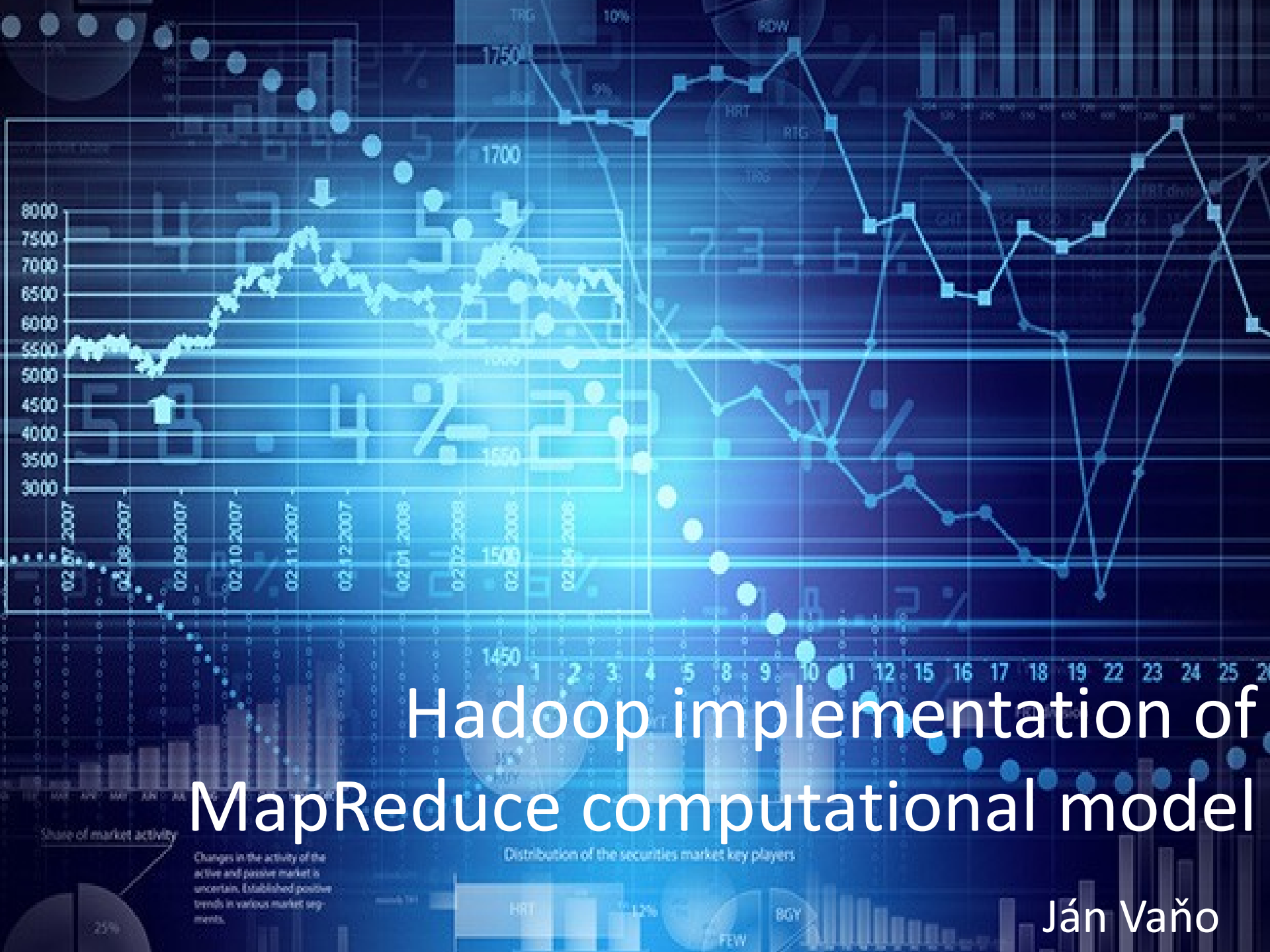

ALSO
BIG
DATA
TOOLS
DISK
TARGET
EXAMPLES
APPLIED
SENSOR
SHARED
MAY
DEFINITION
CURRENT
MOVING
RECONSIDER
WITHIN
ZETTABYTES
PRACTITIONERS
CAPTURE
THOUGHT
INTERNET
BUSINESS
SETS
GENOMICS
DISTRIBUTED
DESCRIBING
RADIO-FREQUENCY
COMPLEXITY
MANAGEMENT
TERABYTES
CASE
ABILITY
PETABYTES
INCLUDE
FC
ELAPSED
SYSTEMS
TOLERABLE
FORMS
SIZE
STORAGE
MPP
PARALLEL
SAN
QUALITIES
SINCE
MASSIVELY
GROW
SEARCH
DIFFICULTY
CONNECTOMICS
ORGANIZATIONS
SET
USE
RELATIONAL
LARGER
CONTINUES
SOCIAL
INDEXING
TENS
COMPLEX
CITATION
ANALYTICS
NOW
BIOLOGICAL
PROCESSING
UBIQUITOUS
CAPACITY
BURIED
WORLD'S
HUNDREDS
DESKTOP
RECORDS
NETWORKS
DATABASES
SOLID
TYPES
CURRENTLY
GARTNER
OPPORTUNITIES
WORKING
AMOUNT
EVERY



# Hadoop implementation of MapReduce computational model

Ján Vaňo

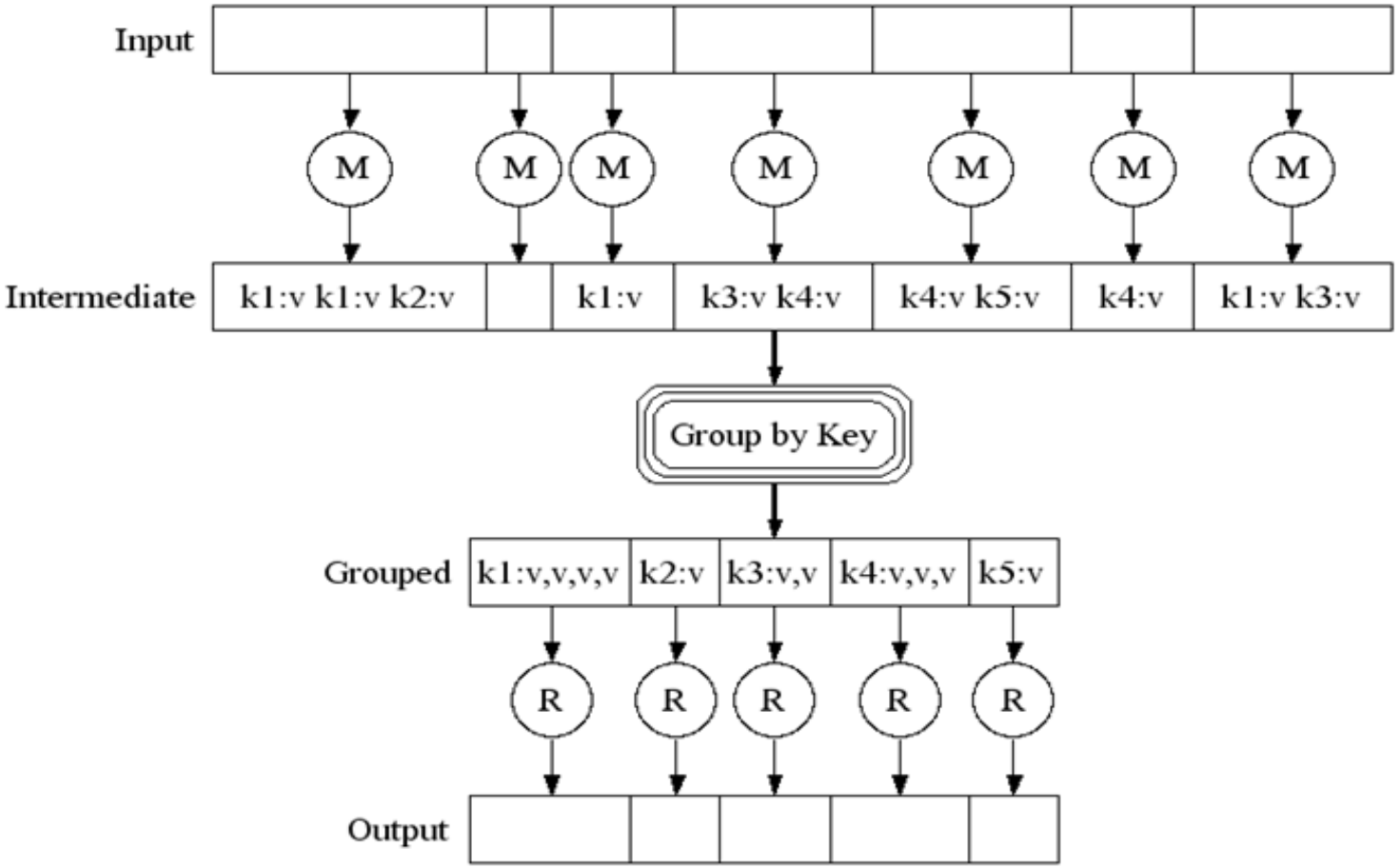
# What is MapReduce?

