جلسه هجدهم سلسله نشست های رایانش ابری، اینترنت اشیاء و کلان داده ها

جلسه ششم کلان داده ها:

Intro to Apache Hadoop

By: Shahab Safaee
Computer Software Engineering PhD
Email: safaee.shx@gmail.com


cibtrc.ir


t.me/cibtrc


www.instagram.com/cibtrc/

Agenda

- Introduction
- Fast Historical Facts
- Evolution of the Hadoop
- Apache Hadoop
- Difference between Hadoop 1 and Hadoop 2
- Hadoop 3 Benefits
- Hadoop Framework
- Hadoop Ecosystem
- MapReduce Paradigm
- Hadoop Execution Engine
- MapReduce Word Count Process

Introduction

- Apache Hadoop is a processing framework that exclusively provides batch processing.
- Hadoop was the first big data framework to gain significant traction in the open-source community.

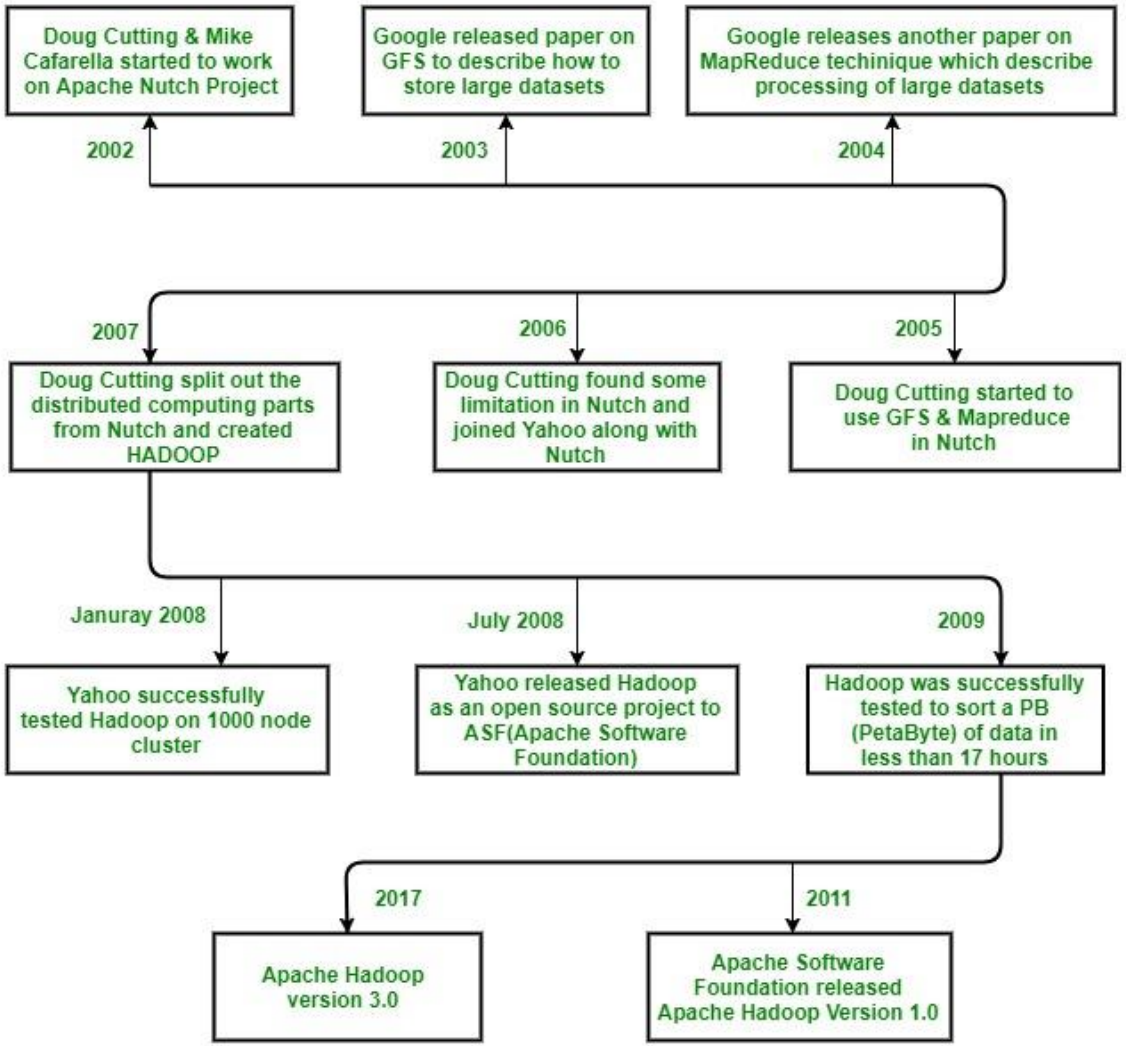
Fast Historical Facts (1)

- The code that eventually become Hadoop was written by Doug Cutting and Mike Cafarella, open source developers working in the search tech community, as part of Nutch project.
- "Hadoop" was the name of a yellow toy elephant owned by the son of Doug Cutting. Cutting's son, then 2, was just beginning to talk and called his beloved stuffed yellow elephant "Hadoop".

Fast Historical Facts (2)

- Yahoo! Was the first user of Hadoop in large-scale production, and cutting did early work on Hadoop there.
- In 2006, Hadoop was released by Yahoo and today is maintained and distributed by Apache Software Foundation (ASF).
- Eventually, Cutting joined Cloudera as its cheief architect and remains there to this day.

Evolution of the Hadoop



Apache Hadoop (1)

- Modern versions of Hadoop are composed of several components or layers, that work together to process batch data:
  - HDFS:
    - HDFS is the distributed file system layer that coordinates storage and replication across the cluster nodes.
    - HDFS ensures that data remains available in spite of inevitable host failures
  - YARN:
    - YARN, which stands for Yet Another Resource Negotiator, is the cluster coordinating component of the Hadoop stack. It is responsible for coordinating and managing the underlying resources and scheduling jobs to be run.
  - MapReduce:
    - MapReduce is Hadoop's native batch processing engine.

Apache Hadoop (2)

- Apache Hadoop and its MapReduce processing engine offer a well-tested batch processing model that is best suited for handling very large data sets where time is not a significant factor.
- The low cost of components necessary for a well-functioning Hadoop cluster makes this processing inexpensive and effective for many use cases.
- Compatibility and integration with other frameworks and engines mean that Hadoop can often serve as the foundation for multiple processing workloads using diverse technology.

Difference between Hadoop 1 and Hadoop 2

1. Components: In Hadoop 1 we have MapReduce but Hadoop 2 has YARN(Yet Another Resource Negotiator) and MapReduce version 2.

