Universidade do Porto

Faculdade de Engenharia

FEUP

Trabalho n.º 1: P2P - SDIS

Relatório de implementação

Sistemas Distribuídos

3.º ano do Mestrado Integrado em Engenharia Informática e Computação

Elementos do Grupo:

Carlos Frias – 040509116 – ei04116@fe.up.pt

Porto, 25 de março de 2012

1. Introdução

Com a elaboração deste trabalho pretende-se construir uma aplicação de partilha de ficheiros online, baseada em comunicação IP – multicast.

O mulicast é um serviço de rede no qual um único fluxo de dados pode ser enviado simultaneamente para vários destinatários. Este serviço apresenta várias vantagens sobre as tecnologias concorrentes, nomeadamente o unicast e o broadcast. Nestes dois últimos há um desperdício de largura de banda. Com o multicast, a infra-estrutura fica responsável por transportar o fluxo de dados para todos os receptores interessados, replicando apenas quando necessário. Por este motivo o multicast é geralmente direccionado para aplicações do tipo um-para-muitos e muitos-para-muitos (sendo este o caso da aplicação em causa). Com a utilização do multicast, o processo de transmissão simultânea para um grande número de receptores é facilitado. Numa rede multicast, pode-se enviar um único pacote de informação para diversos computadores, ao invés de enviar individualmente o pacote para cada um dos computadores.

A aplicação foi realizada em linguagem Java.

2. Implementação

2.1. Funcionalidades Implementadas

A aplicação permite as seguintes funcionalidades:

a) Obter a ligação a uma rede multicast de partilha de ficheiros, especificando o endereço multicast e dois portos para recepção e envio de mensagens: um porto para mensagens de controlo e outro para a transferência de dados;
b) Procurar ficheiros existentes em computadores nessa rede multicast, especificando palavras-chave, que possam estar contidas no seu nome;
c) Efectuar download de um ou mais ficheiros simultaneamente existentes na rede multicast, após pesquisa e ordem de transferência; Esta transferência é realizada recebendo segmentos do ficheiro desejado vindos de computadores distintos;
d) Efectuar upload para computadores da rede, de segmentos de ficheiros existentes na pasta de partilha do próprio computador. Esta transferência é transparente para o utilizador uma vez a aplicação não necessita de ordem deste para a realização da mesma;
e) Fraccionar ficheiros para transferência aquando da recepção de pedidos de transferência por parte de outros computadores da rede;
f) Reagrupar os segmentos transferidos para o computador num ficheiro, idêntico ao original, o qual se pretendia efectuar download;
g) Interagir com todo o processo através de uma interface gráfica construída para o efeito.

2.2. Arquitectura da aplicação

O projecto está organizado oito classes, a saber:

a) MainWindow: classe responsável pela apresentação da interface gráfica. Através dela o utilizador pode efectuar todas as funcionalidades.
b) Peer: classe responsável por toda a gestão de troca de mensagens e de dados transferidos, de e para outros computadores. É a classe mais importante do projecto.
c) PeerThread: classe utilizada para permitir o uso de Thread's indispensáveis para efectuar simultaneamente as operações: emissão de mensagens de controlo, recepção de mensagens de controlo, emissão de pacotes de dados e recepção de pacotes dados.
d) LittleEndian e LittleEndianConsts: classe utilizada para converter tipos de dados para o formato de array de bytes little endian. Estas duas classes foram retiradas de http://www.apache.org/licenses/LICENSE-2.0
e) FileManager: classe utilizada para fragmentar e juntar os fragmentos dos ficheiros a transferir.
f) FoundInfo: classe utilizada para guardar informação sobre cada um dos ficheiros que estão a ser transferidos de e para o computador.
g) MainTest: classe utilizada e modificada vezes sem conta aquando da implementação para debug e teste. Após a finalização do projecto esta classe não tem interesse prático.

2.3. Aspectos importantes da implementação

As principais operações e funções estão implementadas na classe Peer. As seguintes são aquelas que considero serem mais relevantes:

a) enviaProcura: função que envia para a rede uma mensagem com o intuito de procurar um ficheiro na rede;
b) enviaFoud: função que envia para a rede uma mensagem resposta a uma procura recebida.
c) enviaGET: função que envia para a rede um pedido de download de um ficheiro após recepção de pelo menos um found e pedido do utilizador.
d) enviaDados: função que envia um pacote de dados para a rede, contento um segmento de um ficheiro a transferir.
e) escutaMensagens: função que trata da recepção de pacotes de mensagens através do porto de controlo (podem ser: procuras, founds ou get's).
f) escutaDados: função que trata da recepção de pacotes de dados através do porto de dados.
g) processaMensagem: função que trata de interpretar (efectuar o parsing e definir as acções seguintes) as mensagens recebidas pelo porto de controlo.

h) processaDadosRecebidos: função que trata de interpretar (efectuar o parsing e definir as acções seguintes) as mensagens dados pelo porto de controlo.
i) action1: acção descrita da especificação do trabalho e que trata de escolher ao acaso um dos segmentos a transferir, transferir esse segmento e remove-lo da lista de segmentos pedidos.

3. Resultados de teste

Ao longo de da elaboração da aplicação e na fase de implementação fui fazendo diversos testes, nomeadamente de debug e resolução de conflitos. Após finalização do aplicativo realizei poucos testes de controlo do funcionamento. Não testei com vários computadores a correr a aplicação em simultâneo, mas testei no mesmo computador várias instâncias do programa correndo simultaneamente, onde se constatou a inexistência de problemas de maior.

4. Conclusões

A realização deste trabalho permitiu desenvolver os meus conhecimentos acerta dos protocolos UDP, nomeadamente a utilização de redes multicast, para além disso permitiu-me o aprofundamento dos conhecimentos sobre Threads em java e mecanismos de sincronização.

O trabalho poderia ser melhorado nos seguintes aspectos: implementação da acção dois descrita na especificação do projecto, refinamento do código utilizado, sinto que há muito código repetido e uma utilização nem sempre eficiente, melhoramento da interface gráfica que por escassez de tempo e por menor relevância para o projecto foi deixada para último plano.

Foram sentidas algumas dificuldades ao longo da realização do trabalho, uma vez que sou trabalhador estudante fiz grupo sozinho, pois nalgumas situações as dicas de um colega fizeram falta.

Em suma e tendo em conta os aspectos acima descrito, considero ter atingido os objectivos propostos para este trabalho.