eightfold.ai

Thank you for taking the time to take on the eightfold.ai engineering challenge! We use this project as a way to understand your coding abilities as well as how you would work within a team here at Eightfold.

Helping sanitize the logs of IRCTC

IRCTC uses a mixed model of serving user-requests: both a forked and a threaded model. Consider a server node in the ICRTC backend infra. It has multiple web-server processes running in that single node and each of these web-servers spawns a thread in it's fixed thread-pool for each user-request. That implies, at a time, they will be multiple concurrent processes and threads running in that node. Their logging system is a little rudimentary and they just flush in every log statement, in real-time, with thread level and process level information. As you can imagine, while these logs are chronological, they will be jumbled, garbled, muddled (more synonyms! 😄) w.r.t to a thread.

It is just sanity for a developer to expect to see the logs, in sequence, for a particular user-request(aka a thread). Now the task at hand is to take one such dump of garbled log files and sanitize them such that all logs of a corresponding user-request are in sequence and chronological. The dev knows about grep and is not looking for grepping a single thread-id, but rather wants all the threads/user-requests bundled together in the logs.

In addition to this, after sanitization, support for the following in terms of an api is expected:

- Given an input of time-range in seconds (t1, t2), give back information of how many threads were active in this time-range in the entire node; what are the IDs of these threads; what are the PIDs of these threads; which file stores the logs of these threads.
- What was the highest count of concurrent threads alive in a second given your program has observed the entire log dump. And which time range was it
- What is the average and stdev of the all threads lifetime given your program has observed the entire log dump.
- Given you have done all this, do you have any suggestions, if any, to the developer folks at IRCTC on how their logging system can be improved

Considerations:

- The input given is a bunch of log files each representing the log dump of a single web-process.
- The entire thread count in the dump will not be countably *infinite*. They will be around the range of O(1000)
- You would not have the RAM to store all of the log dump in memory.
- Each thread has a start delimiter(**START**) and an end delimiter(**END**) in the logs

- Each log follows the format of:
  PROCESS_ID:THREAD_ID::THREAD_NAME LOGGED_TIME- LOG_MESSAGE
  Each thread in a process can be identified uniquely by its thread_id. ThreadName is just a user friendly name given to a thread.
- The API could be a web-api over HTTP or a std input or you have just hard-coded the inputs in a file; it is fine.
- It is fine to use external libraries for any of the individual components. It is just expected that you know what you are importing and how it works at a high-level.

Time:

- You would have 4 hrs to hack on this problem.
- Do not worry if you have missed completing any part.
- If you are not able to understand a part of the assignment and feel that a part of the assignment is not doing a good job in being specific, feel free to make an assumption there and move ahead. Do not spend too much time there.
- Do not worry if you are not able to complete a part of the assignment. It is not all black and white.

Logs: link