HADOOP 1	HADOOP 2
HDFS	HDFS
Map Reduce	YARN / MRv2

Difference between Hadoop 1 and Hadoop 2

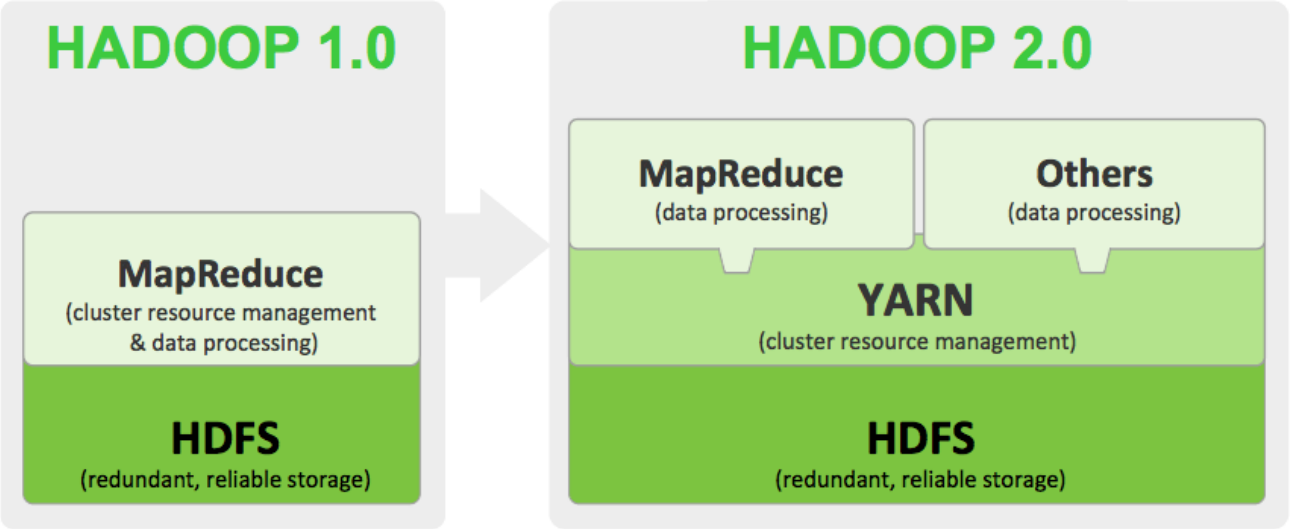
2. Daemons:

HADOOP 1	HADOOP 2
Namenode	Namenode
Datanode	Datanode
Secondary Namenode	Secondary Namenode
Job Tracker	Resource Manager
Task Tracker	Node Manager

Difference between Hadoop 1 and Hadoop 2

3. Working:

- In *Hadoop 1*, there is HDFS which is used for storage and top of it, Map Reduce which works as Resource Management as well as Data Processing. Due to this workload on Map Reduce, it will affect the performance.
- In *Hadoop 2*, there is again HDFS which is again used for storage and on the top of HDFS, there is YARN which works as Resource Management. It basically allocates the resources and keeps all the things going on.



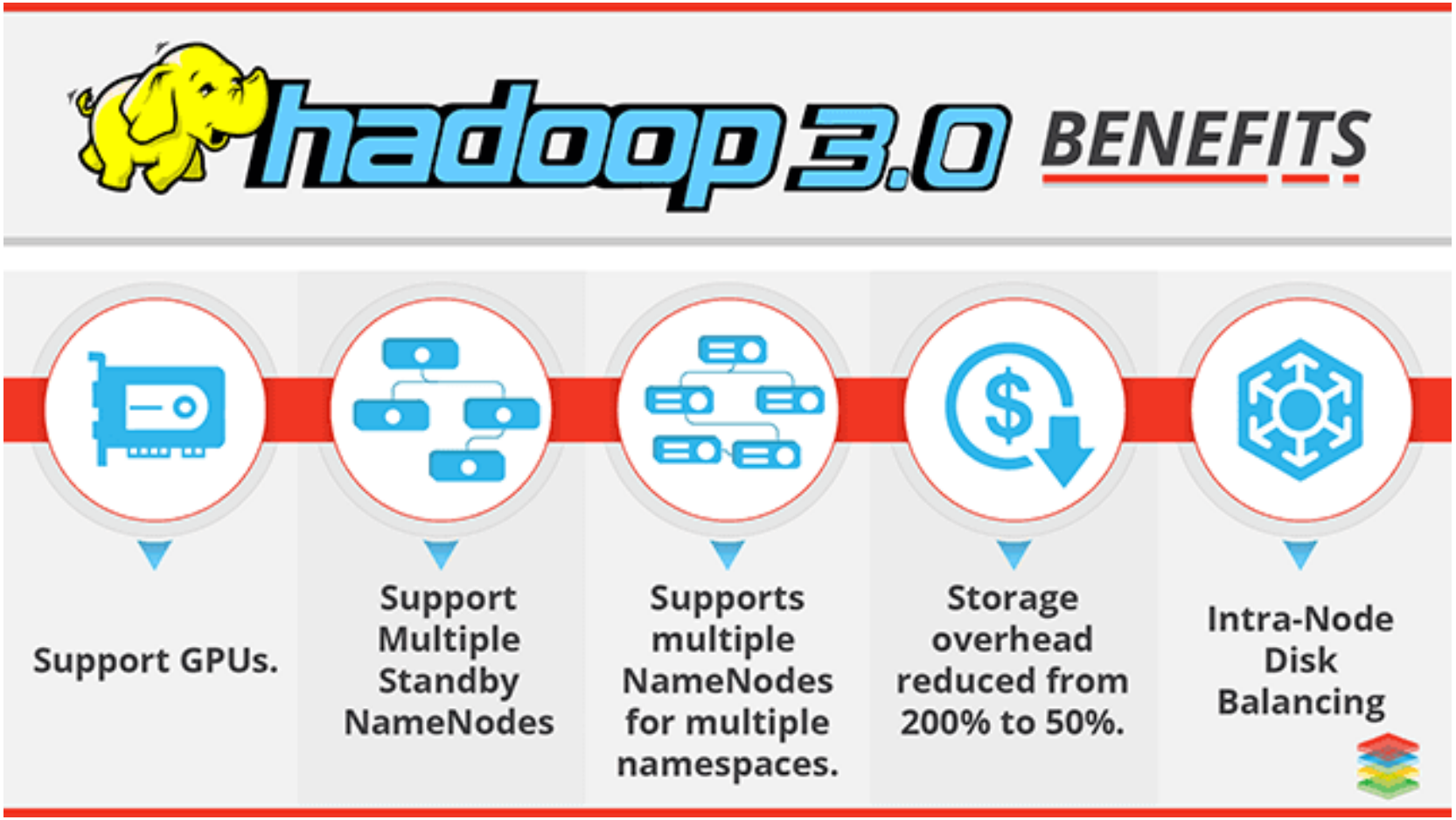
Difference between Hadoop 1 and Hadoop 2

4. Limitations:

Hadoop 1 is a Master-Slave architecture. It consists of a single master and multiple slaves. Suppose if master node got crashed then irrespective of your best slave nodes, your cluster will be destroyed. Again for creating that cluster means copying system files, image files, etc. on another system is too much time consuming which will not be tolerated by organizations in today's time.

