Technical Assignment | Product Backend Engineer @Align

Coupon Issuance System

Overview

Implement a system that **issues a limited number of coupons on a first come first served basis at a specific time**. This challenge addresses concurrent control and performance optimization issues commonly encountered in real services, and evaluates your problem-solving approach and technical decision-making.

Problem Description

You need to develop a coupon issuance system that enables creating campaigns with configurable parameters. Each campaign specifies the **number of available coupons** and **a specific start date and time** when coupons can be issued on a first-come-first-served basis.

The expected traffic immediately after a campaign launch is approximately 500-1,000 requests per second. The system must meet the following requirements:

- **Issue exactly the specified number of coupons per campaign** (no excess issuance)
- Coupon issuance must **automatically start at the exact specified date and time**
- Data consistency must be guaranteed throughout the issuance process
- **Each coupon must have a unique code (up to 10 characters, consisting of Korean characters and numbers)**.

Basic Requirements

- Use **connectrpc** (https://connectrpc.com/) and **go**.
- At least, implement following RPC service and methods:
    - **CreateCampaign**: Create a new coupon campaign
    - **GetCampaign**: Get campaign information including all issued coupon codes (**only include successfully issued ones**)
    - **IssueCoupon**: Request coupon issuance on specific campaign

Challenges

- Implement a concurrency control mechanism to solve data consistency issues under high traffic conditions (500-1,000 requests per second).
- Explore and design solutions for various edge cases that might occur.
- Implement testing tools or scripts that can verify concurrency issues.

Submission Requirements

- Github Repository
- README.md

Notes

- We focus more on your problem-solving approach and design decisions rather than perfect implementation.
- Even if not all features are perfectly implemented, a clear solution and rationale for the core challenges is important.
- Some features can be replaced with mock implementations if necessary, but the reasons and actual implementation direction must be clearly documented.
- If you have any questions, feel free to reach out to `dongeon.kim@coxwave.com`