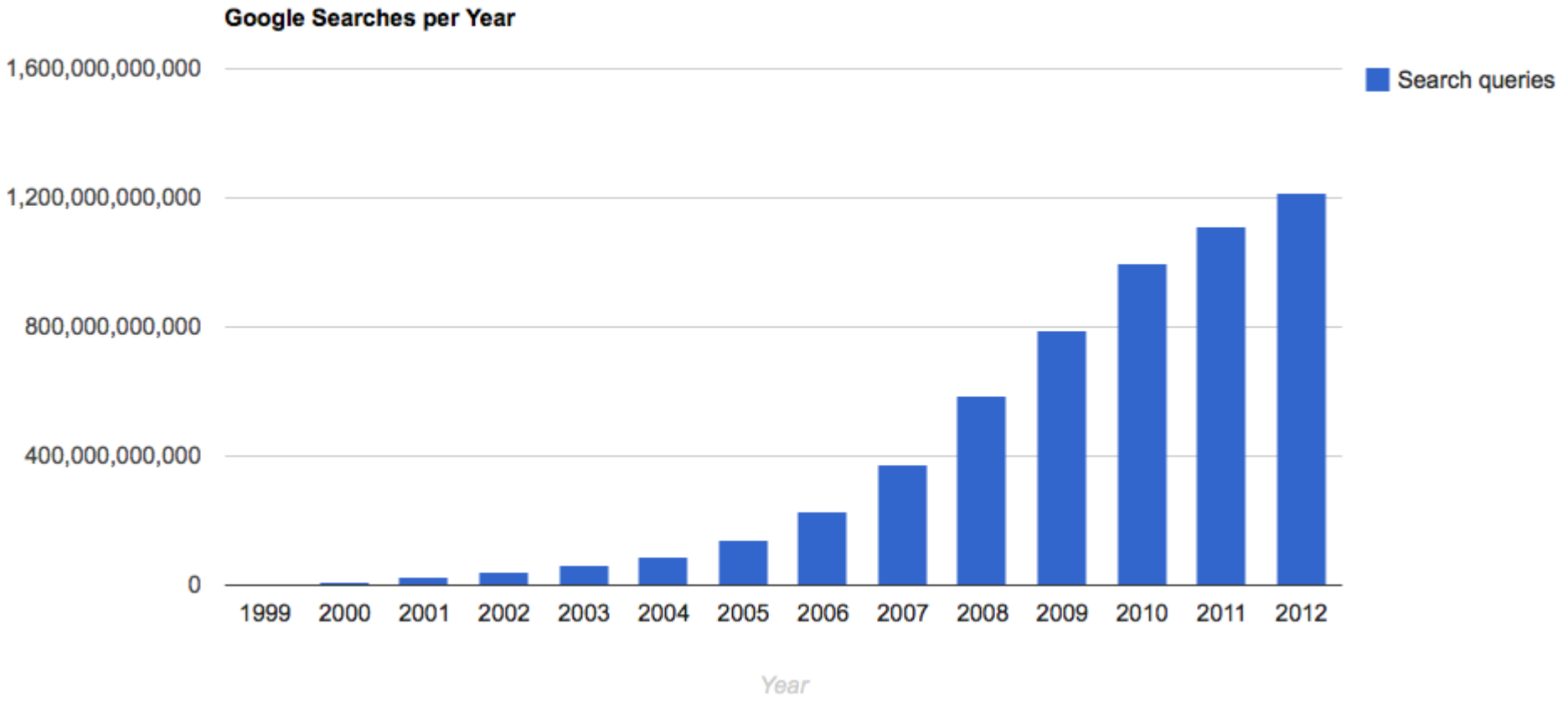
MapReduce:
Simplified Data Processing on Large Clusters
(Woohyung Han, Youngdeok Seo)

Outline

What is MapReduce?
MapReduce Architecture
MapReduce Example
Summary

Motivation


Google Searches per Year
1,600,000,000,000
1,200,000,000,000
800,000,000,000
400,000,000,000
0
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
Year
Search queries

Motivation

- A Single machine can't serve all data
  - need distributed system
- Parallel programming
  - Threading is hard!
    - how to communicate between nodes?
    - how to scale out?
    - how to handle machine failures?

Solution : MapReduce!

What is MapReduce?

- Pioneered by Google
- Parallel **programming model** meant for large clusters
  - Parallelization
  - Fault Tolerance
  - Data Distribution
  - Load Balancing
- User only implements Map() and Reduce()

What is MapReduce used for?

