TÉCNICO LISBOA

Projecto de Sistemas Distribuídos
Segunda entrega

Repositório A-02-12-14

Grupo de SD PagAmigo 12

Andre Bispo	66941	andrefsbispo@hotmail.com
Diogo Pinto	69905	diogo.reis.pinto@gmail.com
Ricardo Leitao	69632	ricardo.f.leitao@ist.utl.pt

Grupo de SD LargaCaixa 14

João Amorim	69310	joaoamorim_14@hotmail.com
João Silva	70038	silvajoaobruno@gmail.com
Pedro Soldado	70184	pedrosoldado_512@hotmail.com

Introdução

No projecto conjunto das cadeiras de Engenharia de Software e Sistemas Distribuidos foi proposta a implementação da rede social SoNet, que permitiria o uso de serviços externos de pagamentos (PagAmigo) e armazenamento de conteúdos(LargaCaixa). Na primeira entrega do projecto de SD, foi pedida a implementação dos serviços externos PagAmigo e LargaCaixa através de webServices. Porém, estas duas soluções desenvolvidas têm limitações importantes no que diz respeito a aspectos de tolerância a faltas e segurança.

A segunda entrega do projecto vai ao encontro destas limitações e do modo como podem ser previstas e colmatadas.

Transacções Distribuídas

Algoritmo de implementação de transacções distribuidas

Para a realização das transacções distribuidas, recorremos ao algoritmo Two-Phase Commit (2PC), que pressupõe a existência de um coordenador da transacção e um conjunto de participantes.

Numa primeira fase, o coordenador da transacção vai certificar-se que todos os participantes se encontram aptos para realizar uma transacção. Assim, o coordenador vai enviar um pedido a cada um dos participantes da transacção, que lhe deverão responder com um voto, positivo ou negativo.

No caso de todos os participantes terem respondido positivamente, o coordenador procede à ordem de commit de todos os participantes. Em caso contrário, encontramo-nos numa situação de tolerância a faltas: o coordenador pode optar por tentar enviar de novo os pedidos de

voto ou abortar a transacção. As decisões envolvidas na tolerância a faltas são um dos objectivos a desenvolver na nossa implementação.

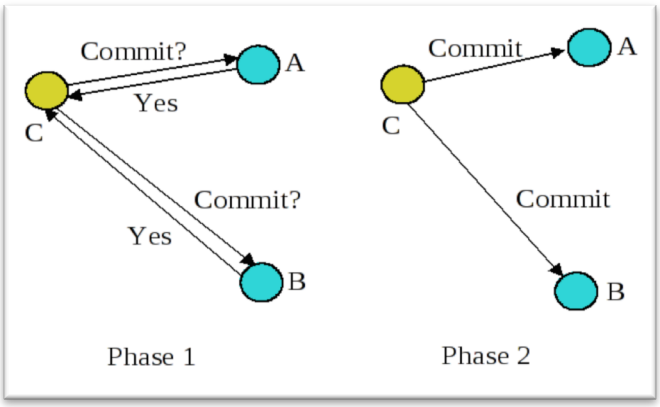

Figura 1 – Algoritmo de 2PC, com C a representar o coordenador da transacção e A e B dois participantes

Solução implementada

Para a implementação do algoritmo, o coordenador de todas as transacções é o serviço PagAmigo, visto ser o serviço que recebe todos os pedidos de pagamento e conhece os participantes nas transacções envolvidas. Estes participantes são os serviços bancários.

Como referido anteriormente foi necessário tomar decisões relativamente à tolerância a faltas. Tendo em conta o *trade-off* entre eficácia e eficiência de utlização do serviço existem dois extremos da solução: na percepção de uma falta o coordenador aborta de imediato a transacção ou tenta recuperá-la, repetindo sucessivamente o pedido até obter uma resposta afirmativa.

Na solução implementada tentamos encontrar um equilibrio entre estes dois extremos. No caso da existencia de uma falta o coordenador repete até duas vezes o pedido. Se no final destas duas tentativas não obtiver qualquer tipo de resposta a transação é abortada e realizado o respectivo rollback.

Foi necessário ainda implementar um mecanismo de timeout da ligação do PagAmigo com cada um dos bancos, de modo a definir qual é o tempo máximo que o Coordenador espera pela resposta de cada um dos participantes. O tempo de Timeout foi definido para 2 segundos, levando a que, mesmo que uma transacção seja abortada, o tempo que o cliente espera pela resposta nunca é superior a 6 segundos.

```
efectuaPagamento(origem, destino, montante) //Transfere montante de conta origem para destino
        While( tentativas <  3 )
                Resposta = Transfere(origem,destino);
                If ( !Resposta )
                        tentativas++;
                else
                        voto++;
        if(voto==2)
                commit();
        else
                rollback();
```

Figura 2 – Pseudo-Código da solução implementada

Relativamente aos pagamentos realizados no mesmo banco, isto é, quando a conta origem e destino pertencem ao mesmo banco, esta transacção é interpretada como local, não havendo necessidade de existência de um coordenador. Assim, este tipo de pagamentos ocorre de forma ACID mas não no âmbito do algoritmo 2PC.

