Software Engineer: Take-Home Project

achieve.ai's software engineers are responsible for developing high-quality, high-performance applications that operate at a massive scale. You should be able to understand system-level designs and then implement them at the component level.

To help us identify this ability, we've created a small project for you. This project should be a fun programming exercise and, at the same time, help you understand a real-world problem that we face.

This project is the most critical part of our selection process. Doing it well will ensure that we call you for an interview and will significantly improve your chances of joining our team.

Context

At achieve.ai, we generate a lot of logs in text format. Recently, we've switched to a Structured Log format and the logs are output as a single JSON document per line. Even though this logging format is concise, the size of the log file has increased significantly, and so has the parsing time.

Problem Statement

We want to build a simple tool that will convert this JSON-based log to columnar format, i.e., we want to create one file per column.

Example, if we have the following 2 log lines:

```
{"timestamp": "2022-01-14T00:12:21.000", "Field1": 10, "Field_Doc": {"f1": "xyz"}}
{"timestamp": "2022-01-18T00:15:51.000", "Field_Doc": {"f1": "abc", "f2": 1.7}}
```

It will generate 4 files:

1. timestamp.column
2. Field1.column
3. Field_Doc.f1.column
4. Field_Doc.f2.column

The column file format is as follows:

- string fields are separated by a new line '\n' character. Assume that no string value has new line characters, so no need to worry about escaping them
- double, integer & boolean fields are represented as a single value per line
- null, undefined & empty strings are represented as an empty line

Example content of timestamp.column:

2022-01-14T00:12:21.000
2022-01-18T00:15:51.000

Example content of Field_Doc.f1.column:

xyz
abc

Note: The fields in the log will be dynamic. Do not assume that these are all the expected properties, or that this is the max. level of nesting.

Goal

The goal of this project is to be able to convert huge JSON-based log files into columnar files. Performance is of the utmost importance, and in the future, we may want to use this tool to accept streaming logs and write them to columnar files. What this implies is that the conversion to columnar files should be performed in single pass.

The size of JSON log file will range between **4GB to 48GB** (sample log files of different sizes).

Bonus points for:

- explaining the bottlenecks in your tool and how you could improve the performance (if you were given a month to optimize further)
- listing out & handling edge cases
- creating any UI to interact with this tool

Your Submission

Code

- The code can be written in any language (provided the code is clean and of high quality)
- Any setup-related instructions should be mentioned clearly in the README.md
- Share your code as a .zip file
- It might be blocked as an attachment, so share a publicly accessible link [Google Drive, OneDrive, etc.] (make sure you test that it can be accessed in incognito mode)
- Please do not upload this project on a public repo

Summary Document

- A brief explanation of your submission
- Table demonstrating the actual time & space complexity
    - time taken for conversion & max. RAM usage for various file sizes (use this folder)
- Do not share links, send direct copies

Testable CLI / API

We should be able to provide the path to a log file and your tool should generate the expected output

Demo (Screenshots / Videos)

Attach screenshots or videos to demonstrate how your submission works

Deadline

You have 7 days to send your submission once you receive this document

Email

Email the files with the subject to tech-jobs@achieve.ai with your resume.

We are looking forward to your submission.

Best of luck & happy thinking!