Hadoop

MapReduce

Agenda

- Introduction
- Hadoop
- Examples

Introduction

MapReduce programming model

MapReduce

Hadoop is an implementation of the MapReduce programming model

map (k1,v1) → list(k2,v2)

reduce (k2, list(v2)) → list(v2)

Algorithms must adhere to this model

Pseudo code

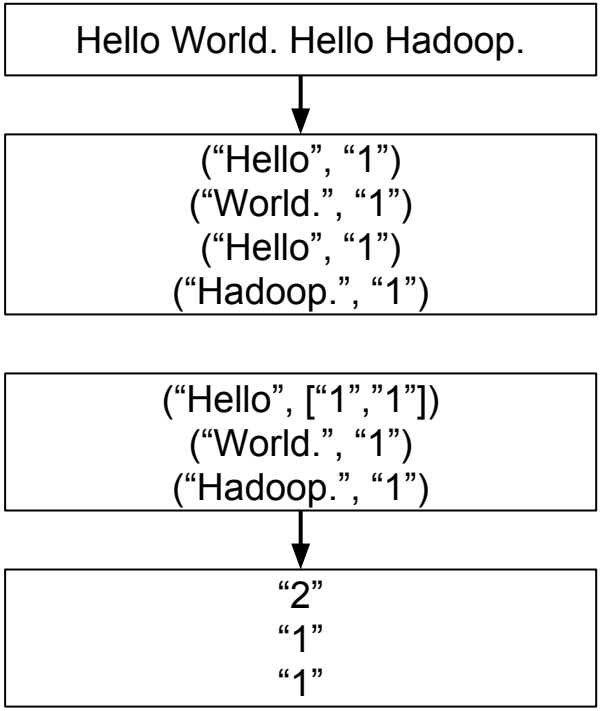
```
map(String key, String value):
    for each word w in value:
        EmitIntermediate(w, "1");

reduce(String key, Iterator values):
    int result = 0;
    for each v in values:
        result += ParseInt(v);
    Emit(AsString(result));
```

Pseudo code

```
map(String key, String value):
    for each word w in value:
        EmitIntermediate(w, "1");

reduce(String key, Iterator values):
    int result = 0;
    for each v in values:
        result += ParseInt(v);
    Emit(AsString(result));
```



Benefits

- Functions are deterministic
  - Executing map or reduce on the same input results into the same output
- Functions only depend on the input (no global state)
  - Implementations don’t need to be aware of each other

Function implementations can be distributed over large clusters of commodity machines

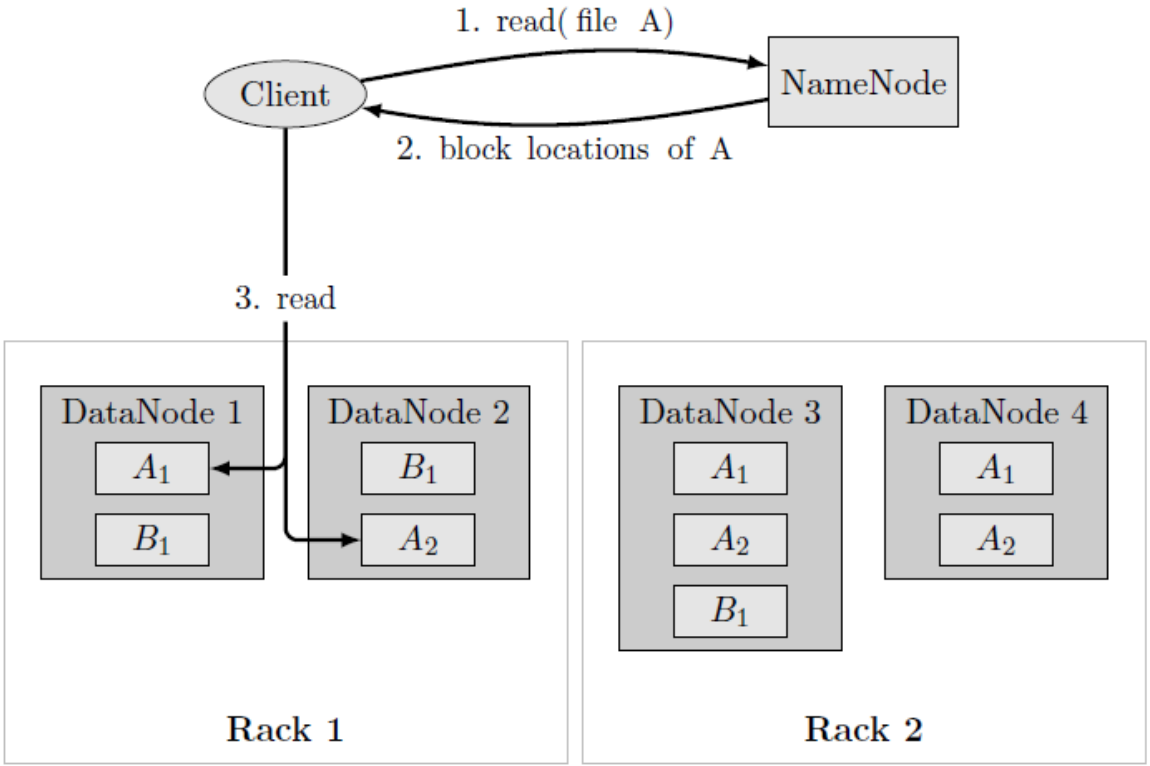
Hadoop

MapReduce and HDFS

Hadoop

- Created by Doug Cutting
  - Inspired by Google
  - Named after the elephant pet of his son
- Apache top level project
  - Hadoop Common
  - **Hadoop Distributed File System (HDFS)**
  - **Hadoop MapReduce**
  - Hadoop YARN