- A computational model published in a paper by Google in 2004
- Based on distributed computation
- Complements Google‘s distributed file system (GFS)
- Works with key:value pairs

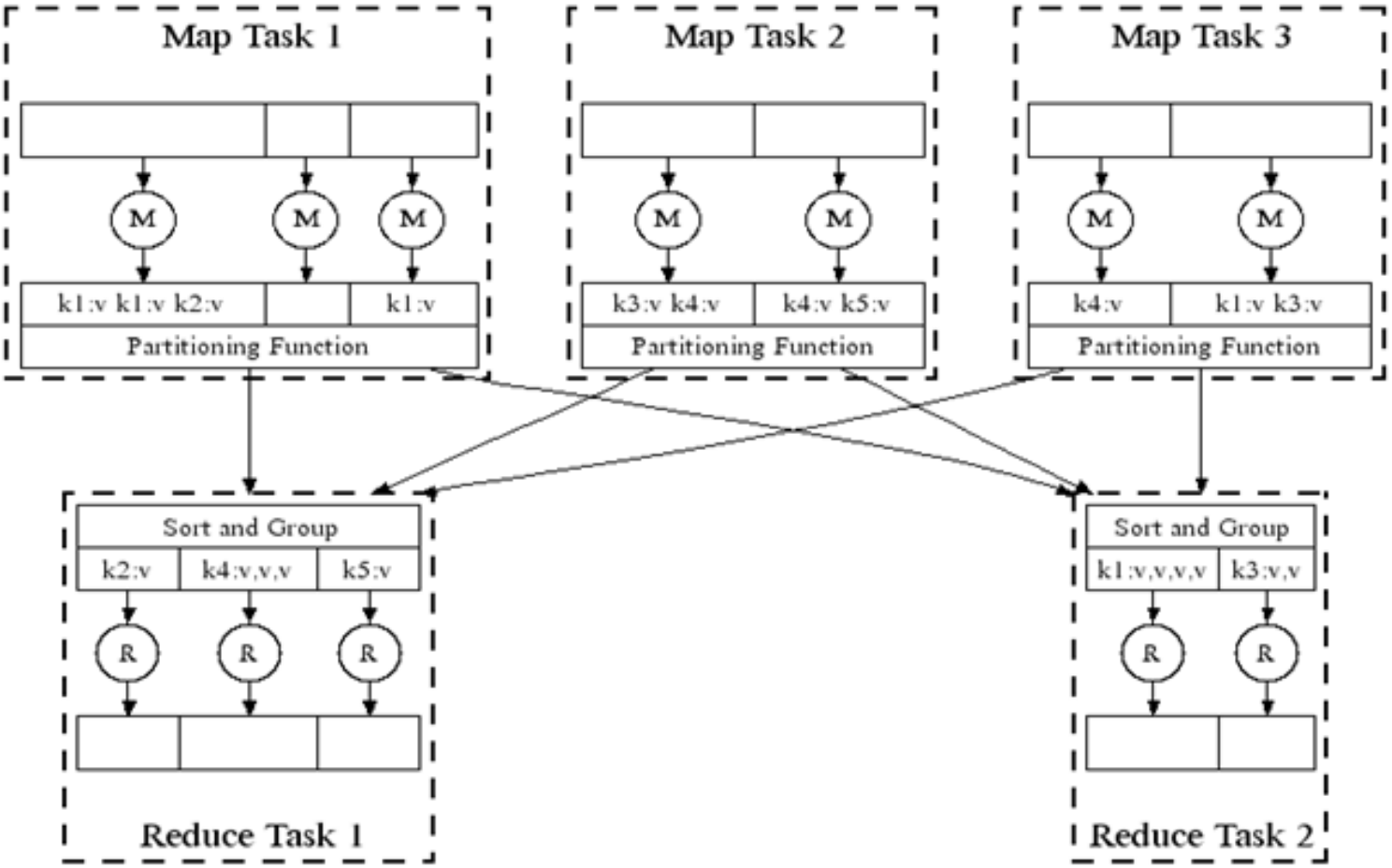
# Why MapReduce?

- It is the ‚answer‘ for Big Data problem
- Runs on commodity hardware
- Very scalable solution

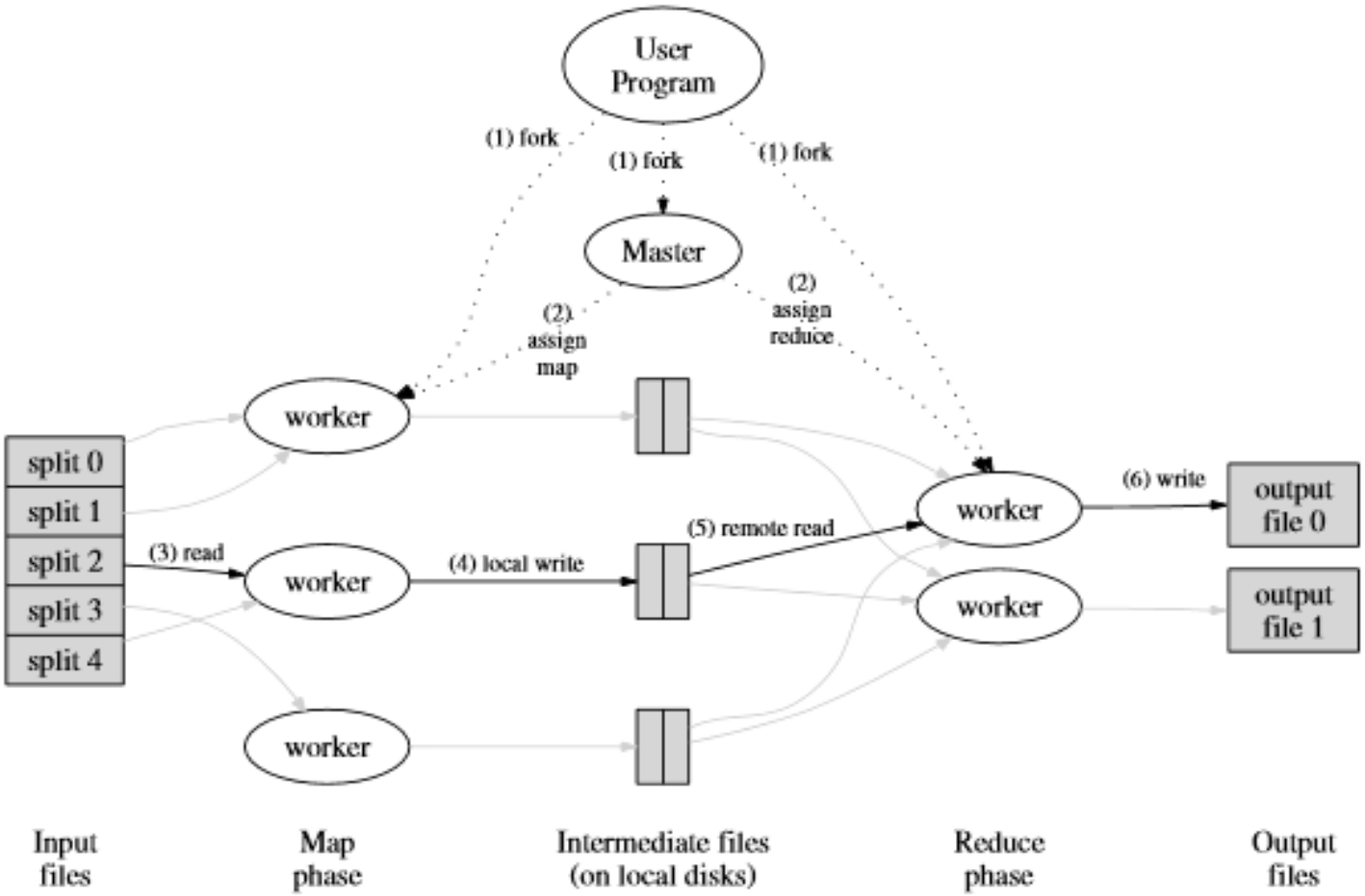
# MapReduce model



# MapReduce model



# MapReduce model



# MapReduce alternatives

- Hadoop – Top-level Apache project
- Spark – University of Berkeley project
- Disco – Open source Nokia project
- MapReduce-MPI – US Department of Energy project
- MARIANE – academic project of University of Binghamton
- Phoenix – University of Stanford project
- BashReduce - MapReduce for std. Unix commands

# MapReduce vs. RDBMS

| | Traditional RDBMS | MapReduce |
|---|---|---|
| Data Size | Gigabytes | Petabytes |
| Access | Interactive and batch | Batch |
| Updates | Read and write many times | Write once, read many times |
| Structure | Static schema | Dynamic schema |
| Integrity | High | Low |
| Scaling | Nonlinear | Linear |

# Data Structure

- Structured Data – data organized into entities that have a defined format.
  - Realm of RDBMS
- Semi-Structured Data – there may be a schema, but often ignored; schema is used as a guide to the structure of the data.
- Unstructured Data – doesn’t have any particular internal structure.
- MapReduce works well with semi-structured and unstructured data.

