Tolerância a Faltas Distribuída: Pong

Alexandre Bernardo 26849 | Artur Ribeiro 28301 | João Carvalho 27847

Abstract

Hoje em dia e pelo facto dos sistemas distribuídos possuírem múltiplas partes de hardware e de software que funcionam conjuntamente, as hipóteses de algumas dessas partes falhar é bem maior do que num sistema simples.

Surgiu então, uma crescente necessidade de dotar as aplicações distribuídas, de mecanismos de tolerância a faltas. Neste artigo é apresentada uma proposta de desenvolvimento e integração de um jogo multi-utilizador tolerante a faltas, baseado em chamadas a procedimentos remotos ás replicações do servidor de jogo, e em particular, a optimização da comunicação entre clientes e servidores.

1. Introdução

A computação distribuída consiste em adicionar o poder computacional de diversos computadores interligados por uma rede de computadores para processar colaborativamente determinada tarefa de forma coerente e transparente.

Grande parte dos sistemas distribuídos de hoje, são baseados em chamadas a procedimentos remotos. Neste artigo é proposta uma solução para que possa ser criado um jogo multi-utilizador tolerante a faltas, baseado nas chamadas a procedimentos remotos.

A Chamada a procedimentos remotos é o tipo de protocolo para chamada remota de procedimentos em qualquer lugar da rede ou uma chamada de função para o método de transferência de controle de parte de um processo para outra, permite a divisão de um software em várias partes, partilha de arquivos e directórios, mas não é solução óptima para este tipo de aplicações, uma vez que se o servidor parar, as aplicações deixam de ter resposta e o sistema para. Para isso é necessário a introdução de mecanismos que permitam a tolerância a faltas.

Sistemas tolerantes a falhas são dispositivos que foram desenhados e construídos para operar com sucesso mesmo quando houver uma falha ou partes com defeito.

Neste caso iremos partir de uma aplicação distribuída, mas não tolerante a faltas. Serão utilizados 4 clientes e um servidor de jogo, que não contêm quaisquer mecanismos de tolerância a faltas.

Uma das formas de permitir a tolerância a faltas passa pela replicação dos servidores para que no caso de falha de um, os clientes não sejam prejudicados, assim o servidor deixa de ser um único processo e passa a ser um conjunto de processos replicados.

Assim sendo, iremos debruçar-nos sobre as componentes de gestão da distribuição e replicação, de modo a permitir a tolerância a faltas e à optimização da comunicação inerente ás aplicações distribuídas.

Nas secções seguintes serão apresentados os componentes da arquitectura, sobre os quais desenvolvemos o nosso sistema, as opções de implementação e possíveis soluções.

2. Arquitectura

2.1. Pong distribuído

A nossa versão deste jogo tem 4 jogadores (clientes) e um servidor. Cada cliente guarda o estado do jogo que lhe é enviado periodicamente. O servidor contém o estado do jogo, processando os pedidos dos clientes que o possam alterar, e reenviando o estado completo de aos clientes periodicamente. Os jogadores são assim, independentes sendo possível existir estado apenas com um jogador.

A comunicação efectuada, é inicialmente ponto-a-ponto, entre os clientes e o servidor, o que consititui o aspecto fundamental a ser alterado pela nossa proposta de arquitectura para garantir tolerância a faltas, visto que em caso de falha do servidor (único) todo o jogo seria comprometido.

2.2. Implementação

Para garantir tolerância a faltas no servidor, utilizaremos replicação activa do servidor. Podendo ter 2 ou mais servidores, o estado enviado pelos clientes terá de ser o mesmo para cada um, e terá de se garantir que cada cliente apenas recebe uma mensagem com cada estado. A solução para esta questão passa pelo utilização do paradigma de comunicação em grupos. Seria constituído um grupo por todos os servidores de jogo, para o qual os clientes enviariam os pedidos (os pedidos seriam enviado para um servi-

dor único que propagava o estado aos restantes servidores).

Para implementar este paradigma, utilizaremos a plataforma *Appia*, que é uma ferramenta de suporte de comunicação. A sua missão é definir uma relação padrão a ser respeitada por todas as camadas e facilitar uma comunicação entre elas.

Apesar do Appia estar direccionado para suportar diversos paradigmas da comunicação, é oferecido com suporte à comunicação de grupos. Para o conseguir, o *Appia* oferece diversos protocolos. Os protocolos foram desenvolvidos para fornecer todos os mecanismos necessários para reforçar a sincronização virtual.

A sincronização virtual é um paradigma da comunicação de grupo que basicamente apresenta todos os participantes de um grupo, permite ver a sociedade do grupo no formulário das vistas, e que todos vêem a mesma alteração da vista pela mesma ordem.

Este sistema concretiza o modelo tradicional cliente-servidor, com chamadas a procedimentos remotos (RPC´s), permitindo a tolerância a faltas através da replicação do servidor de jogo, de forma transparente para o cliente.

Para a concretização deste sistema, assumimos a comunicação em grupo, e iremos abordar as primitivas de comunicação bem como as diferentes técnicas para gerir a replicação do servidor de jogo. [1]

3. Conclusão

Para garantir tolerância a faltas distribuída, no nosso jogo Pong, utilizaremos replicação activa do servidor através do suporte para comunicaçao em grupos oferecido pela plataforma *Appia*.

Referências

[1] Alexandre Pinto, *Appia Group Communication*. http://appia.di.fc.ul.pt, Oct 2005.

[2] Fernando Felício, Susana Guedes, Valter Conceição, *Jogo Multi-Utiilizador Em Tempo-Real: 4Pong*. December 9, 2003.

[3] José Mocito, Liliana Rosa, Nuno Almeida, *Pong a quatro jogadores, distribuído e tolerante a faltas*.

[4] M. João Monteiro, Sandra Teixeira, *Tolerância a faltas em jogos de computador multi-utilizador*.

[5] Pedro Vicente, João Martins, *Arquitectura de um Sistema de Chamadas a Procedimentos remotos a Servidores Replicados*.