Paper Summary

MapReduce was a way of programming that had been developed at Google to address the need to process simple routines on the very large amounts of data generated by the advent of the internet. This was done by abstracting away the work needed to parallelize computations in a fault tolerant way on a cluster of commodity hardware that individually were not reliable. MapReduce programs at a high level included two main phases. First is the map phase, where mapper jobs took input in the form of key-value pairs and transformed them into intermediate key-value pairs. At the end of the of the map phase, the program would group together all intermediate key-value pairs together by their key. Second is the reduce phase, where each reducer jobs took each group of intermediate key-value pairs and transformed them into the desired result. Many applications can be expressed with this simple programing model that has notably few constraints. These include: counting occurrences of words in a large number of files, finding instances of a certain pattern in many documents, reversing the web linking graph and sorting in a distributed way. I think the biggest strength of MapReduce is its unconstrained way of abstracting away parallelization across a cluster of computers that leads to the need for the user to only write code that is core to the application. The biggest weakness is its requirement for users to think about problems in a entirely new way to be able to solve them in the MapReduce framework.