CSC 369: Distributed Computing

Alex Dekhtyar

May 6

Day 14: Java Hadoop API

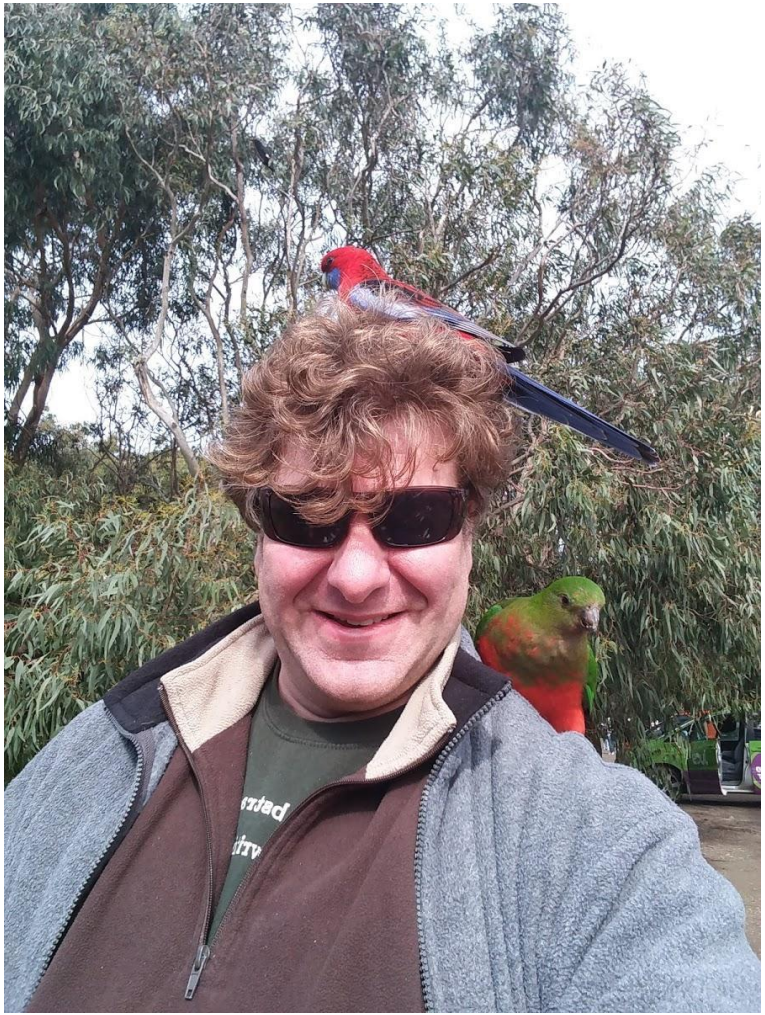
CSC 369: Distributed Computing

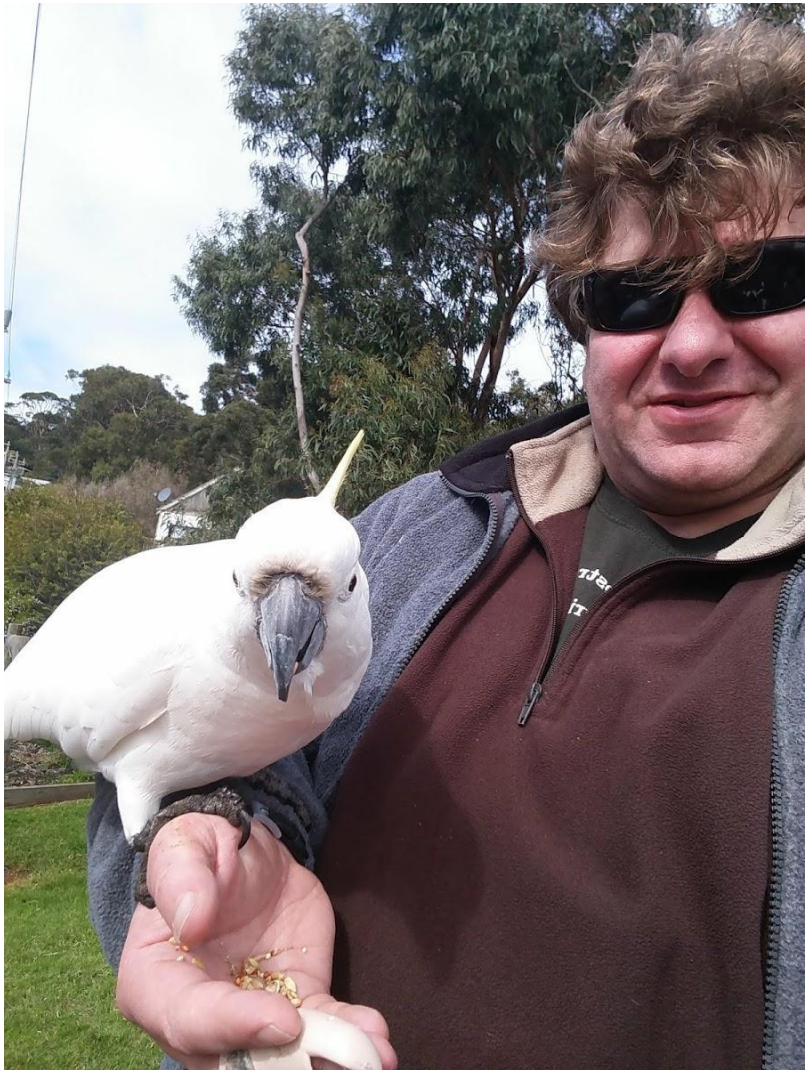
Alex Dekhtyar

May 6

Day 14: Java Hadoop API

HAPPY EQUATOR DAY!





