
2º Parte do Projeto de Sistemas Distribuídos

A47

URL do repositório do GitHub: <https://github.com/tecnico-distsys/A47-SD18Proj>



83521

Mariana Mendes



83539

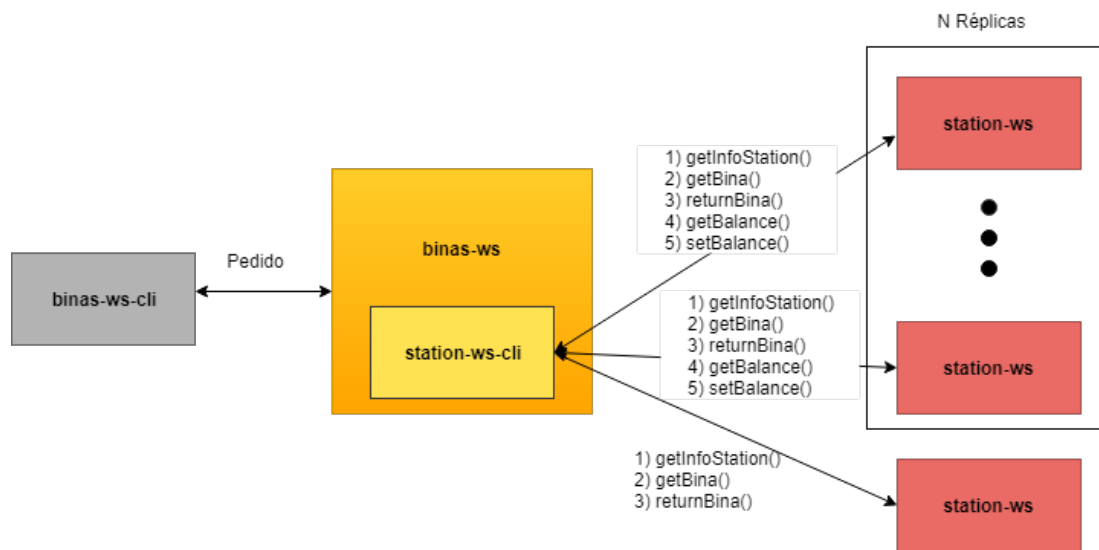
Pedro Caldeira



83540

Pedro Lopes

Figura da solução de tolerância a faltas



Replicação

Na conceptualização aplicada existem N estações estáticas que vão servir como réplicas, como indicado no enunciado. O número de estações, N, é inicializado no arranque do Binas-ws e pressupõe-se que representa as N primeiras estações. Exemplo: sendo $N = 3$, as réplicas terão os nomes "A47_Station1", "A47_Station2" e "A47_Station3". A comunicação para implementar a replicação é realizada através dos métodos adicionais, `getBalance` e `setBalance`.

O método `getBalance` recebe o `userEmail` e retorna uma estrutura, `AccountView`, que tem três atributos: o crédito da conta associada ao `userEmail`, a sua tag e o `clientId`. Este `clientId` permite ao `binas-ws` executar vários pedidos em paralelo (multi-threading).

O método `setBalance` recebe como argumentos o `userEmail`, o crédito e a tag a atualizar na conta do user associado, e o `clientId`. Neste caso os dados referentes ao user associado apenas são atualizados se a tag dada for maior que a tag já existente no user ou, caso estas sejam iguais, se o `clientId` dado seja superior ao já existente.

Breve Explicação da Solução

O protocolo está implementado no lado do `binas-ws` e é baseado no *Quorum Consensus*.

Na função de leitura (`getBalance()`), o pedido é enviado assincronamente para todas as estações estáticas gestoras de réplicas. São feitas até 3 tentativas para o estabelecimento da conexão com as estações, no caso de falha. Caso este valor seja ultrapassado, a tentativa de conexão é abortada, assumindo que a estação está em baixo. De seguida, é esperada a receção das respostas até atingir o quórum, ou seja, $N/2 + 1$. No caso de serem usados protocolos de transporte de menor fiabilidade, existe a possibilidade de mensagens serem perdidas na rede e, portanto, temos um mecanismo de reenvio dos pedidos para evitar que isto aconteça. Por fim, tendo obtido o quórum, as respostas são verificadas e é devolvido o valor do crédito correspondente à maior tag. Em caso de empate destas, é devolvido o que tiver maior `clientId`.

