Report: Cyclomatic Complexity

The system is implemented with a master-worker work-stealing architecture. The system is implemented using RESTful webservices and in python

The master gets all the files in all the commits from the cloned Github repository and then give each file to the workers. Workers are in-charge of calculating the Cyclomatic Complexity of each file and then returns the complexity.

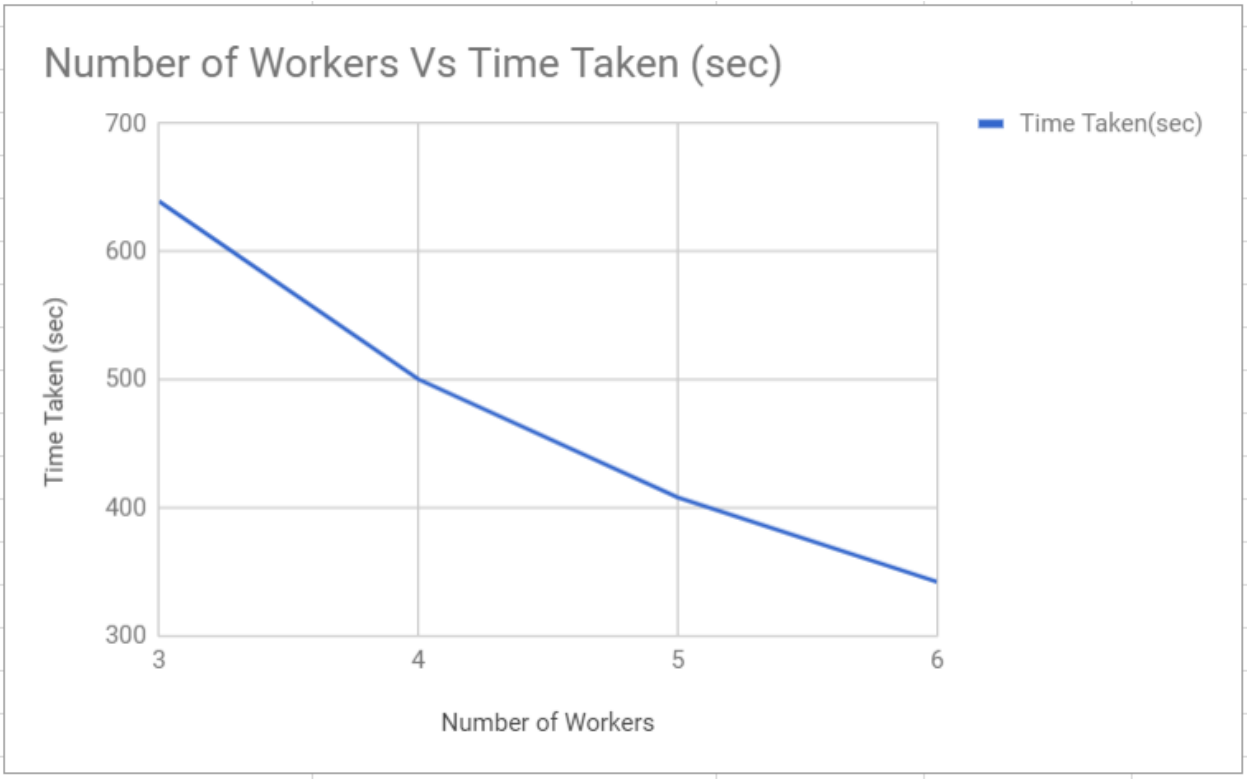
The workers approach the master for the work. Once the work is available, the master distributes the work among the workers. Once a worker finishes its work, it approaches the master for more. The master assigns the next work if any. If a new worker joins in between the execution is also given the work by the master if any.

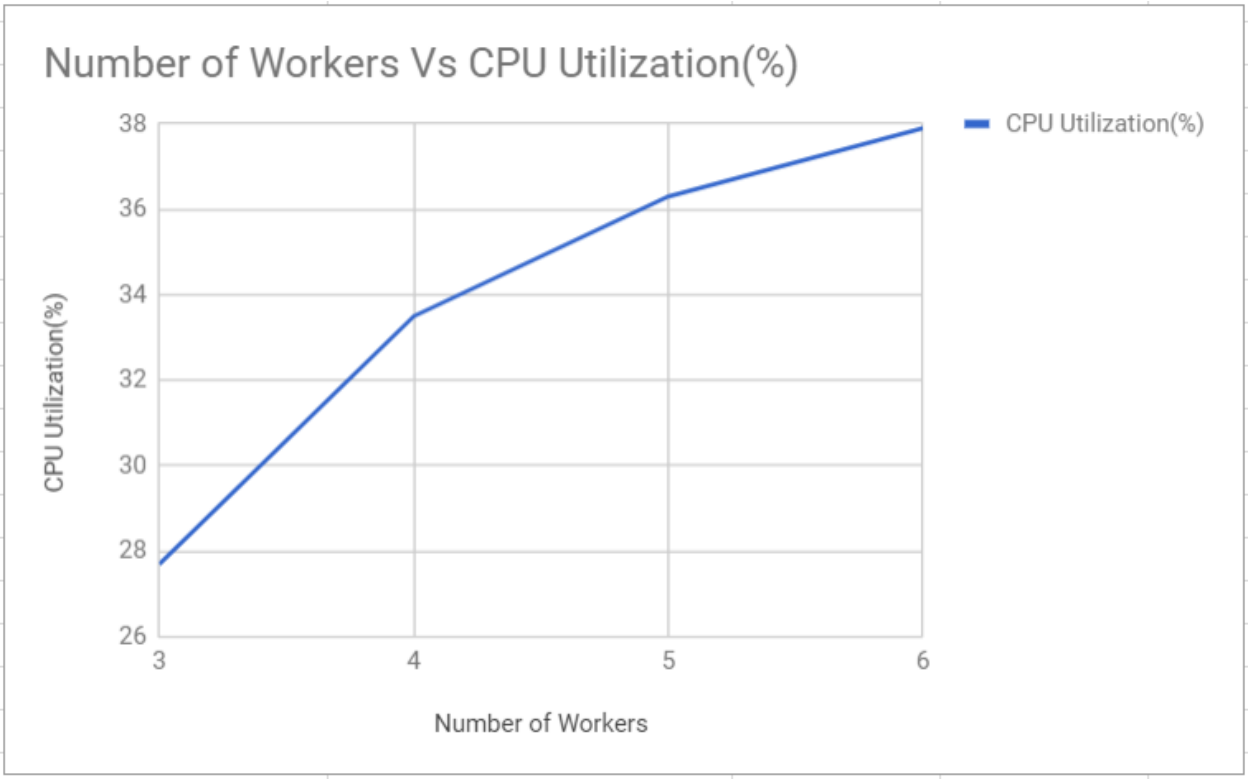
After all the work is distributed, the worker checks if he has received all the result of all the works. Once it is got, it takes the average of the result and the time taken for all the work to be completed.