# What is Hadoop?

- Software platform that lets one easily write and run applications that process vast amounts of data
- Hadoop is most popular implementation of MapReduce so far

# Why Hadoop?

- It has been Apache top-level project for a long time (6 years)
- Hadoop Ecosystem
- Hadoop exclusive technologies

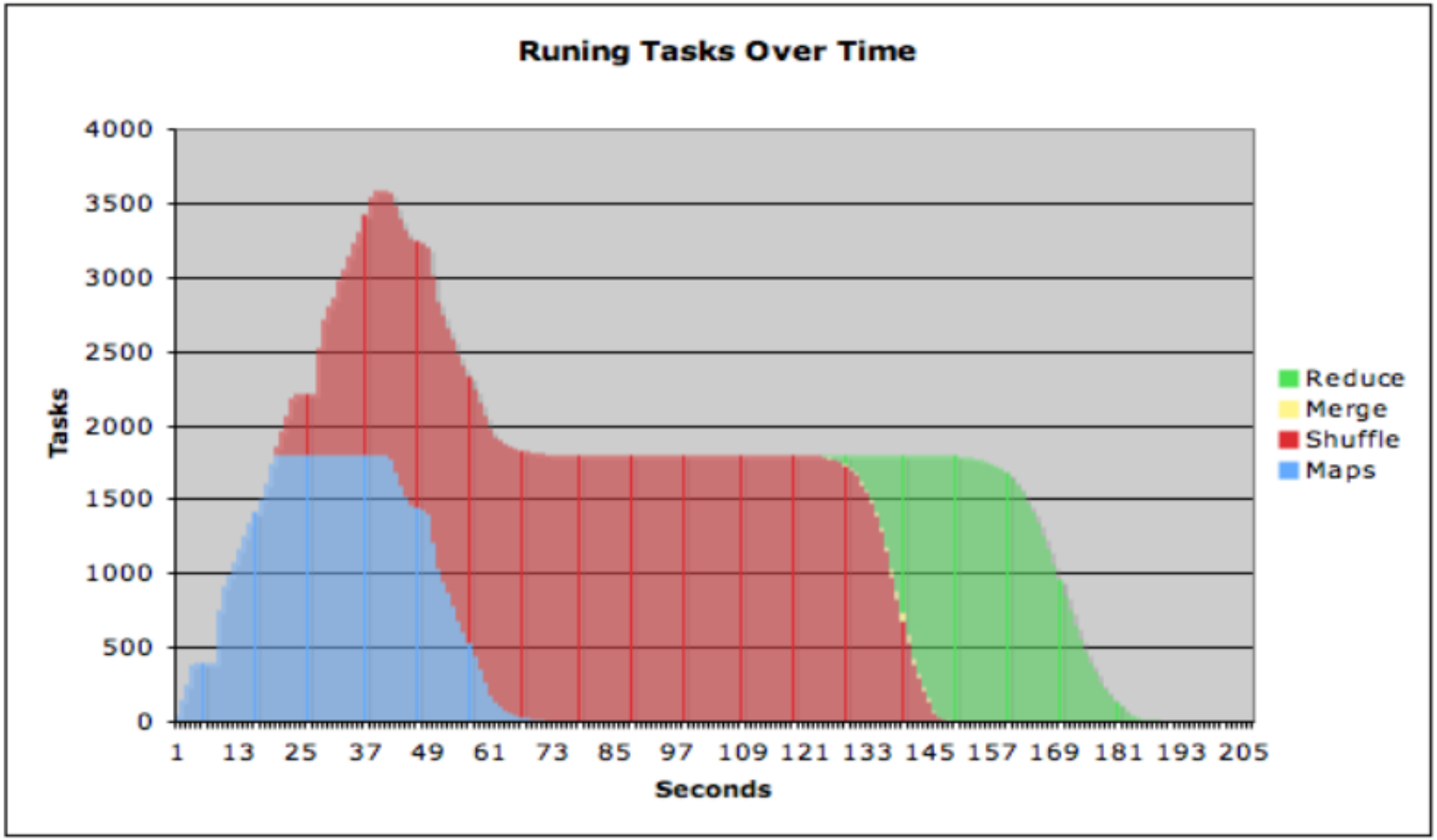
# Why Hadoop?

- Scalable: It can reliably store and process petabytes
- Economical: It distributes the data and processing across clusters of commonly available computers (in thousands)
- Efficient: By distributing the data, it can process it in parallel on the nodes where the data is located
- Reliable: It automatically maintains multiple copies of data and automatically redeploys computing tasks based on failures

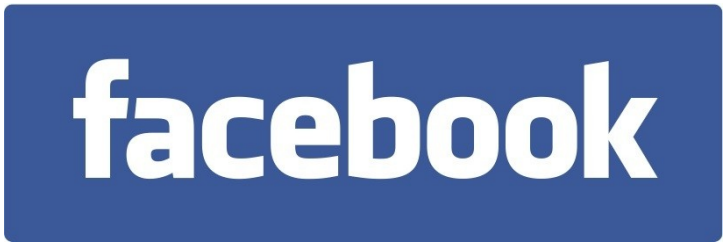
# Brief history

- 2002 - Project Nutch started (open source web search engine) – **Doug Cutting**
- 2003 - GFS (Google File System) paper published
- 2004 - Implementation of GFS started
- 2004 - Google published MapReduce paper
- 2005 - Working implementations of MapReduce and GFS (NDFS)
- 2006 - System applicable beyond realm of search
- 2006 - Nutch moved to Hadoop project, Doug Cutting joins Yahoo!
- 2008 - Yahoo!s production index generated by 10,000 core Hadoop cluster
- 2008 - Hadoop moved under Apache Foundation
- April 2008 - Hadoop broke world record - fastest sorting of 1 TB of data (209 seconds, previously 297)
- November 2008 - Google's implementation sorted 1 TB in 68 seconds
- May 2009 - Yahoo! team sort 1 TB in 62 seconds