Hadoop 2 is also a Master-Slave architecture. But this consists of multiple masters (i.e active namenodes and standby namenodes) and multiple slaves. If here master node got crashed then standby master node will take over it. You can make multiple combinations of active-standby nodes. *Thus Hadoop 2 will eliminate the problem of a single point of failure.*

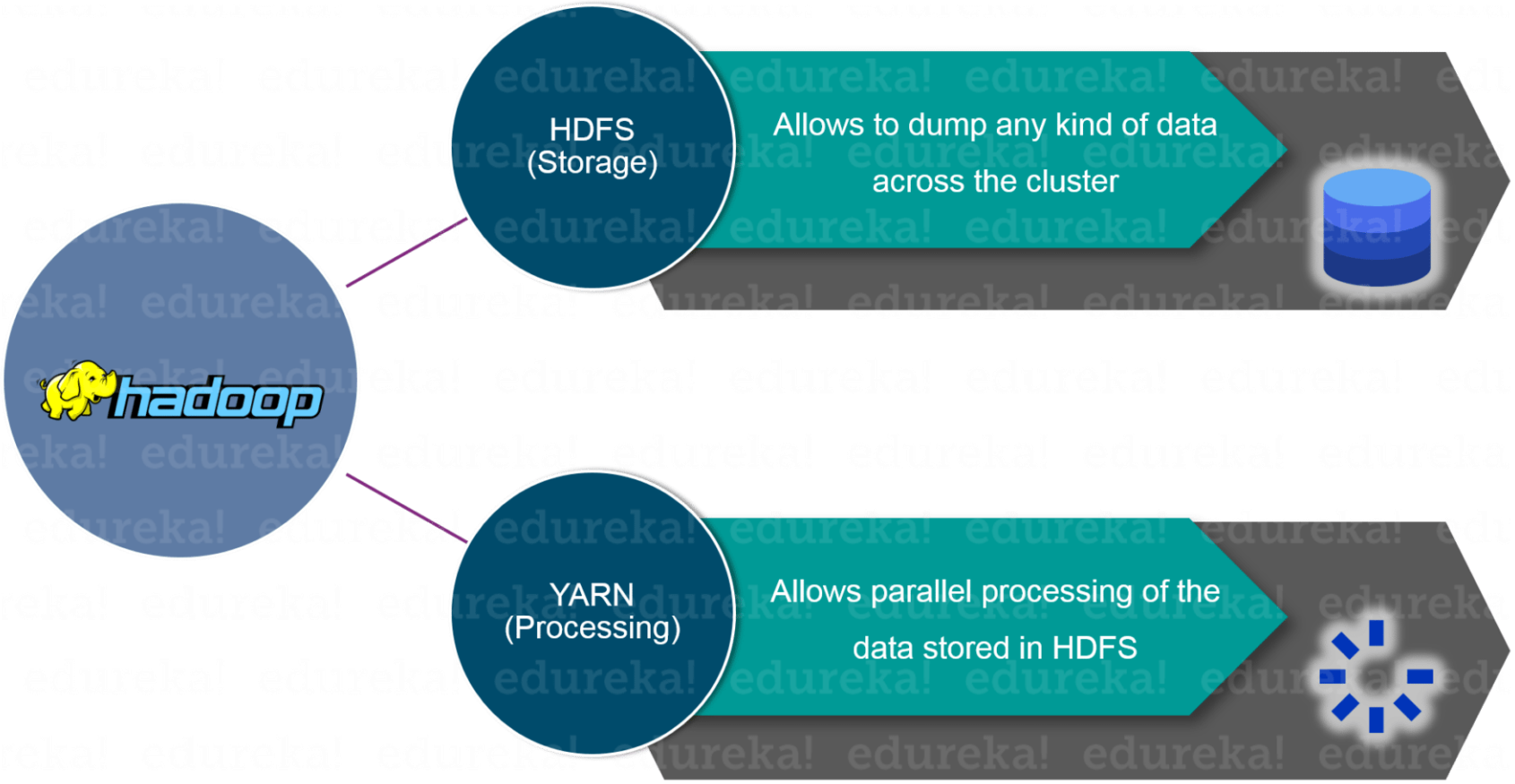
Hadoop 3 Benefits



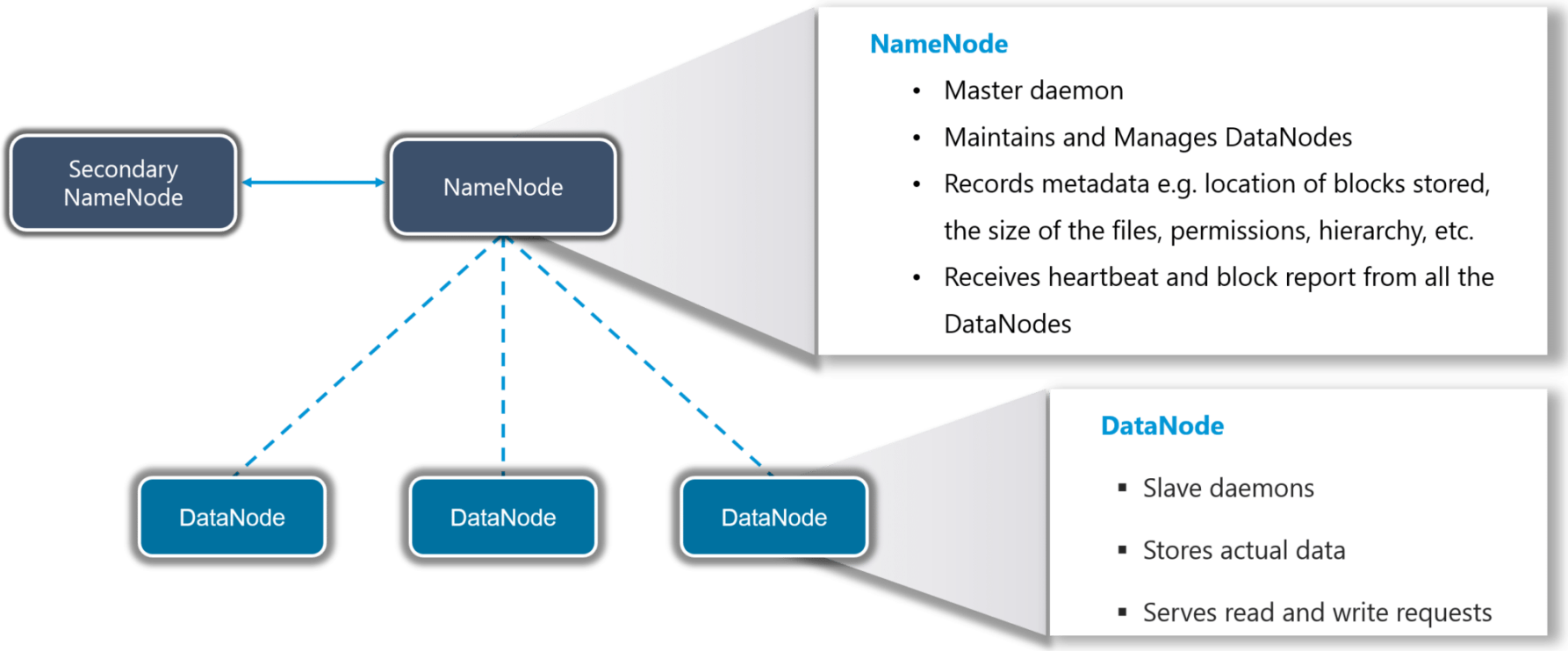
Hadoop 2 vs Hadoop 3

Features	Hadoop 2.x	Hadoop 3.x
Min Java Version Required	Java 7	Java 8
Fault Tolerance	Via replication	Via erasure coding
Storage Scheme	3x replication factor for data reliability, 200% overhead	Erasure coding for data reliability, 50% overhead
Yarn Timeline Service	Scalability issues	Highly scalable and reliable
Standby NN	Supports only 1 SBNN	Supports only 2 or more SBNN
Heap Management	We need to configure HADOOP_HEAPSIZE	Provides auto-tuning of heap

