ISIT312 Big Data Management

MapReduce Framework

Dr Fenghui Ren

School of Computing and Information Technology -
University of Wollongong

MapReduce Framework

Outline

MapReduce

MapReduce is the most important processing framework in Hadoop

Many high-level data processing languages are abstractions of MapReduce, e.g. Pig and Hive or are heavily influenced by MapReduce concepts e.g. Spark

Historically, Hadoop version 1 supported MapReduce only

MapReduce is also a platform and language-independent programming model at the heart of most big data and NoSQL platforms

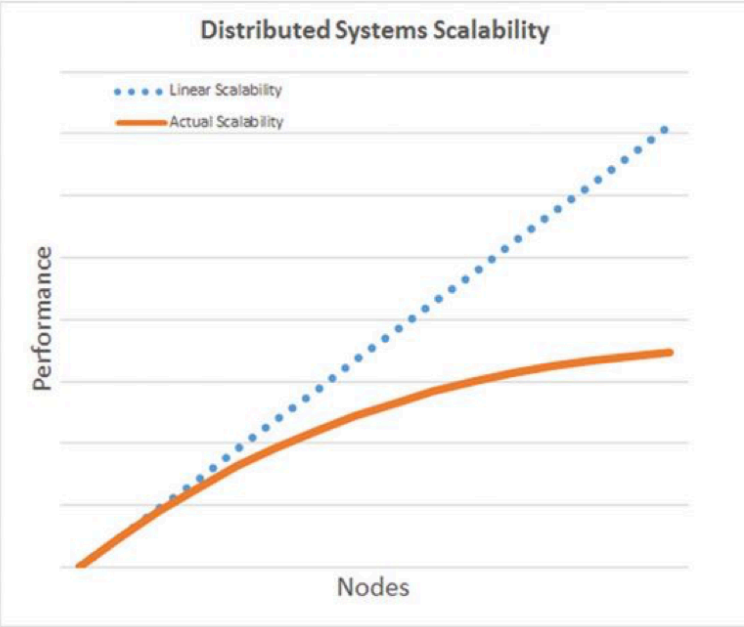
A programming model means a pattern/format in accordance to which we write our programs

The logic of a MapReduce application consists of a Map phase and a Reduce phase

MapReduce

Limitations of early distributed computing and grid computing frameworks:

- Complexity in parallel programming
- Hardware failures
- Bottlenecks in data exchange
- Scalability problem



MapReduce

The 2004 Google MapReduce white papers determined the following design goals of MapReduce

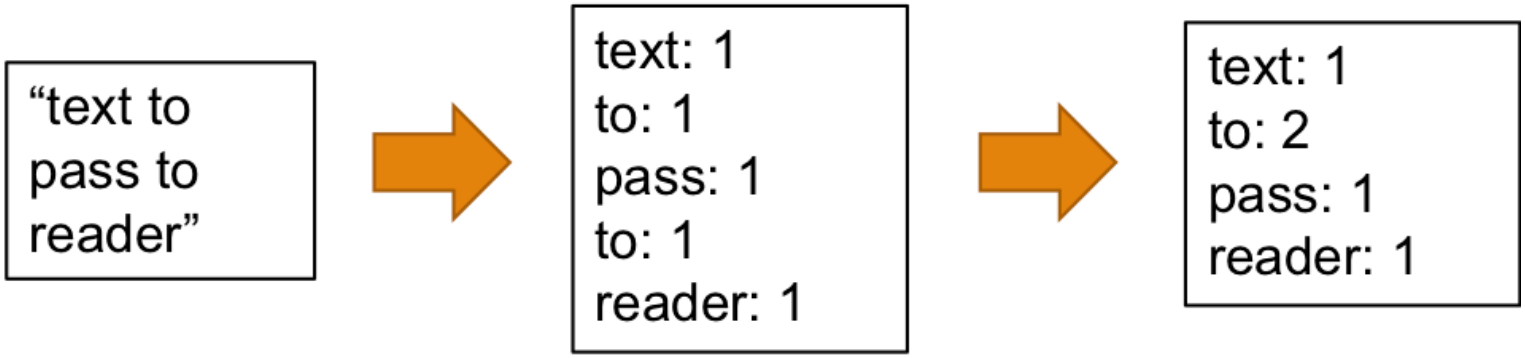
- Automatic parallelization and distribution
- Fault tolerance
- Input/output (I/O) scheduling
- Status and monitoring

