

Sistema de Edição-Subscrição simplificado

Bruno Grilo
i29743@alunos.di.fc.ul.pt

Sara Águas
i31645@alunos.di.fc.ul.pt

Vasco Vilaça
i30393@alunos.di.fc.ul.pt

Grupo 06

Abstract

Os sistemas de edição/subscrição consistem na troca de mensagens baseadas em tópicos. Nestes sistemas, quando os editores pretendem editar algo acerca de determinado assunto, colocam uma etiqueta, com o nome do tópico (e no nosso caso em particular a indicação da qualidade serviço a ser utilizada), em cada mensagem. Os subscritores interessados, realizam a subscrição de um ou mais tópicos do seu interesse, passando a receber mensagens acerca dos tópicos escolhidos.

Existem diferentes formas de abordar este paradigma, uma é utilizando uma arquitectura do tipo cliente/servidor, onde o servidor serve de repositório de dados.

Este artigo apresenta um sistema de edição/subscrição simplificado, que deverá permitir a recepção, em tempo real, da informação, por parte dos subscritores activos. Deve oferecer, à partida, duas Qualidades de Serviço (QoS) distintas, mas deverá ser escalável para qualquer tipo e número de QoS. E, por último mas não menos importante, deverá ser tolerante a faltas.

1. Introdução

Em sistemas distribuídos deve-se garantir que quando um dos componentes falha, o sistema se mantém funcional e coerente. Os sistemas distribuídos tolerantes a faltas, se eficientes, oferecem-nos estas garantias, fazendo uso de técnicas de replicação. Esta técnica consiste na existência de uma cópia com informação acerca do estado actual do sistema em cada um dos seus processos. Isto permite que, ainda que um processo falhe, os outros possam continuar a sua computação de forma correcta.

Com o passar do tempo os sistemas computacionais têm evoluído para sistemas com requisitos temporais.

Neste contexto surgem os sistemas distribuídos de tempo real [5]. Estes sistemas fornecem garantias temporais, operando sobre um sistema de máquinas ligadas por uma rede, o que aumenta o nível de exigência em termos de comunicação e de complexidade da aplicação. Num mundo que sabemos, à partida imperfeito, esta necessidade conduz-nos a uma procura por arquitecturas de suporte à comunicação, de forma a aliviar a complexidade. Este artigo descreve a arquitectura escolhida para o desenvolvimento de um sistema de edição/subscrição de tempo real, com garantias de QoS. Neste sistema, existe uma entidade responsável pela criação de grupos. Estes grupos têm em comum, tópicos de interesse. Esta entidade é personificada por um conjunto de servidores, que tem a seu cargo a atribuição e divulgação dos grupos, bem como a obrigação de manter o seu repositório actualizado e consistente. Os grupos são constituídos por um ou mais elementos (editores e subscritores). A distribuição de mensagens é gerida pelo grupo, mas a comunicação é-lhes transparente, visto que a comunicação entre subscritores e publicadores não passa por este conjunto de servidores. Quando um editor publica uma mensagem, todos os restantes membros do grupo esperam e devem recebê-la, desde que estejam activos. Um grupo pode acoplar mais do que um tema, sendo possível a coexistência de subscritores com diferentes interesses no mesmo grupo, e é a plataforma de comunicação em grupo que assegura, que cada um deles só recebe a informação que subscreveu.

O APPIA é uma framework, desenvolvida em Java, cuja arquitectura modular assenta na construção de pilhas de protocolos, que podem ser construídas ou reconfiguradas em tempo de execução pelas aplicações, oferecendo-lhes diferentes qualidades de serviço. Estas pilhas de micro-protocolos assumem a forma de canais onde a informação circula sob a forma de eventos. No nosso sistema usare-mos esta framework para dar suporte à comunicação em grupo.

O restante artigo encontra-se estruturado da seguinte forma: a secção 2 descreve sucintamente o problema e os seus requisitos próprios, a Secção 3 refere Conceitos relevantes, a Secção 4 descreve aspectos relativos ao processo de comunicação, a Secção 5 descreve pormenores da arquitectura, a Secção 6 descreve , a secção 7 refere tolerância a faltas e tipos diferentes de falhas, a secção 8 conclui o nosso artigo, apresentando conclusões.

2. Descrição do problema e requisitos próprios

O sistema a implementar fornece serviços a dois tipos de clientes, são eles os editores e os subscritores. Aos editores é permitido publicar informação sob a forma de eventos que deverá ser recebida em tempo real por todos os subscritores interessados. Cada evento é distribuído com uma qualidade de serviço distinta, sendo possível ao editor escolher uma de duas, a saber: ordem total e sincronia na vista [Secção 3]. Esta qualidade de serviço é armazenada sob a forma de uma descrição XML da pilha APPIA que oferece essa qualidade de serviço. Esta opção irá facultar, no futuro, a extensão do sistema para qualquer tipo e número de QdS.

