

Інтеграційні програмні системи

Завдання на розрахунково-графічну роботу

Для зарахування роботи необхідно продемонструвати проект, беклог якого було представлено у першій лабораторній роботі на початку семестра. Окрім реалізації історій з беклога проект має відповідати наступним вимогам.

- Проект має бути відкритим, доступним як git-репозиторій на GitHub.
- Збірка програмного коду проекту має бути автоматизованою.
- У проекті мають бути приступні автоматичні тести, які можна запустити за допомогою однієї з задач системи збірки.
- Проект має розроблятися за принципом безперервної інтеграції. Для цього GitHub репозиторій інтегрується з сервісом Travis.
- У програмному коді проекту має бути реалізовано механізм експоненціальної витримки.

Єдиною з вимог до проекту, яка не розглядалася протягом лабораторних робіт, є експоненціальна витримка (exponential backoff). Про неї йтиметься далі.

Експоненціальна витримка (Exponential Backoff)

Механізм експоненціальної витримки використовується у великій кількості сучасних систем, однак зовсім не є новим поняттям. У комп'ютерних мережах, наприклад, даний механізм використовуються для організації доступу до спільної шини (у CSMA/CA). Ми розглянемо даний механізм у контексті клієнт-серверної взаємодії, яка завжди проявляється при інтеграції з зовнішньою системою чи зовнішнім компонентом. Клієнтом називається сторона, яка ініціює комунікацію.

При помилці взаємодії між клієнтом та сервером логіка може вимагати повтору спроби виконання невдалої операції. Ідея експоненціальної витримки полягає в тому, щому кожна така наступна спроба відбувається з більшим інтервалом за попередні з метою мінімізації навантаження на серверну сторону та об'єму некорисних даних, що передаються при таких спробах. Розглянемо 2 приклади таких сценаріїв.

З'єднання з базою даних

Ваш компонент працює, виконуючи запити до бази даних, яка знаходиться на віддаленій машині. У певний момент часу зникає зв'язок з цією машиною. Для відновлення роботи, ваш компонент намагається повторити встановлення з'єднання з базою даних. Перший раз - через 10 сек після помилки. Якщо спроба не вдала, наступного разу - через 20 сек, через 40 сек і т.д., досягаючи максимального періоду 6 хв. У такому випадку, коли сервер бази даних відновлює свою роботу, ви можете розраховувати на автоматичне відновлення роботи всієї системи. Початковий, максимальний періоди та логіка нарощення затримки цілком залежать від вашого конкретного випадку та системи.

Чат

Чат-сервер отримує повідомлення від клієнтів, і, після перенаправлення до інших клієнтів, додає їх у чергу для виконання додаткової обробки. Наприклад, для збору статистичних даних. В обробнику даної черги виникає неочікувана помилка, і, з якоїсь причини, він не може відновитися. Це одразу ніяк не впливає на основну функцію сервера - він продовжує перенаправляти повідомлення. Однак черга постійно накопичується, адже ніхто не забирає з неї дані. Врешті-решт буфер повністю заповнюється, і помилка виникає вже при основній роботі сервера.

Коли чат-клієнт отримує помилку, та намагається повторити спробу взаємодії. Відмінність у цьому випадку полягає в тому, що чат-клієнтів у сервера велика кількість. І помилку отримали усі практично в один момент часу. Відповідно, усі намагаються повторити операції, і також роблять це майже синхронно, хоч і з затримками. Це призводить, до постійних піків навантаження на серверну сторону, що може тільки погіршити ситуацію.

Тому у данному випадку дуже важливим є додавання випадковості у вибір періоду затримки, так щоб розподілити запити на сервер рівномірно, при цьому збільшуючи математичне очікування для вибраного періоду кожного наступного разу. Щось подібне, насправді, відбувається і в CSMA/CA.

При здачі розрахунково-графічної роботи необхідно продемонструвати сценарій успішної роботи вашої ситсеми. Далі відтворити випадок з помилкою на сервері (наприклад, просто його вимкнувши). Показати, що клієнт виконує спроби повтору (наприклад, з допомогою логування). І врешті-решт відновити роботу сервера, показати, що взаємодію між клієнтом та сервером відновлено.

Оформлення роботи

Робота має включати:

1. Титульну сторінку
2. Короткий опис проекту
3. Опис того, яка система збірки використовується у проекті.
4. Перелік та опис задач, які виконуються на сервері безперервної інтеграції.
5. Графік, який ілюструє вибрані інтервали для повтору спроб при експоненціальній витримці.

На осі X - номер спроби, на осі Y - вибраний інтервал.

Ви маєте отримати щось схоже на експоненціальну криву

