MapReduce & Hadoop

- Juanjo Mostazo
- SeedRocket BCN
- June 2011

M/R: Motivation

- Process big amount of data to produce other data
- Scale up VS Scale out

M/R: What is it?

- Different programming paradigm
- Based on a google paper (2004)
- Automatic parallelization and distribution
- I/O Scheduling
- Fault tolerance
- Status and monitoring

M/R: Basics

- Input & Output: set of key/value pairs
- Big amount of data group & sort
- Job = Two phases = Mapper & Reducer

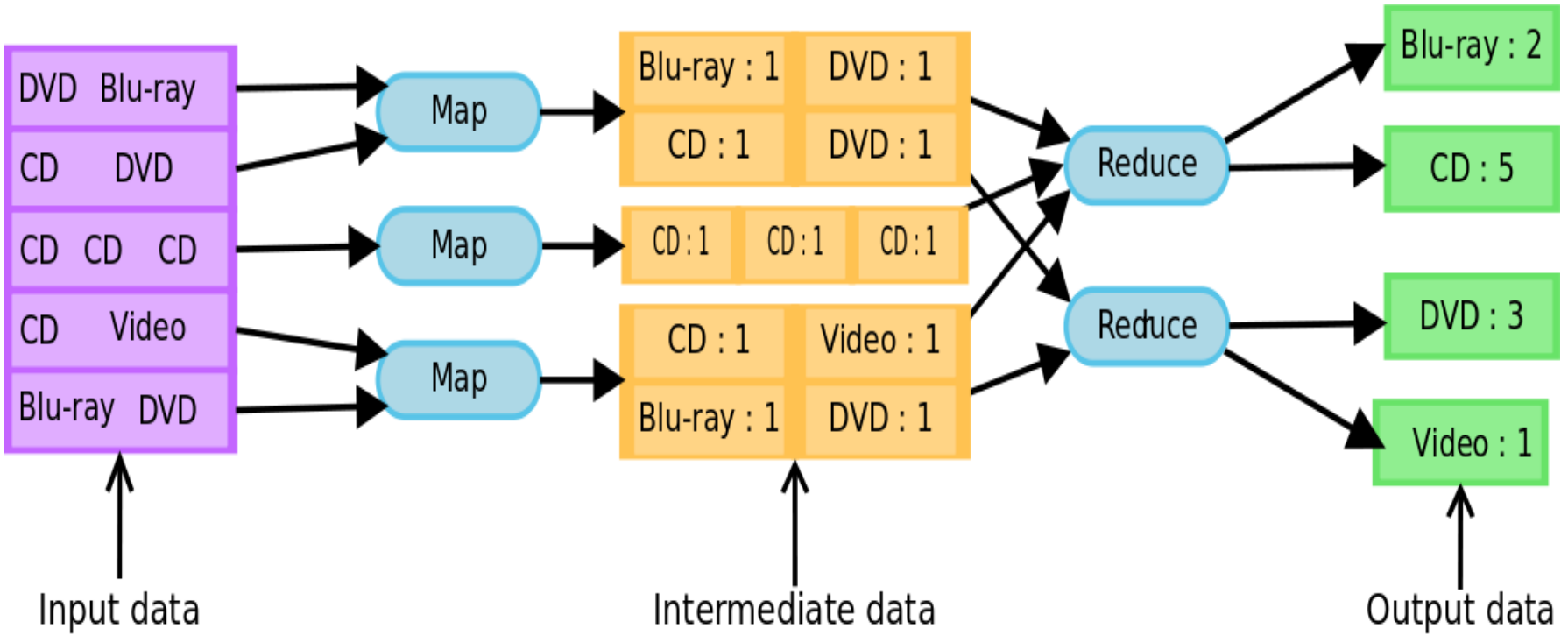
- Map (in_key, in_value) →
  list(interm_key, interm_value)

- Reduce (interm_key, list(interm_value)) →
  list (out_key, out_value)

