Concretização de um protocolo de difusão atómica em sistemas com ligações intermitentes

Sérgio Cardoso
Escola Sup. Gestão de Santarém
Politécnico de Santarém
s.cardoso@mail.telepac.pt

Luís Rodrigues
Faculdade de Ciências
Universidade de Lisboa
ler@di.fc.ul.pt

Resumo

O artigo propõe a concretização de um protocolo de difusão atómica sobre um sistema de comunicação baseado em filas de mensagens, em particular sobre *Microsoft Message Queue Server*. Pretende-se deste modo oferecer um mecanismo de aplicação genérica, sobre um produto comercial de forte implantação, que permita suportar a coerência forte em redes que exibem ligações intermitentes.

1. Introdução

Cenários exibindo ligações intermitentes são comuns em sistemas de comunicação móvel, tornando complexa a coordenação entre os diversos nós. Por esta razão, adoptam-se frequentemente modelos de coerência fraca. No entanto, alguns problemas, como a gestão de dados replicados com equivalência a *uma-cópia* (isto é, que se comportam como se de uma cópia centralizada se tratasse), requerem protocolos que assegurem uma coerência forte.

Os sistemas de comunicação baseados em filas de mensagens fracamente acopladas, como Microsoft Message Queue Server, entre outros, apresentam a vantagem de suportarem interacção em tempo diferido, e parecem particularmente adequados a cenários exibindo ligações intermitentes, tais como a computação móvel (mas não só). No entanto, estes sistemas comerciais apresentam um suporte limitado para a coordenação *multi-participante* em sistemas tolerantes a faltas.

Este trabalho pretende aumentar um sistema de filas de mensagens com uma camada concretizando um protocolo de difusão atómica. Este protocolo assegura que todos os participantes de um grupo recebem as mesmas mensagens, exactamente pela mesma ordem. Esta primitiva é particularmente útil para concretizar a exclusão mútua distribuída ou para manter dados replicados com equivalência a *uma-cópia*.

O desenvolvimento deste serviço facilita a adição de características de tolerância a faltas em serviços construídos sobre o modelo da interacção baseada em filas de mensagens mas dependentes de um único servidor centralizado. Isto pode permitir, de um modo simples,

aumentar a disponibilidade de aplicações que necessitam de operar em redes sujeitas a ligações intermitentes, como é característico dos sistemas de computação móvel.

2. Sistemas baseados em filas de mensagens

Os sistemas baseados em filas de mensagens permitem a interacção entre componentes em tempo diferido e emergiram como uma alternativa viável ao modelo de chamadas a procedimentos remotos na construção de aplicações em que os clientes e os servidores não necessitam de interagir de modo síncrono. O modelo é particularmente adequado para as aplicações que estão sujeitas a uma conexidade intermitente ou/e ao baixo débito nas ligações. É este o caso dos ambientes de computação móvel e do funcionamento desligado.

Nestes sistemas os participantes comunicam através de um componente intermédio, designado por fila de mensagens. Um dos participantes, designado por produtor, coloca as mensagens na fila. Estas serão posteriormente retiradas da fila, pelo outro participante, designado por consumidor.

Na realidade, a fila de mensagens é uma abstracção, que é suportada num conjunto de gestores cooperantes que se executam nas máquinas dos produtores e dos consumidores. Deste modo, uma aplicação pode produzir mensagens mesmo que a máquina do consumidor esteja inacessível, depositando-as numa fila local. Quando a conexidade é re-estabelecida, o sistema encarrega-se de enviar a informação para um representante da fila na máquina do consumidor. Uma vez colocadas em qualquer fila, as mensagens adquirem persistência, uma vez que estas são construídas sobre meios de armazenamento estável.

Este modelo de comunicação por filas de mensagens é útil para o desenvolvimento de soluções em ambiente móvel, uma vez que permite esconder das aplicações as dificuldades inerentes à gestão da conexidade. A aplicação limita-se a produzir e consumir mensagens para e de um canal cujo representante local está sempre disponível. Por sua vez, o canal assegura a propagação das mensagens utilizando políticas adequadas à infra-estrutura de suporte à comunicação. No caso da computação móvel, em vez de exigir a manutenção de uma ligação activa entre o produtor e o consumidor, o canal de mensagens deve recorrer a modos de difusão alternativos, por exemplo, utilizando propagação do tipo epidémico. Finalmente, o canal assegura a tradução dos formatos da mensagens, fornecendo suporte à heterogeneidade.

3. Difusão atómica com ligações intermitentes

Uma solução para o problema da difusão atómica para sistemas móveis terá necessariamente de ser tolerante a faltas para permitir o progresso do sistema, mesmo que um dos componentes esteja falhado ou desconexo.