Housekeeping

Lab 4 (mini-project): due Sunday night

Lab 5: due tonight (grace period tomorrow)

Lab 6: full lab coming out Friday

Grading: slowly happening…

Hadoop Java API

Hadoop API

Current Version is 3.2.1.

hadoop
hdfs
yarn

Command-line tools
We limit ourselves to `hadoop jar`

Hadoop Java API

`org.apache.hadoop`

Let’s concentrate on things we absolutely need

Hadoop Java API

`org.apache.hadoop`	
`org.apache.hadoop.mapreduce`	Core MapReduce classes
`org.apache.hadoop.mapreduce.lib.input` `org.apache.hadoop.mapreduce.lib.output`	Inuput/Output parsing
`org.apache.hadoop.io`	atomic type wrappers
`org.apache.hadoop.conf`	Job configuration
`org.apache.hadoop.fs`	File system classes

`org.apache.hadoop.mapreduce`

`org.apache.hadoop.mapreduce.Job` MapReduce Job

`org.apache.hadoop.mapreduce.Mapper` Extensible Mapper

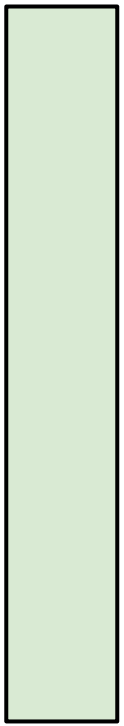
`org.apache.hadoop.mapreduce.Reducer` Extensible Reducer

`org.apache.hadoop.mapreduce.Partitioner` Parent class for Partitioning tasks

`org.apache.hadoop.mapreduce.InputFormat`
`org.apache.hadoop.mapreduce.OutputFormat` Parent classes for Input/Output Formats

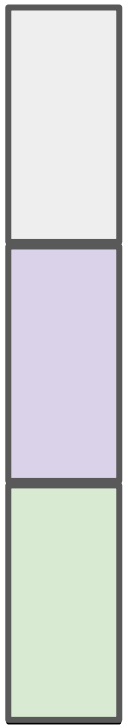
`org.apache.hadoop.mapreduce.InputSplit` Parent class for Input Split

How it works



Input File

How it works



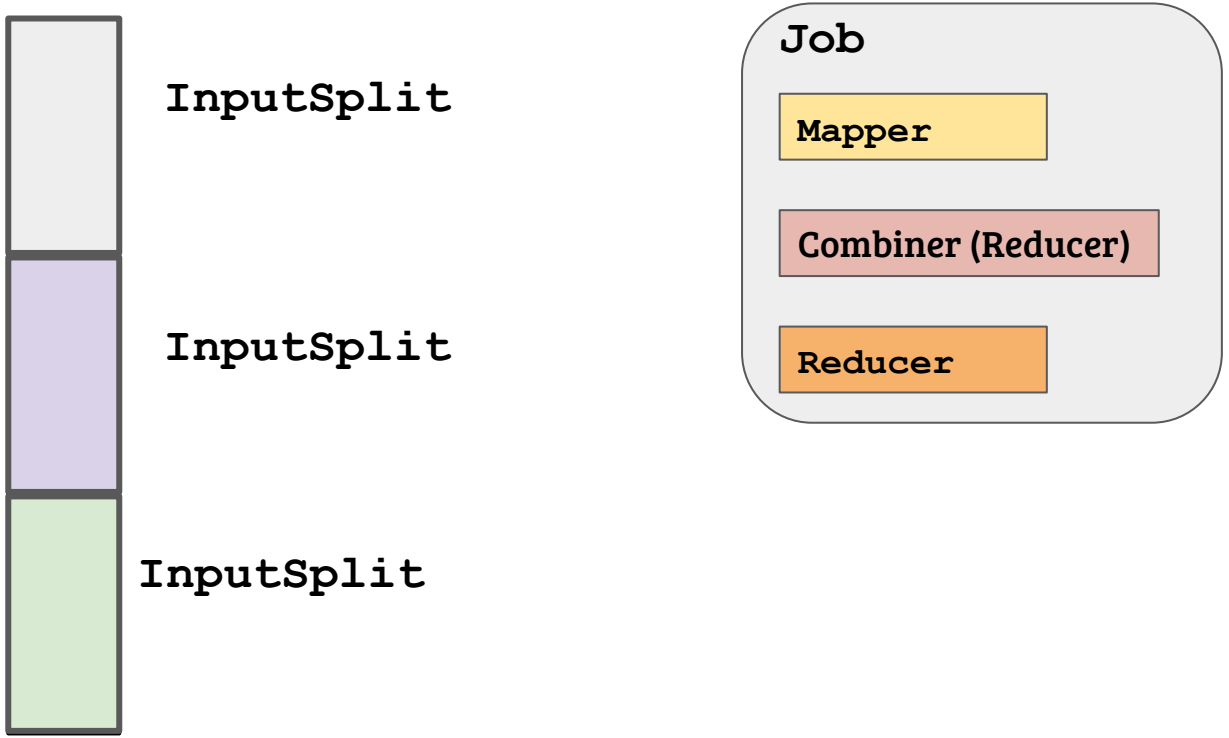
InputSplit

InputSplit

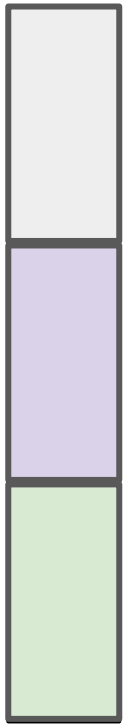
InputSplit

Input File

How it works


InputSplit
InputSplit
InputSplit
Input File
Job
Mapper
Combiner (Reducer)
Reducer

