

Instituto Politécnico do Porto
Escola Superior de Tecnologia e Gestão de Felgueiras



Disciplina de Sistemas Distribuídos

Ano Letivo de 2013/2014

P2World

Aplicação de troca de ficheiros em rede p2p

8090228, Luís Sousa

8090242, Ricardo Barbosa

Dezembro, 2013

Data de Recepção	
Responsável	
Avaliação	
Observações	

P2World

Aplicação de troca de ficheiros em rede p2p

8090228, Luís Sousa

8090242, Ricardo Barbosa

Dezembro,2013

Resumo

Aplicação criada no âmbito da unidade curricular de Sistemas Distribuídos utilizando os conceitos lecionados nas aulas durante o semestre.

O tema da aplicação assenta no princípio de comunicação em rede entre aplicações, para isso cada aplicação recorre a *sockets* e a protocolos de comunicação tais como UDP e TCP.

De forma a poder oferecer um serviço de *multitask* são utilizadas *threads* que nos permitem executar várias ações em simultâneo. Este funcionamento permite à aplicação a receção e envio de múltiplos ficheiros em simultâneo pois cada tarefa é executada independentemente das outras que também estão a ser executadas em paralelo.

1. Introdução

1.1 Contextualização

O sistema p2p não contém uma infraestrutura central, o que significa que não depende de um servidor para efetuar ações, em vez disso esse trabalho de servidor está distribuído por todos os *peers* presentes na rede.

Com uma arquitetura assim, o sistema fornece um maior nível de disponibilidade e de performance bem como um maior nível de segurança pois não sofre ataques centralizados como seria de esperar numa arquitetura cliente-servidor.

2. Funcionalidades

A aplicação dispõe das seguintes funcionalidades

2.1 Lista de ficheiros

- Efetuar pedidos de listas parciais a todos os *peers* que se encontrem ligados;
- Efetuar a transferência de todas as listas parciais existentes no momento do pedido;
- Atender o pedido de um outro *peer* a requisitar a sua lista parcial;
- Enviar a sua lista parcial para um ou múltiplos *peers*;
- Construir uma lista global de ficheiros com base nas listas parciais recebidas;

2.2 Troca de Ficheiros

- Efetuar um pedido a outro *peer* de um dado ficheiro presente na sua lista de ficheiros;
- Efetuar uma transferência de um ficheiro vindo de outro *peer*;
- Atender pedidos de transferência de ficheiros de um ou mais *peers*;
- Enviar ficheiros em simultâneo para múltiplos *peers*;
- Transferência de vários ficheiros em simultâneo a partir de múltiplos *peers*;

3. *Peer*

Cada *peer* está conectado a uma rede que é comum a todos os *peers* presentes na nossa aplicação. A partir desta rede é possível obter a lista de ficheiros de cada *peer* bem como é possível saber e guardar informações de cada *peer* para uma futura ligação.

O endereço da rede é, por defeito, 224.0.0.1 (*all-hosts group*).

As informações de cada *peer* consistem em saber o seu endereço e a porta em que está a aceitar pedidos de transferência de ficheiros, esta informação é vital pois será com base nesse endereço e porta que será estabelecida uma ligação para requerer uma transferência de um certo ficheiro.

Após estabelecida a conexão com o *peer* que tem o ficheiro pretendido, é apresentado um protocolo de comunicação onde é possível confirmar ou cancelar a transferência. Em caso afirmativo é então transferido o ficheiro, em caso negativo é cancelada toda a ação.

4. *Threads*

De forma a cada *peer* poder realizar tarefas de envio e receção em simultâneo são utilizadas *threads* que executam cada tarefa dessas paralelamente à execução do programa principal, ou seja, nenhuma outra tarefa é interrompida apenas porque é necessário obter a listagem de ficheiros (conceito de *multitasking*).

Com este tipo de implementação também é possível o envio/receção de vários ficheiros/listas em simultâneo recorrendo a uma *thread pool*.

4.1 *Threads* sempre em execução

- “Estar à escuta” de pedidos de envio de lista de ficheiros;
- Executar cada pedido de transferência de ficheiro;
- Encaminhar pedidos de transferência de ficheiros para serem executados por uma *thread pool*;
- Estar preparada para realizar a comunicação de forma a transferir um ficheiro;

4.2 *Threads* de ocasião

Para além das *threads* que estão em execução durante todo o ciclo de vida da aplicação, também são criadas *threads* para a realização de tarefas com execução finitas.

- Transferência/Envio de um ficheiro;
- Transferência/Envio de uma lista de ficheiros;
- Pedido de ficheiro/lista de ficheiros;

5. Sockets

5.1 MulticastSocket

É utilizado um *multicast socket* por cada *peer* de modo a ligarem-se à rede onde poderão obter as informações sobre os outros *sockets* ligados à mesma, informação vital para futuras ações de troca de ficheiros.

Este *multicast socket* utiliza o protocolo UDP (*User Datagram Protocol*), o que significa que não fornece nenhuma garantia que o pacote onde contém o pedido é recebido/entregue.

Por esta razão este *socket* apenas é usado para informar os outros *peers* que houveram pedidos de envio de lista de ficheiros.

5.2 Server Socket

Para troca de informação que não pode ficar perdida na rede, é utilizado um *server socket* que assenta no protocolo TCP (*Transmission Control Protocol*) que assegura o envio dos dados.

Este *socket* é utilizado para:

- Receber as listas parciais de cada *peer* da rede;
- Receber todos os pedidos de ficheiros;
- Envio de ficheiro;
- Utilização do protocolo de comunicação da aplicação;

5.3 Socket

Na outra extremidade da conexão existe um *socket* que se liga ao *server socket* previamente criado e é utilizado para troca de informação.

É utilizado para:

- Envio da lista de ficheiros para um outro *peer*;
- Envio de um pedido de ficheiro;
- Receção de um ficheiro;
- Utilização do protocolo de comunicação da aplicação;