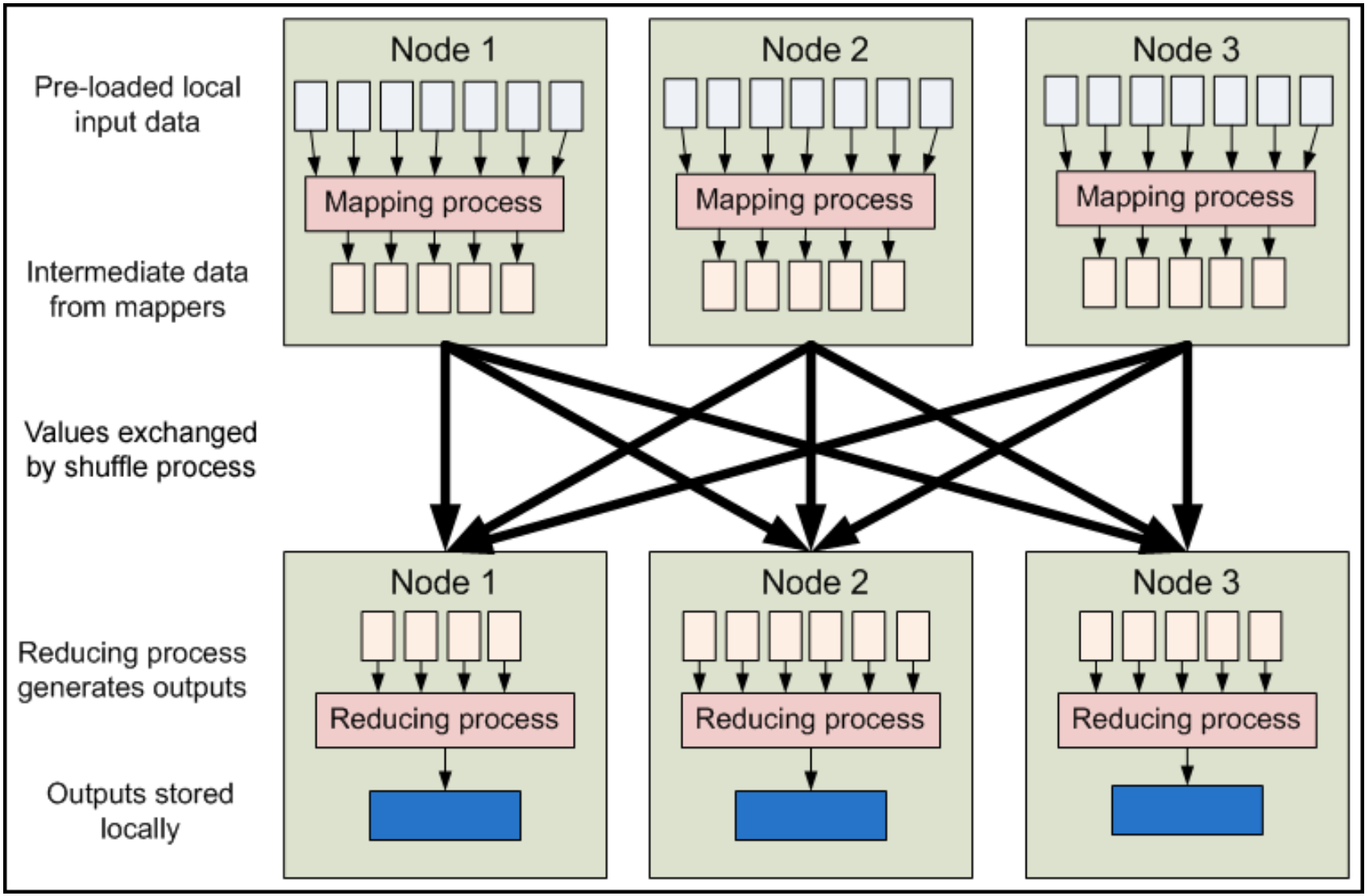
M/R: Example (word counter)

```
map(String input_key,
      String input_value):
   for each shape s in input_value:
      EmitIntermediate(s, 1);

reduce(String interm_key,
         Iterator interm_values):
   result = 0;
   for each v in interm_values:
      result += v;
   Emit(interm_key, result);
```



