

Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

Campus João Pessoa

Unidade Acadêmica de Informática

Sistemas Distribuídos - PPGTI

Aluno: Laerte Bernardo de Medeiros

Prof: Ruan Delgado

ATIVIDADE PRÁTICA - RPC

Características

O sistema desenvolvido permite que clientes se conectem ao servidor e realizem operações síncronas, como inclusão, consulta, remoção e listagem de elementos das listas. A persistência dos dados é garantida por meio do armazenamento em um arquivo JSON (data.json), o que assegura que os dados não sejam perdidos em caso de interrupções no serviço. A exclusão mútua é implementada usando mutexes, garantindo que várias operações concorrentes sejam tratadas de maneira segura e consistente.

1. Escalabilidade

Em termos de escalabilidade, o sistema apresenta algumas limitações inerentes ao seu design. Utilizando um único arquivo JSON para armazenar todas as listas, o desempenho pode ser afetado à medida que o número de listas e a quantidade de dados aumentam. O acesso concorrente ao arquivo pode se tornar um gargalo, especialmente em ambientes com muitos clientes simultâneos. Além disso, o uso de mutexes para sincronização pode limitar a escalabilidade, pois as operações são bloqueadas enquanto uma operação de leitura ou escrita está em andamento.

2. Disponibilidade

A disponibilidade do sistema é afetada pelo fato de que ele depende de um único servidor para todas as operações. Se o servidor falhar, os clientes não poderão acessar ou modificar as listas até que o servidor seja restaurado. No entanto, a implementação da função TryReconnect no cliente adiciona uma camada de resiliência. Essa função permite que o cliente tente se reconectar ao servidor em caso de falhas de conexão, melhorando a disponibilidade percebida pelo usuário final.

3. Consistência

A consistência é garantida pelo uso de mutexes, que asseguram que as operações em listas individuais sejam realizadas de forma atômica. No entanto, em um ambiente distribuído, a consistência poderia ser comprometida em casos de múltiplos servidores. A

solução para isso seria a implementação de mecanismos para manter a consistência dos dados nos servidores, o que pode introduzir complexidade adicional ao projeto.

Pontos de Falha e Tratamento

Os principais pontos de falha do sistema incluem a dependência de um único servidor e o uso de um único arquivo para persistência de dados. Qualquer falha no servidor ou caso o arquivo JSON seja corrompido, ocorreria uma indisponibilidade do serviço ou perda de dados. Para diminuir esses riscos, pode-se implementar mecanismos de backup e replicação dos dados.

Os logs implementados no sistema são fundamentais para monitorar e diagnosticar problemas. Eles registram operações realizadas, conexões e desconexões de clientes, bem como erros que ocorrem durante as operações. Esses logs são essenciais para identificar pontos de falha e realizar a manutenção do sistema.

Conclusão

Em suma, o sistema possui uma boa base para gerenciar listas distribuídas com múltiplos clientes. No entanto, a escalabilidade é limitada pela arquitetura atual. Melhorias na escalabilidade poderiam ser alcançadas com o aumento do número de servidores e a implementação de mecanismos que garantam a consistência, embora isso possa impactar no nível de complexidade e na disponibilidade. A função TryReconnect e os logs detalhados fornecem resiliência e ferramentas de diagnóstico, melhorando a robustez geral do sistema e a melhoria da experiência do usuário.