Map Reduce Programming Model

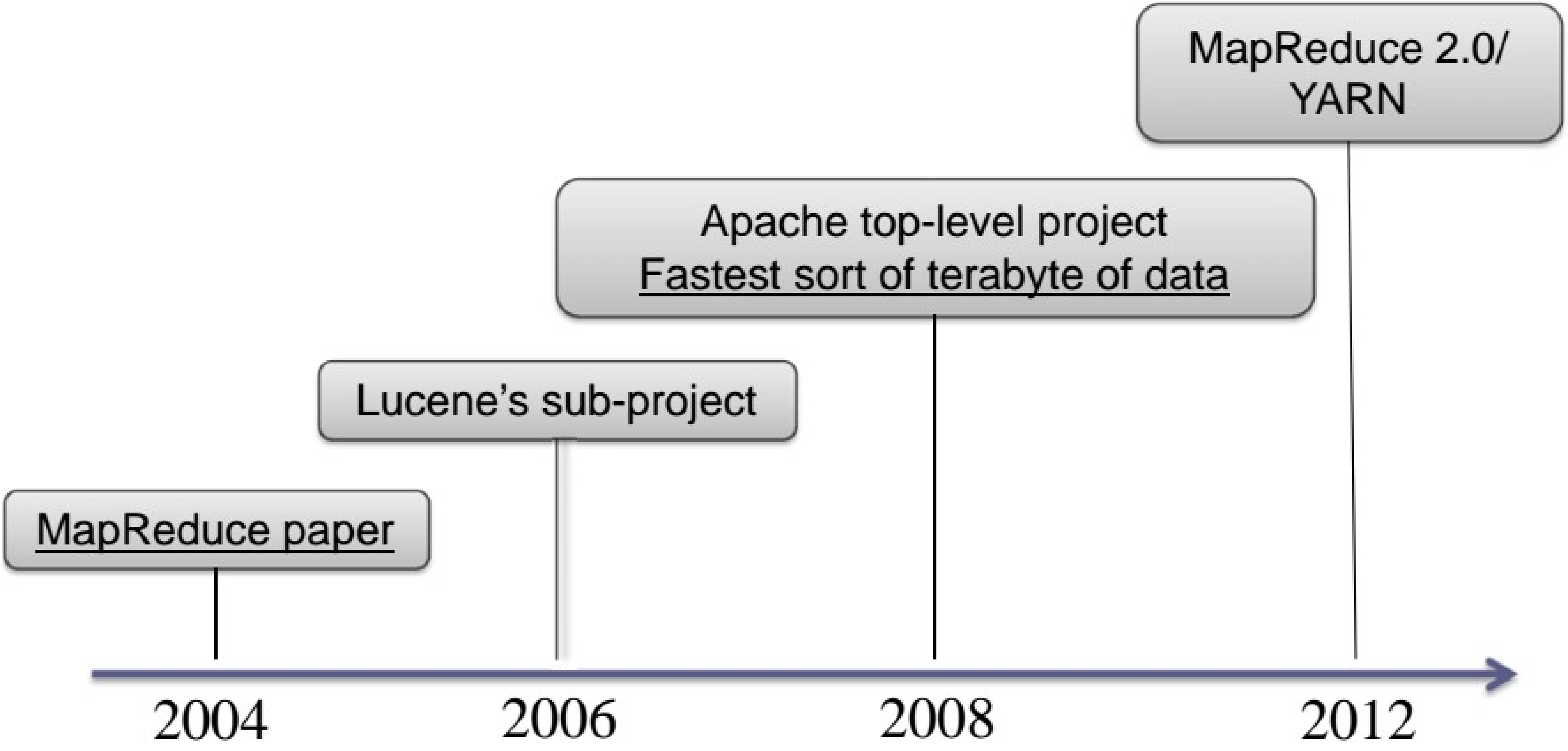
Agenda

- Introduction
- History
- Traditional Vs Map Reduce Approach
- Map Reduce Model
- Logical Data Flow
  - Weather Data set
  - Word count data set
- Advantages

Introduction

- **Model for processing large amounts of data in parallel**
  - On commodity hardware
  - Lots of nodes
- **Derived from functional programming**
  - Map and reduce functions
- **Can be implemented in multiple languages**
  - Java, C++, Ruby, Python (etc...)

