

Servidores 4Pong tolerantes a faltas

Francisco Azevedo
i28714@alunos.di.fc.ul.pt

Luís Aleixo
i27761@alunos.di.fc.ul.pt
Grupo TFD02

Miguel Seabra
i28292@alunos.di.fc.ul.pt

Abstract

O número de sistemas distribuídos de tempo-real tem vindo a aumentar com o tempo tal como a necessidade de fiabilidade inerente a estes. Neste artigo é apresentada uma implementação do jogo 4Pong onde um conjunto de servidores, que comunicam entre si, mantém uma cópia do estado do jogo. Deste modo a falha de um servidor não origina o fim do jogo.

1. Introdução

A necessidade de um sistema tolerante a faltas distribuído é particularmente útil em ambientes em que se pretende garantir o bom funcionamento do sistema aquando a falha de um componente. Uma das várias abordagens para resolver este problema é a replicação de informação, cada processo tem uma cópia do estado do sistema e quando um processo falha os outros podem continuar o seu correcto funcionamento.

Baseando-nos nesta técnica projectamos uma possível arquitectura para o 4Pong onde em um conjunto de servidores, cada um mantém uma cópia do estado actual do sistema e utiliza a comunicação em grupo para comunicar com os restantes servidores. Deste modo o número de servidores pode ser reduzido a um valor mínimo de "um", sem que se dê um falha do jogo.

A plataforma de suporte á comunicação utilizada no desenvolvimento deste projecto é o APPIA [1].

Este artigo está organizado da seguinte forma, a secção 2 descreve a forma geral do 4Pong. A secção 3 descreve a arquitectura do sistema. A secção 4 descreve os aspectos da comunicação fiável em grupo. A secção 5 descreve os mecanismos de coerência de estado. A secção 6 descreve os mecanismos de recuperação e detecção de falhas. A secção 7 contém detalhes particulares da implementação. A secção 8 contém uma descrição da pilha protocolar utilizada. Por fim a secção 9 são referidos alguns aspectos interessantes a desenvolver no futuro.

2. O jogo 4Pong

O 4Pong é uma evolução do jogo tradicional Phong. É jogado dentro de um quadrado em que cada lado funciona como uma baliza. Cada utilizador é associado a um lado do quadrado e controla um paddle, que protege a sua baliza e que apenas se pode mover paralelamente a sua parede. Existe uma bola que se move no campo que faz ricochete quer nas balizas quer nos paddles. Cada jogador protege a sua baliza com o paddle, e quando a bola bate na sua parede marca golo. No fim do jogo ganha o jogador que tiver a menor pontuação.

Apesar de ser um jogo com um máximo de quatro jogadores este pode ser jogado com número mínimo de dois jogadores. Durante o jogo é possível que saiam ou entrem jogadores, quando um jogador entra no jogo a sua pontuação é iniciada a zero.

3. Arquitectura do Sistema

A arquitectura do sistema consiste num conjunto cliente - servidor. De modo a que os jogadores não sejam afectados pelas falhas de um ou vários servidores torna-se necessário tolerar essas mesmas falhas. Para tal utilizaram-se métodos de comunicação em grupo e de replicação de servidores. Desta forma todos os servidores tem uma cópia do estado geral do sistema e comunicam entre si de modo a possibilitar a tolerância a falta de um ou mais servidores.

De modo a facilitar a comunicação utilizou-se o APPIA como plataforma de suporte á comunicação.

4. Comunicação Fiável em Grupo

Dado que a versão a ser implementada por nós é uma solução multi-servidor utilizaremos um sistema de gestão dos vários processos. O sistema APPIA fornece a funcionalidade de gestão de grupos com recurso a varias técnicas de sincronismo virtual [2].

4.1. Noção de Grupo

A utilização de sincronismo virtual [2] permite a cada utilizador de um dado grupo conhecer todos os participantes da sua vista e facilita a gestão de entrada e saída de novos elementos. A utilização deste tipo de técnicas permite também detectar falhas dos elementos do grupo.

