Guião de demonstração - Tolerância a Faltas

Grupo T24 – Komparator

Obtenção do código:

1. No repositório do grupo do GitHub, cujo URL está no relatório de segurança, clicar em "Clone or download";
2. Clicar em "Download ZIP";
3. Extrair o conteúdo, uma pasta de nome T24-Komparator-master;

Instalação e compilação:

1. Abrir um terminal e entrar na pasta T24_komparator-master
2. Inserir no terminal o comando *mvn clean generate-sources install -DskipITs*
3. Esperar até surgir "BUILD SUCCESS" no terminal;

Demonstração do funcionamento normal da replicação:

1. Abrir mais 5 terminais e ir até à pasta T24-Komparator-master;

<u>Pôr os dois suppliers a correr:</u>

2. Abrir mais 4 terminais e ir até à pasta T24-Komparator-master;
3. No primeiro terminal inserir o comando *cd supplier-ws*
4. No primeiro terminal inserir o comando *mvn compile exec:java*
5. Esperar até surgir "Awaiting connections" no terminal;
6. No segundo terminal inserir o comando *cd supplier-ws*;
7. No segundo terminal inserir o comando *mvn compile exec:java -Dws.i=2;*
8. Esperar até surgir "Awaiting connections" no terminal;

<u>Pôr os dois mediators a correr:</u>

9. No terceiro terminal inserir o comando *cd mediator-ws;*
10. No terceiro terminal inserir o comando *mvn compile exec:java*
11. Esperar até surgir "Awaiting connections" no terminal;
12. No quarto terminal inserir o comando *cd mediator-ws*;
13. No quarto terminal inserir o comando *mvn compile exec:java -Dws.i=2*;
14. Esperar até surgir "Creating stub…" no terminal;

(Nota:Tanto nos dois suppliers como nos dois mediators é indiferente a ordem pela qual se põem a correr os mesmos).

Correr os testes da 2ºentrega para demonstrar replicação:

15. No quinto terminal inserir o comando *cd mediator-ws-cli;*
16. No quinto terminal inserir o comando *mvn install;*
17. Esperar até surgir "BUILD SUCCESS" no terminal;

O objectivo da demonstração da replicação normal é mostrar que, não ocorrendo nenhuma falta no sistema, estão a ser efectuadas com sucesso as actualizaçãos do mediator secundário. Para além do seu estado ser actualizado, o mediator seundário é responsável por detectar a falha do mediador primário. Para isso são mandadas provas de vida do mediador primário para o secundário. Esta troca de mensagens pode ser observada através dos prints que são feitos para o terminal. Para melhor observação dos mesmos pode-se comentar a cadeia de handlers das aplicações. (O *MessageIDHandlder* não pode ser comentado dado que é responsável pela transmissão dos identificadores dos pedidos entre o cliente e o servidor).

Demonstração da tolerância a faltas:

A tolerância a faltas do sistema será demonstrada causando a paragem súbita do servidor primário através de sigkill (CTRL-C).

Observação: Pode -se causar a paragem do servidor primário premindo a tecla "Enter". Apesar de ser uma paragem mais controlada faz com que o mediador secundário detecte também a falha no mediador primário e o substitua.

1.Tendo o código já compilado e instalado executar os passos da demonstração do funcionamento normal da replicação:

→ Pôr os dois suppliers a correr;

→ Pôr os dois mediators a correr;

→ Correr os testes da segunda entrega;

2. Durante a execução dos testes premir "CTRL-C" no terminal onde está a correr o mediador primário e verificar que esta pára abruptamente. O mediador secundário detecta a falha e publica-se no UDDI substituindo-o. O cliente do mediador liga-se ao UDDI para poder encontrar e ligar-se ao novo servidor (mediador secundário). Este processo pode ser observado através dos prints nos terminais e pela correcta continuação da execução dos testes. Tal como no caso anterior, pode-se comentar as cadeias de handlers excepto o MessageIDHanlder.

3.Esperar até surgir "BUILD SUCCESS" no terminal;

4.Poder-se-á se repetir a demonstração da tolerância a faltas mas executando apenas uma classe de testes ou mesmo apenas um teste. Deste modo é testado com mais precisão o correcto funcionamento do sistema (actualização de estado do mediador secundário e provas de vida). As operações que alteram o estado da aplicação e que requerem a chamada às funções de actualização são a “buyCart”, “addToCart” e “clear”. Estas operações são executadas nas classes BuyCartIT.java e AddToCartIT.java.

→ Correr apenas uma classe: mvn -Dtest=*classToTest* test

→ Correr apenas um teste: mvn -Dtest=*classToTest#tesToRun* test

5.No projecto foi implementada a semântica no-máximo-1-vez. Isto significa que no servidor existe um timeout para cada mensagem enviada. No caso da não obtenção de resposta a mensagem é reenviada até obtenção da mesma com mesmo identificador de pedido que a anterior. Este processo pode ser observado tal como nos casos anteriores através dos prints que são feitos para o terminal durante a execução dos testes, caso ocorra alguma situação de timeout.

6.Relacionado com o messageId pode -se verificar também que ao fazer CTRL-C, o numero de identificação da mensagem que está a ser enviada nesse momento é o mesmo da que será enviada para o mediador secundário depois de ele se registar no UDDI e substituir o primário.