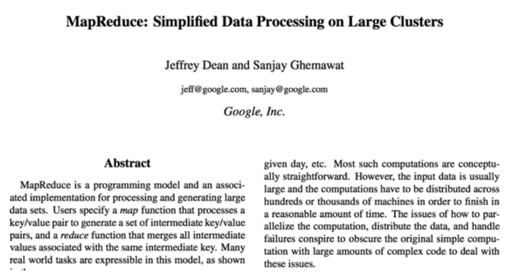
Map-Reduce and Scaling Big Data Processing

Contents

- Understanding Map Reduce
  - – Functional programming patterns applied for scalability
- Hadoop
  - – Map-reduce in Hadoop
    - Python
    - Java
  - – HDFS
  - – Yarn
  - – Pig and Hive
- Further reading

Original 2008 Google Paper



MapReduce: Simplified Data Processing on Large Clusters

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

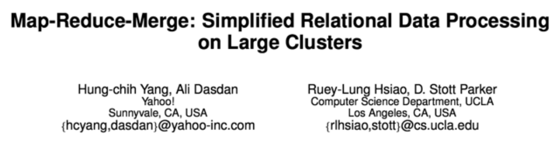
Google, Inc.

Abstract

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown

given day, etc. Most such computations are conceptually straightforward. However, the input data is usually large and the computations have to be distributed across hundreds or thousands of machines in order to finish in a reasonable amount of time. The issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.

Yahoo 2007



Map-Reduce-Merge: Simplified Relational Data Processing on Large Clusters

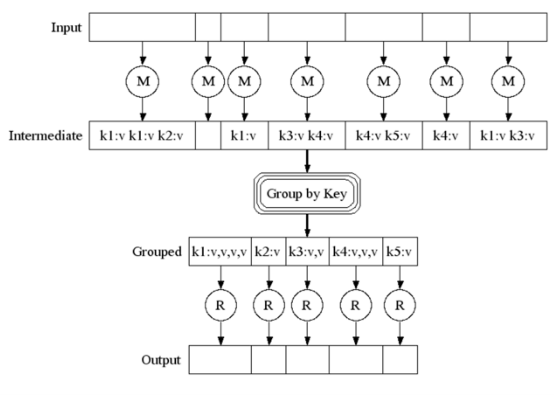
Hung-chih Yang, Ali Dasdan
Yahoo!
Sunnyvale, CA, USA
{hcyang,dasdan}@yahoo-inc.com

Ruey-Lung Hsiao, D. Stott Parker
Computer Science Department, UCLA
Los Angeles, CA, USA
{rlhsiao,stott}@cs.ucla.edu

Class Exercise

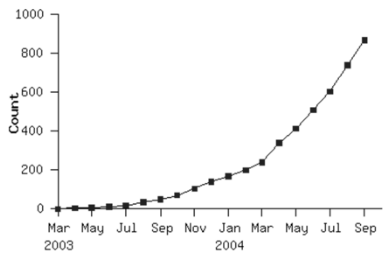
- Find a small piece of paper and write your university and day/month of birth on **one.**
- You don't need the year

Portsmouth
5 October



Google's early use of MR

Map Reduce programs in their code repository



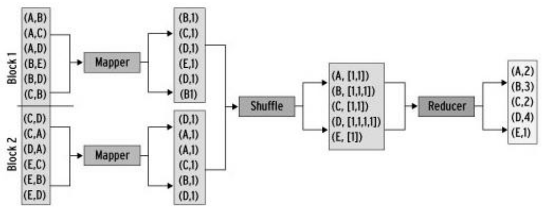
Map Reduce example in words

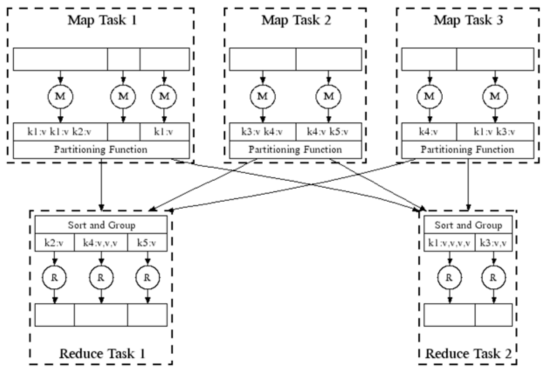
- Do a word count on 1000 books:
  - First count each book (**map** wc onto book)
  - Then **reduce** the outputs to a global wordcount across all books

