(a) HW1 Q1 Summary, primary paper [Dean04a]

(b) Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. In Proceedings of the USENIX Symposium on Operating Systems Design and Implementation, pages 137–150, San Francisco, California, USA, December 2004. USENIX.

(c) https://moodle.ant.isi.edu/pluginfile.php/4504/mod_folder/content/0/Dean04a.pdf?forcedownload=1

(d) This paper designs a new abstraction, called Mapreduce, to parallelize the computation, distrubute the data, and handle failure conspire on a large cluster of machines.

(e) The idea about how to design such a general workflow is quite interesting. The implementation, given by the paper, which only needs user to code map and reduce function, is really graceful.

(f) Idea-Design-Experiment-Analysis. This paper arises the new idea first, gives the implementation, measures the performance, and shares their experience about Mapreduce.

(g) Honestly, this paper is really great, easy to understand. If the auther could talk about more detail about the implementation (some parts of this paper is too concise), it would be better.

(h) In figure 1 of this paper, each mapper and reducer write their output into a seperate file. Why not write the output into the same file?

(i) The paper mentions if the master node fails, we will redo the work. Is there any other better way to deal with this?

(j) How to handle consistency is an important problem of distributed system. I am considering whether writing to seperate files is to avoid handling consistency.

(k) This paper is really important, because the idea of mapreduce have been used widely in industry to parallelize large datasets.

(l) Most similar to The tail at scale [Dean13a]. Both of them talk about how to deal with large datasets.

(m) More details about implementation, like how to guarantee order, how to partition data, etc.

(n) Give an idea, implement the system, measure the performance, and use the idea in different tasks.

(o) An approach to let programmers without experience with parallel and distributed systems to use such model.

(p) I think how to develop a general Mapreduce interface is the biggest challenge of this paper.