Logfile analysis Challenge

Objective

In this challenge, you will need to perform analysis on a logfile.

The logfile is a text file with each line containing one log. A sample of logfile is shown below. The program that produces this log has the following logic

- It writes "Everything looks good" when the server responds correctly to a particular url
- It writes the error, error name, and some stack trace when there is an error.

In general a log line has following components:

- Timestamp
- PID = The process ID
- [234ms] = The time it took for the response.
- UID = User ID which made this request
- INFO/DEBUG/ERROR = The log level
- url = the strings prefixed by / are the urls
- Whatever remains is the message

```
2019/09/12 09:00:00 [PID 1234] [234ms] [UID xS2671] [INFO] /home
Everything looks good
2019/09/12 09:00:00 [PID 1234] [23ms][UID xS2671] [ERROR] /accounts
Error on Line 23 at accountshandler.py Logtrace below. Error name:
InvalidValueError
    Line 23: Something messed up, Exception Raised.
2019/09/12 09:00:00 [PID 1234] [200ms] [UID xS2671] [INFO] /profile
Everything looks good
```

You will need to perform the following analysis on the given logfile

- For a given time period, create
  - a histogram of the errors raised.
  - a 50 percentile, 90 percentile, 95 percentile stats about the response time of successful responses.
- A service/script that consumes from the logfile in a stream and produces a line of log everytime a rule is broke. The rules are to be provided by a json/yaml file. The example of rules are below. The idea is that the user can edit the json file and create new rules, edit existing rules and re-run the script so the new rules will be validated
  - Response time > 100ms for more than 5 seconds
  - Number of errors > 5 in 1 minute interval

You can create one or more programs to get these done. You are free to make reasonable assumptions whenever necessary. Your program(s) will be evaluated on a generic linux system, so you should be using either a language like python, ruby, go etc or a bash script that uses most commonly available linux open source tools.

Discussion Questions

In addition to a code submission, please provide short answers to the following discussion questions in a README (plain-text or Markdown format):

- Briefly describe the conceptual approach you chose! What are the trade-offs?
- What's the runtime performance? What is the complexity? Where are the bottlenecks?
- If you had more time, what improvements would you make, and in what order of priority?

Preparation

Download the sample logfile from this link.

Assessment Criteria

In no specific order:

- If your solution satisfies the requirements
- How the code and functionality is tested
- The understandability and maintainability of your code
- The cleanliness of design and implementation
- Time performance on a standard laptop
- Answers to the discussion questions.