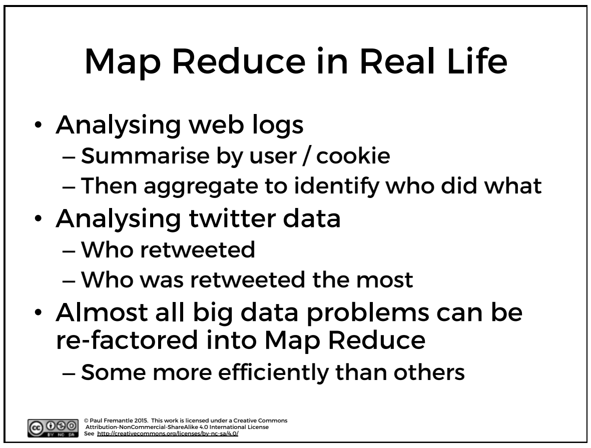
Efficiency

- Reduce phase:
  - We can theoretically process each word in parallel
- How?
  - Shuffle / Sort the results from the map phase by key (word)
  - Partition by keys
  - Parallelize the reduce phase

Map/Shuffle/Reduce







Map Reduce in Real Life

- Analysing web logs
  - Summarise by user / cookie
  - Then aggregate to identify who did what
- Analysing twitter data
  - Who retweeted
  - Who was retweeted the most
- Almost all big data problems can be re-factored into Map Reduce
  - Some more efficiently than others

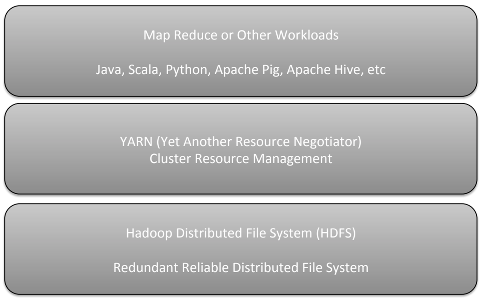
Tuning

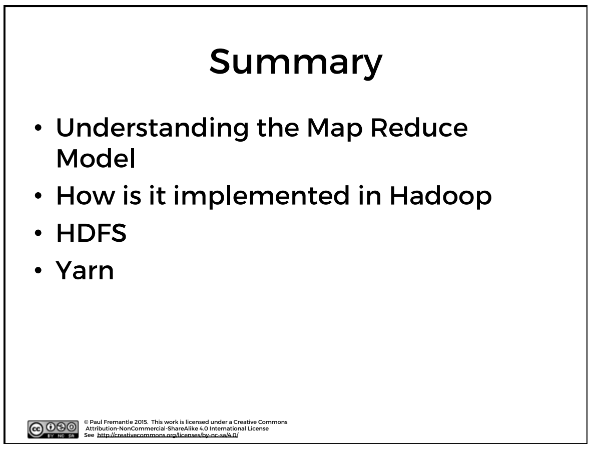
- Fault tolerance
  - Simply re-execute work that fails
- Performance:
  - Partitioning the data
  - Moving the work to near the data

Apache Hadoop

- The most famous and popular Map Reduce framework
  - Open Source
    - Written in Java, but supports other languages
  - Runs Map Reduce workloads across a cloud or cluster of machines
  - Supports a distributed filesystem to store data for these jobs
  - Provides reliability when servers in the cluster fail

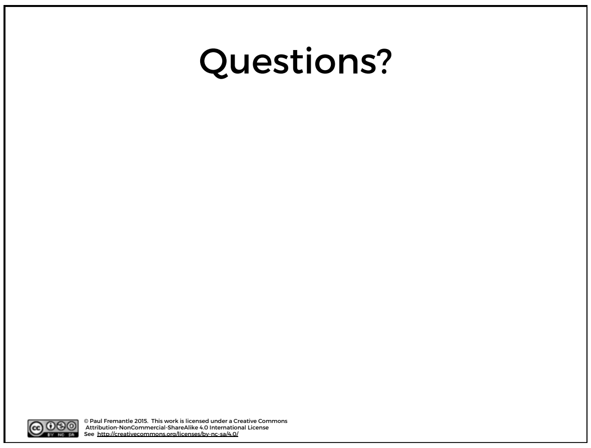
Components of Hadoop


Map Reduce or Other Workloads
Java, Scala, Python, Apache Pig, Apache Hive, etc
YARN (Yet Another Resource Negotiator)
Cluster Resource Management
Hadoop Distributed File System (HDFS)
Redundant Reliable Distributed File System



Summary

- Understanding the Map Reduce Model
- How is it implemented in Hadoop
- HDFS
- Yarn



Questions?