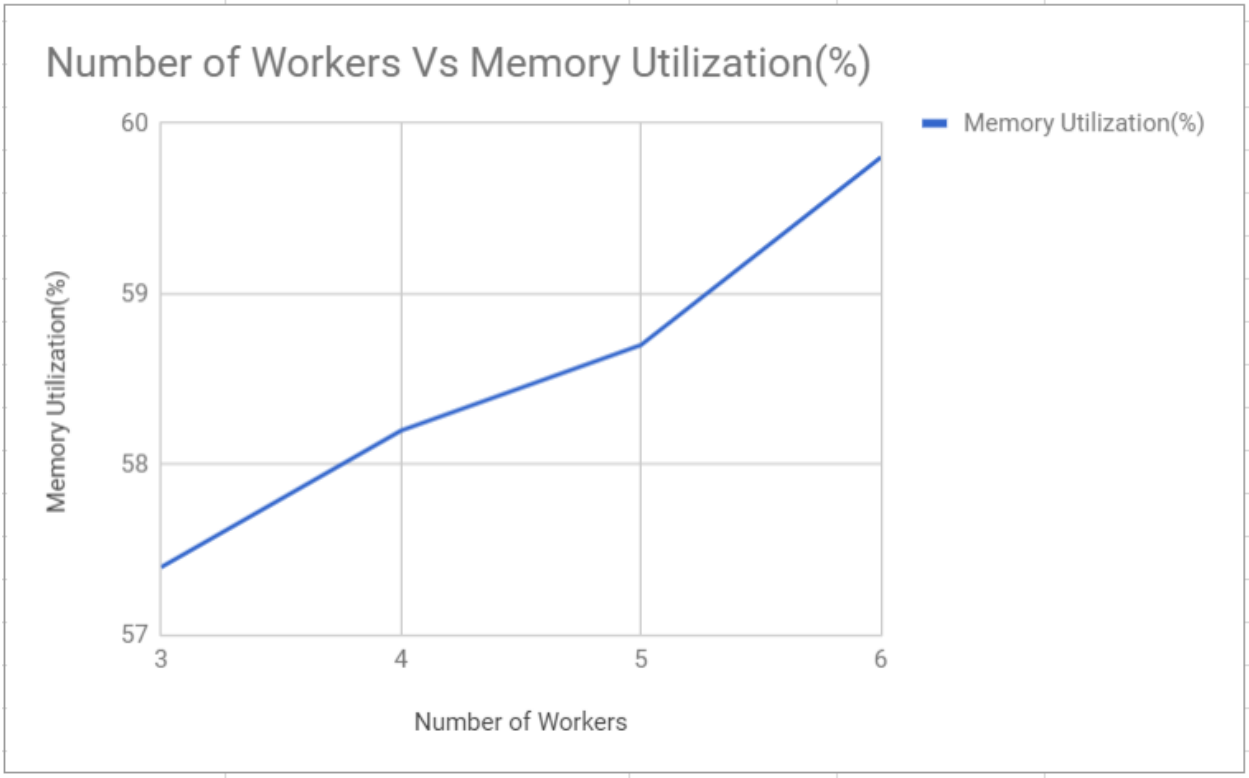
Different Plots are created to see the effect of number of workers. They are:

1. Number of Workers Vs Time Taken
2. Number of Workers Vs CPU Utilization
3. Number of Workers Vs Memory Utilization

The Plots are given below.




Number of Workers Vs CPU Utilization(%)
CPU Utilization(%)
38
36
34
32
30
28
26
CPU Utilization(%)
3
4
5
6
Number of Workers


Number of Workers Vs Memory Utilization(%)
Memory Utilization(%)
60
59
58
57
Memory Utilization(%)
3
4
5
6
Number of Workers

Number of Workers	CPU Utilization(%)	Memory Utilization(%)	Time Taken(sec)
6	37.9	59.8	342.1783628
5	36.3	58.7	408.1262789
4	33.5	58.2	500.4283938
3	27.7	57.4	639.3549671