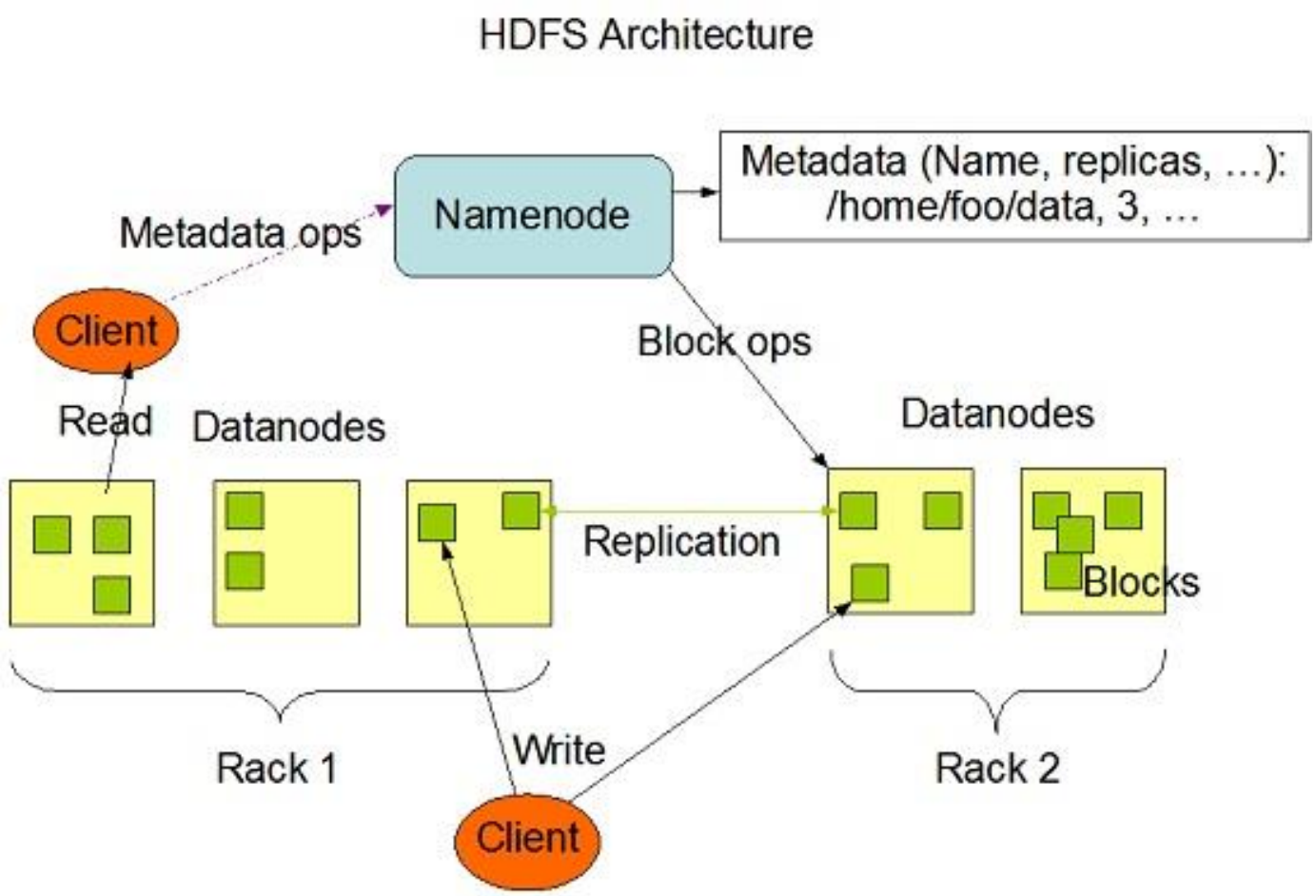
Hadoop Framework



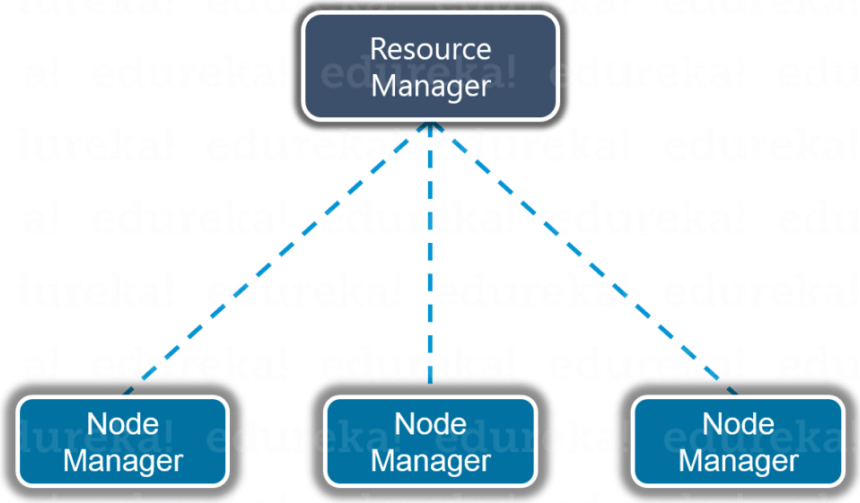
HDFS Component



HDFS Component



Yarn Component



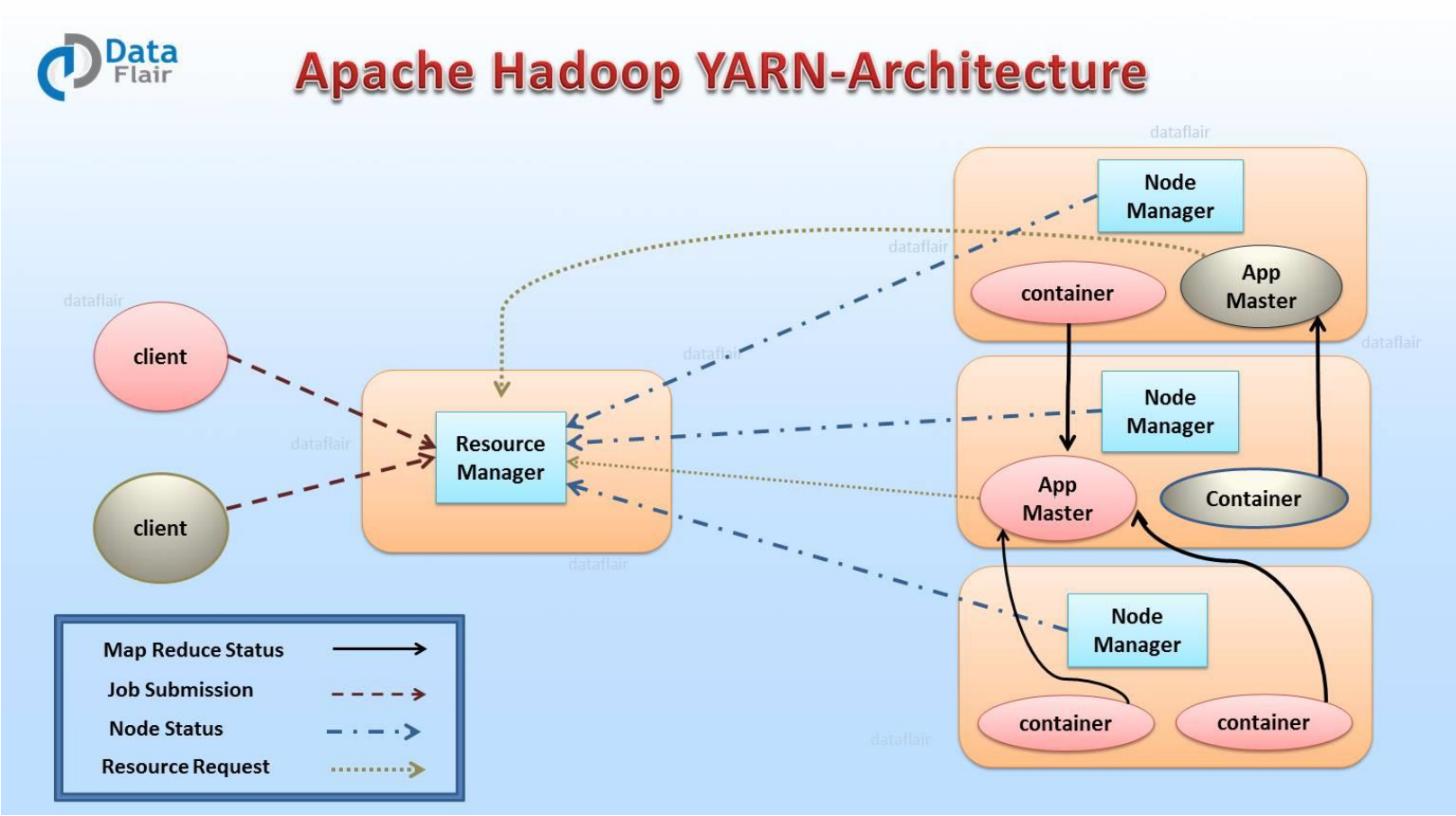
ResourceManager