M/R: Workflow


Pre-loaded local input data
Intermediate data from mappers
Values exchanged by shuffle process
Reducing process generates outputs
Outputs stored locally
Node 1
Mapping process
Node 2
Mapping process
Node 3
Mapping process
Node 1
Reducing process
Node 2
Reducing process
Node 3
Reducing process

Hadoop: What is it?

- Framework based on GMR / GFS
- Apache project
- Developed in Java
- Multiple applications
- Used by many companies
- And growing!

Hadoop: HDFS dist. systems

- Data is sent to computation (whereas here computation goes to data)
- Nodes must get info from each other
- Very difficult to recover from partial failure

Hadoop: HDFS concepts

- Distributed file system.

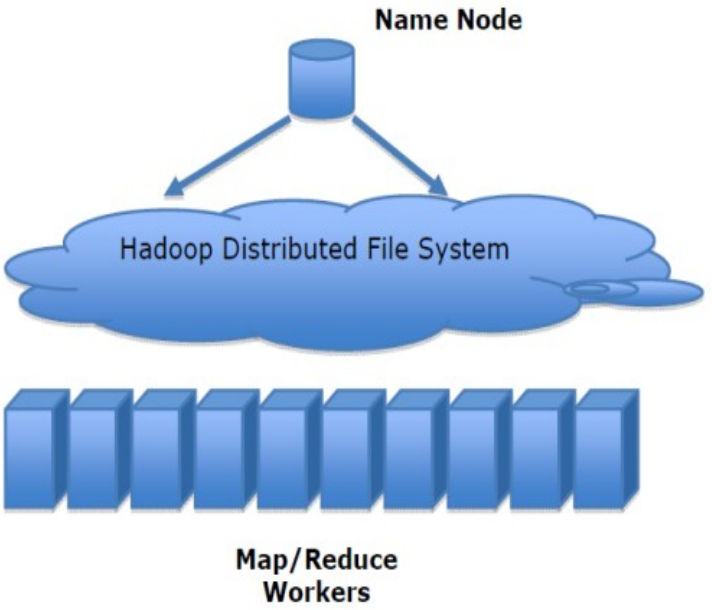
Layer on top ext3, xfs…

- Works better on huge files
- Redundancy (default 3)
- Bad seeking, no append!
- Good rack scale. Not

