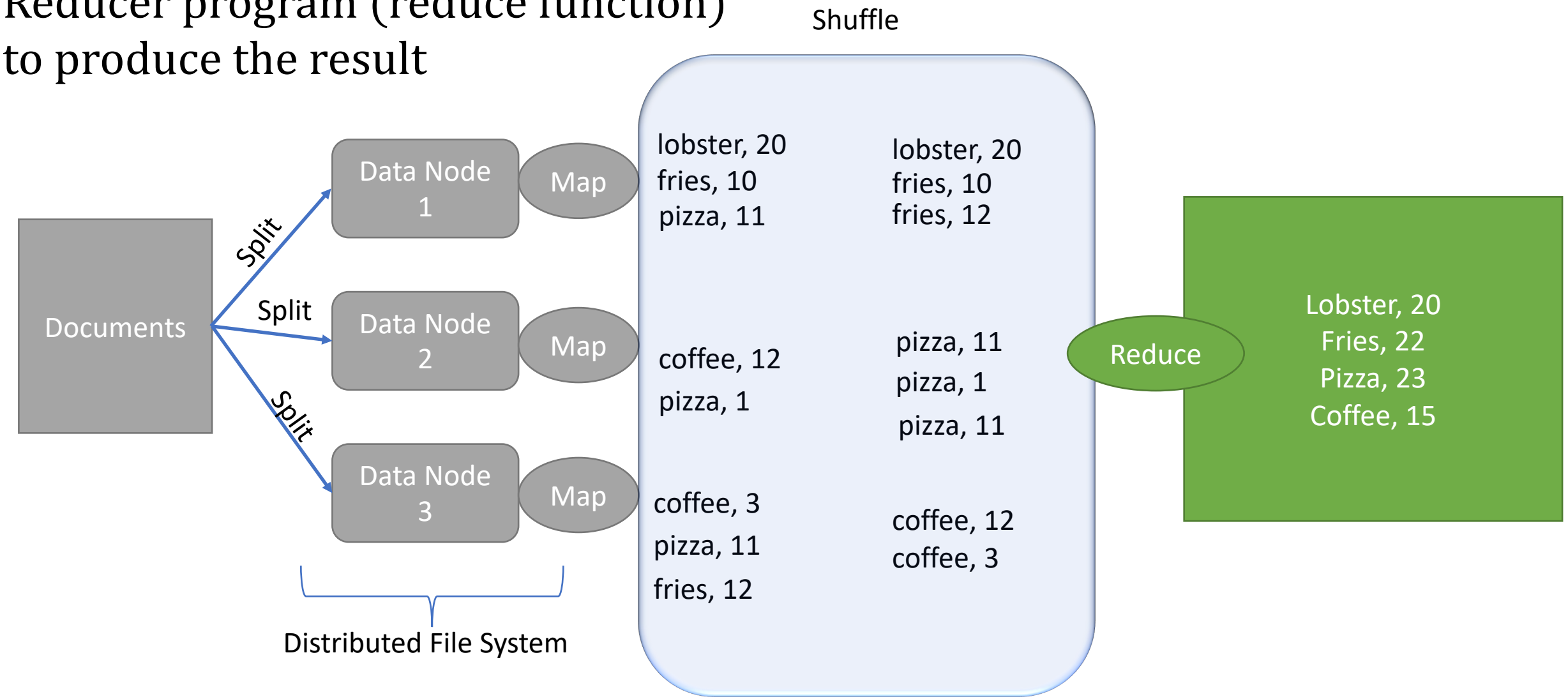
Step 2: Run a mapper program (Map function) in each of the node and obtain key, value pair of food item and number of restaurants



Step 3: Shuffle, and organize the key, value pairs.



Step 4: combine results using Reducer program (reduce function) to produce the result



MapReduce – Contd.