Hadoop MapReduce History



Why Map Reduce?

- **Traditional approach for line oriented data set- unix script awk**
- **Challenges with this**
  - Not suitable for larger sets
  - Takes more time
  - Does not scale up with production
- **Solution**
  - Parallel processing

Why Map Reduce?

- **Parallel processing requires**
  - Dividing the work into equal-size pieces chunks- Hbase
  - Assign each chunk to a process- Mapper & Reducer
  - Combining the results from independent processes- Combiner
- **The processing capacity of a single machine is limited- N Nodes**
- **Who runs the overall job? How do we deal with failed processes?- Taken care by Hadoop Framework**

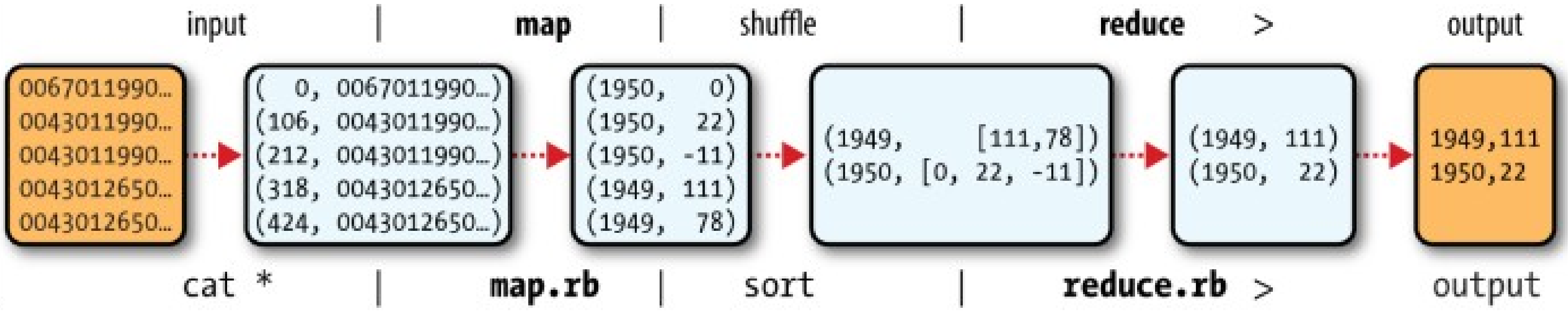
Map and Reduce

- **Breaking the processing into two phases:**
  - the map phase and the reduce phase.
  - Each phase has key-value pairs as input and output, the types chosen by the programmer.
- **The programmer also specifies two functions:**
  - The map function and the reduce function.