- Receives the processing requests
- Passes the parts of requests to corresponding NodeManagers

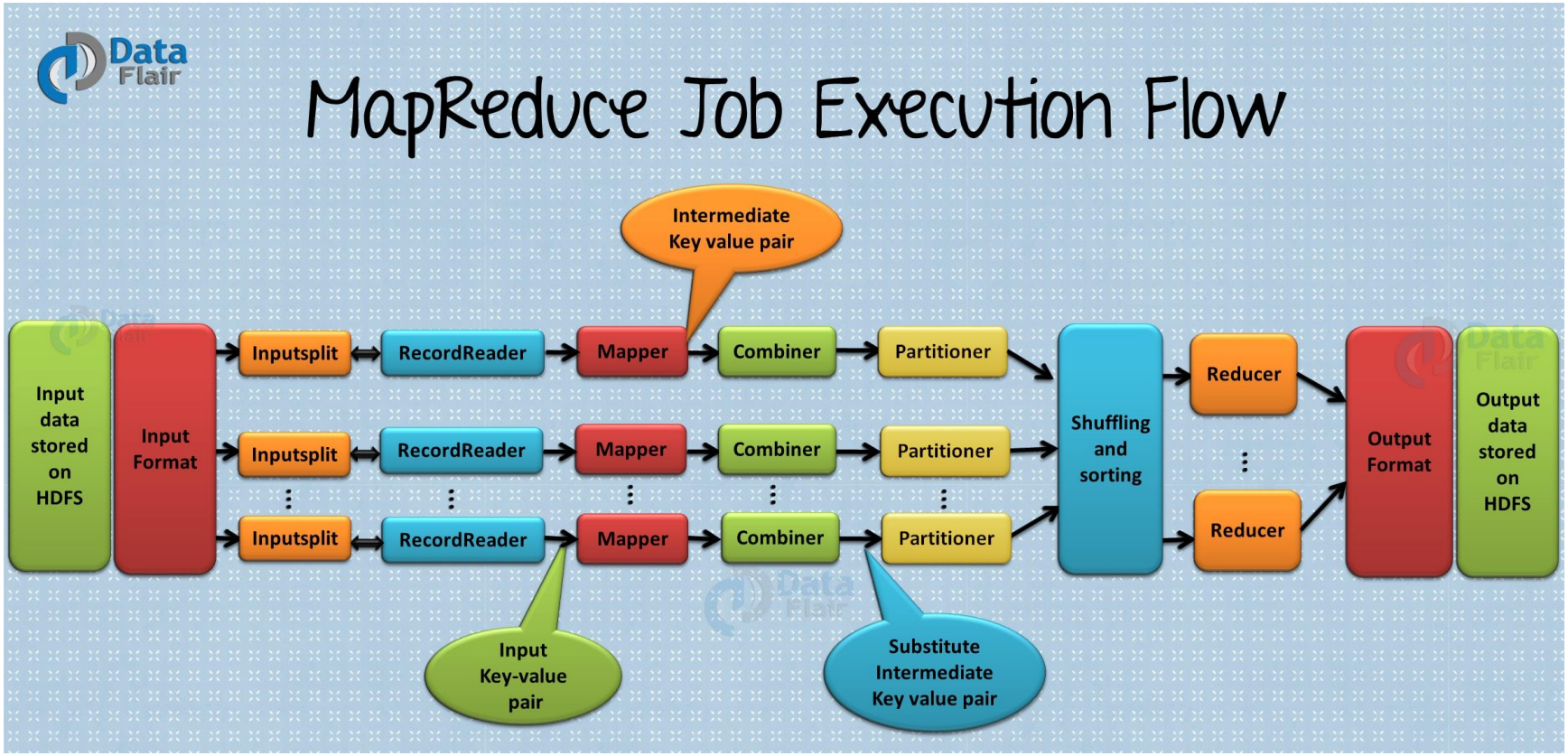
NodeManagers

- Installed on every DataNode
- Responsible for execution of task on every single DataNode

