MapReduce: Simplified Data Processing on Large Clusters

The paper discusses the map and reduce programming model in detail. After a brief introduction of the mapReduce a description on the programming model with examples are discussed. It gives us an overview of how mapReduce is being implemented on google clusters.

Towards the end of the paper we can see the performance measure and users of the mapReduce programming.

Implementation wise mapReduce is simple and efficient that allows the programmer with very little to no experience to also take advantage of the system. It abstracts away the details of parallelization and optimization and provides a hassle free experience to the users. The mapReduce can be highly scalable on the common machines as well. Numerous machines can process large amounts of data simultaneously.

The reduce function uses intermediate key value pairs produced by map that are then merged to output a smaller list of values. With this specific functionality of mapReduce, large lists now fit in memory more easily.

Although there can be multiple interfaces of mapReduce implementation, the paper concentrates on Google’s computing environment where a vast number of computers are connected together with a switched ethernet.

The authors discuss the implementation details of map function and how it splits the input data and enables multiple machines to process the data chunks. A master data structure plays an important role in the mapReduce system. It tracks all the map and reduce tasks and also identifies the worker machine. The master is also responsible for the system tolerance as it maintains the worker state.

Among the mapReduce users are the grep and sort computation which performs on a terabyte of data. The computation time recorded was 150 seconds for the grep command to execute while the sort program took 891 seconds. The mapReduce program performance is impressive even if there is multiple machine failure.

Lastly, the paper concludes by listing the attributes that made the mapReduce a successful programming model at Google and the learnings that came along with developing such a programming model.