**CycloMatic Complexity**

The cyclomatic complexity gives a measure of the number of decision points in the program. Decision points are those from where the task execution can proceed in different directions. It is software metric which is used to measure the complexity of the source code. It measures the number of independent paths in a program. I explained cyclomatic complexity with the help of the Master Slave Program.

**Working:**

In this, I have explained the efficiency of work done when the number of slaves increases.

* As the number of slaves increase, the efficiency increases.
* Similarly, when we increase the nodes in the program, then the efficiency increases.

**Work Flow of the program:**

**Masters side:**

1. Master issues Workers with commit SHA's for a given repository and accepts the average Cyclomatic-Complexity of a commit from Worker.
2. Then, Issues newly instantiated Workers with a URL to access commits.
3. Retrieves SHA's for every commit in my Internet Apps Chat Server repository.
4. Calculate and print the average cyclomatic-complexity for the repository.

**Slave side:**

1. A Slave calculates the average Cyclomatic-Complexity for a given commit in a Github repository.
2. The Master issues each Slave with a repo URL upon instantiation, and a commit SHA each time it asks for work. The Slave performs CC analysis on every file in the given commit and returns the average value to the Master.
3. The Slave asks for work forever until the list of commits has been exhausted.