# 1TB sort by Hadoop



# Who uses Hadoop?













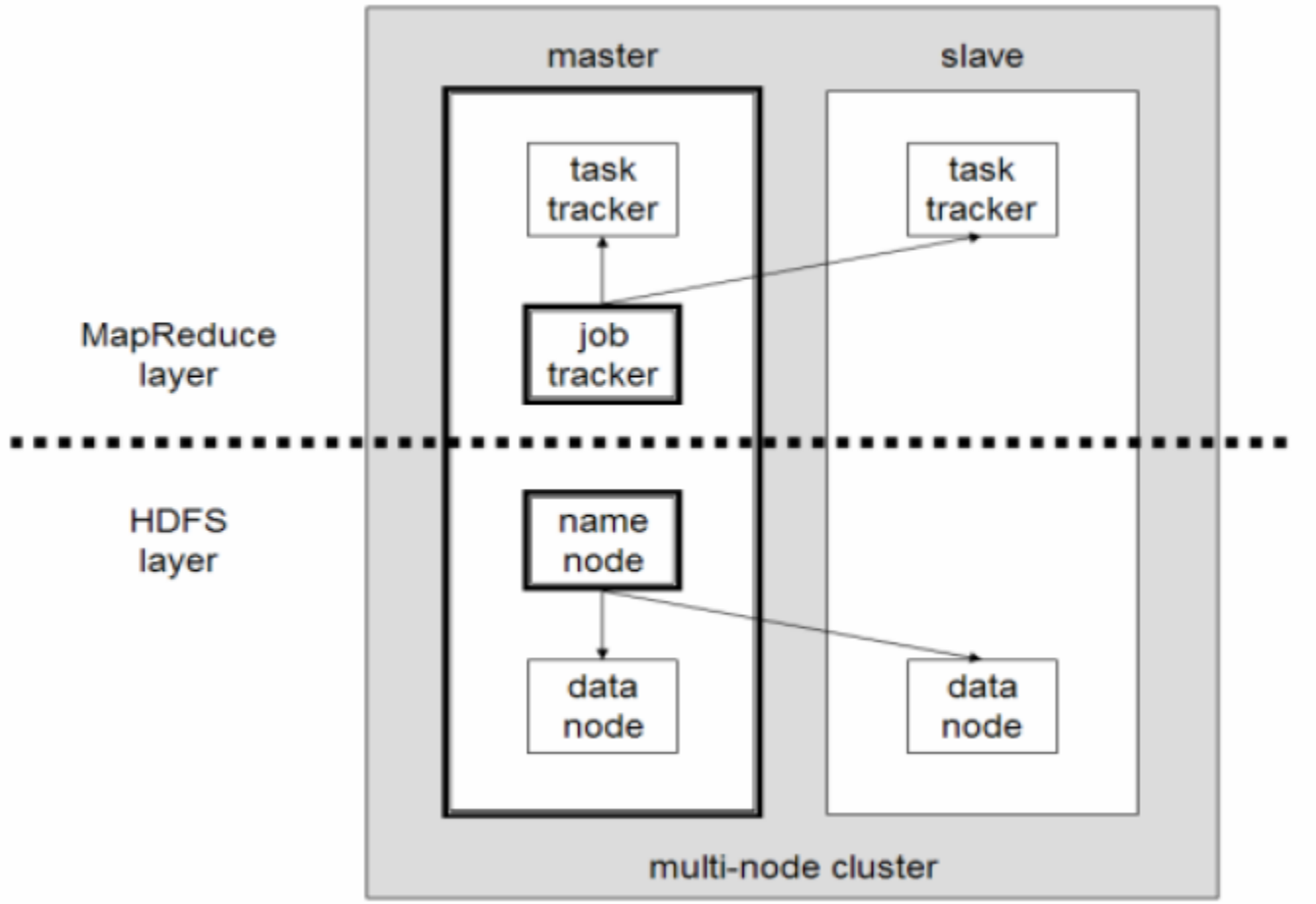
# Assumptions

- Hardware will fail
- Processing will be run in batches. Thus there is an emphasis on high throughput as opposed to low latency
- Applications that run on HDFS have large data sets. A typical file in HDFS is gigabytes to terabytes in size
- It should provide high aggregate data bandwidth and scale to hundreds of nodes in a single cluster. It should support tens of millions of files in a single instance
- Applications need a write-once-read-many access model
- Moving Computation is Cheaper than Moving Data
- Portability is important

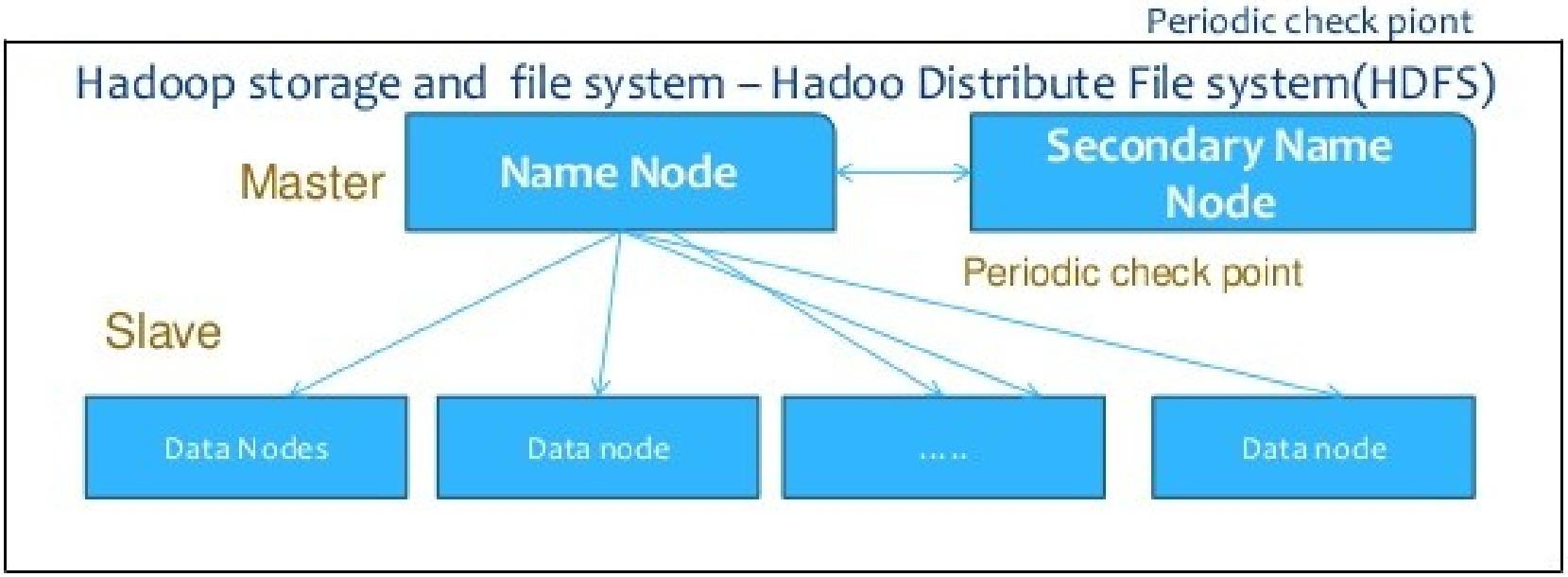
# Hadoop modules

- Hadoop Common - contains libraries and utilities needed by other Hadoop modules
- Hadoop Distributed File System (HDFS) - a distributed file-system that stores data on the commodity machines, providing very high aggregate bandwidth across the cluster
- Hadoop MapReduce - a programming model for large scale data processing
- Hadoop YARN - a resource-management platform responsible for managing compute resources in clusters and using them for scheduling of users' applications
  - Provides base for MapReduce v2

