

# COMP 543, Tools and Models for Data Science

Chengyin Liu, c193

## *Research #4, MapReduce and the Google File System*

This paper [1] describes Google File System (GFS), which is a scalable distributed file system designed to meet Google's internal data storage needs. GFS provides fault tolerance while running on commodity machines and delivers high aggregate performance. GFS is designed for specific applications, and it relaxes some consistency models which increases the flexibility of design. A GFS cluster consists a single master and multiple chunk servers. Unlike other file systems, GFS does not have a per-directory data structure, but logically represents its namespace as a lookup table mapping full pathnames to metadata.

Based on this distributed file system, Jeffrey and Sanjay developed MapReduce, a model of cloud computing. As described in their paper [2], MapReduce is a distributed computing technology and a simplified parallel model, which is mainly used for large scale concurrent program parallelism. A Map function is used to classify data and group them, then the data with the same key are processed by a Reduce function. And this paper also shows that the basic principle of MapReduce is the localization of information processing.

However, it is not always easy to program map-reduce jobs, especially when a series of map-reduce operations need to be optimized. Therefore, google proposed FlumeJava [3] in 2010, which is a data-oriented computing framework. FlumeJava defines two data collections: PCollection representing an immutable data set, and PTable as a subclass of PCollection. It also defines several basic data operations for the above data collections.

To make concurrent programming easier, the design foundation of FlumeJava is to incorporate map-shuffle-combine-reduce as a meta element and provide some basic arithmetic functions. When users have a series of concurrent tasks to run, FlumeJava will try to optimize them by using classic loop fusion optimization and database push down optimization. As a result, the system is more efficient since the number of repetitive tasks reduces a lot.

**References:**

- [1] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. *The Google file system*. Vol. 37. No. 5. ACM, 2003.
- [2] Dean, Jeffrey, and Sanjay Ghemawat. *"MapReduce: simplified data processing on large clusters."* Communications of the ACM 51.1 (2008): 107-113.
- [3] Chambers, Craig, et al. *"FlumeJava: easy, efficient data-parallel pipelines."* ACM Sigplan Notices. Vol. 45. No. 6. ACM, 2010.