O problema da difusão atómica é, em muitos modelos, equivalente ao problema do acordo distribuído, o qual não possuí solução determinista em sistemas assíncronos sujeitos a falhas **[FLP 85]**. Felizmente, este tem sido alvo de um estudo aprofundado nos anos recentes, tendo imergido modelos que clarificam em que cenários é possível a sua resolução **[Chandra 96]**.

Este trabalho aproveita os resultados recentes obtidos nesta área **[Hurfin 98, Aguilera 98]**. Em particular, o trabalho irá concretizar o protocolo de difusão atómica para sistemas com falha e recuperação descrito em **[Rodrigues 98]**.

3.1 Protocolo de difusão atómica

O protocolo de difusão fiável funciona em rondas. Em cada ronda é acordado um conjunto de mensagens para entregar ao utilizador. A ordem total de entrega de mensagens é garantida pelo número de sequência da ronda e, dentro de cada ronda, por uma ordenação determinista das mensagens desse conjunto (usando um identificador único de mensagem).

De modo a assegurar que todos os participantes escolhem exactamente o mesmo conjunto de mensagens para entregar em cada ronda é utilizado um protocolo de acordo distribuído capaz de operar em sistemas com falhas e recuperações. Para a descrição do mecanismo de ordenação total basta referir a interface deste serviço, baseada em duas primitivas *"Propõe"* e *"Decide"* (a complexidade destes protocolos não permite a sua síntese no curto espaço disponível, remetendo-se o leitor para a bibliografia existente **[Hurfin 98, Aguilera 98]**). A primitiva "Propõe" aceita dois argumentos: (1) *um inteiro que identifica a ronda do protocolo de acordo que está em execução*, e, (2) *o valor proposto*, no nosso caso um conjunto de mensagens para ordenar. Quando o protocolo termina, a primitiva *"Decide"* retorna os identificadores das mensagens decididas.

Para conseguir o acordo, é necessário que existam períodos em que uma maioria dos participantes consegue interagir, caso contrário o progresso não é assegurado. Note-se que não é necessário que todos os participantes estejam simultaneamente acessíveis durante todo este período, basta que consigam trocar a informação necessária para a terminação do acordo.

Para garantir que todas a mensagens são ordenadas alguma vez, é necessário garantir a sua disseminação por todos os participantes. O serviço de filas de mensagens é utilizado também para suportar a difusão de mensagens a ordenar. Quando uma mensagem é recebida, é colocada numa fila de mensagens *"PorOrdenar"*. Em cada ronda de acordo, cada participante propõe as mensagens que estão na sua fila de mensagens por ordenar. As mensagens ordenadas após o acordo são removidas desta fila e entregues ao utilizador.

3.2 A arquitectura proposta

O serviço assenta em dois componentes fundamentais, tal como ilustrado na *Figura 1:*

a) um protocolo de transporte, concretizado pelo *MSMQ*, que garante a comunicação em modo assíncrono entre processos;
b) um protocolo para a resolução do problema da difusão atómica, designado *AB-MSMQ*.

O *AB-MSMQ* garante a difusão atómica da informação entre os vários *RED Server*, utilizando o modelo de comunicação assíncrona assente em filas de mensagens. *O Red Client* é um interface que implementa funcionalidades próprias de uma dada aplicação, capaz de comunicar com algum dos servidores existentes, mantendo com eles o mesmo modelo de

comunicação por filas de mensagens. A difusão atómica existe só no contexto da comunicação entre servidores replicados ou especificamente no modo de configuração *"agente-completo"*. Para os outros casos fala-se de difusão fiável.

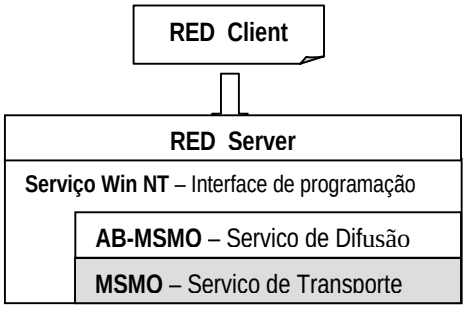

Figura 1 – Arquitectura do Serviço e integração com a aplicação RED

O serviço é inspirado no *Generic Multicast Transport Service (GTS)* proposto em **[Maffeis 96]**. Este é um serviço de transporte genérico destinado à difusão em grupo de mensagens, que oferece suporte a um conjunto alargado de protocolos de transporte. Em comparação, o *GTS* oferece suporte a um leque mais alargado de primitivas de comunicação. Por outro lado, o *GTS* utiliza mecanismos de ordenação elementares, assentes na existência de um processo coordenador com a função de determinar de modo centralizado a ordem relativa das mensagens, que não garantem o progresso perante a falha de um único nó. Ao recorrer ao consenso e à propagação das mensagens em modo epidémico, a solução proposta não sofre estas limitações.

