

# Sistemas Distribuídos

Segunda Parte: Tolerância a Faltas

Grupo A17 <a href="https://github.com/tecnico-distsys/A17-SD18Proj">https://github.com/tecnico-distsys/A17-SD18Proj</a>		
		
83448	83475	83504
Dorin Gujuman	Hugo Guerreiro	Manuel Vidigueira

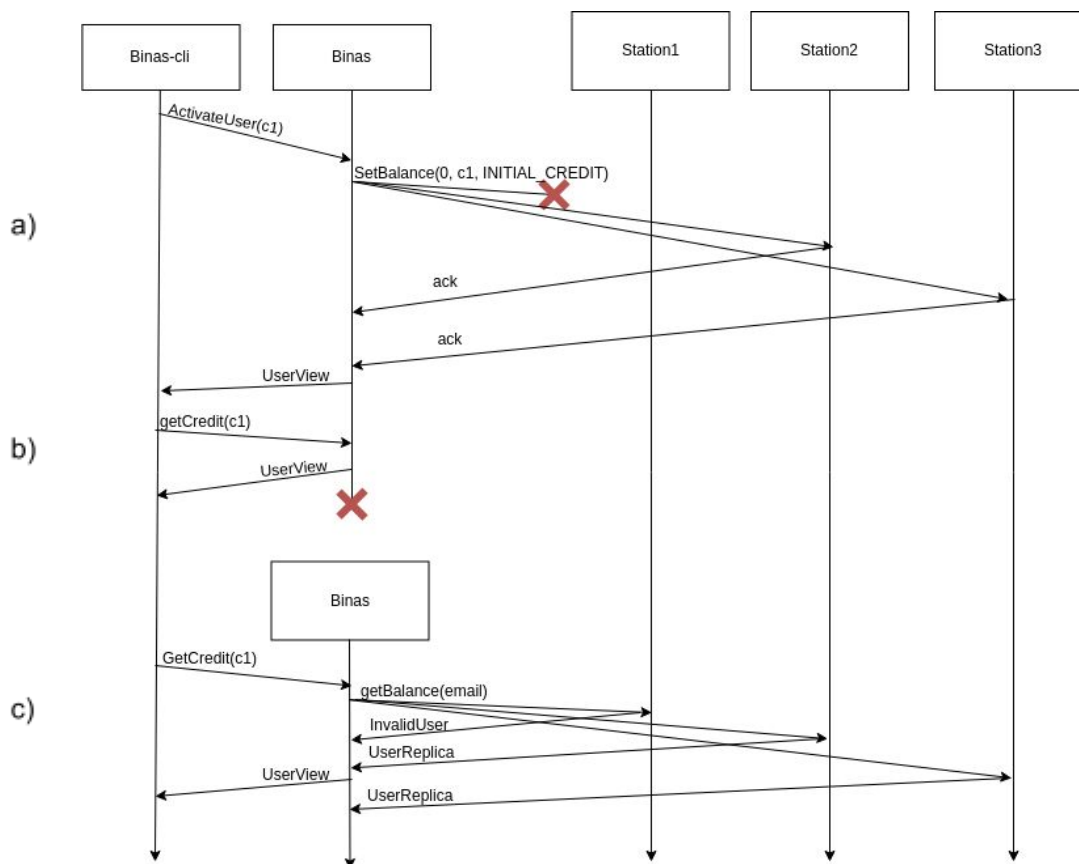


Figura 1) Solução de tolerância a falhas

## 1) Solução

### 1.1) Tolerância a falhas

Para obtermos consistência e tolerância a falhas no nosso sistema utilizámos replicação activa onde os saldos dos utilizadores são replicados nas estações à la Quorum Consensus, deste modo suportamos, para N estações,  $N/2 - 1$  falhas.

### 1.2) Métodos adicionados

Alterámos o wsdl do binas-cli para adicionar dois novos métodos à estação: `int getBalance(email)` e `void setBalance(UserReplica)`. O método `setBalance` recebe uma instância de um `UserReplica` que se trata de uma estrutura que contém o sequence number, email do utilizador e saldo. O método `getBalance` recebe o email do utilizador e devolve o valor do seu saldo.

As chamadas a estes dois métodos são feitas no servidor Binas de forma assíncrona pelo que para recebermos as respostas fazemos polling. Escolhemos fazer polling pois a thread que chama os métodos não irá executar mais nenhuma instrução para além de aguardar pelas respostas, logo não iríamos tirar partido da utilização de chamadas com funções de callback.

### 1.3) Otimizações face a Quorum Consensus normal

#### 1. Sequence number por utilizador

- Permite-nos fazer "QC writes" (ver Notas 1) em paralelo para utilizadores diferentes

2. Eliminação do “client ID” (Ver Notas 3.1)
3. “Cache”
  - a. O Binas guarda o sequence number de cada utilizador, o que permite:
    - i. Eliminar a fase de leitura de um “QC write” (Ver Notas 3.1)
    - ii. Melhorar a concorrência face a um sequence number global
  - b. O Binas guarda o crédito corrente de cada utilizador, o que permite:
    - i. Responder imediatamente a pedidos de “getCredit” sem consultar as estações
4. Eliminar a fase de write-back de “QC reads”
  - a. Devido a 3.a.i, a única altura em que é feito “QC read” é se o Binas não possuir em cache o utilizador referido. Como um “QC write” só é feito quando o utilizador é ativado, a cache mantém-se consistente ao longo de toda a sua vida.(Ver Notas 3.2)

## 2) Explicação da figura

Na figura 1 estão ilustradas algumas trocas de mensagens do nosso sistema com as 3 estações ativas. Identificamos 3 momentos distintos da execução:

a) É efetuada um pedido de ativação de um utilizador pelo que é necessário garantir que os dados são replicados. Deste modo, esperamos por um quorum de respostas (2 em 3) das estações antes de confirmar ao utilizador o seu registo. Além de replicarmos o saldo do cliente nas estações também o registamos na cache mantida no Binas. É de notar que uma das escritas não chegou à estação 1, pelo que o utilizador não é registado lá.

b) Neste momento é feito um pedido para obter o crédito corrente do cliente. Dado que a cache está sempre atualizada podemos responder imediatamente sem precisarmos de pedir e esperar pelas respostas das estações. Após a execução deste pedido, o servidor Binas falha.

c) O servidor Binas recupera e é feito um pedido de leitura para um cliente que o servidor Binas não irá ter em cache pois está vazia. Nesta circunstância, não podemos devolver imediatamente o saldo pois o cliente existe e está replicado nas estações. Para evitar informar erroneamente que o utilizador é inválido temos de obter um quorum de respostas das estações, atualizamos a cache e devolvemos o resultado obtido com a tag mais recente. Como este quorum de leitura interseja o quorum de escrita anterior, conseguimos obter o valor correto e tolerar ambas a falha silenciosa da estação.

## 3) Notas

1. Chamamos informalmente “QC read” ao processo de pedido de leitura e espera por uma maioria de respostas do protocolo Quorum Consensus normal (sem writeback), e de “QC write” ao de escrita (sem a fase preliminar de leitura).
2. O servidor Binas é o único “front end”, e garante que as escritas nas estações são atômicas (no sentido em que uma “QC write” para um utilizador nunca é feita em simultâneo a outra escrita ou leitura para o mesmo utilizador).
3. Assumimos que o servidor Binas só pode falhar depois de ter recebido a resposta ao último pedido feito.