Worker number	File size	Average Idle time	Average Map time	Average Reduce time
1 worker	22 files/15.8MB	0s	10 s	6.3s
	47 files/32.7MB	0s	22s	12s
	69 files/56.2MB	0s	36.2s	19.2s
	92 files/71.3MB	0s	46 s	23s
2 workers	22 files/15.8MB	1 s	5.5s	3.2 s
	47 files/32.7MB	1 s	11 s	6.5s
	69 files/56.2MB	1 s	19 s	11s
	92 files/71.3MB	1 s	24 s	14 s
4 workers	22 files/15.8MB	1s	4s	2s
	47 files/32.7MB	1.5s	8s	5s
	69 files/56.2MB	1.5 s	14 s	8.5 s
	92 files/71.3MB	1.75 s	18.5 s	11.25s
8 workers	22 files/15.8MB	3 s	3s	2s
	47 files/32.7MB	2.5s	8s	4s
	69 files/56.2MB	3.5 s	14 s	7s
	92 files/71.3MB	3.875 s	20.75 s	10.875 s
16 workers	22 files/15.8MB	4 s	3.56 s	1.11 s
	47 files/32.7MB	4.5s	8 s	2.5s
	69 files/56.2MB	5.75 s	19 s	6 s
	92 files/71.3MB	2.125 s	28 s	21s

In the analysis, I input 4 dataset which include 22 files, 47 files, 69 files and 92 files. The idle time is increasing by the growing of the files, and the idle time is also decided by the number of workers which more workers mean longer idle time for each worker. At the same time, growing worker number will decrease the average map time and average reduce time, this trend is clearly when there is 1 or 2 workers, but when the number of worker reach 8, the time of MapReduce will not decrease. I think for this workload, 4 or 8 workers can reach the best effectiveness.