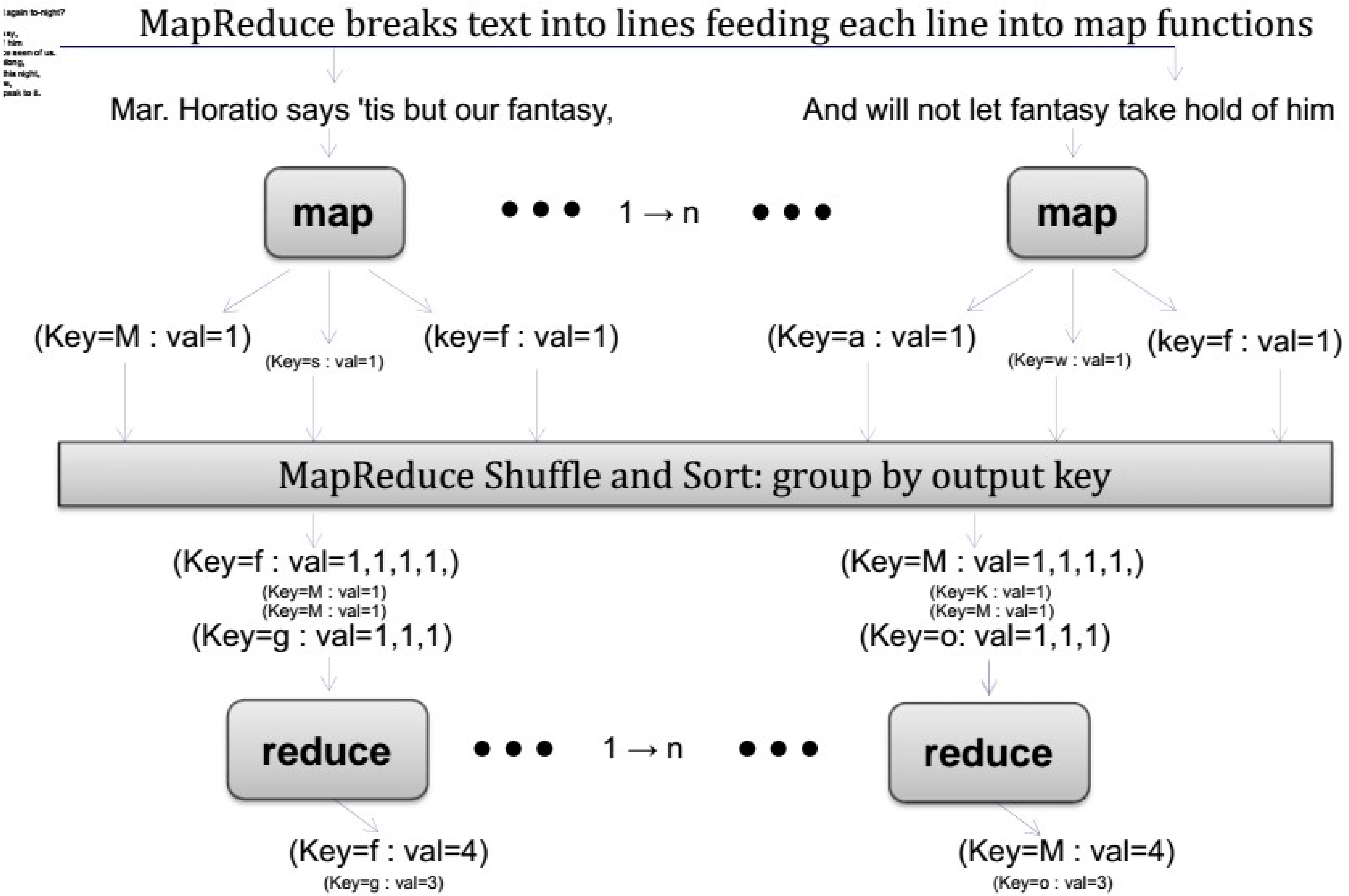
MapReduce Model

- **Imposes key-value input/output**
- **Defines map and reduce functions**
  - map: (K1,V1) → list (K2,V2)
  - reduce: (K2,list(V2)) → list (K3,V3)
- **Map function is applied to every input key-value pair**
- **Map function generates intermediate key-value pairs**
- **Intermediate key-values are sorted and grouped by key**
- **Reduce is applied to sorted and grouped intermediate key-values**
- **Reduce emits result key-values**

MapReduce logical data flow- Weather Data



MapReduce logical data flow- Word Count



MapReduce breaks text into lines feeding each line into map functions

Mar. Horatio says 'tis but our fantasy,

And will not let fantasy take hold of him

map • • • 1 → n • • • map

(Key=M : val=1)
(Key=s : val=1)
(key=f : val=1)
(Key=a : val=1)
(Key=w : val=1)
(key=f : val=1)

MapReduce Shuffle and Sort: group by output key

(Key=f : val=1,1,1,1,)
(Key=M : val=1)
(Key=M : val=1)
(Key=g : val=1,1,1)

(Key=M : val=1,1,1,1,)
(Key=K : val=1)
(Key=M : val=1)
(Key=o: val=1,1,1)

reduce • • • 1 → n • • • reduce

(Key=f : val=4)
(Key=g : val=3)

(Key=M : val=4)
(Key=o : val=3)

MapReduce Framework Advantages

- **Takes care of distributed processing and coordination**
- **Scheduling**
  - Jobs are broken down into smaller chunks called tasks.
  - These tasks are scheduled
- **Task Localization with Data**
  - Framework strives to place tasks on the nodes that host the segment of data to be processed by that specific task
  - Code is moved to where the data is

MapReduce Framework Advantages

- **Error Handling**
  - Failures are an expected behavior so tasks are automatically re-tried on other machines
- **Data Synchronization**
  - Shuffle and Sort barrier re-arranges and moves data between machines
  - Input and output are coordinated by the framework

Resources

- **Hadoop: The Definitive Guide**
  - Tom White (Author)
  - O'Reilly Media; 4th Edition.