Foi concebida uma aplicação distribuída para ilustrar o potencial do serviço. Trata-se do **RE**plicador (de) **D**ados – **RED**, uma aplicação destinada a permitir a distribuição de conteúdos, com equivalência a *uma-cópia*, por um número indeterminado de servidores, em condições de conectividade variáveis. O *RED* pretende demonstrar:

- A utilização do *AB-MSMQ* como meio de alcançar difusão atómica nestes sistemas.
- A adequação do modelo de filas de mensagens para a construção de serviços de transporte em sistemas assíncronos, e sujeitos a condições variáveis de conectividade.
- A boa utilização do serviço em ambiente de computação móvel.

De modo a ilustrar o funcionamento do pedido, iremos construir uma aplicação que permite realizar a reserva de recursos de modo distribuído. Este tipo de aplicação poderá ser útil em cenários onde agentes itenerantes promovem a venda de um conjunto finito de bens. Devido ao reduzido número de unidades disponíveis para venda, opta-se por não pré-reservar unidades para cada agente. Pelo contrário, as unidades disponíveis são geridas de um modo distribuído, usando o protocolo de difusão atómica para ordenar os pedidos de reserva.

Cada agente mantêm uma lista das unidades disponíveis, o que lhe permite obter uma estimativa da disponibilidade do pedido. Caso existam unidades, o agente envia um pedido de reserva, de acordo com o modo como tenha sido configurada a aplicação. Note-se que a garantia de reserva só é obtida após o pedido ser ordenado de ordem total, uma vez que vários agentes podem tentar reservar o mesmo item de modo concorrente. Dado que a ordenação

requer a conexidade de uma maioria dos participantes, poderá existir um intervalo de tempo significativo entre o momento em que os pedidos são conhecidos e o momento em que se consegue chegar a acordo acerca da sua ordenação. De modo a diminuir a incerteza da estimativa de disponibilidade por parte de cada agente, são dados a conhecer á aplicação os pedidos ainda não ordenados.

A aplicação pode ser configurada de dois modos distintos. No modo *"agente-completo"*, os agentes interagem directamente entre si para estabelecer a ordenação de mensagens. No modo *"agente-ligeiro"*, a ordenação é estabelecida por um conjunto de servidores com os quais os agentes comunicam usando mensagens ponto-a-ponto. A arquitectura permite não só que os agentes se desliguem dos servidores, e que contactem servidores diferentes em momentos diferentes, mas também que os próprios sofram desconexões.

4. Conclusões

Neste artigo é apresentada uma arquitectura para a concretização de um serviço de difusão atómica para sistemas sujeitos a ligações intermitentes. O sistema utiliza mecanismos de transporte baseados no modelo de interacção por filas de mensagens. Ilustra-se ainda o potencial do serviço com uma aplicação distribuída concebida predominantemente para a computação móvel.

Referências

[Aguilera 98] M. Aguilera, W.Chen and S.Toueg, Failure detection and consensus in the Crash-Recovery Model, Proc 12th Int. Symposium on DIStributed Computing, pp. 231-245, Setembro 1998.

[Chandra 96] T. Chandra e S. Toueg, Unreliable failure detectors for reliable distributed systems, ACM Journal, 43(2):225-267, Março 1996.

[FLP 85] Fischer, Lynch and Paterson, Impossibility of Distributed Consensus with one faulty process, ACM Journal, 32(2):374-382, 1985.

[Hurfin 98] M. Hurfin, A. Mostefaoui and M. Raynal, Consensus in Asynchronous Systems where processes can Crash and Recovery, Proc. 17th Symposium on Reliable Distibuted Systems, Outubro 1998.

[Maffeis 96] Silvano Maffeis , Walter Bischofberger and Kai-Uwe Mätzel. A Generic Multicast Transport Service to suport disconnected operation, ACM Wireless Networks Journal, 1996.

[Rodrigues 98] Luís Rodrigues and Michel Raynal. Non-blocking atomic broadcast in Asynchronous Crash-Recovery Distributed Systems, Technical Report DI-FCUL 99-1, 1999.

[Turek 92] J. Turek, The many faces of consensus in Distributed Systems, IBM T.J.W. Research Center, 1992.