A um editor, enquanto tal é-lhe permitido anunciar e publicar. A acção de anunciar, consiste em contactar o servidor de forma a indicar o assunto a publicar e a respectiva qualidade de serviço que pretende utilizar, e obter destes o nome do grupo que deve utilizar para esse fluxo de informação. A acção publicar é a concretização do envio de uma mensagem para o grupo[Secção 3] appia em causa.

Um subscritor, apenas pode subscrever um ou mais tópicos. Para tal, necessita de contactar o servidor indicando o tópico de seu interesse, de forma a obter deles o nome do grupo e a QdS que lhe está associada. De seguida, deverá criar a sua pilha APPIA com o nome genérico e QdS associada, juntando-se ao grupo com esse nome. Acontece que no appia não existe um mecanismo explícito que permita juntar um elemento ao grupo. Desta forma, o elemento que pretende entrar num grupo cria uma vista em que é o único membro e espera que o mecanismo de união de vistas junte a sua vista à vista 'principal' (à do grupo que já está em funcionamento).

O servidor, é na realidade composto por conjunto de servidores que existe com o intuito de solucionar problemas de tolerância a faltas, assim cada um deles vai possuir a sua própria cópia local das associações entre assuntos e respectivas QdS e nomes de grupo, bem como a obrigação de manter o seu repositório actual-

izado e consistente. Ao conjunto de servidores cabe a tarefa da atribuição e divulgação dos nomes de grupo. Os servidores devem possuir um parâmetro de configuração, que permita controlar o número de nomes de grupo disponíveis, tendo em atenção que este número deve ser superior ou igual ao número de QdS oferecidas. O que não quer dizer que não podem existir mais assuntos a publicar, pelo contrário, se necessário um grupo pode ter mais que um assunto a ser publicado, desde que ofereça a mesma QdS. E nestes casos os subscritores devem estar preparados para filtrar, ignorando o que não querem. A plataforma de comunicação é responsável pela distribuição das mensagens segundo QdS distintas. Neste sistema, estarão inicialmente disponíveis a ordem total e a sincronia na vista. A saber: ordem total garante que as mensagens são entregues a todos os interessados pela mesma ordem e sincronia na vista garante que uma nova vista só é instalada depois de entregues todas as mensagens que foram enviadas na vista actual.

3. Conceitos relevantes

Aqui iremos definir um conjunto de conceitos que utilizaremos ao longo deste artigo.

3.1. Noção de Grupo

Um grupo é um sistema que engloba:

- suporte de filiação, o que permite adicionar/retirar elementos de um grupo e obter uma vista do grupo (elementos que compõem esse grupo num determinado instante);
- suporte de comunicação permite a comunicação entre todos os elementos do grupo, através de comunicação ponto a ponto ou multicast [5].

De forma a simplificar a comunicação entre editores e subscritores prevê-se vantajosa a utilização de um sistema de comunicação em grupo, o qual é conseguido com a utilização das camadas de suporte à comunicação em grupo, fornecidas pelo appia.

3.2. Ordem total

Este tipo de QdS garante que todos os processos entregam todas as mensagens pela mesma ordem [4].

3.3. Sincronia na vista

Este tipo de QdS garante que antes de uma nova vista ser instalada, todas as mensagens que foram pre-

viamente enviadas têm de ser entregues por todos os membros activos da vista anterior [4].

3.4. Replicação passiva

Neste tipo de replicação [5] e no nosso caso em particular, existe um coordenador com o qual são realizadas todas as comunicações. É ele o responsável pela gestão dos grupos appia. Todas as alterações realizadas na lista de grupos são transmitidas às suas réplicas, para que elas actualizem os seus dados, antes de ser dada uma resposta pelo coordenador ao utilizador (editor/subscritor). A comunicação para realizar estas actualizações só acontece quando existe alguma alteração dos dados. A falha do coordenador implica a eleição de um novo coordenador de entre as suas réplicas.

4. Processo de comunicação

No processo de comunicação - para suportar a comunicação recorre-se à utilização dos protocolos de comunicação em grupo existentes no Appia, afigura-se como uma boa solução, uma vez que se trata de implementações robustas e testadas. Além de que a utilização do mecanismo de vistas facilita largamente a entrada e saída de elementos de um grupo, o que corresponde exactamente às nossas necessidades.