5. Mecanismos de Coerência de Estado

No jogo 4Pong a cada instante o estado do jogo é constituído pela posição da bola, posição dos paddles dos utilizadores e pontuações. Visto ser um sistema que assenta numa rede de computadores é possível que duas mensagens cheguem a destinos diferentes por ordem diferente. Devido à existência de um conjunto de servidores replicados torna-se necessário evitar divergências de estados entre estes.

De forma a resolver este problema utiliza-se um protocolo de ordenação total, deste modo, todas as mensagens enviadas do grupo são entregues pela mesma ordem a todos os elementos desse mesmo grupo. É necessário também introduzir um coordenador de jogo. O papel deste consiste em determinar quando a bola deve ser movida e transmitir essa decisão ao grupo.

Dado que este projecto terá como base o APPIA não será necessário desenvolver protocolos de ordem total pois já são fornecidos pela aplicação em questão.

6. Falhas

No caso de processo falhar o jogo não é comprometido pois cada processo tem uma cópia do estado actual do jogo e não é dependente de outros. Deste modo mesmo existindo uma falha os outros processos continuam o seu correcto funcionamento.

6.1. Detecção de Falhas

Se um servidor falhar é necessário que os outros servidores tomem conhecimento dessa falha. Para tal usa-se o conceito de vista para detectar servidores que tenham saído do jogo. Sempre que uma nova vista é detectada cada processo compara os membros actuais com os membros da vista anterior determinando os servidores em falta.

6.2. Recuperação de Falhas

Quando um servidor se encontra em condições de funcionar correctamente tenta juntar-se ao grupo. As camadas de comunicação do sistema APPIA encarregam-se de proceder à actualização da vista e de darem a conhecer a todos os processos dessa vista as novas entradas.

Quando um processo reintegra uma vista não contem qualquer informação sobre o estado do sistema, deste modo o seu estado é actualizado mediante a comunicação com os restantes participantes dessa vista.

7. Aspectos particulares

Após se ter descrito de forma geral a arquitectura do sistema seguintes secções apresentam-se alguns detalhes particulares.

7.1. Eleição do coordenador

A forma como é eleito um coordenador assenta nos seguintes aspectos:

- Se o servidor for o único na sua vista então é coordenador.
- Se o servidor receber uma mensagem com o estado actual do jogo deixa de ser coordenador.
- Se o coordenador de uma determinada vista morrer então o primeiro elemento dessa vista assume-se como coordenador.

7.2. Sincronização do estado nos servidores réplica

O coordenador envia o estado do jogo aos restantes elementos da sua vista. No entanto, visto que a bola é um elemento em movimento e tendo em conta que a latência da rede é variável, não se pode garantir, à partida, que o estado de dois elementos da mesma vista seja igual. Logo, visto a bola ter velocidade constante, sabendo a direcção desta e a latência de rede entre dois elementos da vista pode-se calcular o espaço percorrido e a posição final da bola nos instantes em que a mensagem percorre a rede.

8. Pilha de Protocolos

No sistema implementado a pilha de protocolos tem as propriedades comuns de um sistema com sincronia virtual. Garante-se também ordem total na entrega de mensagens como já foi descrito anteriormente. A pilha protocolar da aplicação terá então o seguinte aspecto:

ApplicationLayer
TotalAbcastLayer
VSynLayer
StableLayer
HealLayer
InterLayer
IntraLayer
SuspectLayer
MergeOutLayer
GossipOutLayer
GroupBottomLayer
TCPCompleteLayer

A camada TCPCompleteLayer está responsável pela comunicação utilizando o protocolo TCP. O conjunto de 9 camadas acima desta corresponde á comunicação em grupo, a camada TotalAbcastLayer é responsável pela ordenação total das mensagens e finalmente a camada Application-Layer consiste na camada do jogo em si.

9. Referências

[1] L. R. Hugo Miranda, Alexandre Pingo. Application program interface specification of Appia version 1.2. Technical report, Universidade de Lisboa, July 2001.

[2] Pinto, "Appia Group Communication Manual", 2001.