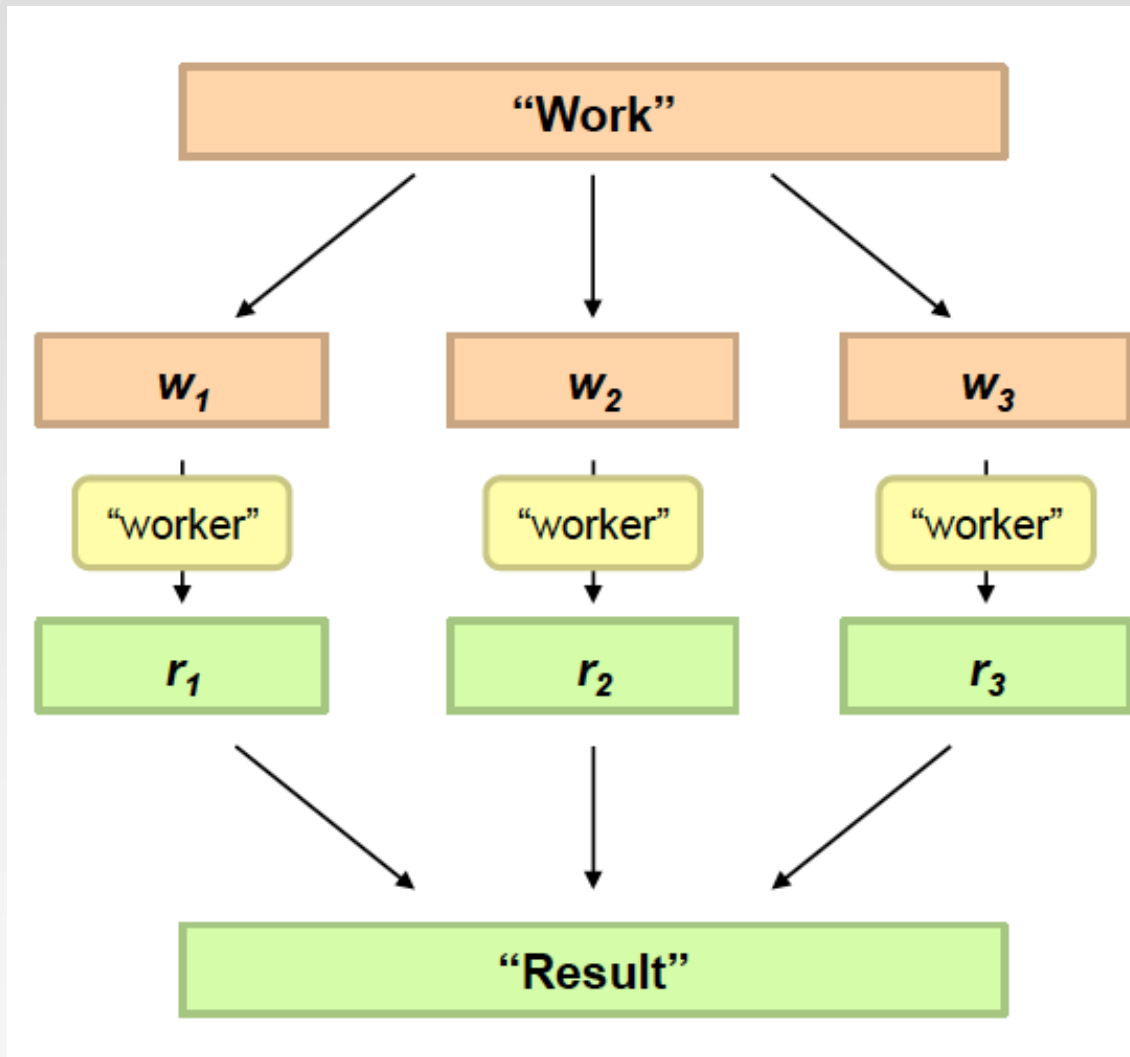


Philosophy to Scale for Big Data Processing

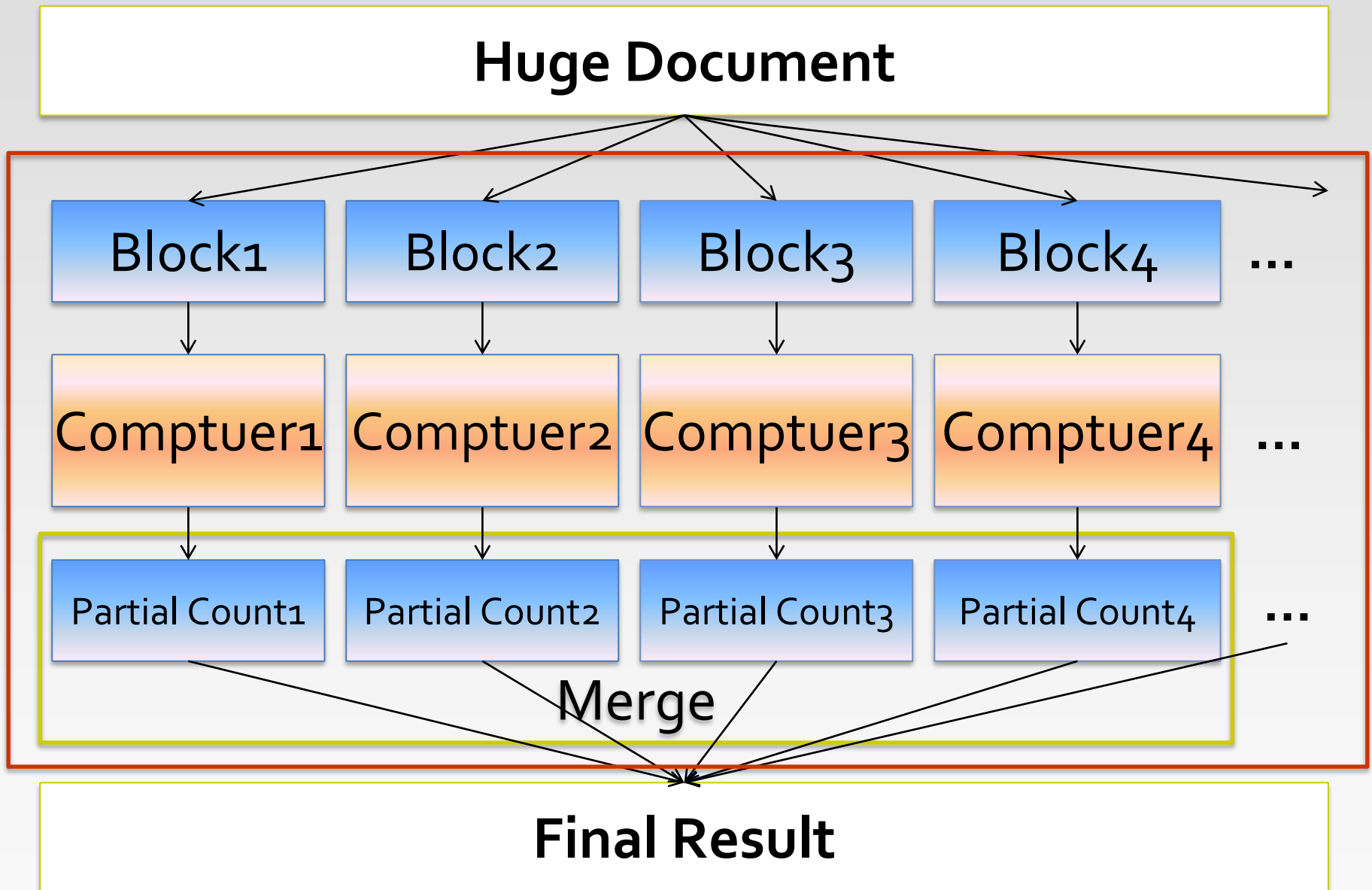


Divide Work



Combine Results

Distributed Word Count



Distributed Word Count

❖ Challenges?

- Where to store the huge document dataset?
- How to split the dataset into different blocks?
 - ▶ How many blocks?
 - ▶ The size of each block?
- What can we do if one node lost the data it received?
- What can we do if one node cannot be connected?
-

Distributed Processing is Non-Trivial

- ❖ How to assign tasks to different workers in an efficient way?
- ❖ What happens if tasks fail?
- ❖ How do workers exchange results?
- ❖ How to synchronize distributed tasks allocated to different workers?