MapReduce

MapReduce model uses key-value pairs for processing data

Key	Value
City	Sydney
Date	[02-03-2017, 02-04-2016]

`WordCount`: MapReduce `Hello World` example



MapReduce Framework

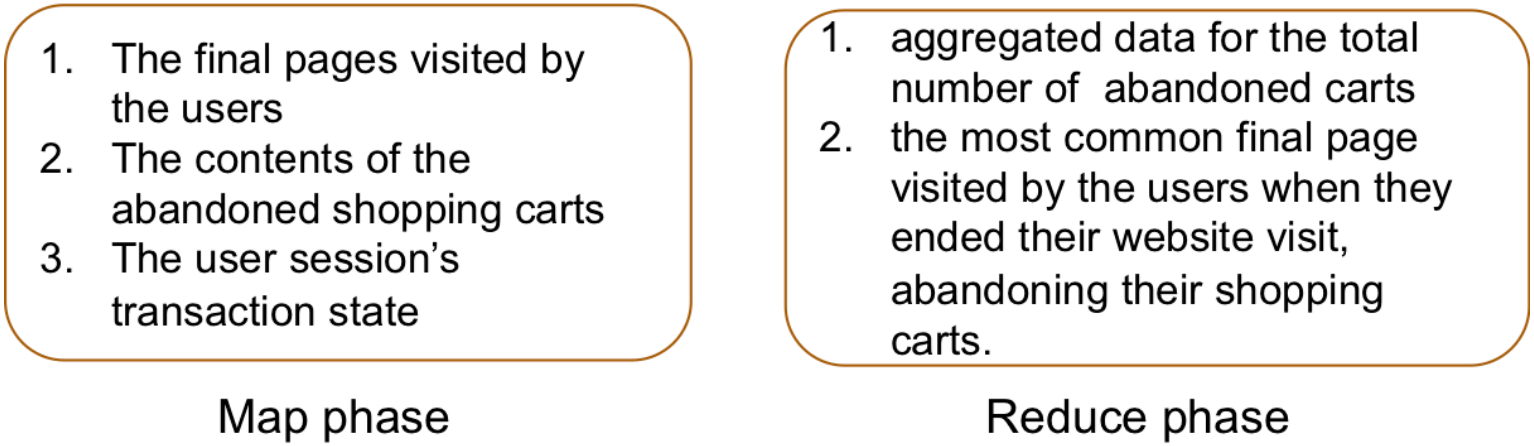
Outline

A Real-World scenario: log data analysis

In online purchasing, users sometimes abandon their shopping carts before completing the purchase

In order to improve their business, companies are usually interested to find out more about the nature of these abandoned purchases

A MapReduce job for this analysis



MapReduce Framework

Outline

MapReduce implementation in Hadoop

Hadoop MapReduce frees the users from the low-level communication and coordination of nodes and processes

Let programmers focus on the MapReduce implementation and a few configuration parameters

As the data file is usually too large to be stored in a single persistent storage device (of the commodity hardware), Hadoop handles the shipment of code to data fragments (aka, data locality)

This can dramatically reduce the overhead of network transmits

MapReduce implementation in Hadoop

Why Hadoop is useful to Big Data ?

- Cost-effective fault-tolerant storage (HDFS)
- Scalability
- Data that is ingested may be interpreted at runtime
- Low cost in storing unstructured and semi-structured data
- Fast transfer of data into storage
- Separation of programming logic and scheduling/management
- Multiple levels of distributed system abstractions: Hive, Pig, Spark
- Multi-language tooling: Java: MapReduce; SQL: Hive; data-flow: Pig; Scala, Python: Spark;

References

White T., Hadoop The Definitive Guide: Storage and analysis at Internet scale, O'Reilly, 2015 (Available through UOW library)

Vohra D., Practical Hadoop ecosystem: a definitive guide to Hadoop-related frameworks and tools, Apress, 2016 (Available through UOW library)

Aven J., Hadoop in 24 Hours, SAMS Teach Yourself, SAMS 2017

Alapati S. R., Expert Hadoop Adiministration: Managing, tuning, and securing Spark, YARN and HDFS, Addison-Wesley 2017