Map Reduce

Introduction

Purpose of this lab is to use map reduce model to find frequency of title-cased words from files. Execution time of different size of input data and number of process are tested in this experiment.

Description

In this experiment, map reduce is composed of a map procedure that reads data and count the title word frequency for several chunks of the data, and a reduce method that combines the result. The reason for not having a shuffler in between map and reduce and not sending text data from input reader to map through channel is that reading the file is quite time consuming comparing the processing of the data. Number of reducer is 1.

Experiment

In this experiment, sequential version of data processing is compared with results of 2, 4, 6 and 8 process on a machine with 8-core CPU. Each version is tested with benchmarking of 8, 12, 16, 20 and 24 files, each around 100MB.
Golang 1.13 is used for this experiment. Process limited is set using `runtime.GOMAXPROCS()`. Number of the goroutine for mapper is the same as the process limit for each execution. Execution time is reported in seconds.

Result

	$p(0)$	$p(2)$	$p(4)$	$p(6)$	$p(8)$
$f(8)$	6.68	4.33	2.60	2.69	2.41
$f(12)$	10.08	6.04	3.50	3.49	3.65
$f(16)$	13.50	8.15	4.48	4.12	3.38
$f(20)$	16.92	9.78	5.79	4.80	4.22
$f(24)$	20.04	13.61	6.42	4.50	4.05

Table 1: Execution time

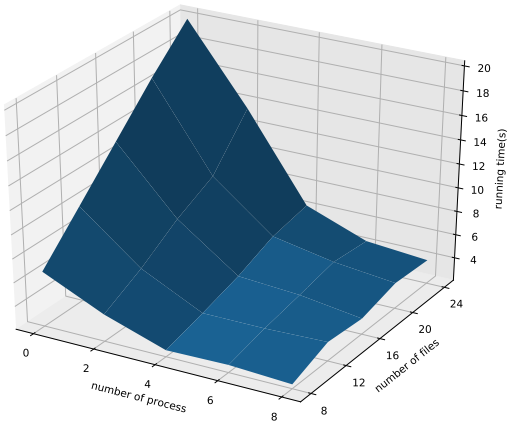

Figure 1: Execution time

Conclusion

From the graph we can see that the execution time increase linearly with the data size, and using more processes to parallelize the model would reduce the execution time of the model.