Image courtesy of Master isolated images at FreeDigitalPhotos.net

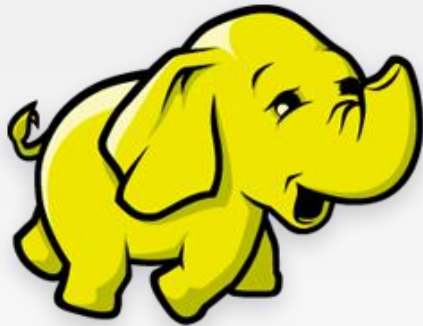
Big Data Storage is Challenging

- ❖ Data Volumes are massive
- ❖ Reliability of Storing PBs of data is challenging
- ❖ All kinds of failures: Disk/Hardware/Network Failures
- ❖ Probability of failures simply increase with the number of machines ...



What is Hadoop

- ❖ Open-source data storage and processing platform
- ❖ Before the advent of Hadoop, storage and processing of big data was a big challenge
- ❖ Massively scalable, automatically parallelizable
 - Based on work from Google
 - ▶ Google: GFS + MapReduce + BigTable (Not open)
 - ▶ Hadoop: HDFS + Hadoop MapReduce + HBase (opensource)
- ❖ Named by Doug Cutting in 2006 (worked at Yahoo! at that time), after his son's toy elephant.



Hadoop Offers

- ❖ Redundant, Fault-tolerant data storage
- ❖ Parallel computation framework
- ❖ Job coordination