4.1. Modelo de comunicação entre componentes

4.1.1 Servidor - servidor

A interacção entre servidores é fundamental para garantir a tolerância a faltas do sistema. Estas interacções são fundamentais para garantir a coerência entre as diferentes réplicas, de forma a garantir que, ainda que um servidor falhe o serviço se mantenha disponível. A disponibilidade do serviço é garantida enquanto pelo menos uma das réplicas (servidor) estiver em funcionamento. Por outro lado, no que diz respeito aos clientes (editores e subscritores) este funcionamento é transparente.

4.1.2 Editor/subscritor - servidor

O editor/subscritor comunica ponto a ponto com o servidor coordenador do grupo de servidores do sistema, para que possa criar ou juntar-se a um grupo appia.

4.1.3 Editor- subscritor

A comunicação entre estes dois intervenientes é assegurada pelas camadas de suporte à comunicação em

grupo do appia, segundo as QdS disponíveis (sincronia na vista ou ordem total). Nos eventos appia circulam mensagens que contêm as informações publicadas pelos editores e em resposta a estas, o subscritor retorna-lhe uma mensagem de confirmação de recepção, também ela dentro de um evento appia.

4.1.4 Criação de grupo

Falamos várias vezes ao longo deste artigo, na criação de grupos por parte do servidor, na realidade o servidor apenas regista internamente o grupo como criado, e atribui-lhe um nome e QdS. O grupo é realmente criado quando o primeiro processo (editor ou subscritor) se juntar a esse grupo.

5 Arquitectura

Para solucionar o problema proposto iremos utilizar uma arquitectura baseada em comunicação em grupo e replicação passiva.

Em cada grupo, implementado pela tecnologia appia, existe um coordenador. O coordenador será automaticamente eleito pelo protocolo responsável. Também o conjunto de servidores estará organizado desta forma. O coordenador dos servidores é ainda responsável por manter as suas réplicas actualizadas, deverá efectivar essas actualizações sempre que se registre alguma alteração [Secção 6]. Deve também utilizar mecanismos de controlo, para detecção de falhas das suas réplicas ou para que elas detectem a sua falha [Secção 7].

A entrada e saída dos grupos é tratada pelo protocolo de filiação de grupo do appia. Nestes casos, é criada uma nova vista e sempre que isto acontece o suporte de sincronia virtual bloqueia todos os membros através do evento 'BlockOk' [3], isto sem prejuízo da QdS associada ao grupo. Quando as condições estiverem criadas, a nova vista é então instalada e o envio de mensagens volta a ser possível.

6 Descrição de opções de implementação

As opções indicadas nesta secção são apenas um esboço do nosso plano e não uma imagem fidedigna da nossa concretização final.

No que se refere à tarefa realizada pelo conjunto de servidores acerca da atribuição e divulgação dos nomes de grupo. Os servidores devem possuir um parâmetro de configuração, que permita controlar o número de nomes de grupo disponíveis, tendo em atenção que este número deve ser superior ou igual ao número

de QdS oferecidas. Optamos por permitir, que uma vez atingido o número máximo de grupos autorizados, fosse possível adicionar novos tópicos a grupos existentes. Sendo possível que um grupo possa ter mais que um tópicos a ser publicado, desde que ofereça a mesma QdS. E nestes casos os subscritores devem estar preparados para filtrar, ignorando o que não querem. A plataforma de comunicação é responsável pela distribuição das mensagens segundo QdS distintas e fornece também mecanismos que permitem a filtragem de mensagens. Esta escolha tem a vantagem de poderem existir tantos tópicos quantos quisermos, a grande desvantagem é a quantidade de mensagens que podem ter de ser descartadas por parte dos subscritores. Isto acontece pelo facto de não possuímos informação, ao nível dos servidores, sobre a composição dos grupos, se a tivéssemos poderíamos reagrupar grupos tendo em conta a convergência de interesses dos elementos dos grupos a unir. Desta forma, seria possível reduzir drasticamente o número de mensagens descartadas. A abordagem mais simples para implementar esta nossa opção é fazer uso do algoritmo Round Robin [5].

No caso de um grupo deixar de ter elementos, optamos por mantê-lo, uma vez que o conjunto de servidores não tem acesso à composição dos grupos, ele apenas é responsável pela sua criação e divulgação. Assim, num grupo podem existir zero ou mais editores e zero ou mais subscritores.

Relativamente à entrega de mensagens, decidimos que não há obrigatoriedade de entrega de mensagens a um processo que falhou, logo quando um processo falhado recupera, não recebe as mensagens que circularam na sua ausência.

De forma a tolerar a falha de um ou mais servidores optou-se por sempre que um novo grupo é criado, actualizar todas as réplicas do servidor coordenador, antes da resposta ser enviada para o cliente (editor/subscritor), garantindo desta forma a coerência da informação. Para o efeito utilizamos comunicação em grupo, e replicação passiva [Secção 3]. Optamos por um esquema de mensagens planeadas ou seja, as mensagens de actualização apenas são enviadas para as replicas quando existirem alterações à lista de grupos. Poderíamos optar por utilizar checkingpointing, mas existia a possibilidade de ter-mos mensagens a ser enviadas sem que houvesse uma alteração efectiva do estado das réplicas.

