

Trabalho Final de Reuso de Software

Carlos Gabriel de Castro Rodrigues - 509655

Kauan Oliveira Perdigão Lopes - 514867

Fevereiro 2025

1 Descrição do problema

Em sistemas distribuídos que possuem múltiplos serviços ou componentes responsáveis por executar tarefas, pode ocorrer um desequilíbrio na distribuição da carga de trabalho. Isso pode resultar em alguns componentes sobrecarregados enquanto outros permanecem ociosos. Esse problema leva a desperdício de recursos, aumento no tempo de resposta e redução da eficiência do sistema. Para garantir uma melhor utilização dos recursos disponíveis, é necessário um mecanismo que distribua dinamicamente as tarefas entre os componentes de forma equilibrada.

2 Descrição da solução

O padrão criado para solucionar o problema é o "Pattern Balancer" que define um intermediário para gerenciar a distribuição de tarefas entre componentes ou serviços disponíveis. Esse intermediário avalia a carga atual de cada componente e atribui novas tarefas de maneira otimizada, considerando fatores como carga de trabalho, capacidade de processamento e prioridade.

A implementação inclui:

- Tarefa: Representa uma unidade de trabalho com um tempo de execução definido.
- Componente: Representa um serviço ou unidade de execução que processa as tarefas.
- Balanceador De Tarefas: Atua como o intermediário que decide qual componente deve executar cada tarefa, garantindo um balanceamento de carga eficiente.

3 Exemplos de uso

Foram implementados 3 situações de uso para o padrão, afim de demonstrar o seu funcionamento e utilidade. São eles:

- **Servidores:** Representa um sistema de balanceamento de carga, onde múltiplos servidores processam requisições de usuários. O balanceador escolhe o servidor menos ocupado para evitar atrasos e melhorar a performance.

Utilidade: Usado em hospedagem de sites, APIs, serviços em nuvem e qualquer sistema que precisa distribuir requisições entre servidores.

- **Fábrica:** Simula um sistema de linha de produção, onde diferentes máquinas processam tarefas. O balanceador distribui as tarefas entre as máquinas menos ocupadas, garantindo um fluxo de produção mais eficiente.

Utilidade: Aplicado em indústrias para otimizar a produção e evitar sobrecarga em uma única máquina.

- **Cluster:** Simula um cluster de processamento, onde vários nós computacionais executam jobs (tarefas complexas). O balanceador distribui os jobs entre os nós menos ocupados, maximizando o desempenho do sistema.

Utilidade: Aplicado em Big Data, Inteligência Artificial, Computação em Nuvem e Data Centers para processar grandes volumes de dados de forma eficiente.

4 Diagrama de classes

