Internet applications Assignment 2
Cyclomatic Complexity of a git Repository

There are 3 services involved in my applicaiton.
The manager service:

- Provides a rest api to communicate with the Docker Manager Api and the Individual nodes.

The Docker Manager Service:

- Creates the specified number of docker nodes using the specified image and returns the ip's for all nodes.

The Worker Service:

- Python script that gets the cyclomatic complexity of a given file, uses flask for the webserver and lizard to compute complexity.

Usage:
Initialization:

- Make a request to /init with the following query params:
- docker: <url for docker service>
- num_workers: <Number of workers to start>
- repo: <Git Repo>
- Returns a session id

Status:

- /status?session=<session_id>
- Provides current status of workers [ build, clone, ready, busy, finished ]

Analyze

- /analyze_all?session=<session_id>
- Makes all worker nodes start working

Lookup

- /lookup?session=<session_id>
- Returns timings for all commits that have been processed.

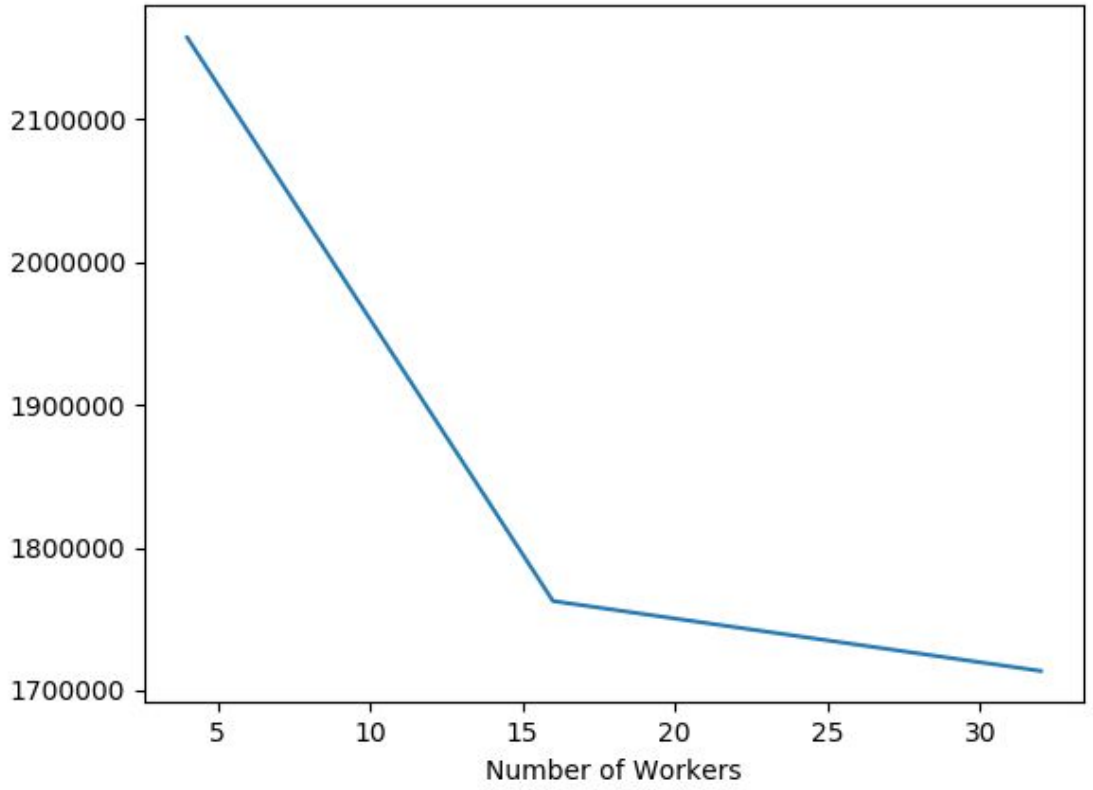
Performance Improves with the number of workers to a point,
My best performance was 32 workers on a large repository
Any more and the resources used by each node start to affect overall performance

My approach for analyzing repos is as follows:

- Clone a repo
- Get list of commits
- For each commit, process all files
- For each file get the complexity
- Store in database

An alternative approach would be to assign a commit to a node and run through all commits that way, which would potentially be faster

Time is in Milliseconds, python cut off the label sorry!




Repo: https://github.com/request/request