HDFS

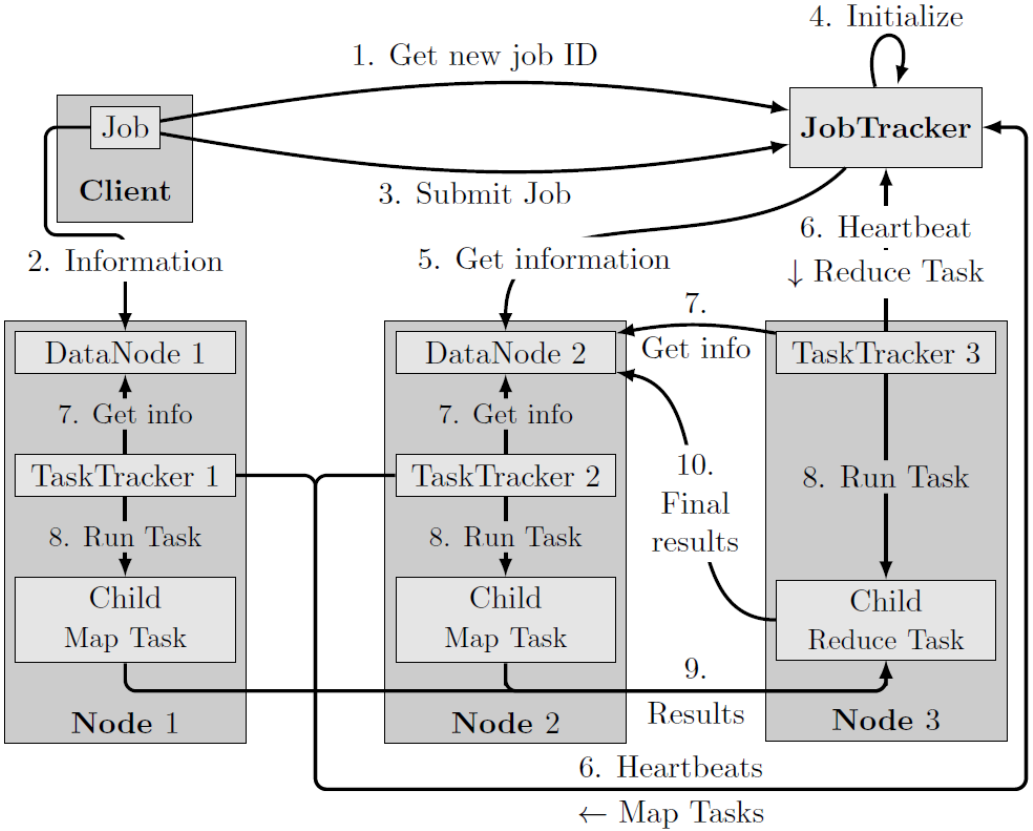


- Files split in blocks
  - 64 MB
- Fault tolerance
  - Master/Slave
  - Heart beats
  - Block replication
- Example
  - 64 MB < File A <= 128 MB
  - File B <= 64 MB

HDFS

- POSIX style
  - hadoop fs -mkdir <path>
  - hadoop fs -ls <path>
  - hadoop fs -du
  - hadoop fs -put <local> <remote>
  - hadoop fs -get <remote>
  - hadoop fs -chmod …
  - hadoop fs -chown ...

MapReduce (old)



- Fault tolerance
  - Master/Slave
  - Heartbeat
  - Repetition
- Job
  - Mapper
  - Reducer
  - Configuration
- Data locality
  - Computation on nearest data node

Hadoop

Examples

Wordcount

Eclipse

Sort

- Shuffle phase between map and reduce
  - Transfer files from Mappers to Reducers
  - Merge values with same key
    - (“Hello”,”1”), (“Hello”,”1”) → (“Hello”, [“1”,”1”])
  - Sort merged key value pairs
    - (“Foo”, [“1”,”1”]), (“Bar”, [“1”,”1”]) → (“Bar”, [“1”,”1”]), (“Foo”, [“1”,”1”])
- Sort words in a text file
  - Word is delimited by white spaces

Wordcount

Thank you
Any questions?

Literature

- MapReduce: Simplified Data Processing on Large Clusters (Jeffrey Dean and Sanjay Ghemawat)
  - Paper from Google
  - http://research.google.com/archive/mapreduce.html
- Data-Intensive Text Processing with MapReduce
  - Design recommendations for MapReduce algorithms and much more
  - http://lintool.github.com/MapReduceAlgorithms/MapReduce-book-final.pdf