- At google
  - Index construction for Google Search
  - Article clustering for Google News
  - Statistical machine translation
- At Yahoo!
  - Yahoo! Search
  - Spam detection for Yahoo! Mail
- At Facebook
  - Data Mining
  - Spam Detection

Outline

~~What is MapReduce?~~

MapReduce Architecture

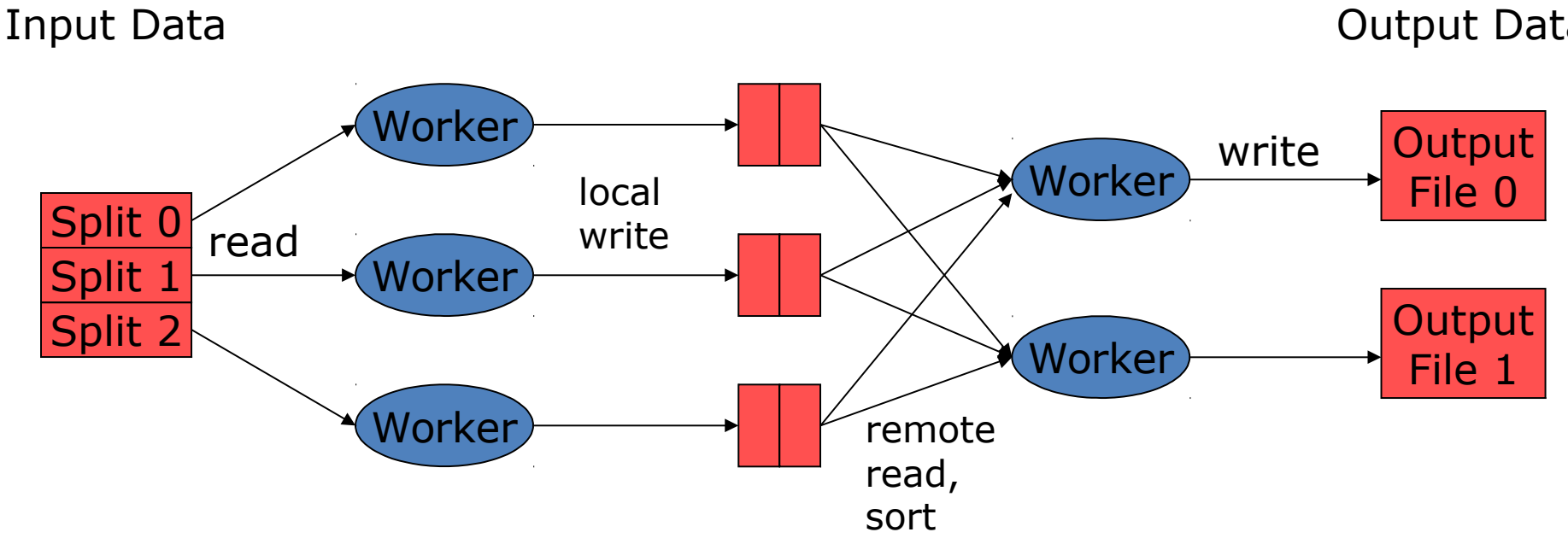
MapReduce Example

Summary

Challenges

- Node failure
  - Mean time between failures for 1node = 1years
  - Mean time between failures for 10k nodes = 53min
  - Solution : Fault-tolerance
- Network traffic
  - Solution : Push computation to the data
- Programming distributed system is hard
  - Solution : Make Data-parallel programming model that well defined.

MapReduce workflow



Map

extract something you care about from each record

Reduce

aggregate, summarize, filter, or transform

Map()

- Inputs a key/value pair
  - Key is a reference to the input value
  - Value is the data set on which to operate
- Evaluation
  - Function defined by user
  - Applies to every value in value input
    - Might need to parse input

Map Example

```
def map(key, value):
    list = []
    for x in value:
        if test:
            list.append( (key, x) )
    return list
```

Reduce()

- Accepts the Mapper output
- Aggregates value on the key
  - Merge all intermediate values associated with the same key