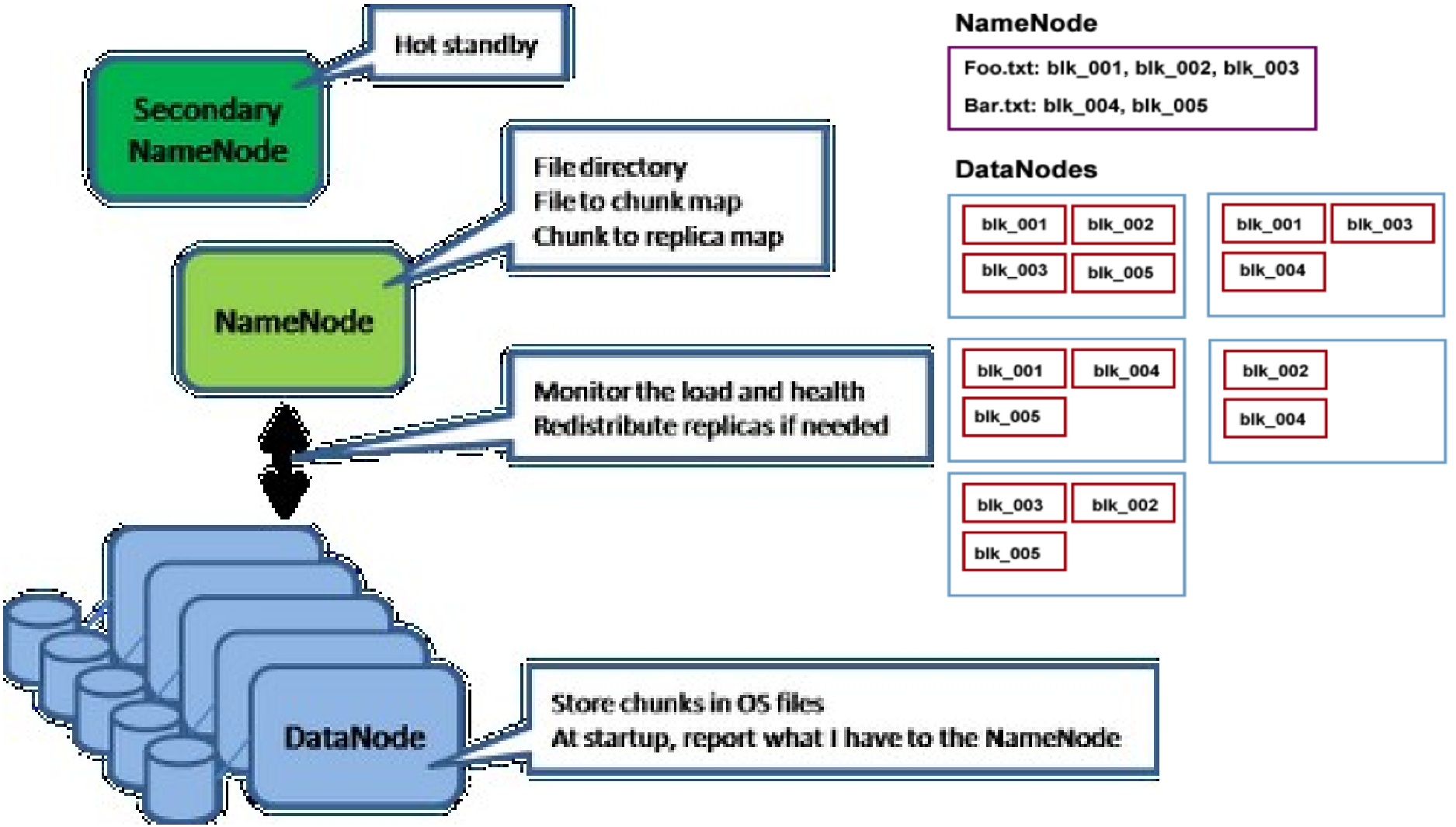
good data center scale

- File divided in 64Mb –

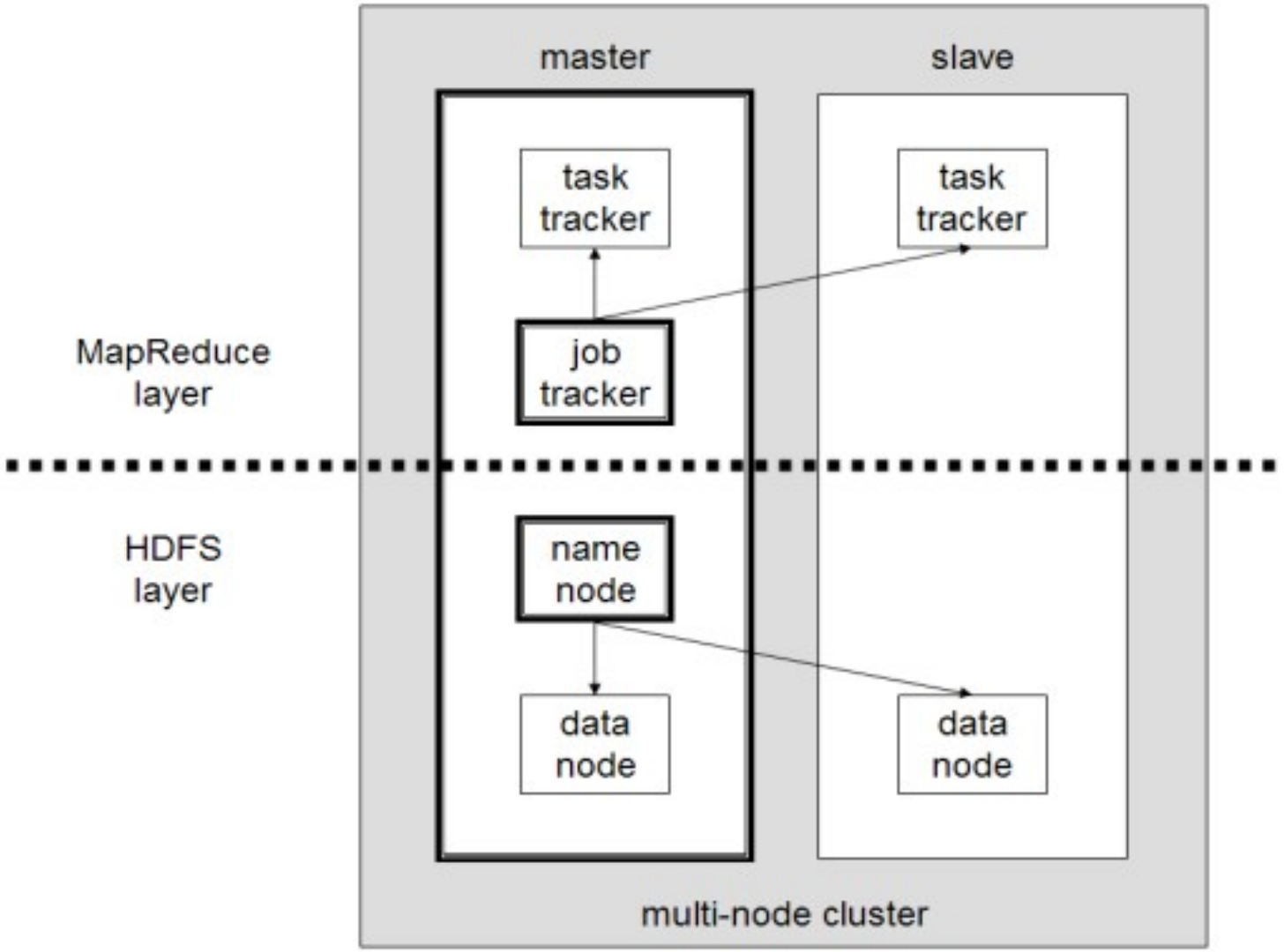
128Mb blocks



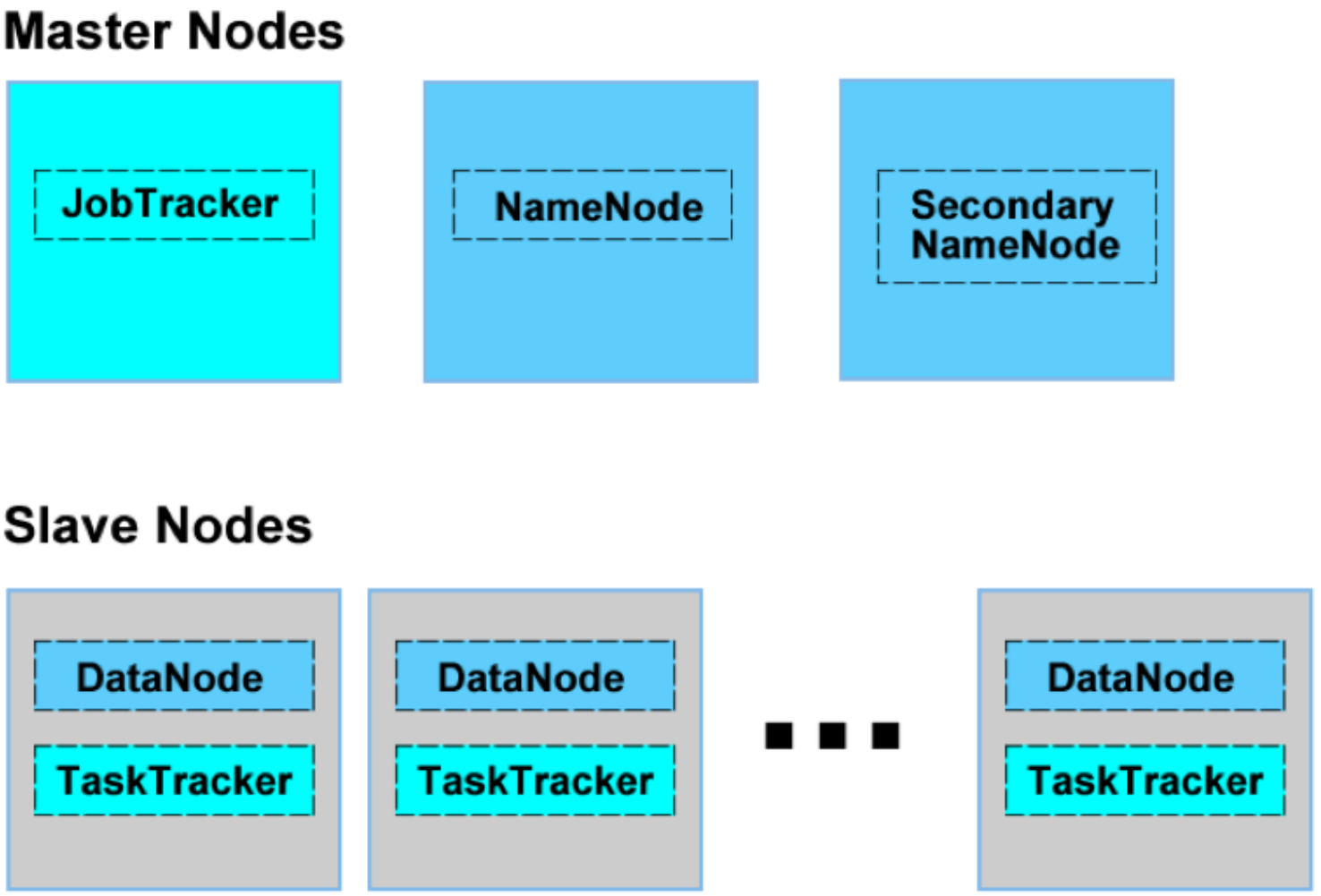
Hadoop: HDFS Architecture



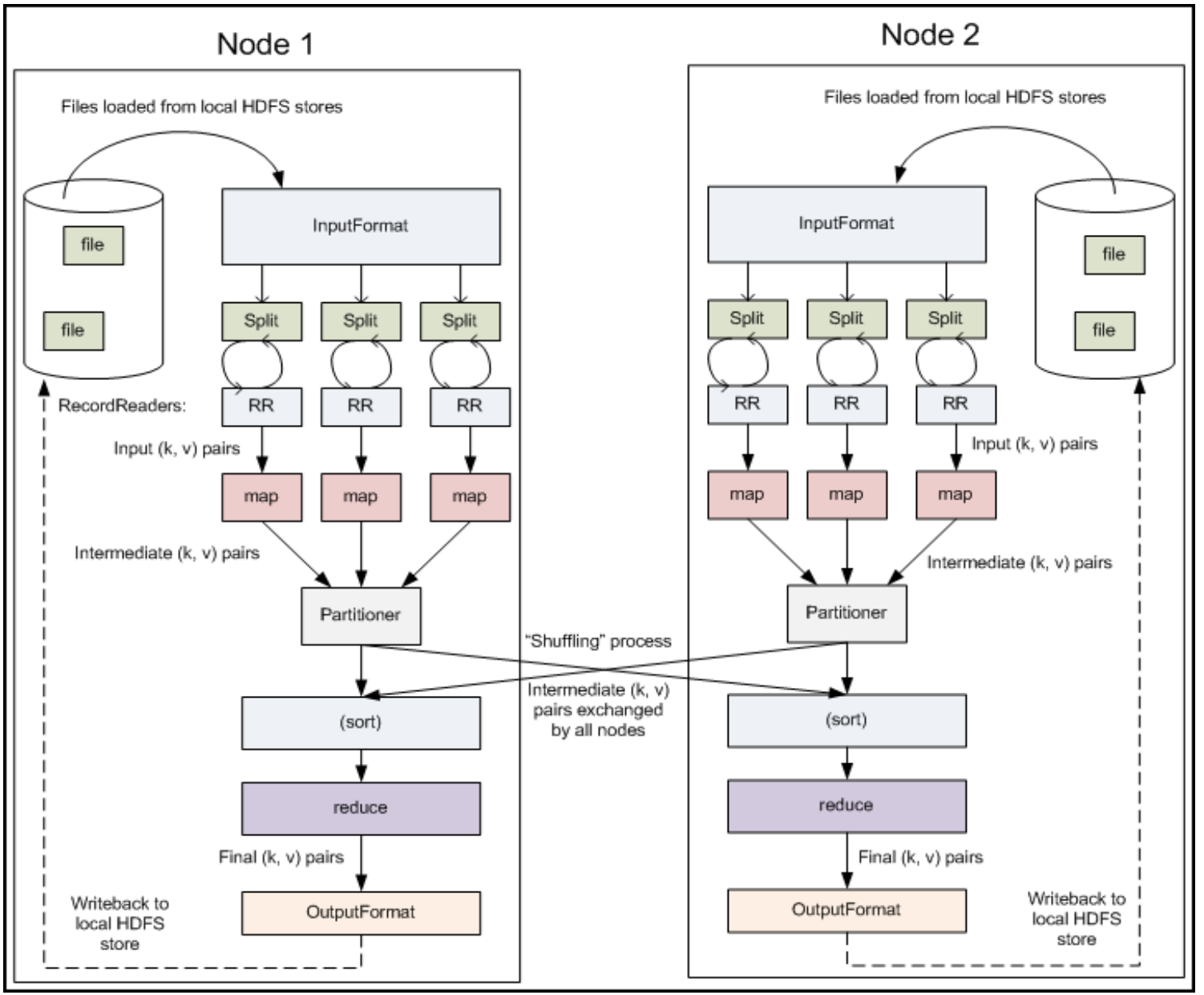