Na função de escrita (`setBalance()`), é feita uma leitura inicial (usamos a função de leitura `getBalance()`) de forma a saber qual é a maior tag guardada. Tal como antes, são feitas 3 tentativas para o estabelecimento da conexão com cada uma das estações. Caso este valor seja ultrapassado, a tentativa de conexão é abortada, assumindo que a estação está em baixo. Posteriormente, são feitos requests a todas as estações gestoras de réplicas e no campo da tag, é usado o valor da tag máxima obtida incrementada em 1 unidade. Fica então à espera do quórum, ou seja, que $N/2 + 1$ estações respondam ao pedido. Novamente, tal como antes, no caso de serem usados protocolos de transporte de menor fiabilidade, existe a possibilidade de mensagens serem perdidas na rede e, portanto, temos um mecanismo de reenvio dos pedidos para evitar que isto aconteça. Quando o quórum é obtido, a função retorna e o programa continua livremente.

Para cada pedido do cliente é gerado um `clientID` sequencial que é único ao longo do pedido, de forma a que seja possível a existência de vários pedidos em paralelo.

Semântica Utilizada

A semântica utilizada na implementação da solução é “*No-Máximo-Uma-Vez*”, no caso do `setBalance`. Isto porque, caso a tag seja menor do que a que já está registada a atualização da informação fornecida não ocorre. No caso do método `getBalance`, a semântica utilizada é “*Talvez*”, uma vez que não é passada qualquer tag e não há nenhuma garantia que esta função tenha sido executada no lado da `station-ws`. Pode não executar e o pedido pode ser repetido. No entanto, caso o quórum já tenha sido obtido, esse valor é descartado mesmo que seja recebido.

Troca de Mensagens

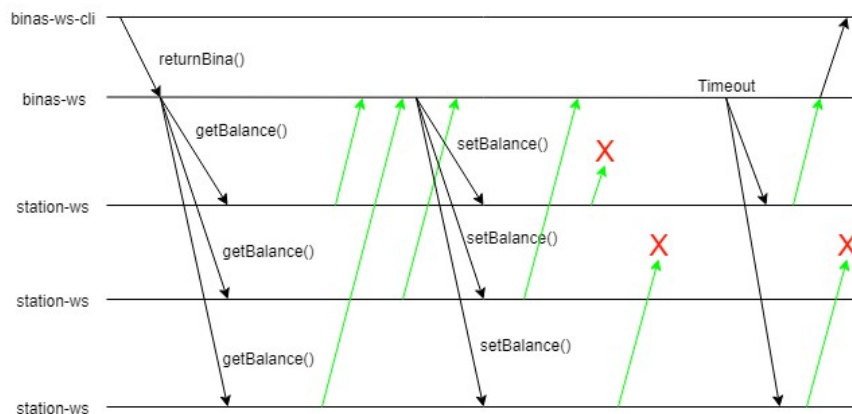


Diagrama referente à troca de mensagens durante a execução do método `returnBina()`.

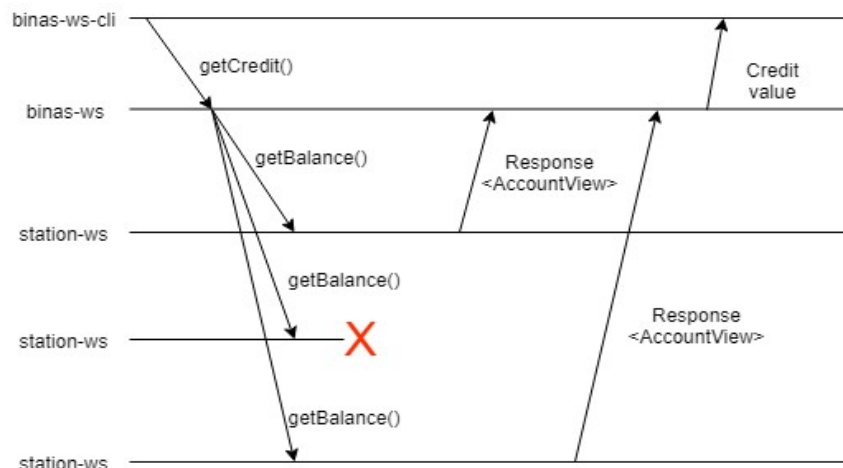


Diagrama referente à troca de mensagens durante a execução do método `getCredit()`.