Reduce Example

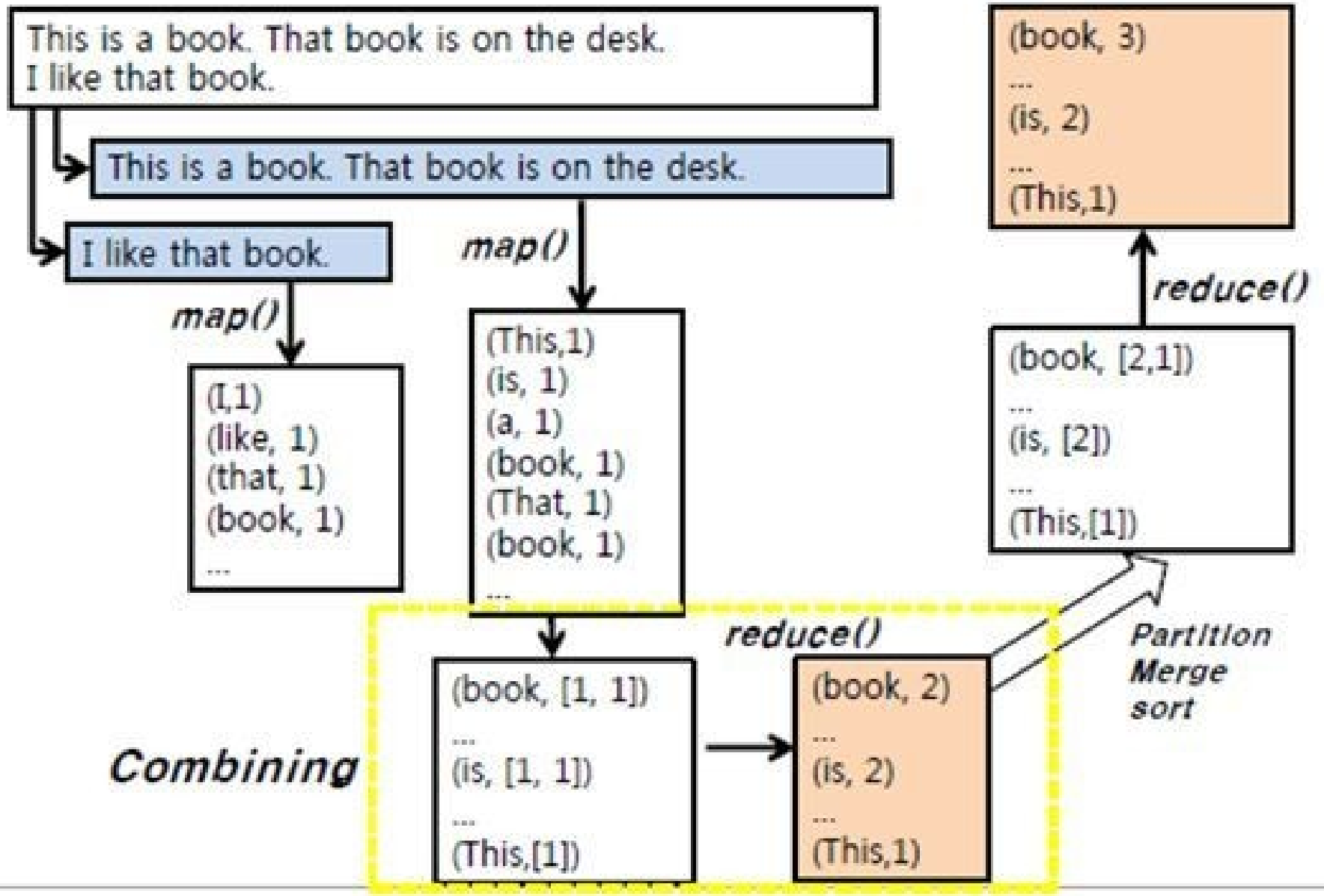
```
def reduce(key, listOfValues):
    result = 0
    for x in listOfValues:
        result += x
    return (key, result)
```

Outline

~~What is MapReduce?~~
~~MapReduce Architecture~~
MapReduce Example
Summary

MapReduce Example

- Word Count
  - Count how many words include
    - To be, or not to be, that is the question.
      - to : 2
      - be : 2
      - or : 1
      - …


This is a book. That book is on the desk.
I like that book.
This is a book. That book is on the desk.
I like that book.
map()
map()
(I,1)
(like, 1)
(that, 1)
(book, 1)
...
(This,1)
(is, 1)
(a, 1)
(book, 1)
(That, 1)
(book, 1)
...
reduce()
(book, [2,1])
...
(is, [2])
...
(This,[1])
(book, 3)
...
(is, 2)
...
(This,1)
Partition
Merge
sort
Combining
(book, [1, 1])
...
(is, [1, 1])
...
(This,[1])
reduce()
(book, 2)
...
(is, 2)
...
(This,1)

Outline

~~What is MapReduce?~~
~~MapReduce Architecture~~
~~MapReduce Example~~
Summary

Summary

- We introduced MapReduce
  - programming model for processing large scale data

- MapReduce provides :
  - a general-purpose model to simplify large-scale computation
  - Allows user to focus on the problem without worrying about details