Hadoop: Architecture v1



Hadoop: Architecture v2



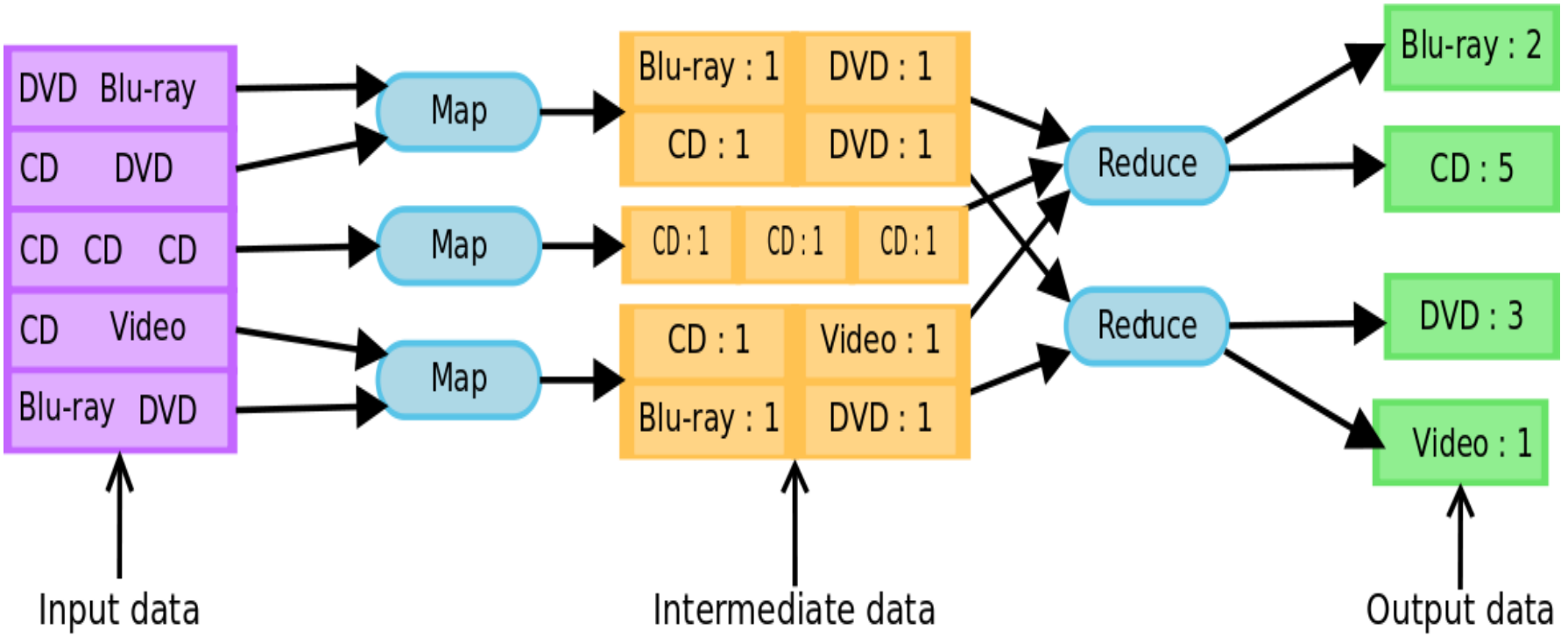
Hadoop: Architecture v3



M/R: Example (word counter)

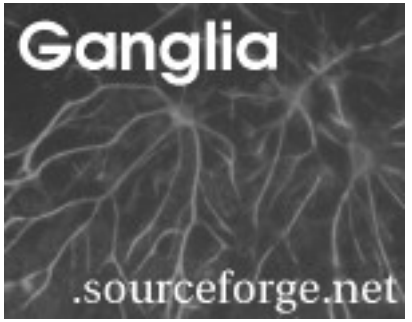
```
map(String input_key,
      String input_value):