How it works

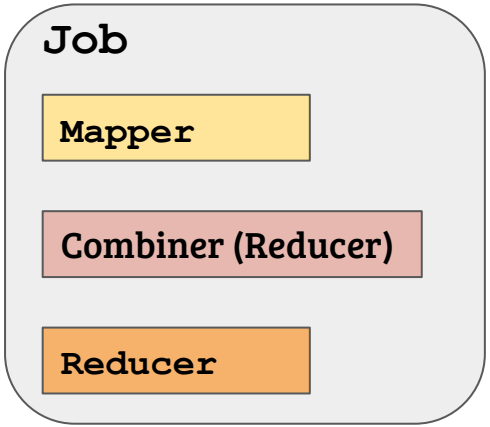


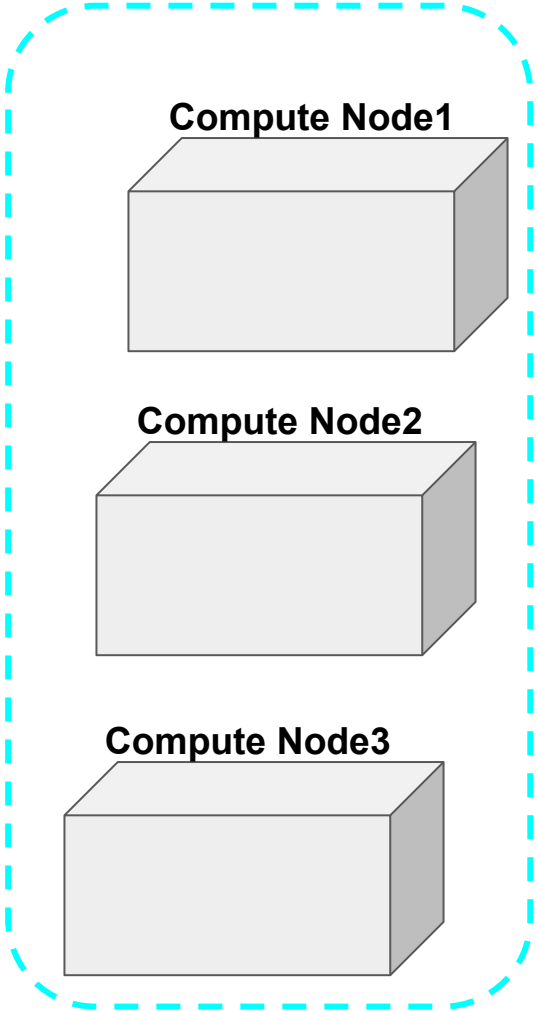
InputSplit

InputSplit

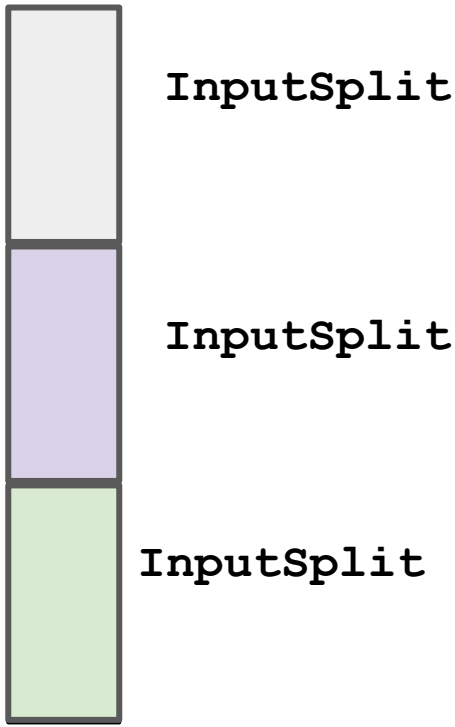
InputSplit

Input File

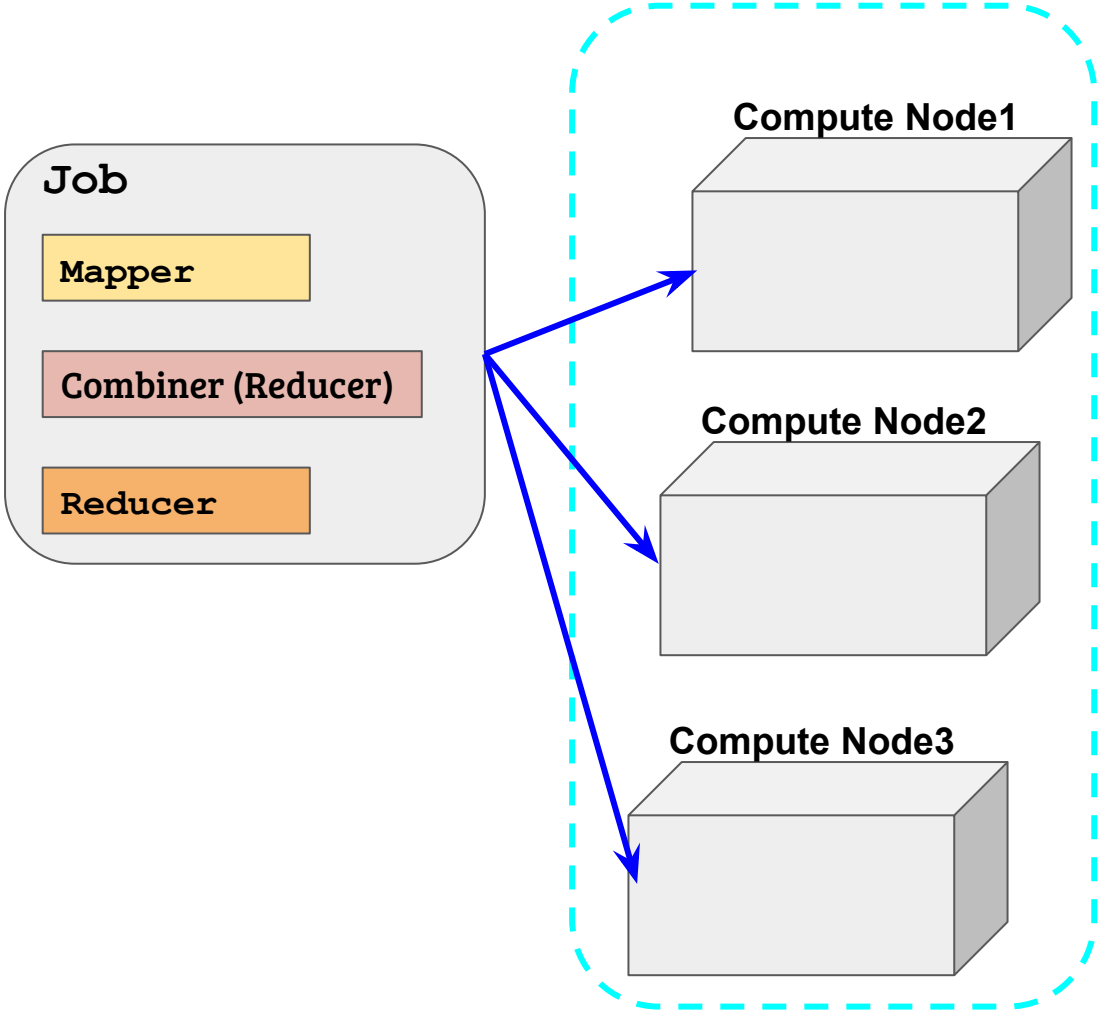




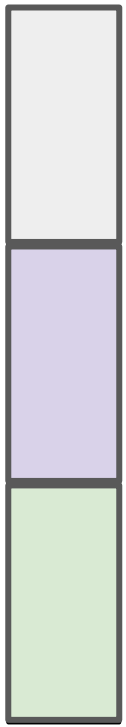
How it works



Input File



How it works

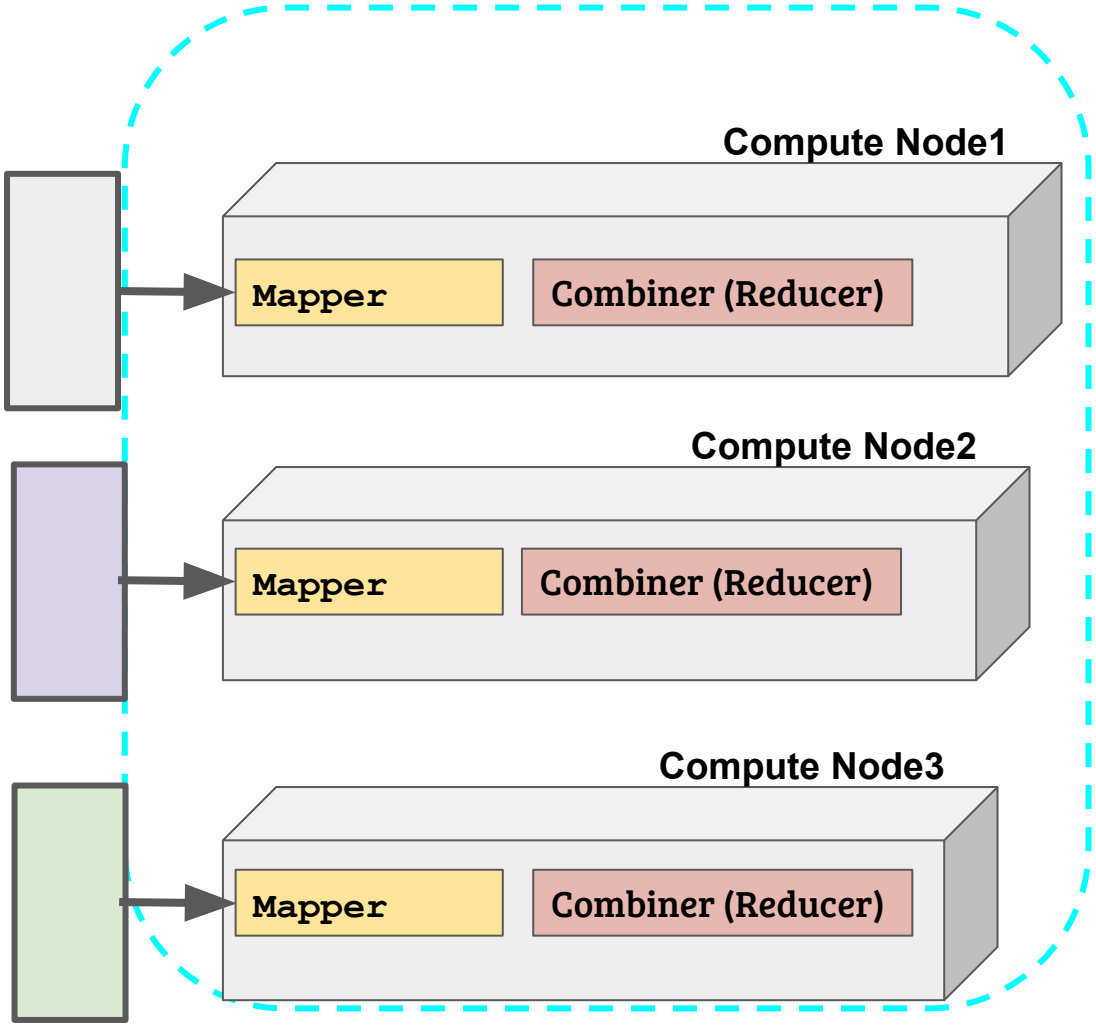


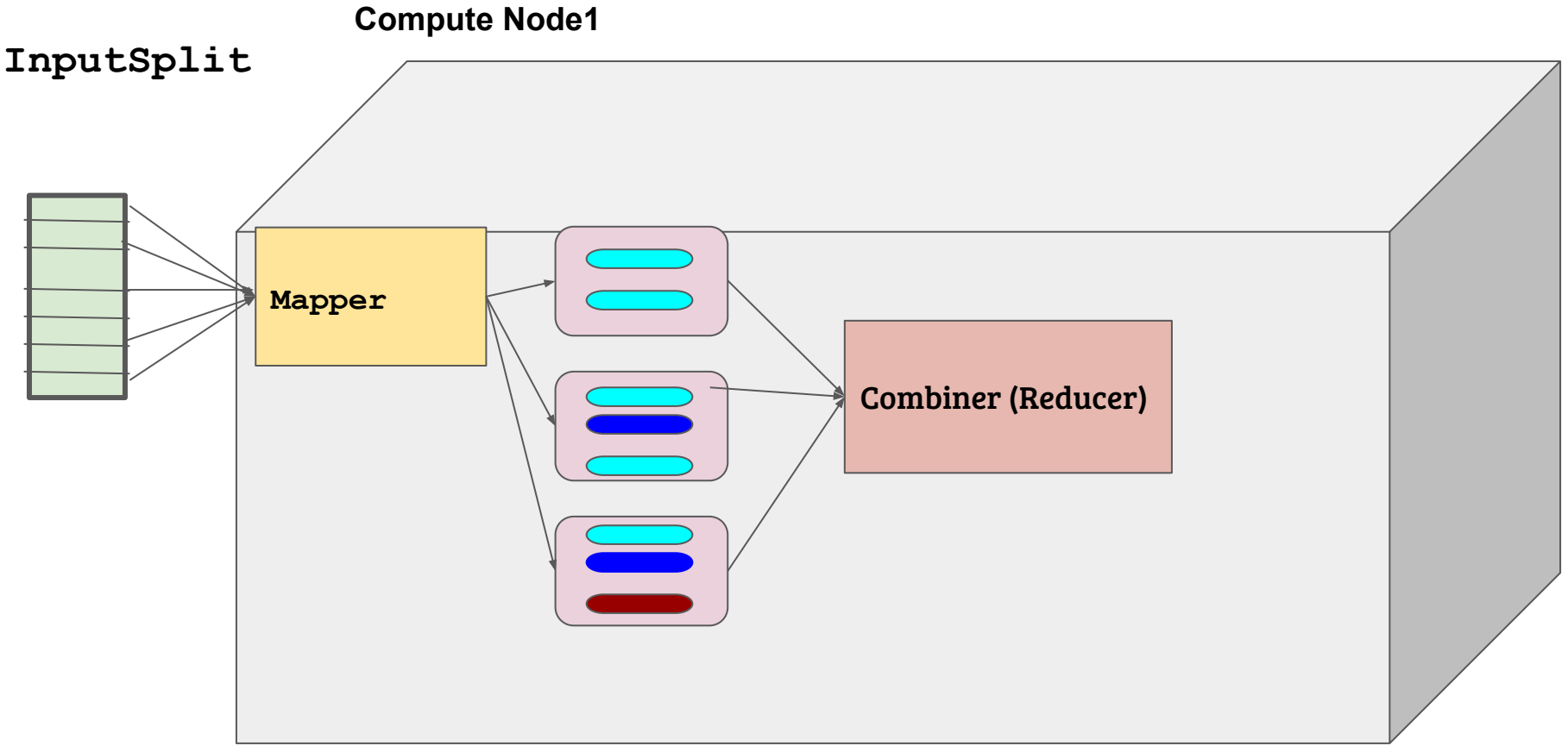
InputSplit

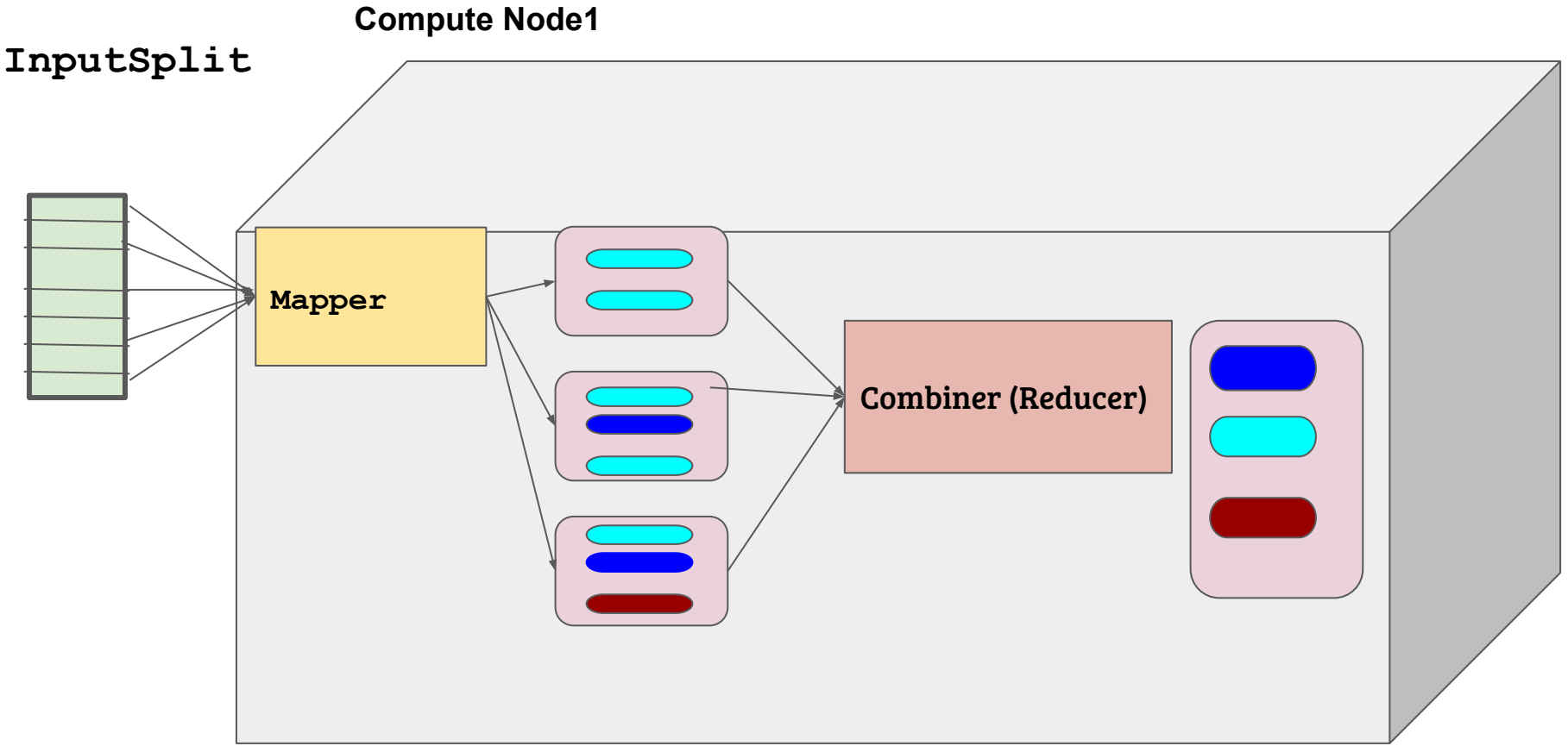
InputSplit

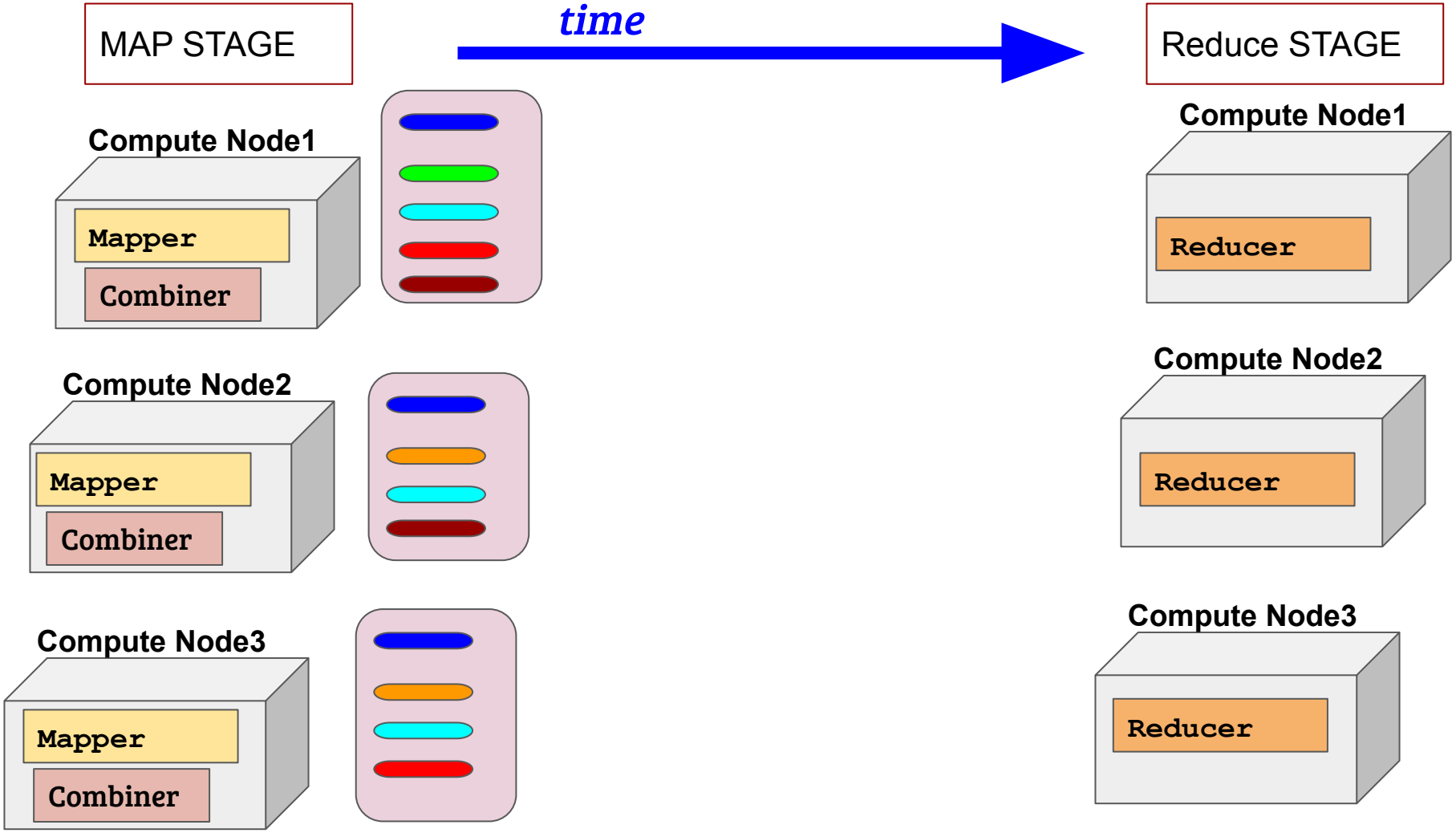
InputSplit

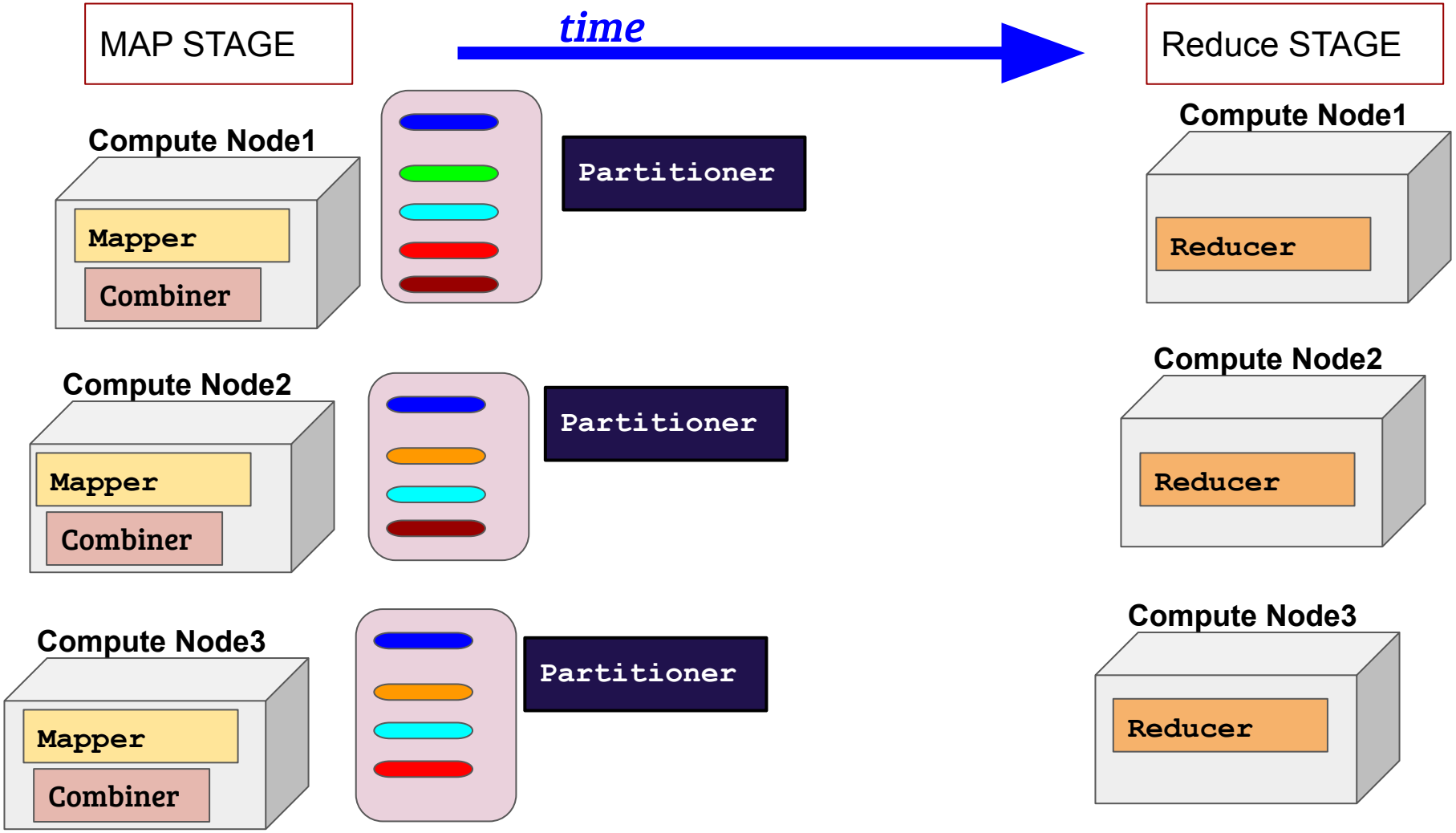
Input File

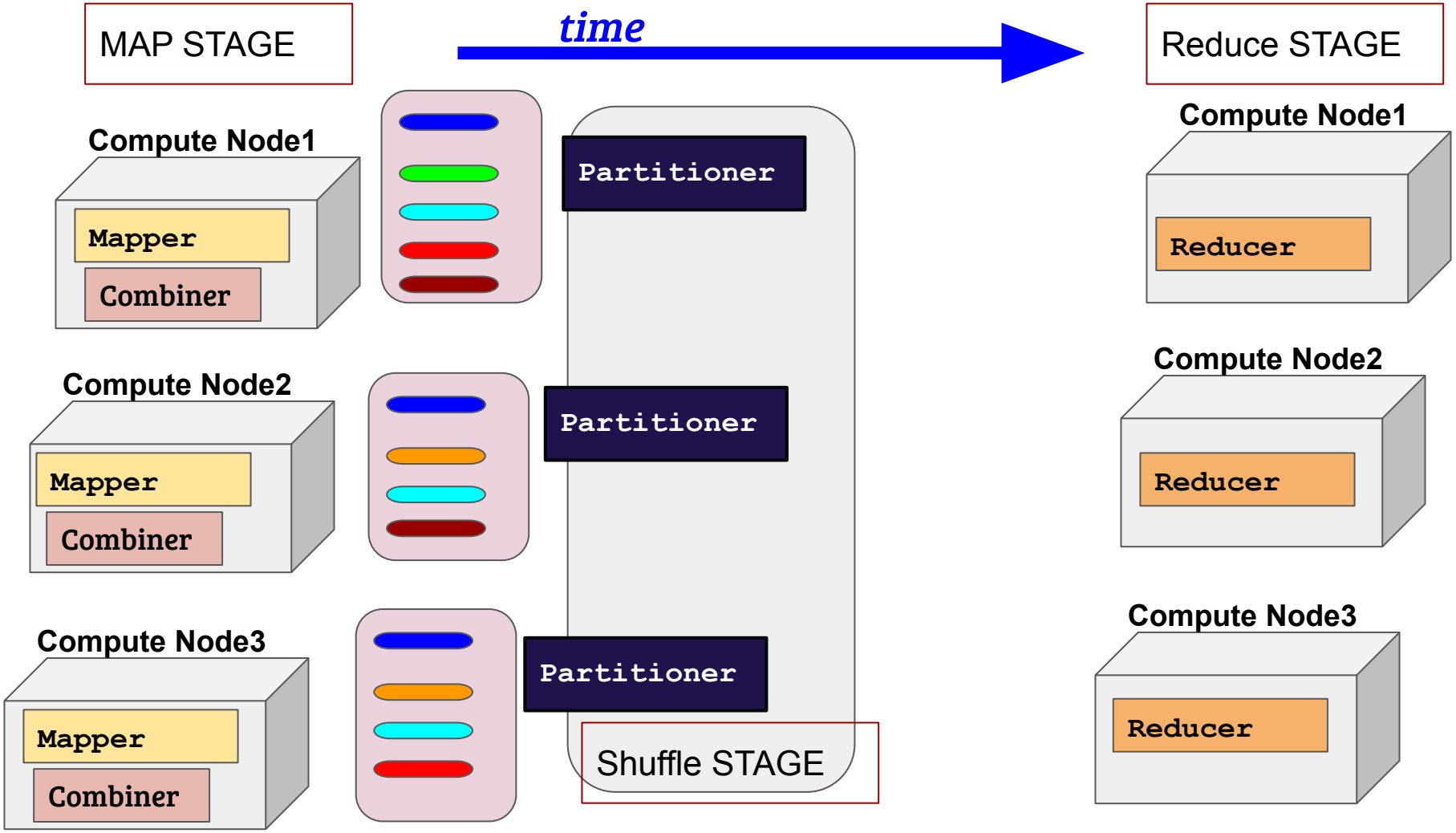


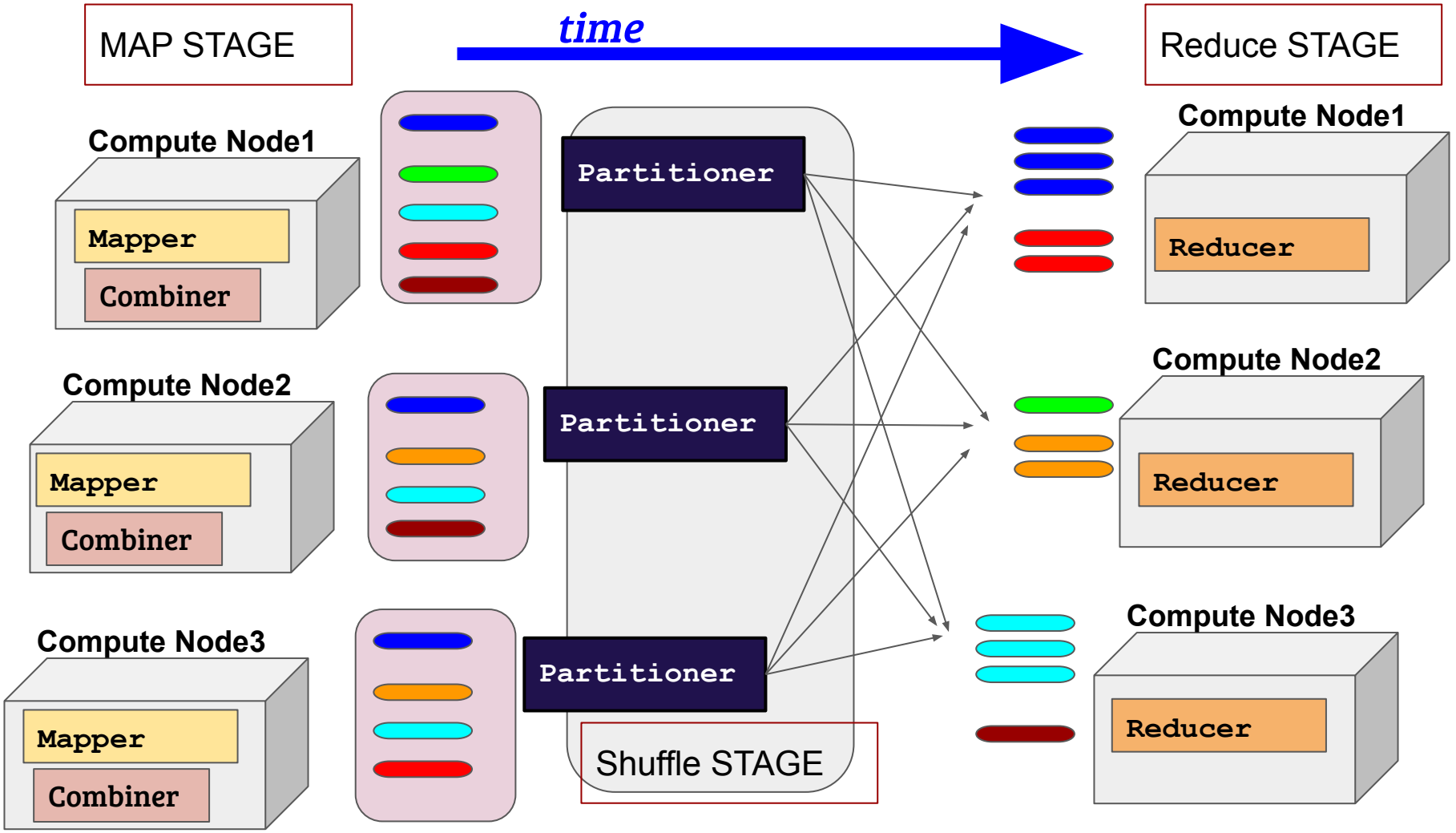

Compute Node1
InputSplit
Mapper
Combiner (Reducer)


Compute Node1
InputSplit
