

Sistemas Distribuídos Confiáveis

Tolerância a Faltas

Relatório de Desenvolvimento

António Anjo
a67660

João Monteiro
a67740

Ricardo Fernandes
pg32986

30 de Abril de 2017

Resumo

Relatório de desenvolvimento do primeiro trabalho prático da Unidade Curricular de Sistemas Distribuídos Confiáveis: Tolerância a Falhas. Neste documento apresentar-se-ão algumas das decisões tomadas durante o processo de desenvolvimento, assim como eventuais dificuldades encontradas.

Conteúdo

1	Introdução	2
2	Tolerância a faltas	3
2.1	Replicação	3
2.1.1	Mensagens Duplicadas	3
2.1.2	Replicação Passiva	4
3	Implementação e Funcionamento	5
3.1	Troca de Mensagens	5
3.2	Linhas e Segmentos	5
4	Conclusão	6

Capítulo 1

Introdução

O trabalho consiste na implementação em Java, usando o protocolo de comunicação em grupo Spread, de um par cliente/servidor tolerante a faltas.

A aplicação a desenvolver é um sistema de controlo de tráfego ferroviário. Considerando linhas ferroviárias divididas em segmentos numerados, o sistema deve garantir que nunca há duas composições num mesmo segmento assim como a existência de um segmento de linha livre entre duas composições.

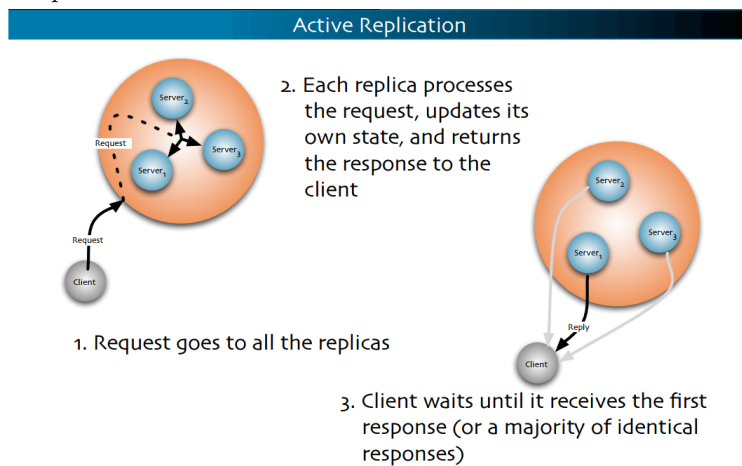
É esperado que composições possam efetuar pedidos de acesso a segmentos da linha, assim como enviar notificações de entrada ou saída de um segmento. O servidor deve poder manter uma lista de posições atuais das composições assim como um histórico de alarmes correspondentes a notificações de situações de perigo.

Capítulo 2

Tolerância a faltas

2.1 Replicação

O aspeto deste sistema que o torna tolerante a faltas é a utilização de um protocolo de **replicação ativa**, sendo a ideia deste protocolo que exista um grupo de no mínimo 2 servidores e, quando o cliente pretende fazer um pedido, este seja enviado a todas as réplicas.



Cada uma das réplicas processa então o pedido, atualiza o seu próprio estado e responde ao cliente. Fica assim ao cargo do cliente esperar pelas respostas que receber e tratá-las como entender.

2.1.1 Mensagens Duplicadas

É importante notar que, dado que os pedidos chegam a múltiplos servidores e todos eles vão responder, o cliente vai receber múltiplas respostas para cada pedido.

Se esta situação não for devidamente tratada, isto pode levar a problemas no sistema, como o cliente ler a resposta ao pedido errado.

Como solução a este problema, cada pedido é etiquetado com um número de sequência, que é copiado para a resposta pelo servidor. Assim, o cliente ignora todas as respostas cuja etiqueta não correspondam ao pedido esperado.

2.1.2 Replicação Passiva

Uma alternativa ao protocolo de replicação utilizado seria a **replicação passiva**. Este tipo de replicação introduz o conceito de servidor primário.

Neste tipo de replicação, o pedido do cliente chega apenas ao servidor primário, sendo aqui processado. O servidor atualiza o seu estado e envia depois uma mensagem de atualização de estado às réplicas. Cada réplica atualiza o seu estado e envia uma mensagem de acknowledgment ao servidor primário. Só nesta fase é que este envia a resposta ao cliente. Em caso de falha do servidor primário, uma das réplicas toma o seu lugar.

Este tipo de replicação tem a vantagem de ser transparente para o cliente e resolve o problema mencionado anteriormente das mensagens replicadas. No entanto, a sua dificuldade acrescida de implementação levou nos a manter o uso da replicação ativa.

Capítulo 3

Implementação e Funcionamento

3.1 Troca de Mensagens

A comunicação entre as composições e o sistema de controlo tráfego é feita utilizando o protocolo de comunicação em grupo Spread.

O Spread é um toolkit open source destinado à utilização em aplicações distribuídas que requerem alta disponibilidade, performance e robustez de comunicação entre grupos de intervenientes.

Na aplicação desenvolvida, os vários servidores constituem um SpreadGroup, isto é, um grupo para o qual uma mensagem pode ser multicast para todos os seus elementos simultaneamente. Cada composição tem a ela associada também o seu próprio SpreadGroup privado, o qual vai estar especificado nas mensagens que envia.

Assim, quando uma composição pretende comunicar com os servidores, faz multicast da sua mensagem para todas as réplicas, as quais vão processar o pedido, obter o grupo privado da composição que o enviou através do método `getSender` e transmitir para esse grupo a resposta ao pedido.

3.2 Linhas e Segmentos

Como sugerido no enunciado do trabalho, o sistema está pré-configurado com um conjunto de linhas e respetivos segmentos.

Construíram-se 3 linhas unidireccionais, sendo que os seus segmentos são numerados por ordem crescente no sentido da linha.

- Linha 1 - 6 Segmentos
- Linha 2 - 8 Segmentos
- Linha 3 - 11 Segmentos

Ao entrar no sistema, cada composição escolhe uma linha, um segmento de entrada e um segmento de saída, utilizando o sistema de controlo de tráfego para entrar na linha e navegar até ao seu destino.

Cada segmento tem a ele associado um valor binário que indica se está, em dado momento, ocupado ou livre.

Capítulo 4

Conclusão