Implementation complements HDFS structure

- **Job tracker:** central control program
- **Task tracker:** reduces tasks on a node
- **Batch processing:** runs tasks from beginning to end with no user interaction

This happens when user submits a job

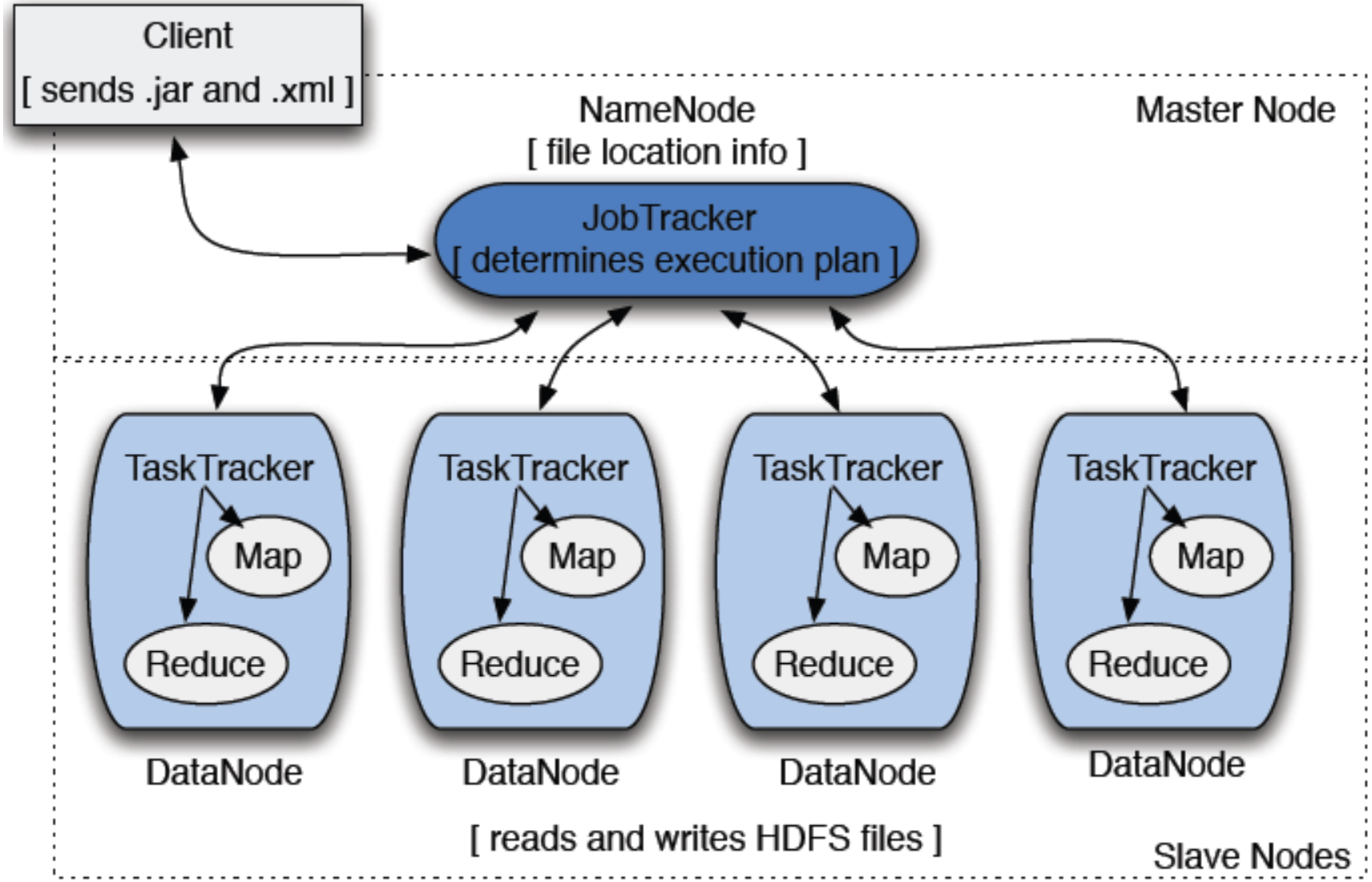


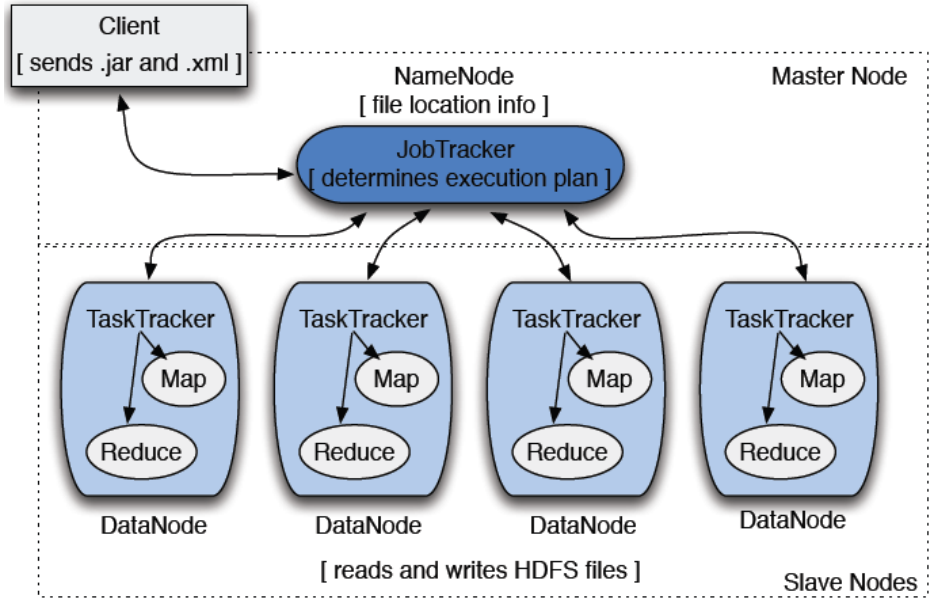
Image Source: http://jjeong.tistory.com/600

#1: A client node (client application) submits a MapReduce job to the job tracker.

#2: The job tracker communicates with the name node to determine which data nodes contain the blocks that should be processed for this job.

#3: The job tracker determines which task trackers are available for work. Each task tracker can handle a set number of tasks. Remember, many MapReduce jobs from different users can be running on the Hadoop system simultaneously, so a data node may contain data that is being processed by multiple mappers from different jobs all at the same time. Therefore, the task tracker on that node might be busy running mappers for other jobs when this new request arrives. Because the data is replicated on multiple nodes, the job tracker may be able to select from multiple nodes for the same data.