Mapper
Combiner (Reducer)


MAP STAGE
time
Reduce STAGE
Compute Node1
Mapper
Combiner
Compute Node2
Mapper
Combiner
Compute Node3
Mapper
Combiner
Compute Node1
Reducer
Compute Node2
Reducer
Compute Node3
Reducer


MAP STAGE
time
Reduce STAGE
Compute Node1
Mapper
Combiner
Partitioner
Compute Node1
Reducer
Compute Node2
Mapper
Combiner
Partitioner
Compute Node2
Reducer
Compute Node3
Mapper
Combiner
Partitioner
Compute Node3
Reducer


MAP STAGE
time
Reduce STAGE
Compute Node1
Mapper
Combiner
Compute Node2
Mapper
Combiner
Compute Node3
Mapper
Combiner
Partitioner
Partitioner
Partitioner
Shuffle STAGE
Compute Node1
Reducer
Compute Node2
Reducer
Compute Node3
Reducer


MAP STAGE
time
Reduce STAGE
Compute Node1
Mapper
Combiner
Compute Node2
Mapper
Combiner
Compute Node3
Mapper
Combiner
Partitioner
Partitioner
Partitioner
Shuffle STAGE
Compute Node1
Reducer
Compute Node2
Reducer
Compute Node3
Reducer

`Mapper` in a nutshell

```
protected void    setup(org.apache.hadoop.mapreduce.Mapper.Context
                  context)

protected void    map(KEYIN key, VALUEIN value,
                  org.apache.hadoop.mapreduce.Mapper.Context context)

protected void    cleanup(org.apache.hadoop.mapreduce.Mapper.Context context)
```

```
void              run(org.apache.hadoop.mapreduce.Mapper.Cont
                  ext context)
```

```
run(InputSplit s, Context c):

    setup(s,c);

    for each record in s do:
       map(record, c);
    end for;