Programmers

No longer need
to worry about



**Q: Where file is
located?**

**Q: How to handle
failures & data lost?**

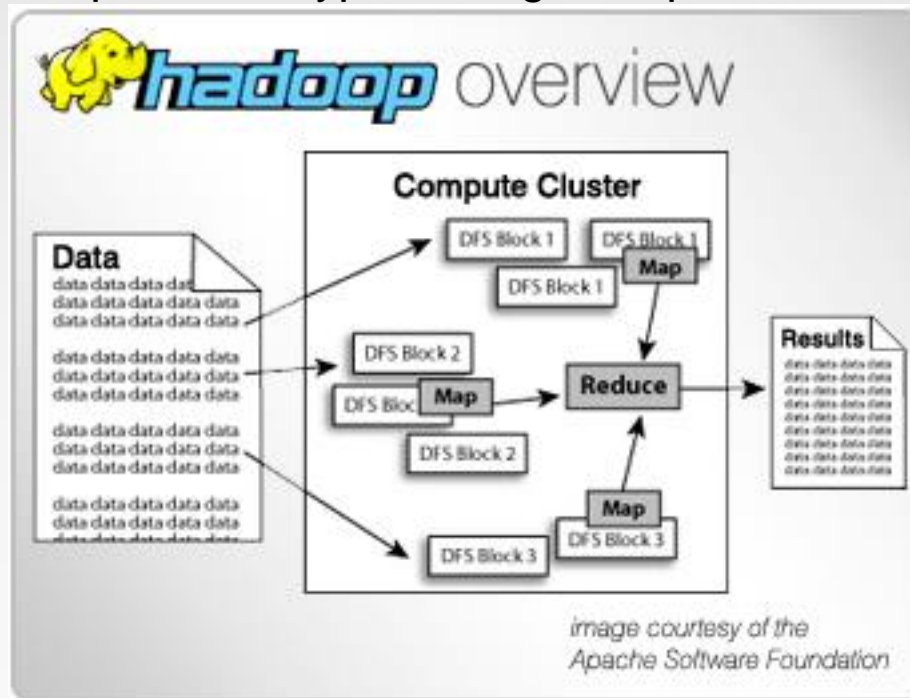
**Q: How to divide
computation?**

**Q: How to program
for scaling?**

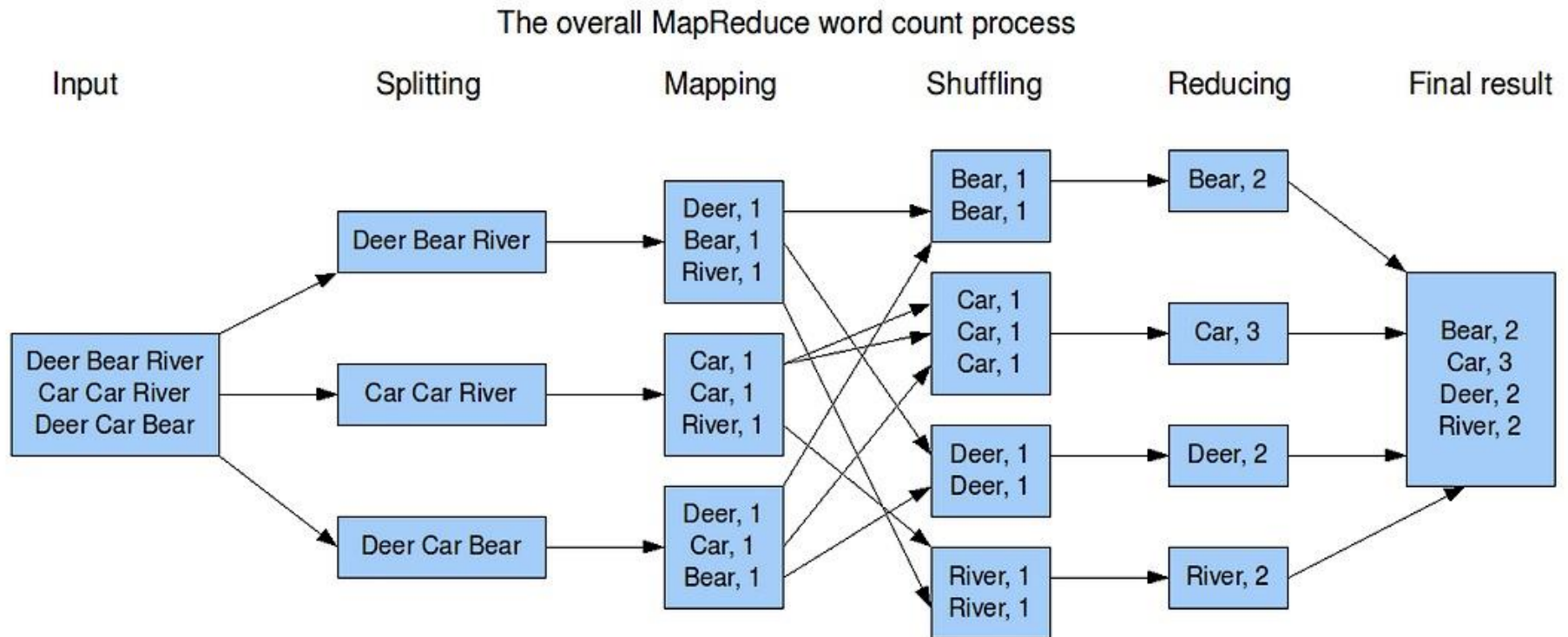
... ..

Why Use Hadoop?

- ❖ Cheaper
 - Scales to Petabytes or more easily
- ❖ Faster
 - Parallel data processing
- ❖ Better
 - Suited for particular types of big data problems



MapReduce Example - WordCount



- ❖ Hadoop MapReduce is an implementation of MapReduce
 - MapReduce is a computing paradigm (Google)
 - Hadoop MapReduce is an open-source software

Companies Using Hadoop



The New York Times



Hadoop 1.x

❖ Data storage (**HDFS**)

- Runs on commodity hardware (usually Linux)
- Horizontally scalable

Processing (**MapReduce**)

Parallelized (scalable) processing
Fault Tolerant

Other Tools/Frameworks

HBase

Hive

... ..