   for each shape s in input_value:
      EmitIntermediate(s, 1);

reduce(String interm_key,
         Iterator interm_values):
   result = 0;
   for each v in interm_values:
      result += v;
   Emit(interm_key, result);
```



Hadoop: EcoSystem

- FLUME



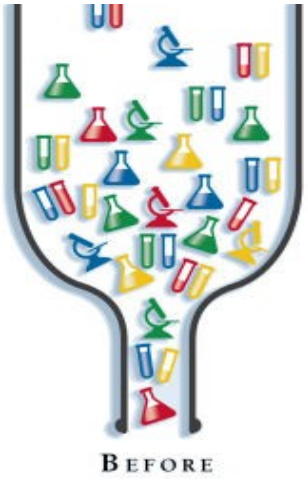
- ZOOKEEPER

Hadoop: Advanced stuff

- Distributed caches
- Partitioner
- Sort comparator
- Group comparator
- Combiner
- Input format & Record reader
- MultiInput
- MultiOutput
- Compression (LZO)

Hadoop: Conclussions

- Simplify large-scale computation
- Hide parallel programming issues
- Easy to get into & develop
- Deeply used & maintained by community
- Possibility yo throw away RDBMs! (Bottleneck)



MapReduce & Hadoop

- Juanjo Mostazo
- juanj.mostazo@gmail.com