#4: The job tracker then contacts the task trackers on each of those nodes to begin mappers and reducers to complete that node's portion of the task.



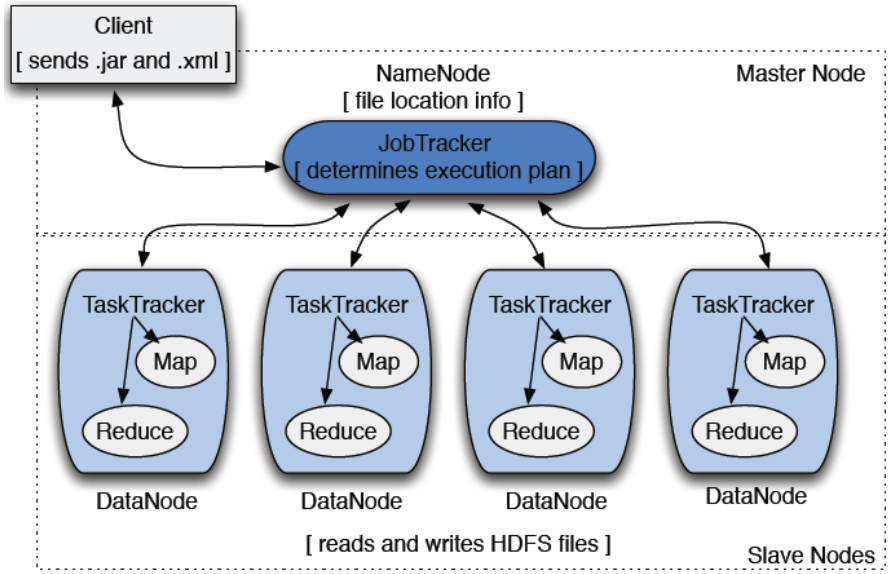
#5: The task tracker creates a new JVM (Java virtual machine) to run the map and reduce functions. This way, if a function fails or crashes, the entire task tracker is not halted.

#6: The task tracker sends heartbeat messages to the job tracker to let the job tracker know that the task tracker is still working on the job (and about the nodes availability for more jobs).

#7: The job tracker monitors the heart beat messages to determine if a task manager has failed. If so, the job tracker can reassign that portion of the task to another node.

#8: When the entire job is finished, the job tracker changes status to indicate that the job is completed.

#9: The client node periodically queries the job tracker until the job status is completed.



End of Lecture Questions

1. Is there any alternative of Hadoop? Why is Hadoop popular?

2. Which technology or tool is popular in Big Data analysis?