CMPSC 441
Distributed Systems

Dr. Aravind Mohan
Allegheny College

March 29, 2018
Lecture 11 - Introduction to Map Reduce
What is Map Reduce?

- MapReduce is a software framework / simple programming model based on Java language.
- Used for writing applications to process huge amounts of data, in parallel, on large clusters of commodity hardware.
- It does it in a reliable and fault-tolerant manner.
- It abstracts the complexity of distributed processing and parallelism from developer.
- Computation/ processing is taken to nodes where data resides in the cluster. This is called ’Data Locality’.
Map reduce works by breaking the processing into three phases:
1. Map phase where multiple Map tasks are created and sent to nodes in the cluster where the data resides.
2. Data is processed in parallel in the cluster.
3. Shuffle and Sort phase where intermediate data from all the map tasks is shuffled and sorted.
4. Reduce phase where results from all the Maps are consolidated, further processed if required and final results are generated.
Map Reduce Stages

- In Map Reduce, data is processed in Key-Value pairs

**Mapper**
1. Converts input key/value pairs to a set of intermediate key/value pairs
2. Each Map task usually operates on single block (input split) in HDFS

**Shuffle and Sort**
1. Shuffle and Sorts intermediate data from all mappers. Data is sorted by Key
2. This stage happens when all the Map tasks are completed and before Reduce tasks start

**Reducer**
1. Operates on shuffled/sorted intermediate data i.e. Map tasks output
2. Consolidates and Produces final output
Map Reduce Execution Flow [1]

- Input splits are identified from data blocks for a file to be processed
- Framework tries to send the Map tasks to slave nodes where actual data split resides
- Number of Map tasks created by framework is equal to number of input splits
Map Reduce Execution Flow [2]

- Each Map task works on a ‘Input Split’ of data
- Mappers output intermediate data
- Shuffle process sort and merge the intermediate data from all the Mappers
- Reduce takes output from shuffle process and produces final results
Map Reduce Framework
Execution of Map Reduce Jobs

1. Copy Input files
2. Submit Job
3. Get Input files info
4. Create Splits
5. Upload Job Information
6. Submit Job

User

Client

DFS

Job.xml. Job.jar.

Input Files

Job Tracker
Execution of Map Reduce Jobs

1. Client
2. Submit Job
3. Read Job Files
4. Job Tracker
5. Initialize Job
6. Create maps and reduces
7. Job Queue
8. Reduces
9. Maps
10. As Many Maps as splits
11. Input Splits
Execution of Map Reduce Jobs

- Job Tracker
  - Job Queue
    - Picks tasks (Data Local if possible)
  - Heart beat
  - Task tracker- H1
    - Heart beat
    - Assign Tasks
    - Heart beat
  - Task tracker- H3
    - Heart beat
    - Heart beat
  - Task tracker- H3
  - Task tracker- H4
    - Heart beat
Map Reduce Example

Input File
- Jack Bill Joe
- Jack Don Bill
- Don Don Joe
- Jack Bill Joe
- Jack Don Bill
- Don Don Joe

Input Splits
- K1,V1
  - 0, Jack Bill Joe
  - 14, Jack Don Bill
- K2,V2
  - 16, Don Don Joe
  - 27, Jack Bill Joe
- K3,V3
  - 41, Jack Don Bill
  - 54, Don Don Joe

Maps
- List(K2,V2)
  - Jack, 1
  - Bill, 1
  - Joe, 1
  - Jack, 1
  - Don, 1
  - Bill, 1

Shuffling
- K2,List(V2)

Reducers
- List(K3,V3)
  - Bill, 4
  - Don, 6
  - Jack, 4
  - Joe, 4

Final Result
- Bill, 4
- Don, 6
- Jack, 4
- Joe, 4
Paper Reading

Questions