Introduction to MapReduce

**Outline:**

1, Target and introduction

2, Introduction to MapReduce, How does it work

3, Reference

**1, Introduce to MapReduce**

MapReduce is first mentioned by Google. Inc in 2004. It is an abstract model of method of problem solving in the real world.

It also is a programming model that let users to deal with large scale and complex data set. Program written in this way can be easily scaled. Once it was done, it can be quite convenient to put on more machines in a cluster.

**2, How does it work**

MapReduce basically includes two steps, Map and Reduce.

In simple term, the Map will deal and transform the input data into the intermediate dataset, then the Reduce function will take over and retrieve as the program required.

The whole process in MapReduce can be seen in the following table:

|  |  |  |
| --- | --- | --- |
|  | Input | Output |
| Map | <k1, v1> | <k2, list(v2)> |
| Reduce | <k2, list(v2)> | <k3, v3> |

In the Map Stage, the original dataset (e.g. a list of <k1, v1>) is broken into smaller data blocks which will be copied to different Datanodes in a cluster. After that, each Datanode will use user-defined Map function to deal with these data blocks. Since all those data blocks and computation job to be finished on the multiple Datanodes, the results produced in the Map stage are in arbitrary order.

Now we are using the wordcount example, assume we have three text files, we want to find out how many times of each word appears in these documents. Here, we try to find how many times of the word “foo” appears. Suppose we have “foo” three times in the file 1, two times in the file 2, and none in the file3. In the Map Stage, the result could be like the following table:

|  |  |
| --- | --- |
| **File 1** | **<”foo”, 1,1,1>** |
| **File 2** | **<”foo”, 1,1>** |
| **File 3** | **<”foo”, 0>** |

In the reduce stage, Reduce will collect all the results and use the user-defined Reduce function/program to produce the final result that user expect.

The following image represents the data flow of the MapReduce:

