

Servidor Replicado de Registos

João Custódio
Faculdade de Ciências
Universidade de Lisboa
N.º 28509

João Lopes
Faculdade de Ciências
Universidade de Lisboa
N.º 25836

Pedro Catarino
Faculdade de Ciências
Universidade de Lisboa
N.º 26587

October 23, 2004

Abstract

Um servidor replicado de registos (SRR) permite armazenar informação em várias máquinas/processos distintos mantendo, na medida do possível, um estado coerente entre eles. Os servidores mantêm um conjunto de registos, que vão sendo alterados por clientes e, essas alterações, induzem uma mudança de estado nos servidores. Nos servidores, são contemplados dois modelos de funcionamento, que permitem a propagação das alterações efectuadas pelos clientes: a propagação melhor-esforço ("best-effort") e a atómica ("atomic"). Estes dois modelos de funcionamento descrevem formas distintas de processar uma alteração e definem o grau de coerência dos servidores.

1 Introdução

1.1 Contexto

Actualmente, existem diversas arquitecturas de comunicação, entre as quais se destacam aquelas que possuem uma elevada complexidade e aquelas que definem um "standard", pelo facto de serem as mais utilizadas. Tendo em conta o aumento do número de sistemas distribuídos existentes e uma necessidade cada vez maior de fiabilidade e portabilidade do "software", a área de tolerância a faltas distribuída está em constante evolução, com novos paradigmas a surgirem todos os anos. A aplicação descrita neste artigo, pretende demonstrar que é possível criar um sis-

tema, composto por várias máquinas/processos, onde existe um estado partilhado que permite a tolerância a ocorrência de faltas, ou seja, a falha num componente do sistema, não prejudica o seu funcionamento global.

1.2 Descrição Trabalho

Para a comunicação são considerados dois modos de funcionamento distintos: melhor-esforço e atómico. Estes modos de funcionamento vão definir a forma como a comunicação entre os servidores é efectuada, ou seja, o modo como é processada a alteração de um registo e, simultaneamente, o modo como o cliente observa essas mesmas alterações. Na definição dos modos de funcionamento é usada a tecnologia Apia (comunicação em grupo fiável). Assume-se que não existe nenhum modo mais forte (i.e., que garante com maior probabilidade a coerência do estado partilhado) do que outro, embora à primeira vista não o pareça. No modo atómico, uma alteração de estado é efectuada "simultaneamente" em todas as máquinas servidor, impossibilitando a ocorrência da situação em que o cliente efectua uma alteração a um registo numa máquina e, depois, obtenha outro valor para o mesmo registo quando se liga a outra máquina diferente. Aparentemente, este modo parece ser mais forte quando comparado com o modo melhor-esforço, onde não existe este tipo de preocupação e apenas se garante que, eventualmente, os estados dos servidores convergem para um estado partilhado e coerente. No entanto, o facto de um modo ser mais

forte do que outro não é relevante, se tomarmos em consideração a natureza das aplicações que podem correr sobre esta plataforma; no final, é a própria aplicação que dita qual o melhor modo de funcionamento para uma correcta execução da mesma. Para ilustrar esta ideia, pode-se pensar nos seguintes exemplos: numa aplicação bancária, o melhor modo será o atómico, para evitar que exista incoerência no estado de uma determinada conta; num serviço de "newsgroups", existe uma maior flexibilidade relativamente à actualização do estado, visto que não é crítico que, quando uma mensagem é entregue a um servidor, todos os clientes devem ver essa mensagem, independentemente do servidor a que se ligue.

1.3 Estrutura

Contents

1	Introdução	1
1.1	Contexto	1
1.2	Descrição Trabalho	1
1.3	Estrutura	2
2	Trabalho	2
2.1	Arquitectura	2
2.1.1	Comum aos dois modos de propagação	2
2.1.2	Propagação Melhor-Esforço	2
2.1.3	Propagação Atómica	2
3	Conclusão	3

2 Trabalho

2.1 Arquitectura

A concretização vai assentar sobre a tecnologia Appia, que fornece serviços de filiação e comunicação fiável em grupo. Sobre este suporte, vai ser definida uma camada protocolar que vai fornecer o serviço de replicação de servidores. Sobre este serviço vão existir dois modos distintos de operação, como se descreve a seguir.

2.1.1 Comum aos dois modos de propagação

Para garantir a coerência dos valores dos registos em todos os servidores, é necessário garantir que as escritas são efectuadas pela mesma ordem em todas as máquinas do grupo, de modo a evitar certos problemas. Por exemplo, supondo que um cliente altera na máquina s1, o registo r1, e em seguida, um outro cliente altera na máquina s2, o registo r1. Se não há garantia que as escritas são efectuadas pela mesma ordem em todos os servidores, o resultado é que vários servidores vão ter valores diferentes para o registo r1, fazendo com que se fique com um estado global incoerente. Para evitar estas irregularidades, é necessário que exista ordem total na entrega de mensagens, sempre que é efectuada uma escrita. Ou seja, os servidores efectuem as operações de escrita em que a ordem é idêntica em todos os processos correctos. Quanto à leitura, não é necessário que exista o mesmo tipo de garantia que é exigida nas operações de escrita, visto que o modo melhor-esforço define que um cliente pode ler valores diferentes do mesmo registo de várias máquinas.

2.1.2 Propagação Melhor-Esforço

No momento em que o servidor recebe um pedido para efectuar uma escrita, altera o seu registo interno e inicia processo de propagação da alteração pela rede para os outros servidores no seu grupo, de modo a informar os outros elementos da alteração efectuada e sincronizarem os registos com o servidor onde foi efectuada a alteração. Este processo de sincronização é efectuado da seguinte maneira: é enviada uma mensagem para cada servidor, sendo que a única garantia oferecida, é a de que, eventualmente, todos os servidores correctos irão receber a mensagem e alterar o seu registo para o valor indicado. Para efectuar a difusão das mensagens, usamos o algoritmo "Best-Effort Broadcast" [1].

2.1.3 Propagação Atómica

Neste modelo de funcionamento, garante-se que após um cliente ver uma actualização a um registo, qualquer cliente que leia esse registo vai ver essa actualização. A solução que encontramos foi utilizar ordem

total na entrega das mensagens ao grupo de servidores. Se todos os servidores receberem todas as mensagens pela mesma ordem vão sempre responder aos clientes de forma coerente. Assim para além de propagar as mensagens de escrita, também vamos ter de propagar as mensagens de leitura pelo grupo.

3 Conclusão

A aplicação descrita neste artigo, tenta demonstrar uma arquitectura capaz de criar um sistema distribuido, com um estado partilhado que é tolerante à ocorrência de faltas.

References

- [1] N. Carvalho e L. Rodrigues, *Implementing Reliable Broadcast In Appia - A Brief Tutorial*, 3rd November 2003