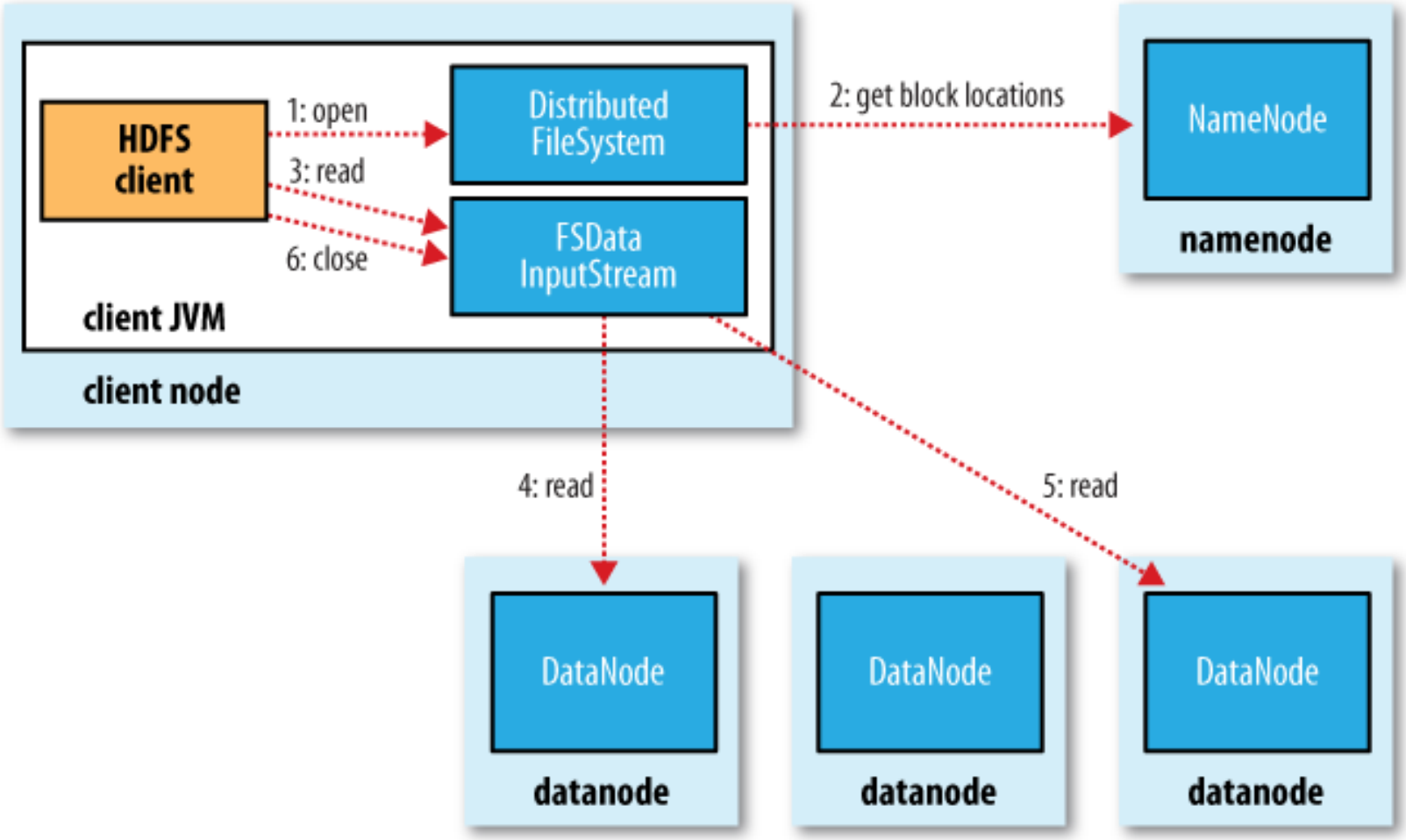
# Hadoop architecture



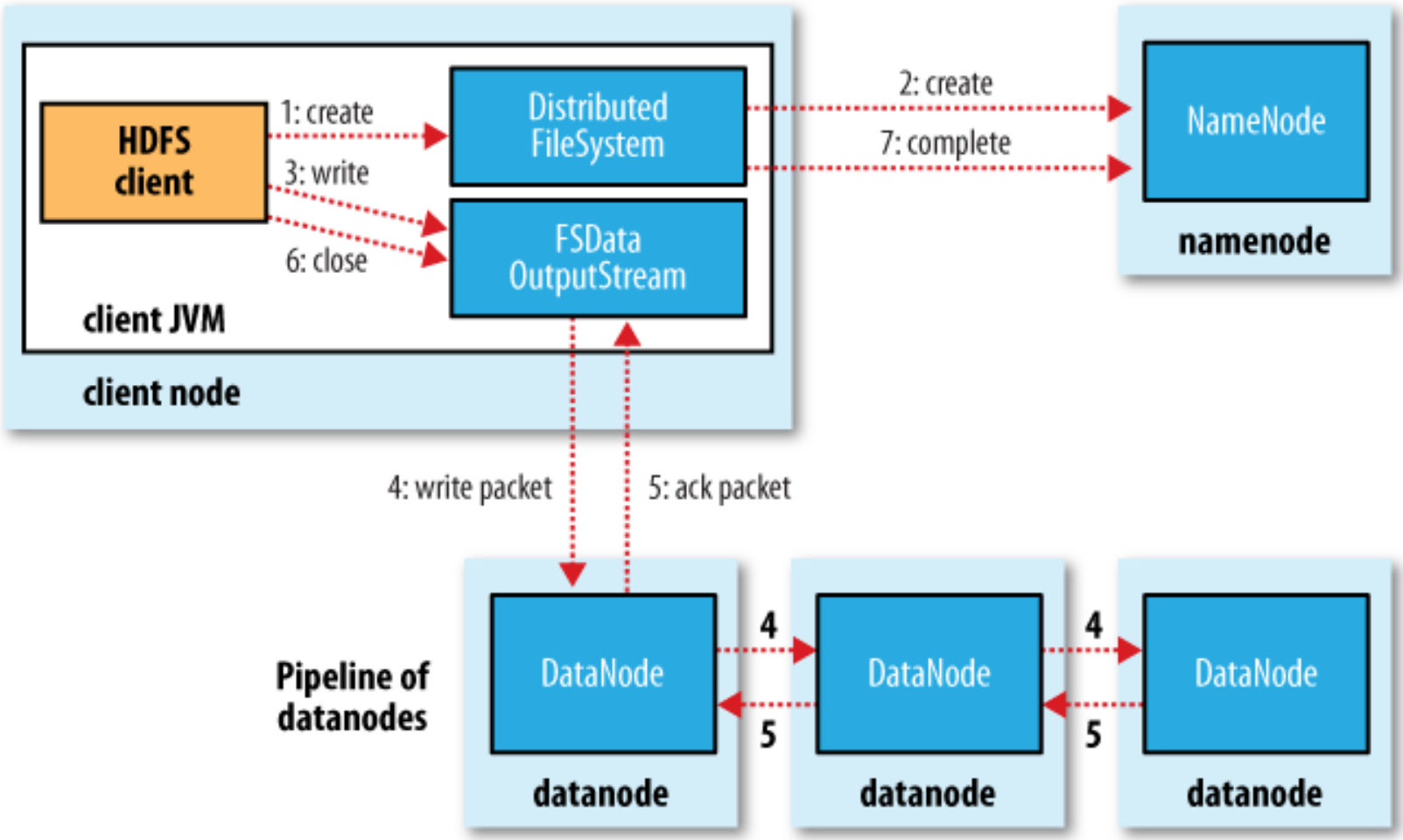
# HDFS architecture



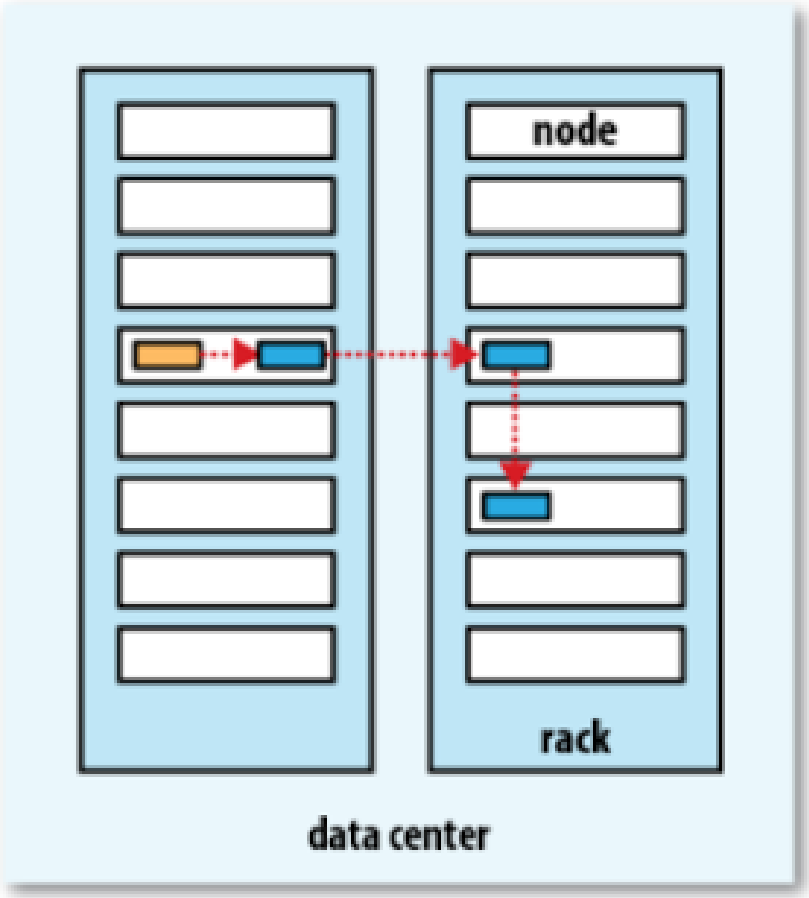
# HDFS architecture (reading)



