Map Reduce: Simplified Data Processing on Large Clusters

Kartikeya Upasani (kuu2101)

Paper Review, 24 October 2016

1 Motivation

Several tasks that operate on large dataset require performing common operations in a distributed manner using unreliable hardware.

2 Goal

Developing an API for distributed tasks (map and reduce functions) that take care of load balancing, error handling, and components failures.

3 Approach

The API provides map and reduce functions that abstract the underlying process of cluster computing and generate result in key-value format from input data. The map function splits the data key range into R parts that are stored in a distributed fashion (using GFS). A master node is spawned that maintains the locations of parts of data, which then spawns several worker nodes. The worker nodes perform the task on the assigned range of data and store the output in GFS, notifying the master node of the positions of keys in the output produced by it. Redundancy is used to ensure that a slow worker does not affect throughput, repetition and atomicity of tasks is used to handle node failure. MapReduce is also cognizant of locality of data, and tries to assign computation tasks to nodes that locally contain the required chunk of data, thereby reducing network bandwidth requirement.

4 Results

Experiments were performed for distributed grep (only read and compute) and sorting (read, compute, and write) operations. The read/write speeds and network bandwidth consumption were all favourably reported in the results.

5 Conclusion

MapReduce is instrumental in abstracting the process of cluster computing, and does not require a developer to have knowledge about parallel computing. It also deals with paraphernalia such as handling failures and redoing incomplete tasks.

6 Comments

An additional notable observation is that the lines of code needed to perform common cluster tasks were greatly reduced. MapReduce also made it possible to separate tasks that were earlier clubbed together for efficiency. This simplifies code and makes modifying it easier.