A SUMMARY OF THE MAPREDUCE PAPER BY JEFFREY DEAN AND SANJAY GHEMAWAT

This paper introduces readers to an algorithm known as MapReduce that is used for data analysis of large amounts of datasets. This algorithm is made up of two parts; a map function and a reduce function. The map function takes in inputs and organizes the input into key/value pairs. The reduce function then takes the key/value pairs produced by the map function and merges all values associated with the same key. The output from the reduce function is an accumulated value given to a particular pair. The result of the reduce function can also be in the form of a key/value pair. Only that, the values are merged into one. The MapReduce algorithm is mainly used for counting occurrences of particular words in large amounts of documents. Other examples outlined in the paper are the use of the MapReduce function for accumulating the frequency for the number of times particular URLs are visited and its use for accumulating inverted indices, amongst many others.

For its implementation, multiple machines are used in the mapping phase. Every machine gets an amount of data, and the mapping is done in parallel. The outputs from the mapping phase are buffered in memory. In moments of time, the buffered outputs are stored on a local disk and then partitioned for the reduce phase. It is also shared amongst some machines and reduction is done in parallel. Before the data is passed to the reduce function, the data is sorted to group all keys. This allows for easy passing of values with similar keys into the reduce function. The reduce function produces the output pairs which are stored in output files to be accessed or passed on to another MapReduce function. A structure called the master is what does the sharing of the data for mapping and reduction, and the machines are known as workers. These workers have particular states at every point in time based on their current work. They are idle, in-progress, or completed. Master and worker failures are handled effectively in the sense that when one worker fails, all workers are notified, and another worker handles that task the failed worker was handling. For the master, copies of itself are made and so when it fails, a backup copy is used to replace it.

Several extensions aid the MapReduce process. Examples are the partitioning function, which helps in the partitioning of data, ordering guarantees, which help in the sorting of map output data, combiner function, which helps in the reduction process, reading various input and output types, amongst many others.

Regarding performance, the rate of input is higher than the rate of sorting and output because of where the data is read from; the local disk. The rate of sorting is higher than the output rate because of the replication of the output data for reliability purposes. When backup tasks are disabled, computation time increases by 44%. Also, when machine failures occur, there is an increase in computation time by 5%.

In a nutshell, the MapReduce algorithm is an easy to use system because of the number of details abstracted. Also, the MapReduce algorithm can be used to solve a variety of problems that is why large companies like Google widely use it.