# HDFS architecture (writing)



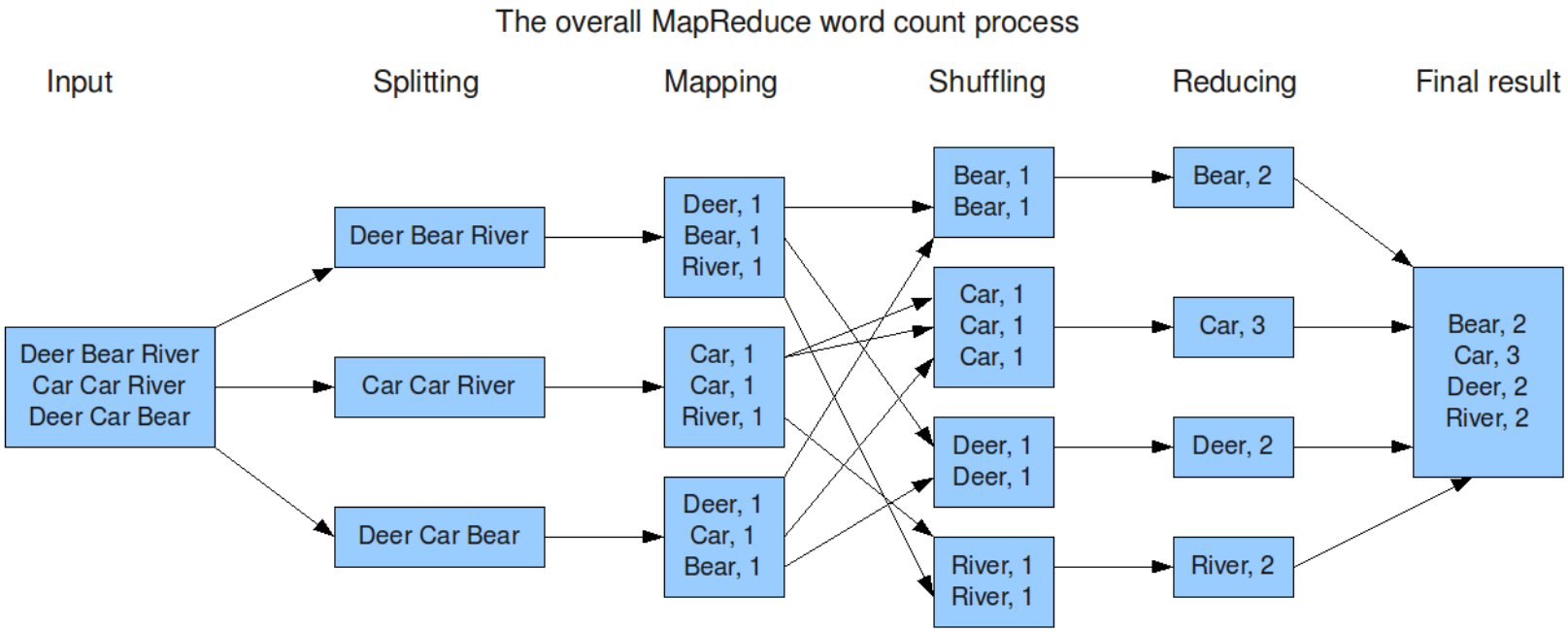
# Data replication in HDFS



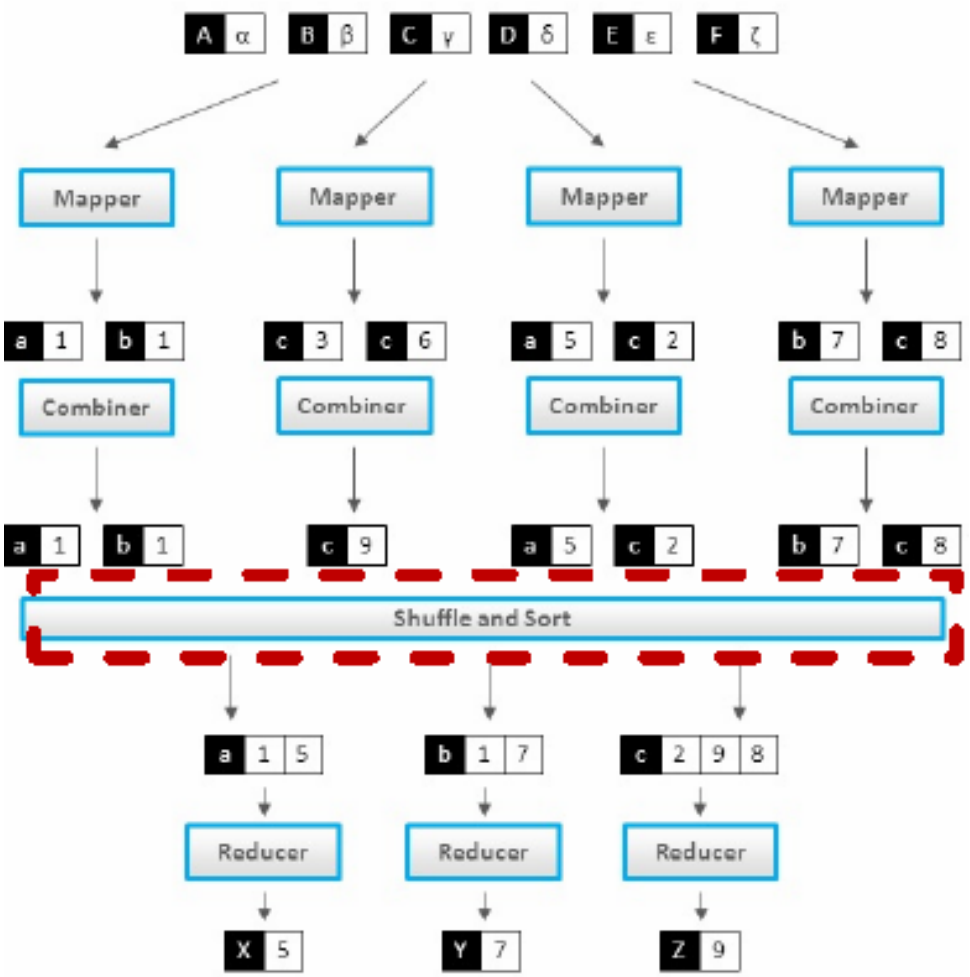
# How to use Hadoop MapReduce?

