Hive Streaming
Backend Home Assignment

Welcome to Hive Streaming backend home assignment! Please find the task and some clarifications below. One thing to keep in mind: please do not be restricted by the details. We value your creativity! If you need to make any assumptions not covered in the task, please do so. We are not limiting your architectural choices, what you build may range from a small POC-like solution to a big complex architecture. We hope you enjoy the task 😊 Have fun!

Task

FilesFromYou is transforming the file sharing business by making it super easy to share files between your friends and family. You share files by installing a small client on your home PC or mobile device. In one of the latest releases, it has turned out that some clients have very high CPU usage, making the FilesFromYou service unusable.

In order to find out what is going wrong we want you to design a service that handles CPU data reported to our backend. The collected data should be processed and aggregated in an intuitive way such that it helps troubleshooting which clients are having a problem with the CPU usage. Service is responsible for generating the report on the data.

Scope

1. Report is intended for Head of Product in FilesForYou so she can get an idea of how bad the issue is (for an API-based report, assume that we build a web UI).
2. Please feel free to omit most of the client implementation, and if you want client to send something, like client id, please feel free to assume it does, and place any constraints on it. You can write a simple client implementation or use any API tools available to simulate the requests.
3. Details like authentication or an actual way to measure CPU consumption can be mentioned but please don't focus on that part of the solution.
4. Aim to design a solution that would cover the problem enough to draw the conclusions about the affected clients, so that the client team can triage their code and locate the faulty changeset.
5. The solution does not have to be fully optimized; we will value a good design over the lowest latency. Discussion around possible optimizations is very welcome.

Considerations

To help you with your design, please look at these considerations. It isn't necessary to prepare full answers for each of the questions but incorporating these points into your solution will help you to cover some things we are interested in seeing in your implementation.

Client

1. How often does the client send data?
2. What happens if the backend is unavailable?
3. What happens if the client fails to send data because it consumed all the CPU?

Backend

1. What is "low" or "high" CPU consumption?
2. How does the API look like, which protocol does it use, and why?
3. Which information does the client pass to the service?
4. How does the service handle scaling, restarts, crashes etc.?
5. Which database or database family to use, if any, and what schema would it have?
6. Do you need to store everything the client sends?
7. How to query persisted data to generate the report?
8. What information in the report allows us to pinpoint the faulty clients?

You are also very welcome to bring up more considerations on top of the list above.

Artifacts

We expect you to provide us with the code for your solution via GitHub, archive or compressed git repo. Other artifacts that you can prepare before the interview:

- Architecture diagrams if you have an idea for a bigger or better design
- Notes on what you would improve if you had more time, resources, or manpower

Tech stack

When it comes to programming languages, we do not want to limit you too much, but at the same time, we need to understand each other. The best language match for our backend is Scala with cats-based stack. If you are familiar with Scala or wish to try it for this assignment, please consider picking it. We can be gentle if you have very little Scala experience – it does not have to be perfect. However, if Scala is not an option, please choose a relatively common and popular backend development language, like Java, Go, Python, NodeJS, Typescript, Ruby etc.

Follow-up

After you submit your solution, we will invite you for a technical interview. We expect that you will be able to present your solution via screensharing over a video call, where you will walk us through your code. It isn't necessary to comment on every line! We would like to see how you laid out and structured your code. If you have any additional artifacts, please present them as well. A live demo is very welcome but not required.