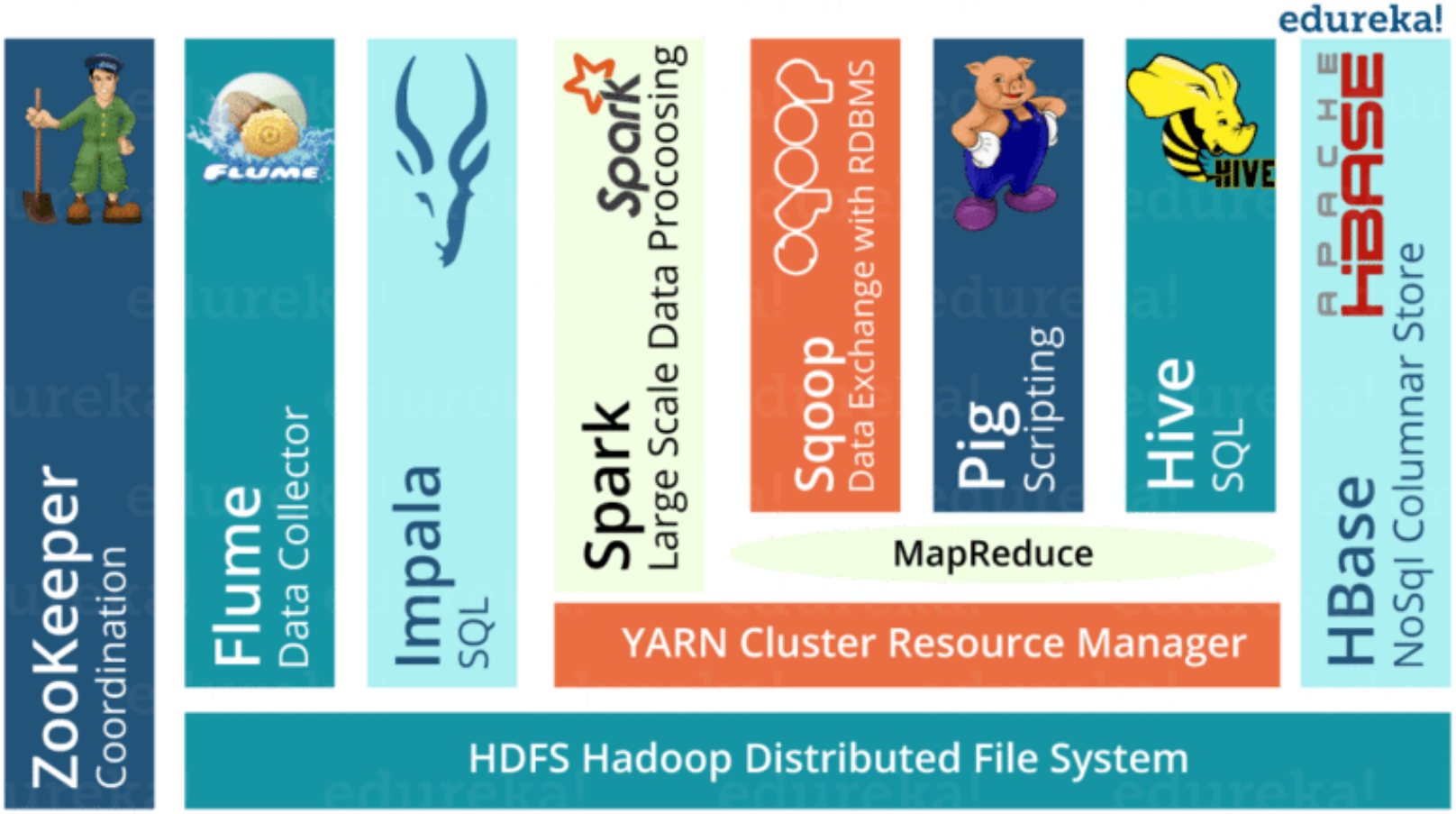
Yarn Component



Map/Reduce Component



Hadoop Ecosystem



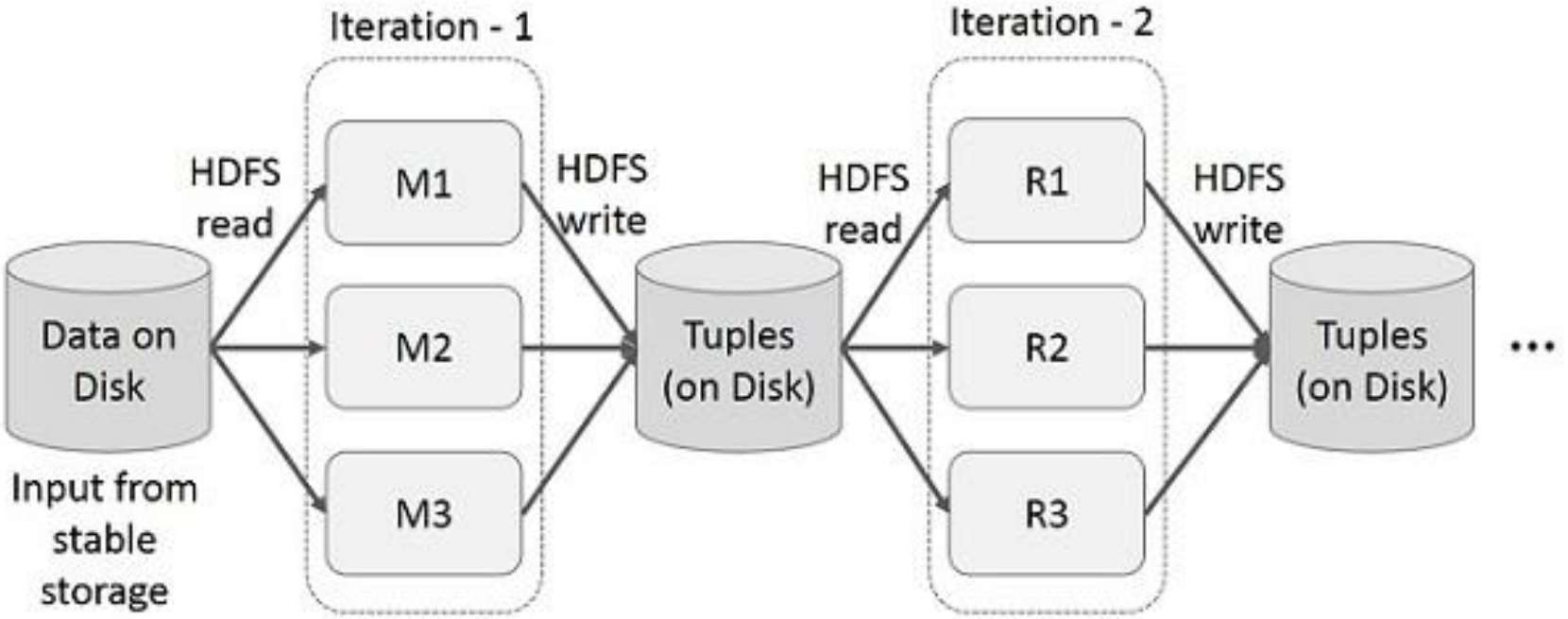
MapReduce Paradigm (1)

- MapReduce the current framework/paradigm for writing data-centric parallel applications in both industry and academia.
- MapReduce is inspired by the commonly used functions - Map and Reduce in combination with the divide-and conquer parallel paradigm.
- For a single MapReduce job, users implement two basic procedure objects Mapper and Reducer for different processing stages.

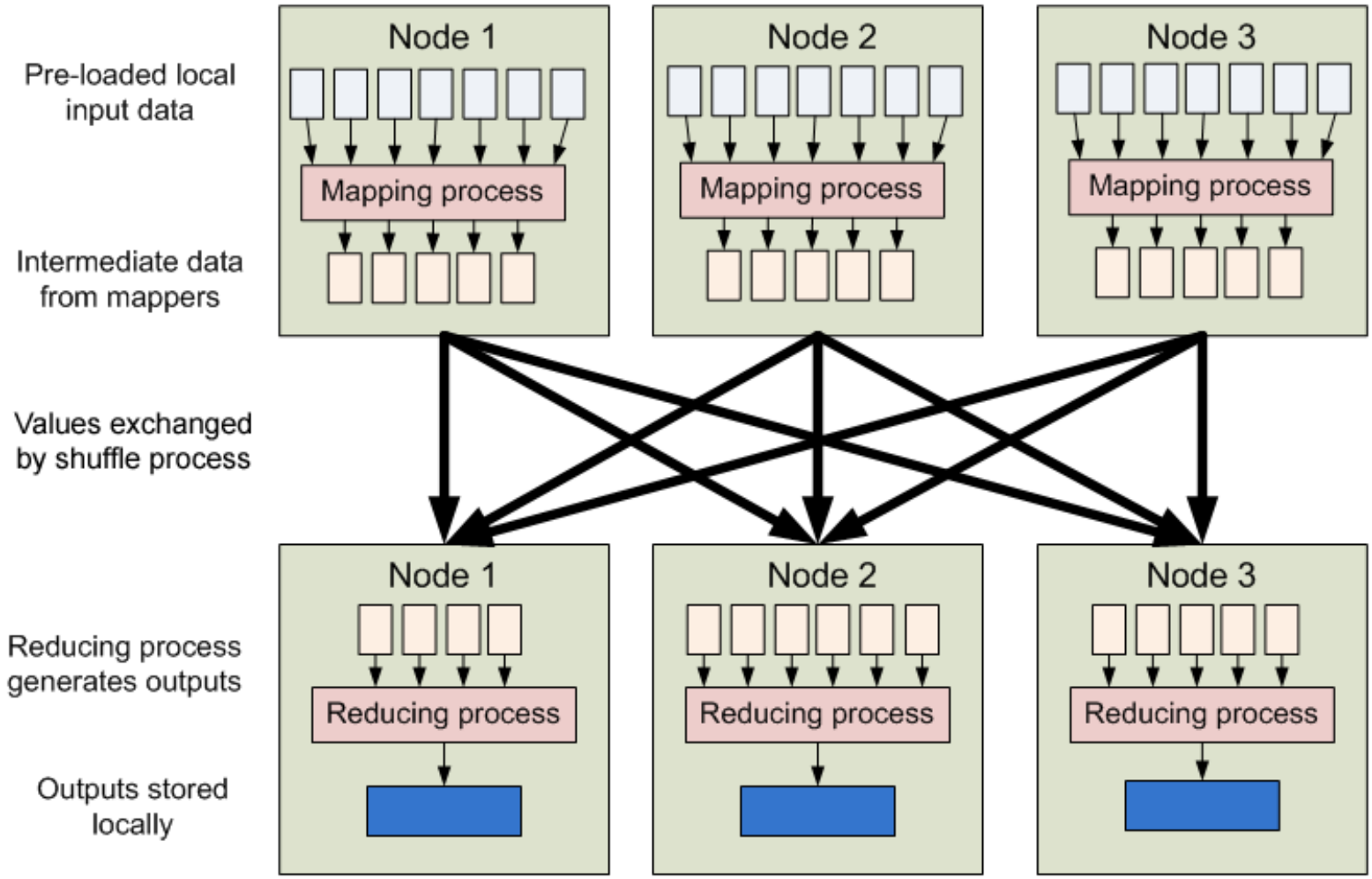
MapReduce Paradigm (2)