- Implement 2 basic functions:
  - Map
  - Reduce
- Implement Driver class

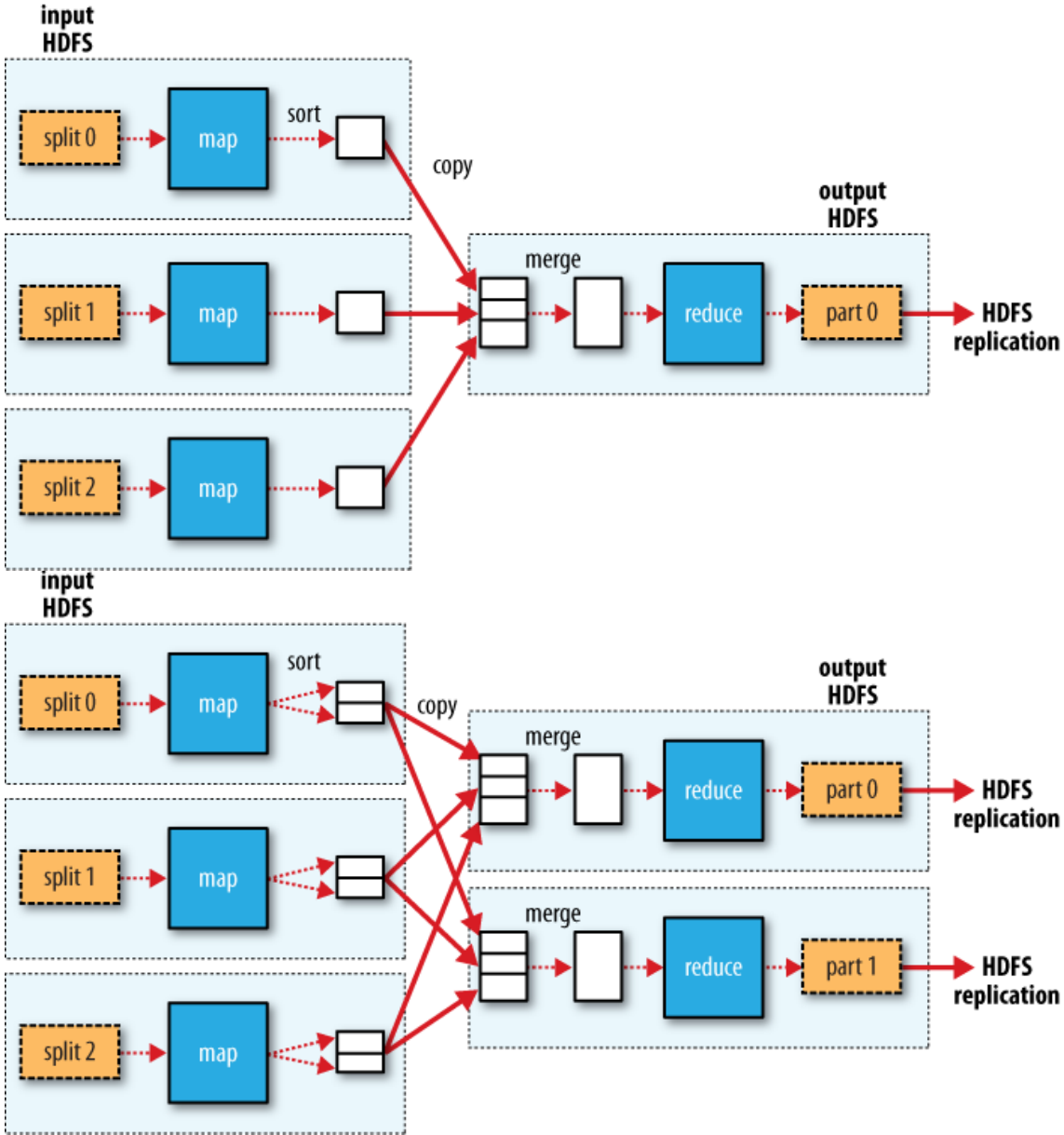
# MapReduce structure



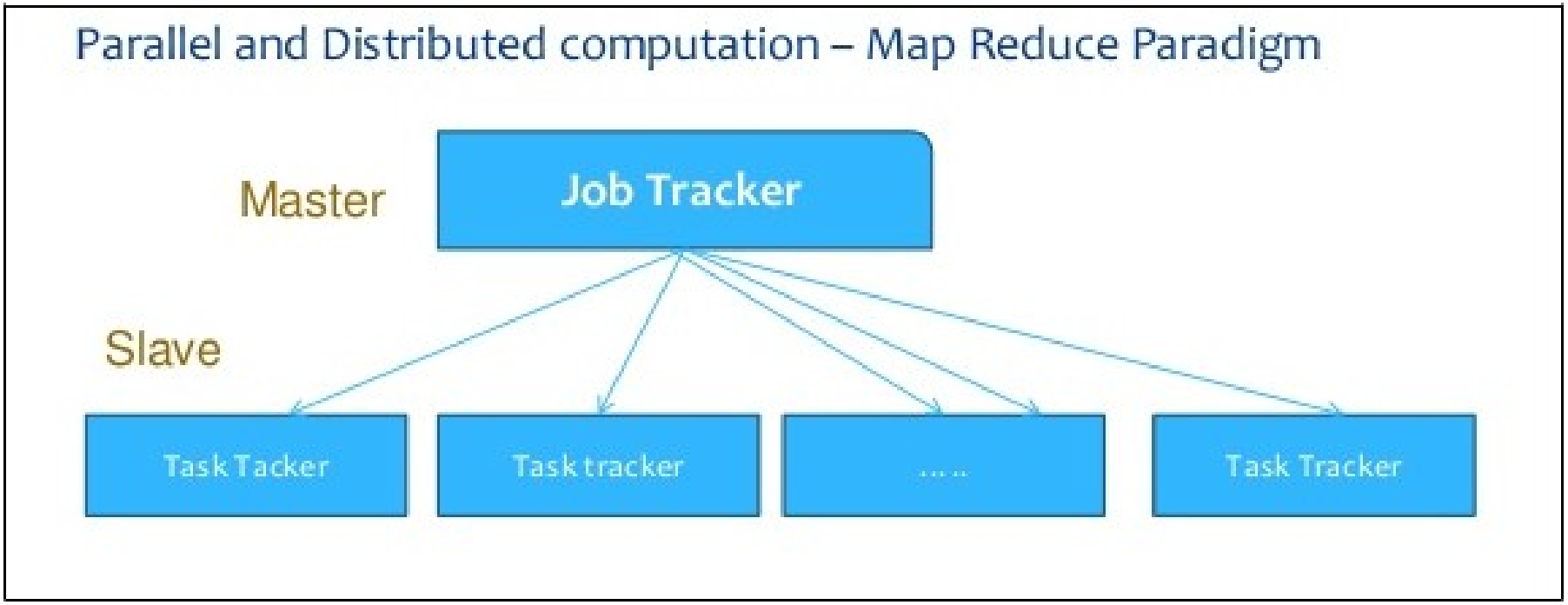
# MapReduce structure



# MapReduce structure



# Job submission (MapReduce v1)



# Job submission (MapReduce v1)

- Client applications submit jobs to the Job tracker
- The JobTracker talks to the NameNode to determine the location of the data
- The JobTracker locates TaskTracker nodes with available slots at or near the data
- The JobTracker submits the work to the chosen TaskTracker nodes
- The TaskTracker nodes are monitored. If they do not submit heartbeat signals often enough, they are deemed to have failed and the work is scheduled on a different TaskTracker

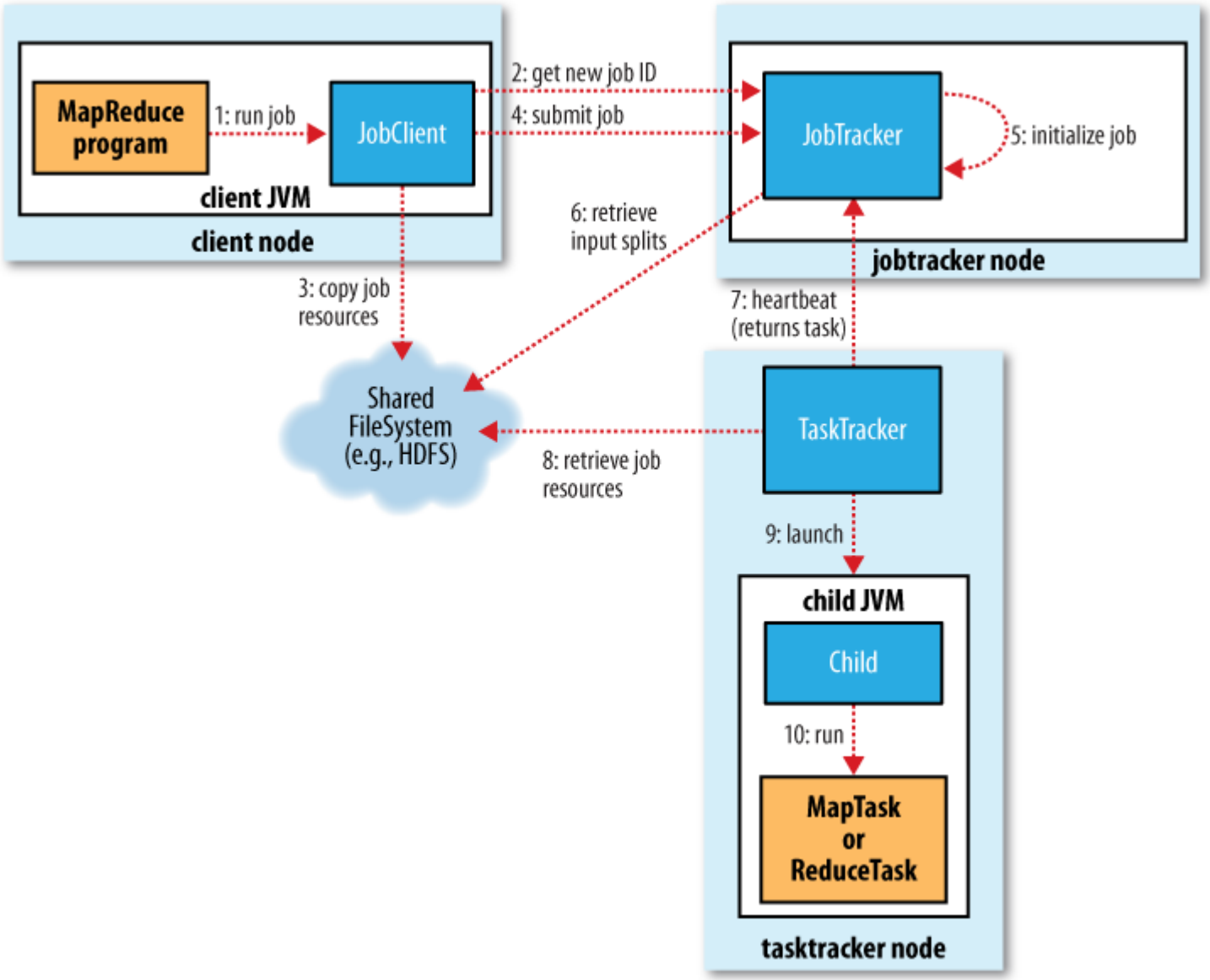
# Job submission (MapReduce v1)

- A TaskTracker will notify the JobTracker when a task fails. The JobTracker decides what to do then: it may resubmit the job elsewhere, it may mark that specific record as something to avoid, and it may may even blacklist the TaskTracker as unreliable

