Hadoop – MapReduce

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The Algorithm

- Generally MapReduce paradigm is based on sending the computer to where the data resides!
- MapReduce program executes in three stages, namely map stage, shuffle stage, and reduce stage.
- During a MapReduce job, Hadoop sends the Map and Reduce tasks to the appropriate servers in the cluster.
- The framework manages all the details of data-passing such as issuing tasks, verifying task completion, and copying data around the cluster between the nodes.
- Most of the computing takes place on nodes with data on local disks that reduces the network traffic.
- After completion of the given tasks, the cluster collects and reduces the data to form an appropriate result, and sends it back to the Hadoop server.

Inputs and Outputs (Java Perspective)

The MapReduce framework operates on <key, value> pairs, that is, the framework views the input to the job as a set of <key, value> pairs and produces a set of <key, value> pairs as the output of the job, conceivably of different types.

Python MapReduce Code

The “trick” behind the following Python code is that we will use the Hadoop Streaming API (see also the corresponding wiki entry) for helping us passing data between our Map and Reduce code via STDIN (standard input) and STDOUT (standard output). We will simply use Python’s sys.stdin to read input data and print our own output to sys.stdout. That’s all we need to do because Hadoop Streaming will take care of everything else!

Execution

Given a MapReduce program with user-defined map and reduce functions, how is that program executed using the MapReduce framework?

To begin, the framework partitions the input data set into M pieces or splits to be processed by different machines. After splitting the input data, multiple copies of the MapReduce program are started on a cluster of worker machines and a single master machine. The master is responsible for assigning work to the worker machines. To start the computation, the master assigns one of the M map tasks to each idle worker.

When a worker receives a map task, it reads the contents of the corresponding input split, parsing the key-value pairs out of the input split and outputting intermediate key-value pairs by invoking the user-defined map function. The intermediate results are kept in memory and periodically flushed to local disk. The location of these results are passed back to the master machine, who forwards these locations to worker machines to run the reduce function.

The MapReduce framework handles errors by restarting worker machines. Since the results of a map worker are stored on local disk, if that machine goes down the results are lost — the master is responsible for scheduling that piece of work on a new worker machine.