- Then the MapReduce program is automatically interpreted by the execution engine and executed in parallel in a distributed environments.
- MapReduce is considered as a simple yet powerful enough programming model to support a variety of the data-intensive programs.

MapReduce Paradigm (3)



MapReduce Dataflow



MapReduce Features (1)

- *Map and Reduce functions*
  - A MapReduce program contains a Map function doing the parallel transformation and a Reduce function doing the parallel aggregation and summary of the job.
  - Between Map and Reduce an implied Shuffle step is responsible for grouping and sorting the Mapped results and then feeding it into the Reduce step.

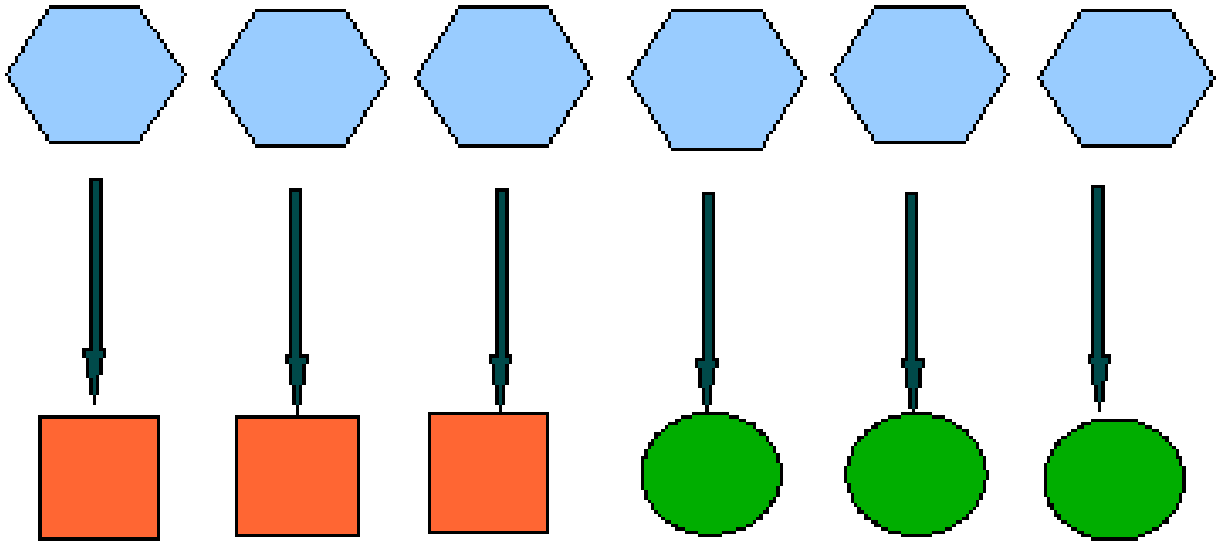
MapReduce Features (2)

- *Simple paradigm*
  - In MapReduce programming, users only need to write the logic of Mapper and Reducer while the logic of shuffling, partitioning and sorting is automatically done by the execution engine.
  - Complex applications and algorithms can be implemented by connecting a sequence of MapReduce jobs.

MapReduce Features (3)

- *Key-Value based*
  - In MapReduce, both input and output data are considered as Key-Value pairs with different types.
  - This design is because of the requirements of parallelization and scalability.
  - Key-value pairs can be easily partitioned and distributed to be processed on distributed clusters.
- *Parallelable and Scalable*
  - Both Map and Reduce functions are designed to facilitate parallelization, so MapReduce applications are generally linearly-scalable to thousands of nodes.

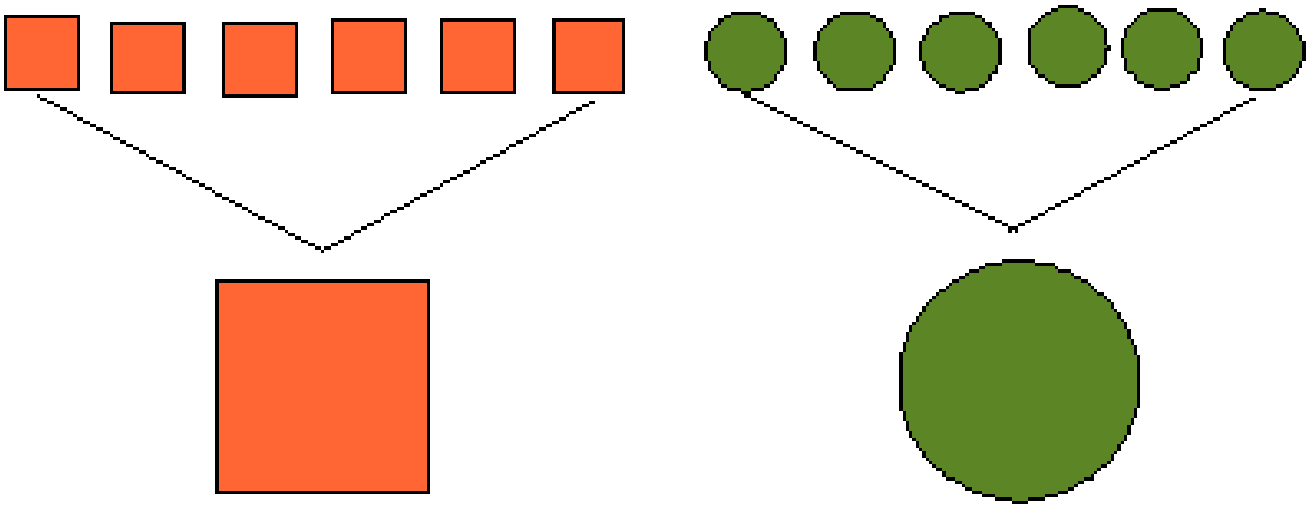
The Programming Model Of MapReduce (Map Phase)



