Task Manager

Overview

Design and implement a scalable system that can schedule and execute tasks.

Functional requirements

- Users can submit a new task for immediate or periodic execution.
- Users can list all the submitted tasks and their current status.
- Users can view a single task's details and status.
- Tasks can be run once or recurring.

Non-functional requirements

- Highly scalable — system should scale for millions of tasks
- Durable — system should not lose tasks information in case of any failure
- Using 3rd party modules and libraries is allowed, but not recommended.
- Solution should be tested.
- Write documentation containing:
    - Explanation of the architecture and why it was chosen
    - Guide on how to run it locally.

Example

The following is a sample API design that can be used as a reference. You're not required to implement that exact API.

1. submit(shell_command, recurring_type, recurring_pattern) - Submit a new task
    - **shell_command**: a command to execute (e.g. "ping -c 4 google.com" - ping google.com 4 times)
    - **recurring_type**: ONCE or RECURRING
    - **recurring_pattern**: If type is RECURRING - the recurring pattern (e.g. "5s"). Otherwise - ignored

2. view(task_id) - View a single task and its status (e.g. PENDING, RUNNING, FAILED or SUCCEEDED)

3. list() - List all tasks and their statuses

Delivery expectations

- Implement the service following the requirements above.
- Publish the source code into the provided Chaos' internal Git server, located at http://tig.chaosgroup.com (registration is required, you may sign in with your Google profile).

The solution will be judged in terms of correctness and code quality.