Todos os requisitos de execução de transacções distribuídas e respectivas tolerâncias a faltas foram assim implementadas com sucesso no projecto.

Replicação

Nesta componente deveria ser assegurada a máxima disponibilidade do serviço LargaCaixa. Recorrendo ao protocolo de primary-backup, seria possível ocorrer uma falha (no servidor primário) sem que isso afectasse o desempenho do serviço.

É feito o deploy de dois servidores idênticos, sendo um deles o servidor primário (o que funciona no ínicio), e secundário (que substitui o primário quando ocorre uma falha neste). Apenas o servidor primário actual serve as invocações do cliente. Foram criadas duas bases de dados independentes, uma para cada servidor.

As duas bases de dados, mantêm-se actualizadas. Isto é conseguido duplicando a mensagem SOAP que é enviada, chegando também à base de dados do servidor secundário, que se actualiza.

Inicialmente optou-se por criar outro Web Service, um controlador que funcionasse como intermediário entre os servidores, e os mantivesse actualizados com o estado um do outro. No entanto, esta solução, como se compreendeu depois, não resolvia o problema em caso de falha (caso este controlador falhasse, perdia-se a ponte de comunicação entre os servidores).

Pensou-se então noutra alternativa para garantir que ambos conhecem o estado um do outro. Implementou-se uma classe de controlo, acessível a ambos os servidores, que informa o servidor secundário caso ocorra uma falha no servidor primário. Esta comunicação é conseguida utlizando multiplos endpoints nos servidores, o que permite ligar os dois servidores a outro endpoint (interface do controlador), estabelecendo-se assim a ponte de comunicação.

Em relação ao modelo de faltas, relacionado com a comunicação entre os servidores e a mudança de servidor primário quando ocorre falhas no actual, encontra-se implementado. No entanto, não foi possível testar exaustivamente o seu funcionamento, não sendo possivel assegurar que todas as situações de falha tenham sido analisadas e tidas em conta.

Segurança

Mecanismos de segurança no Serviço PagAmigo

Para a implementação de mecanismos de segurança no PagAmigo, utilizámos pares de chaves assimétricas e uma Autoridade de Certificação, uma entidade superior e fiável a todas as outras que tem como principais funções a emissão de certificados, gestão e revogação destes. No momento em que é iniciado, o serviço PagAmigo gera o seu par de chaves assimétricas, guardando para si a chave privada e enviando a sua chave pública para a Autoridade de Certificação.

Sempre que é efectuado um pagamento, é gerado um comprovativo, que será encriptado através do algoritmo RSA, utilizando a chave privada do PagAmigo e é feito o pedido à Autoridade de Certificação para assinar digitalmente o comprovativo.

Depois do comprovativo encriptado e assinado, este é enviado para o serviço LargaCaixa, onde deverá ser validado.

Relativamente ao serviço PagAmigo, todo o mecanismo de segurança de comprovativos foi implementado. Apesar de o mecanismo de segurança ter sido completamente implementado no âmbito do projecto, poderia ter sido melhorado. Podia-se ter procedido à autenticação de todos os bancos envolvidos em transacções, mas por falta de tempo, esta implementação não foi realizada.

Mecanismos de segurança no Serviço LargaCaixa

O grupo responsável pela implementação do serviço LargaCaixa ficou responsável pela autoridade de certificação (CA), e criação dos certificados digitais de chave pública.

Foi criada a classe da autoridade de certificação, sendo esta um Web Service acessível aos 2 serviços. Esta classe disponibiliza métodos de aquisição e armazenamento da chave pública do PagAmigo, de envio da chave privada da CA, e da criação de certificados sempre que o PagAmigo necessite (aquando da criação de um comprovativo de transacção).

A estrutura do certificado é baseada no certificado X509 (classe abstracta indicada na página da cadeira). Optou-se por não criar um certificado próprio pois, além de mais simples e possivelmente incompleto, o X509 é standard para grande parte dos exemplos de soluções estudados.

O grupo decidiu que faria mais sentido considerar que apenas existe uma chave pública para cada serviço PagAmigo, ao invés do explicado no enunciado, pois torna a gestão de chaves mais rápida e simples.

Esta funcionalidade não se encontra totalmente implementada, pois a informação obtida nos comprovativos do PagAmigo não é tida em conta quando se obtém o conteúdo adquirido no LargaCaixa (verificação da autenticidade).

No entanto, a distribuição e armazenamento tanto das chaves da Autoridade de Certificação como do PagAmigo e a criação de certificados encontra-se implementada.