MAP

map (in_key, in_value) -> (out_key, intermediate_value) list

The Programming Model Of MapReduce (Reduce Phase)



REDUCE

reduce (out_key, intermediate_value list) ->out_value list

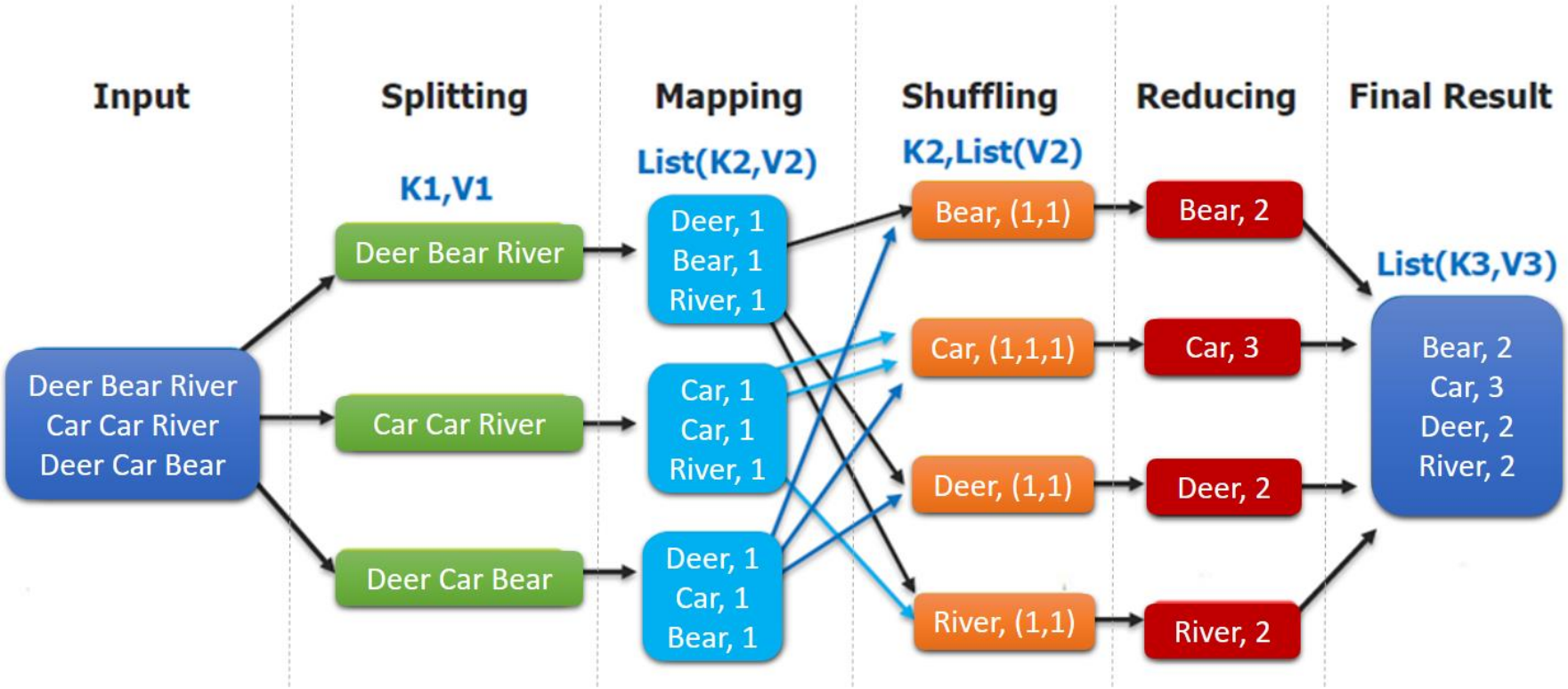
Hadoop Execution Engine (1)

- Hadoop MapReduce
  - Hadoop MapReduce is the core Hadoop ecosystem component which provides data processing.
  - MapReduce is a software framework for easily writing applications that process the vast amount of structured and unstructured data stored in the Hadoop Distributed File system.
  - MapReduce programs are parallel in nature, thus are very useful for performing large-scale data analysis using multiple machines in the cluster. Thus, it improves the speed and reliability of cluster this parallel processing.

Hadoop Execution Engine (3)

- Hadoop MapReduce
  - Hadoop is mainly implemented in Java, therefore, the map and reduce functions are wrapped as two interfaces called Mapper and Reducer.
  - The Mapper contains the logic of processing each key-value pair from the input.
  - The Reducer contains the logic for processing a set of values for each key.
  - Programmers build their MapReduce application by implementing those two interfaces and chaining them as an execution pipeline.

MapReduce Word Count Process



Word Count example in Hadoop

```
public void map(Object key,
                Text value, Context context) {
  String text = value.toString();
  StringTokenizer itr = new StringTokenizer(text);
  while(itr.hasMoreTokens()) {
    word.set(itr.nextToken());
    context.write(word, one);
  }
}

public void reduce(Text key,
       Iterable<IntWritable> values, Context context) {
  int sum = 0;
  for (IntWritable val : values) {
    sum += val.get();
  }
  result.set(sum);
  context.write(key, result);
}
```


Thank
You
Merci
Gracias
TapadhLeat
Köszönöm
Murakoze
aDank
BarakAllahuFiik
Grazzi
Grazie
Shukriya
Buznyg
WaadMahadsantahay
Takk
Enkosi
Nouari
ThintKo
TapadhLeibh
FaafetaiLava
Ngiyabonga
M-Sapo
Bedankt
Zikomo
Mèsi
Chokrane
Kiitos
AsanteSana
Blagodaram
Waita
Rahmat
Dhanyavaad
Xièxie
Dziakuju
Mauruuru
GoRaibhMaithAgat
Trugarez
KyayTzuTinPaTe
Mochchakkeram
Matondo
Welalin
TangioTumas
Aabhar
TeşekkürEderim
Sobodi
AabhariAahe
Barkal
KurreSumanga
GratiasAgimus
Mercé
TesekkurEdirem
Yekeniele
Dhanyabaad
Taiku
TerimaKasih
Mamnun
KeYaLeboha
Vinaka
Tanemirt
Tenki
Spacibo
KhobChaiDeu
Chnorakaloutioun
TänanVäga
Ačiū
KopKhunKrap
Tau
Nanni
Barka
Dankon
Dankie
Dakujem
aNiKié
Gratzias
Motashakkeram
Toda
Mersi
Nizzik ajr
Aguyjé
Danke
Trugéré
Akpé
Misaotra
Tak
Sagolun
Dhanyavadagalu
Grandmercé
Dhanyavadalu
Sulpáy
Diolch
EskerrikAsko
Dankewol
GratiasAgo
Tanmirt
Akiba
Gràcies
Blagodaria
Nandri
Arigatô
Hvala
Spas
Rahmet
MaloTeOfa
Gràçie
Gracie
KopKhunKha
Faleminderit
NajisTuke
Merkzi
Akun
Wado
Asante
Mahalo
Gracinas
KamSahHamnida
CámOn
Efharisto
DankJuWel
Marahaba
Meharbani
ÇoxSagOlun
Diky
Multumesc
Saha
Obrigado
Salamat
TakkFyri
Miigwetch
Nirringrazzjak
Bayarlalaa