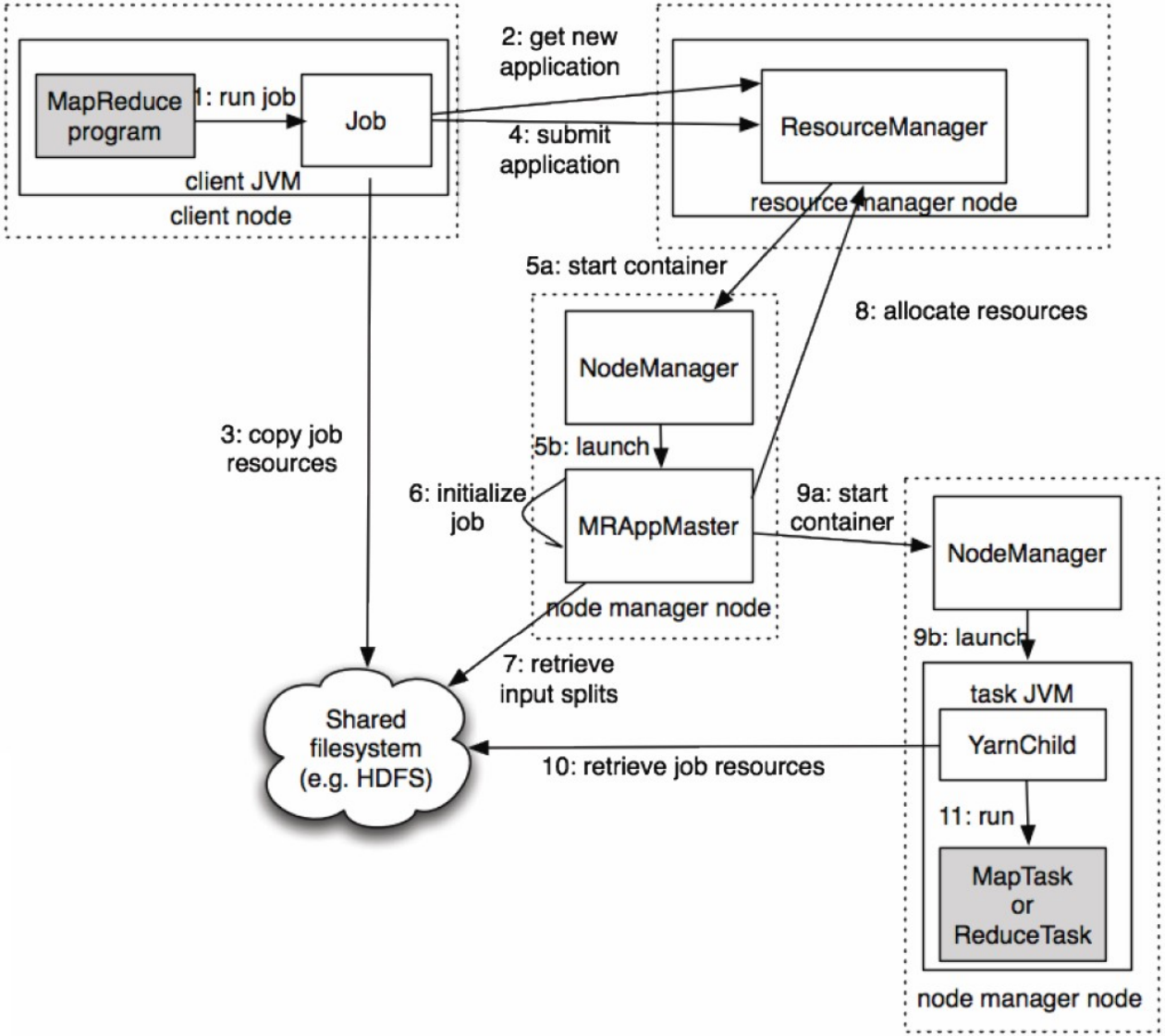
- When the work is completed, the JobTracker updates its status

- Client applications can poll the JobTracker for information

# Job flow (MapReduce v1)



# Job flow (MapReduce v2)



# Hadoop Ecosystem

- Apache Avro (serialization system for persistent data)
- Apache Pig (high-level dataflow querying language)
- Apache Hive (data warehouse infrastructure)

# Hadoop Ecosystem

- Apache HBase (database for real-time access)
- Apache Sqoop (tool for moving data from SQL to Hadoop or opposite)
- Apache ZooKeeper (distributed coordination service providing high availability) – library for building distributed systems

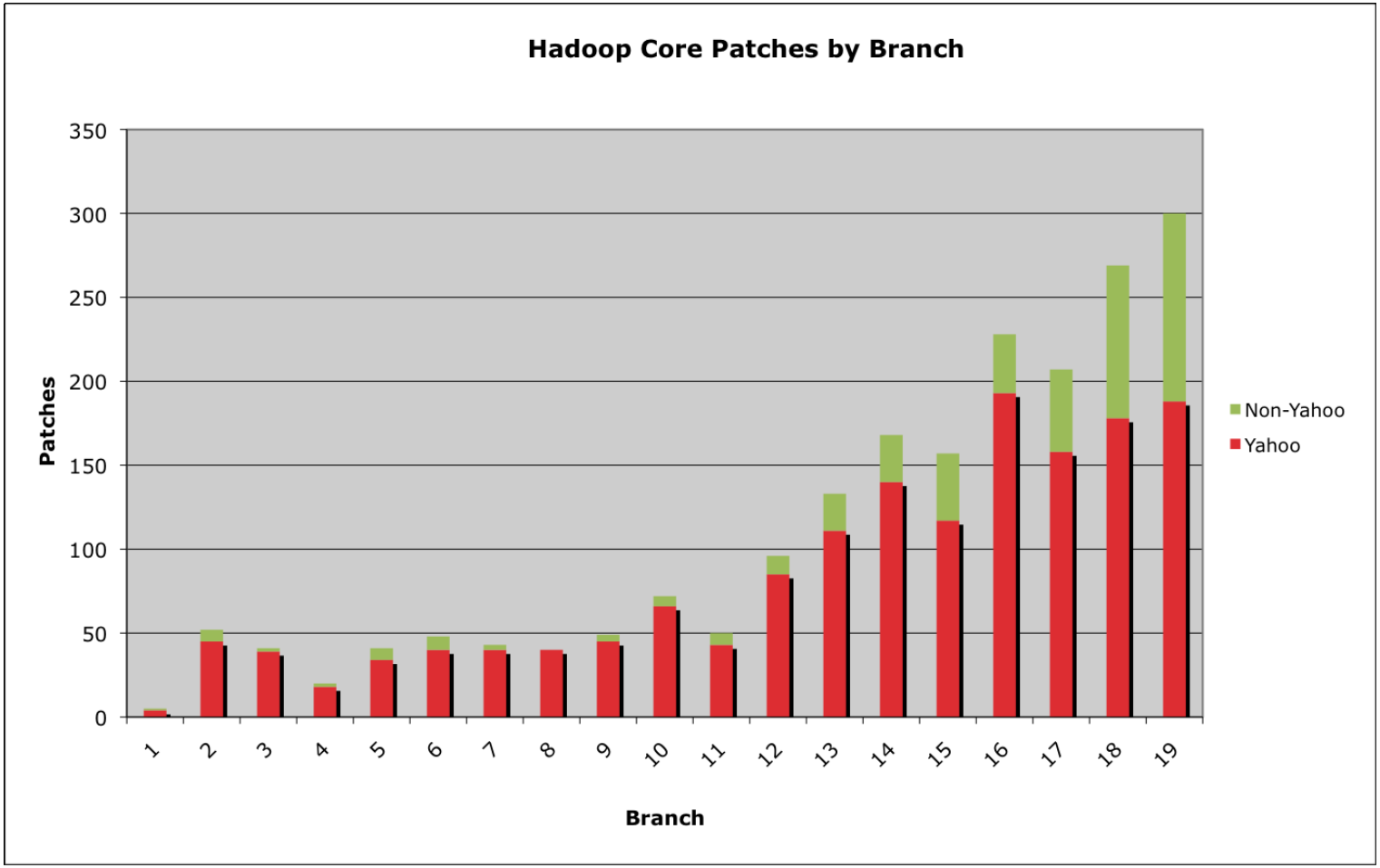
# Hadoop exclusive technologies

- YARN – Yet Another Resource Negotiator
- HDFS federation – possibility of partitioning namespace across several namenodes to support high number of files
- HDFS high-availability – techniques for for removing the namenode as the single point of failure

# Examples of production use

- Yahoo! : More than 100,000 CPUs in ~20,000 computers running Hadoop; biggest cluster: 2000 nodes (2*4cpu boxes with 4TB disk each); used to support research for Ad Systems and Web Search
- Facebook: To store copies of internal log and dimension data sources and use it as a source for reporting/analytics and machine learning; 320 machine cluster with 2,560 cores and about 1.3 PB raw storage

# Size of releases



# Hadoop

+ Framework for applications on large clusters
+ Built for commodity hardware
+ Provides reliability and data motion
+ Implements a computational paradigm named Map/Reduce
+ Very own distributed file system (HDFS) (very high aggregate bandwidth across the cluster)
+ Failures handles automatically

# Hadoop

- Time consuming development
- Documentation sufficient, but not the most helpful
- HDFS is complicated and has plenty issues of its own
- Debugging a failure is a "nightmare"
- Large clusters require a dedicated team to keep it running properly
- Writing a Hadoop job becomes a software engineering task rather than a data analysis task