Para que um editor possa publicar acerca de um tópico, deve contactar o coordenador do grupo de servidores afecto ao sistema, indicando-lhe o tópico e qualidade de serviço associada. Nesta altura uma das seguintes situações pode ocorrer:

- já existe um grupo acerca do tópico pretendido e com a mesma QdS e nessa altura o servidor limita-se a responder-lhe enviando-lhe o nome desse grupo;
- se ainda não existe um grupo com o tópico a publicar, o servidor cria um novo grupo com aquele tópico e QdS proposta, envia uma mensagem de actualização para as suas replicas e aguarda a sua confirmação de recepção, após o que responde então ao editor com o nome do grupo recém criado;
- se o tópico existe mas tem associada uma QdS diferente, o servidor informa o editor da situação fornecendo-lhe de imediato o nome do grupo detectado, cabendo a este a decisão de publicar ou não neste grupo segundo a QdS existente;
- se foi atingido o numero máximo de grupos, o servidor deverá utilizar o algoritmo de Round Robin para o agrupar com outro existente, devendo-lhe responder com a indicação de que é necessário agrupá-lo a um grupo existente, neste caso, se interessado o editor deve responder ao servidor pedindo o agrupamento. O servidor vai então actualizar as suas réplicas e responder-lhe com o nome do grupo a utilizar.

7 Tolerância a faltas

A falha de um editor ou de um subscritor não inviabiliza o funcionamento do sistema, uma vez que é permitido a existência de grupos só com subscritores, só editores ou sem nenhum elemento.

O ponto que pode ser um pouco mais crítico, tem a haver com a disponibilidade do servidor em determinados momentos, ou seja aquando da criação de grupos ou da obtenção da informação relativa aos grupos existentes. Por esta razão optamos ter, não um mas um conjunto de servidores agregados através de um grupo appia, em que existe um coordenador e um conjunto de réplicas actualizadas. Esta escolha tolera a falha de todos os servidores à excepção de um. Se todos falharem não é possível assegurar a criação e divulgação dos grupos, mas os grupos existentes mantêm o seu normal funcionamento. As falhas atrás descritas encaixam-se nas falhas omissivas, e são detectadas pela camada suspect do appia [3].

Outro tipo de faltas possíveis são designadas de assertivas, nestas estão englobadas: as sintácticas, que são normalmente detectadas pela linguagem java (linguagem escolhida para a implementação do sistema), através do levantamento de excepções que tornam essa interacção inválida; e as semânticas, por natureza mais

difíceis de detectar, que se caracterizam pelo envio de valores errados mas possíveis, uma solução para as detectar é utilizando replicação activa e redundância de operações (um pedido origina várias respostas, sendo que o valor correcto deduzido, por exemplo recorrendo a maiorias).

8 Conclusões

Neste artigo apresentámos um sistema de edição/subscrição simplificado, que permite a recepção, em tempo real, da informação, por parte dos subscritores activos. Oferece, à partida, duas Qualidades de Serviço (QoS) distintas, a saber: ordem total que garante, que as mensagens são entregues a todos os interessados pela mesma ordem; e sincronia na vista que garante que uma nova vista só é instalada depois de entregues todas as mensagens que foram enviadas na vista actual. É um sistema escalável para qualquer tipo e número de QoS, graças à utilização da plataforma de suporte ao desenvolvimento e execução de protocolos, o Appia. Utilizamos replicação passiva dos servidores, para prevenir a falha do sistema por falha de um componente crítico.

References

- [1] Nuno Carvalho, Filipe Araújo, Luís Rodrigues *"In-diQoS: um Sistema Publicação com Qualidade de Serviço"*, Julho 2003
- [2] Alexandre Pinto *"Group Communication Protocols Manual"*, Outubro 2005
- [3] API Specification for latest version (3.2.3) *"URL: <http://appia.di.fc.ul.pt/docs/javadoc/>"*
- [4] Rachid Guerraoui, Luís Rodrigues *"Introduction to Reliable Distributed Programming"*, Springer, 2006 - Cap 3., ISBN: 3-540-28845-7
- [5] Paulo Veríssimo and Luís Rodrigues *"Distributed System for System Architects"*, Kluwer Academic Publishers, ISBN 0-7923-7266-2
- [6] Tobias Oetiker, Hubert Partl, Irene Hyna, Elisabeth Schlegl *"The Not So Short Introduction to LATEX2"*, January, 1999