MapReduce: Simplified Data Processing on Large Clusters

OSDI'04

201924169 Ju-Won Park

May 29, 2019

Problem statement

- Programmers want the ability to process large amounts of data in a fast and efficient way without worrying about all the details involved with distributed systems. This programming model supplies that.

Solution approach

- Authors design MapReduce that is a programming model and an associated implementation for processing and generating large datasets. MapReduce contains Map and Reduce function.

Strong points

- A new and very simple abstraction for processing any size of data that fits into this programming model. Many data processing jobs do fit into this category, or can be modeled to.
- Fault tolerance is a first order consideration. The master node monitors workers and restarts chunks of the work if necessary.
- Locality is taken into consideration when scheduling nodes, and performance enhancements (making sure the last few stragglers dont bog down the job).

Weak points / Limitations

- MapReduce is a successful abstraction of many real world applications, but it is not suitable to be implemented in all environments.

Questions

- Map nodes need to write the intermediate data to disk and the Reduce nodes need to read them from disks. So Im wondering why MR could provide better performance over this simple method.
- If the reason is MR could use a parallel I/O, why do MR still need to do the partition?

New ideas / Comments

- We can just use the disk arrays to enhance the I/O performance. I guess only when there are actually a very large number of nodes and a very big input files can the MR provide a much better performance.
- This paper provides a good overview of the map-reduce functionality but it is not very technical. Map-reduce is a good way to parallelize the computation for certain problems. These problems tend to not require perfect answers. If the number of people to visit a URL is off by a few counts, there will not be any repercussions. That is why map-reduce may be appropriate in some instances while use of a database is appropriate in others.