Relatório de Implementação de Espaço de Tuplas Distribuído Tolerante a Falhas

Ismael Coral Hoepers Heinzelmann, Matheus Paulon Novais

June 30, 2024

1 Introdução

Este relatório descreve a implementação de um espaço de tuplas distribuído tolerante a falhas, utilizando JGroups e Java 17. O sistema consiste em clientes que adicionam, removem e leem tuplas do espaço, e servidores que processam essas operações, garantindo a consistência e a integridade dos dados. Baseada na implementação do espaço de tuplas, foi implementada uma aplicação de alocação de salas, que permite usuários solicitarem salas para realizar suas tarefas.

2 Arquitetura do Sistema

A abordagem escolhida para a implementação do espaço de tuplas foi uma arquitetura cliente-servidor, onde o servidor é composto por réplicas, que recebem requisições de operações no espaço de tuplas. Cada requisição feita pelos clientes é atendida pelo servidor com o menor ID, e as operações que forem feitas pelo mesmo são replicadas para outras instâncias de servidor.

2.1 Clientes

Os clientes são responsáveis por enviar requisições para o espaço de tuplas. As operações suportadas incluem:

- **Get:** Requisição bloqueante que suspende a thread do cliente até que a tupla solicitada esteja disponível no espaço de tuplas.
- **Write:** Adição de uma tupla no espaço de tuplas.

- **Read:** Leitura de uma tupla que corresponda a um padrão declarado.

Os clientes estão conectados ao canal de clientes do JGroups, e a interface de cliente está disponível na classe TupleSpaceInterface.

2.2 Servidores

Os servidores recebem as requisições dos clientes e executam as operações no espaço de tuplas. A arquitetura de servidor inclui:

- Eleição de um coordenador (instância com menor ID) para coordenar as operações.
- Replicação das operações para garantir consistência entre as réplicas de servidor.
- Garantia de atomicidade de operações, utilizando locks distribuídos e locks inter processos.
- Garantia de exclusão mútua entre instâncias.
- Garantia de ordem das mensagens utilizando protocos suportados pelo JGroups.

Os servidores estão conectados ao canal de servidores e também ao canal de clientes do JGroups, e sua classe de funcionamento é definida em TupleSpace.

3 Problemas Enfrentados e Soluções Implementadas

3.1 Implementação de um Get Bloqueante no Cliente

Para garantir que os clientes recebam tuplas apenas quando estiverem disponíveis, implementamos um mecanismo de fila e suspensão de threads, onde a solicitação é colocada em uma fila, e assim que uma instância receber um recurso que satisfaça esta solicitação, a mesma é retornada diretamente para o cliente solicitante, que possui uma thread para recebimento de mensagens, que então acorda a thread principal, retornando a execução da aplicação. Mecanismos adicionais foram implementados, como no caso de um cliente sair do canal (ou falhar) suas solicitações são retiradas da fila.

3.2 Integridade do Espaço de Tuplas Durante as Solicitações

Para manter a integridade durante operações de adição e remoção de tuplas, utilizamos os mecanismos de exclusão mútua suportados pelo JGroups, onde o mesmo é controlado pelo coordenador do canal de servidores. Isso garante que as operações sejam atômicas e evita conflitos de sincronização.

3.3 Garantia de Ordem das Operações

Inicialmente achava-se que seria necessário implementar todas as questões de ordem de mensagens manualmente, porém após uma pesquisa, foi encontrado que os canais podem utilizar protocolos específicos de comunicação. Os escolhidos foram o UNICAST3 e NAKACK2, garantindo que as ordem total das mensagens é alcançada e também a garantia de entrega.

3.4 Reentrada de uma Instância de Servidor

Implementamos uma política de reentrada onde novas réplicas de servidores solicitam o estado atual ao coordenador, que interrompe o fluxo das operações para permitir a entrada consistente do novo membro. Após receber o estado, retornar o sucesso para o coordenador, as réplicas retomam as operações. Caso o solicitante falhe em comunicar o sucesso (ou seja, o mesmo resultou em "*crash*"), o sistema retorna ao fluxo usual. Caso alguma falha ocorra com o coordenador, o sistema elege um novo coordenador, que irá retornar o fluxo usual dos servidores.

4 Aplicação Implementada: Alocação de Salas

A aplicação construída sobre o espaço de tuplas implementado é um sistema de alocação de salas. Cada sala é representada como uma tupla no formato:

$$
\begin{aligned}
&(Nome\ da\ sala,\\
&Projetor,\\
&Ar\ Condicionado,\\
&Computadores,\\
&Mesa\ Compartilhada,\\
&Capacidade)
\end{aligned}
$$

Onde os atributos podem ser:

- **Projetor, Ar Condicionado, Computadores, Mesa Compartilhada:** Valores de "sim" ou "não".
- **Capacidade:** Pode ser 25 ou 50.

4.1 Funcionalidades da Aplicação

Os clientes podem realizar as seguintes operações:

- **Reservar Sala:** O cliente escolhe as características desejadas da sala e o sistema aguarda até que uma sala correspondente esteja disponível. Após a reserva, o cliente mantém a posse da sala até liberá-la.
- **Registrar Sala:** Adicionar uma nova sala ao sistema especificando todos os atributos.
- **Verificar Estado de Sala:** Consultar o estado de uma sala específica pelo nome ou características.
- **Listar salas:** Exibe todas as salas disponíveis no momento.

Essa aplicação utiliza as funcionalidades do espaço de tuplas para garantir a consistência e a integridade das operações de alocação de salas, mesmo em cenários de falha.

5 Implementação

A implementação da aplicação de alocação de salas foi realizada em Java 17 utilizando a biblioteca JGroups para comunicação e sincronização entre os clientes e servidores do espaço de tuplas.

- **JGroups:** Utilizado para criar os canais de comunicação entre clientes e servidores, garantindo a entrega confiável de mensagens.
- **Sincronização com Espaço de Tuplas:** Uso de locks e ordenação total de mensagens para assegurar que todas as operações sejam consistentes e ocorram na ordem correta.
- **Comunicação:** Dois canais são instânciados, um canal de clientes, onde os clientes se comunicam com os servidores, e o canal de servidores, onde os mesmos o utilizam para informar uns aos outros as operações em cima do espaço de tuplas.

6 Conclusão

A implementação do espaço de tuplas distribuído tolerante a falhas, juntamente com a aplicação de alocação de salas, demonstra eficiência na gestão de recursos compartilhados em um ambiente distribuído. As soluções implementadas garantem que as operações sejam atomicamente consistentes, ordem de entrega de mensagens e a continuação do funcionamento do sistema mesmo diante de falhas.

A vantagem principal que podemos observar no sistema implementado é a replicação total, garantindo que todo o estado seja replicado facilmente, possibilitando que se qualquer instância de servidor estiver disponível, as tuplas estarão disponíveis. Já a desvantagem o "*overhead*" de replicação, onde cada operação deve ser replicada para N - 1 instâncias.