    cleanup(s,c)
```

Run setup() once

Run map() for each record

Run cleaunp() once

Reducer in a nutshell

```
protected void      setup(org.apache.hadoop.mapreduce.Mapper.Context
                    context)

protected void      reduce(KEYIN key, Iterable<VALUEIN> value,
                    org.apache.hadoop.mapreduce.Mapper.Context context)

protected void      cleanup(org.apache.hadoop.mapreduce.Mapper.Context context)
```

```
void                run(org.apache.hadoop.mapreduce.Mapper.Cont
                    ext context)
```

Shuffle

Sort

SecondarySort

```
run(InputSplit s, Context c):

    setup(s,c);

    for each record in s do:
       map(record, c);
    end for;

    cleanup(s,c)
```

Run setup() once

Run map() for each record

Run cleaunp() once

Hadoop Java API

`org.apache.hadoop`

`org.apache.hadoop.mapreduce` — Core MapReduce classes

`org.apache.hadoop.mapreduce.lib.input`
`org.apache.hadoop.mapreduce.lib.output` — Inuput/Output parsing

`org.apache.hadoop.io` — atomic type wrappers

`org.apache.hadoop.conf` — Job configuration

`org.apache.hadoop.fs` — File system classes

Hadoop Java API

`org.apache.hadoop`	
`org.apache.hadoop.mapreduce`	Core MapReduce classes
`org.apache.hadoop.mapreduce.lib.input` `org.apache.hadoop.mapreduce.lib.output`	Inuput/Output parsing
`org.apache.hadoop.io`	atomic type wrappers
`org.apache.hadoop.conf`	Job configuration
`org.apache.hadoop.fs`	File system classes

org.apache.hadoop.mapreduce.lib.input

Single File Input Format

FileInputFormat	Generic Input File format (others extend it)
TextInputFormat	Text Input
KeyValueInputFormat	User-defined Key-Value Pairs
FixedLengthInputFormat	Fixed Length Records in input
NLineInputFormat	Controls the size of split (in terms of #lines)

org.apache.hadoop.mapreduce.lib.input

Single File Input Format

FileInputFormat	Generic Input File format (others extend it)
TextInputFormat	Text Input
KeyValueInputFormat	User-defined Key-Value Pairs
FixedLengthInputFormat	Fixed Length Records in input
NLineInputFormat	Controls the size of split (in terms of #lines)

Other Important Classes

MultipleInputs	Multiple Files as inputs to a single Mapper
FileSplits	File Partitions