HDFS Storage

Redundant (3 copies)
For large files – large blocks
64 or 128 MB / block
Can scale to 1000s of nodes

MapReduce API

Batch (Job) processing
Distributed and Localized to clusters (Map)
Auto-Parallelizable for huge amounts of data
Fault-tolerant (auto retries)
Adds high availability and more

Other Libraries

Pig
Hive
HBase
Others

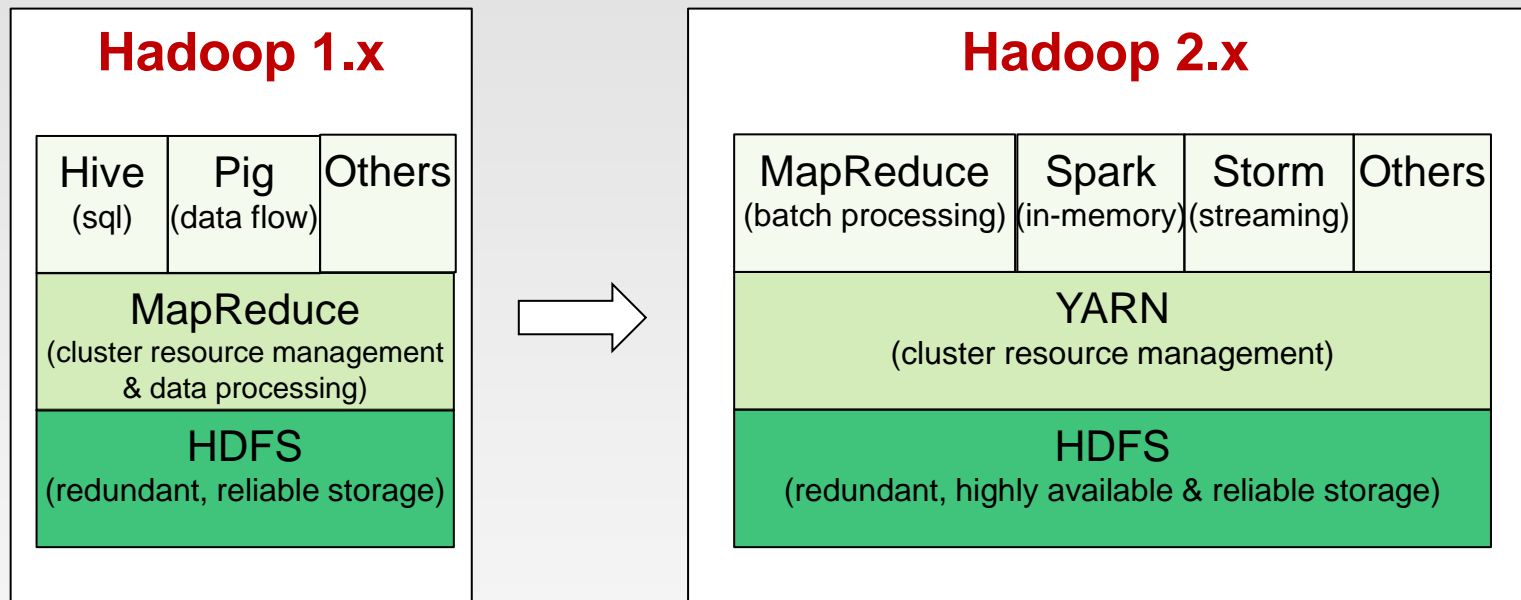
Hadoop 2.x

❖ Single Use System

➤ Batch apps

Multi-Purpose Platform

Batch, Interactive, Online, Streaming



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

Hadoop 3.x

	Hadoop 2.x	Hadoop 3.x
Minimum supported Java version	JAVA 7	JAVA 8/11
Storage Scheme	3x Replication Scheme	Erasure encoding in HDFS
Fault Tolerance	Replication is the only way to handle fault tolerance which is not space optimized.	Erasure coding is used for handling fault tolerance.
Storage Overhead	200% of HDFS (6 blocks of data will occupy the space of 18 blocks due to replication factor)	50% (6 blocks of data will occupy 9 blocks i.e 6 blocks for actual data and 3 blocks for parity)
Scalability	Limited Scalability, can have upto 10000 nodes in a cluster.	Scalability is improved, can have more then 10000 nodes in a cluster.
NameNodes	A single active NameNode and a single Standby NameNode	Allows users to run multiple standby NameNodes to tolerate the failure of more nodes