



Universidade Federal do Ceará

Campus Quixadá

Disciplina: Reuso de Software

Título: Smart Connection Pool - Um Padrão para Gerenciamento Eficiente de Conexões com Banco de Dados

Autores: Eduardo de Oliveira Sousa – 510969, Murilo dos Santos Cunha – 521441

1. Introdução

O gerenciamento eficiente de conexões com o banco de dados é essencial para garantir a estabilidade e o desempenho de sistemas escaláveis. Muitas aplicações, como plataformas de e-commerce e serviços financeiros, lidam com múltiplas requisições simultâneas, exigindo um controle eficiente das conexões para evitar sobrecarga e latência. Este documento apresenta a proposta do **Smart Connection Pool**, um padrão de projeto voltado para a otimização da reutilização de conexões com o banco de dados.

2. Tecnologias Utilizadas

Para a implementação do **Smart Connection Pool**, utilizamos as seguintes tecnologias:

- **Linguagem de programação:** Python
- **Gerenciamento de concorrência:** Multithreading
- **Estrutura de dados:** Fila (queue.Queue) para armazenar conexões disponíveis
- **Testes automatizados:** unittest para validação das funcionalidades
- **Versionamento de código:** Git e GitHub para controle de versão e colaboração

Essas tecnologias foram escolhidas para garantir eficiência, escalabilidade e modularidade na implementação do padrão.

3. Descrição do Problema

Sistemas que utilizam bancos de dados frequentemente enfrentam desafios na administração de conexões. Sem um controle adequado, novas conexões são criadas a cada requisição e posteriormente destruídas, resultando em:

- Alto consumo de recursos computacionais;
- Aumento da latência das operações;
- Sobrecarga do banco de dados;
- Possibilidade de indisponibilidade do serviço devido ao limite de conexões simultâneas.

Esse problema é crítico especialmente para aplicações que lidam com um grande volume de acessos simultâneos. Assim, é fundamental uma abordagem que permita o reaproveitamento de conexões e um controle eficiente da concorrência.

4. Descrição da Solução

A solução proposta é a implementação do padrão **Smart Connection Pool**, um gerenciador de conexões que reutiliza conexões já estabelecidas ao invés de criar novas. A arquitetura funciona da seguinte forma:

- O sistema mantém um pool com um número pré-definido de conexões ao banco de dados.
- Quando um cliente precisa acessar o banco, ele solicita uma conexão ao pool.
- Caso haja conexões disponíveis, uma é fornecida ao cliente. Caso contrário, o cliente precisa aguardar.
- Após o uso, a conexão é devolvida ao pool para reutilização por outras requisições.
- Um mecanismo de controle de concorrência garante que múltiplas requisições sejam processadas de maneira segura e eficiente.

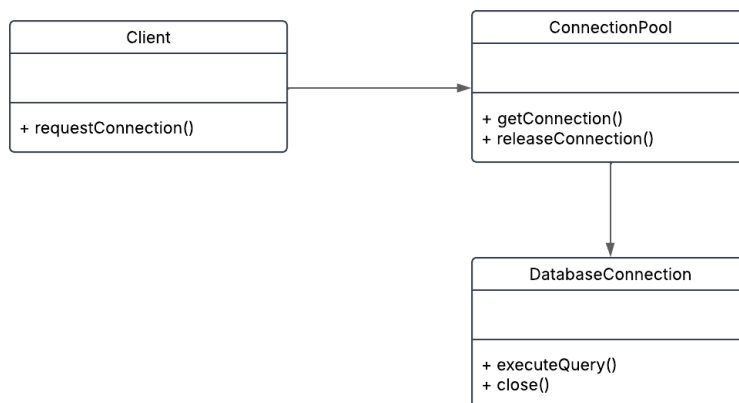
Essa abordagem reduz drasticamente o custo de abertura e fechamento de conexões, melhora a eficiência do sistema e evita falhas decorrentes de sobrecarga do banco de dados.

5. Diagrama de Classes

A arquitetura do **Smart Connection Pool** segue o padrão de reuso de conexões, garantindo que o acesso ao banco de dados seja otimizado. Cada componente do sistema desempenha um papel essencial:

- **ConnectionPool**: Gerencia um conjunto limitado de conexões reutilizáveis, controlando o acesso e evitando sobrecarga.
- **DatabaseConnection**: Representa uma conexão individual com o banco de dados, sendo gerenciada pelo pool.
- **Client**: Representa uma entidade que solicita conexões ao ConnectionPool para executar consultas e operações no banco.
- **Controle de Concorrência**: Garante que múltiplos clientes possam acessar conexões de forma segura e eficiente, evitando bloqueios ou falhas.

Além disso, a arquitetura permite escalabilidade e pode ser aplicada a diferentes cenários, desde pequenas aplicações até grandes plataformas que lidam com milhões de transações simultaneamente. A seguir, apresentamos o diagrama UML representando a estrutura do Smart Connection Pool:



6. Implementação e Exemplos de Uso

A implementação do **Smart Connection Pool** foi desenvolvida em **Python** e está disponível no repositório GitHub do projeto. Para demonstrar sua aplicabilidade, foram criados três exemplos de uso:

6.1 Servidor Web

Neste exemplo, múltiplas requisições simultâneas ao servidor web são processadas utilizando o Smart Connection Pool para gerenciar conexões ao banco de dados.

6.2 Sistema de Relatórios

Um sistema de geração de relatórios onde várias consultas são feitas ao banco sem que novas conexões sejam criadas a cada requisição, reduzindo a carga no banco.

6.3 E-commerce

No contexto de um e-commerce, diferentes usuários realizam compras simultaneamente, e o pool de conexões gerencia as transações de forma eficiente.

7. Conclusão

O **Smart Connection Pool** demonstra ser uma solução eficaz para o gerenciamento eficiente de conexões com o banco de dados, proporcionando um melhor desempenho e reduzindo o desperdício de recursos computacionais. A implementação do padrão em diferentes cenários comprovou sua aplicabilidade e benefícios para sistemas de grande escala.