

MapReduce

Sirui Tan st2957

MapReduce is a programming model as well as an associated implementation for processing and generating large datasets that are amenable to a broad variety of real-world tasks. Motivated by the large amount of data google is facing daily, and inspired by the map and reduce primitives of functional languages, mapreduce is developed and contributed as a simple yet powerful interface that enables automatic parallelization and distribution of large-scale computations.

Mapreduce takes input data in the form of key/value pairs and generate outputs as key/value pairs also. The map phase takes input pairs and produce a set of intermediate key/value pairs. The reduce phase, on the other hand, accepts intermediate key/value pairs and merge them with respect to values to form a possibly smaller set of values.

In Google, MapReduce usually targets to large clusters of commodity PCs connected together with switched Gigabit Ethernet. When being implemented, input is partitioned into multiple splits, which can be processed in parallel by different machines during map phase. Reduce is also distributed by partitioning the intermediate key space into pieces. All these operations are controlled by a special node in the cluster called master node, which is supposed to control all other nodes, worker nodes, by assigning map reduce tasks.

MapReduce library is designed to tolerate machine failure gracefully. The master node check if workers are alive by pinging periodically. Whenever a node fails, failed worker is reset and tasks are re-executed. The semantics of MapReduce ensures deterministic functions would produce the same result even facing failures.

MapReduce takes location information into account so that most input data are read locally. MapReduce tasks are granulated by making the number of map and reduce splits much larger than that of cluster nodes. MapReduce also applies backup tasks to alleviate the problem of stragglers.

Experiments concerning MapReduce include linux-like grep program for finding rare pattern in 10^{10} 100-byte records, sorting one terabyte of data, and many others like large-scale machine learning problems. Most of the experiments shows promising outcome which bolsters MapReduce's efficiency. One of the most successful implementation of MapReduce is rewriting the production indexing system that produces the data structures used for the Google Web search service.