Sistemas Distribuídos

Ano letivo 2019/2020

Relatório
Projeto de Sistemas Distribuídos
Meta 1

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Trabalho Realizado por:

Alexandre Maria Martins Magalhães Teixeira Serra – 2017248031

João Gabriel de Matos Fernandes – 2017247486

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Arquitetura de software

Multicast Server

Threads:

- Multicast Server → É a thread principal do multicast server. Em primeiro lugar, obtém o número do Multicast Server. De seguida, vai buscar a informação aos ficheiros e inicia as restantes threads. Após isto, está à escuta de mensagens provenientes do grupo Multicast.
- Multicast Server Action → executa cada pedido recebido pela thread Multicast Server.
- Web Craweler → faz uma indexação pedida por um administrador
- Multicast Server Control → periodicamente, manda mensagem por multicast aos restantes servidores multicast e aguarda, durante 5 segundos, por uma resposta de todos. De seguida, verifica se todos responderam. Se algum não respondeu, manda mensagem ao servidor RMI a avisar que o número do servidor Multicast que foi abaixo.
- Synchronization → periodicamente, sincroniza com os outros multicast servers informação sobre a indexação feita, URL's, pesquisas efetuadas e utilizadores.
- MulticastAdminPage → periodicamente, atualiza o top 10 das pequisas efetuadas e dos URL's com mais links a apontar para eles.

Sockets:

- Multicast Socket → recebe mensagens do RMI server e de outros multicast servers, respondendo pelo mesmo
- TCP Socket → usado para efetuar a sincronização entre os multicast servers

Organização do código (main e classes):

- MulticastServer → Programa a correr para iniciar o multicast server. As funcionalidades são iguais à thread Multicast Server.
- MulticastServerInfo → classe com o número do servidor, endereço TCP, porto TCP e carga do servidor.
- Search → classe com pesquisa e número de vezes que foi pesquisada.
- URL → classe URL com o URL, título e uma citação do texto da página, o número de links e os mesmo que apontam para este URL.
- User → classe com as informações do utilizadores (username, password, se é administrador, histórico de pesquisas, se está logado, o número de cliente e se tem notificação)

RMI Server

Threads:

- RMIMulticastManager → criada pelo RMIServer, é usada para gerir os MulticastServers, atribuindo-lhes um numero identificador único, mantendo uma lista de MulticastServers ativos e recebendo updates das Real Time Statistics.

Sockets:

- RMIMulticastManager, RMIServer → recebem/enviam mensagens do/para os MulticastServers. O RMIServer traduz essas mensagens em "mensagens" rmi para o RMIClient.

Organização do código:

- RMIServer → mantém a comunicação direta com todos os clientes.
- RMIMulticastManager → thread que recebe e trata de mensagens assíncronas específicas.
- ServerInterface → interface que contém todos os métodos remotos implementados no RMIServer.

RMI Client

Organização do código:

- RMIClient → mantém a comunicação através de rmi com o RMIServer.
- UI → classe que contém toda a interface com o utilizador.
- ClientInterface → interface que contém todos os métodos remotos implementados no RMIClient.

Funcionamento do Servidor Multicast

O funcionamento do servidor multicast baseia-se nas 6 threads que contém, já previamente explicadas, e nos ficheiros de objetos guardados em disco. Estes estão explicado detalhadamente no Javadoc dos ficheiros.

Para as threads funcionarem corretamente, foi estabelecido um protocolo de mensagens que são trocadadas entre os servidores multicast e o servidor RMI. O protocolo está detalhadamente especificado de seguida. No nosso protocolo usámos uma mensagem semelhante a um dicionário, em que temos um valor e uma chave, separados por "|||", e as várias chaves separadas por ";;".

Registo

Pedido do servidor RMI:

type|||register;;clientNo|||*clientNo*;;username|||*username*;;password|||*password*

Respostas do Multicast Server:

Registo aceite:

type|||registerResult;;;clientNo|||*clientNo*;;;status|||valid;;username|||*username*;;isAdmin|||*boolean*

Registo com username já existente:

type|||registerResult;;clientNo|||*clientNo*;;status|||invalid

Login

Pedido do servidor RMI:

type|||login;;clientNo|||*clientNo*;;username|||*username*;;password|||*password*

Respostas do Multicast Server:

Login válido:

type|||loginResult;;clientNo|||*clientNo*;;status|||valid;;username|||*o username*;;isAdmin|||*boolean*;;notification|||*boolean*

Login Inválido:

type|||loginResult;;clientNo|||*clientNo*;;status|||invalid

Index

Pedido do servidor RMI:

type|||index;;clientNo|||*clientNo*;;serverNo|||*serverNo*;;url|||*url*

Respostas do Multicast Server:

type|||indexResult;;clientNo|||*clientNo*;;status|||started

Search

Pedido do servidor RMI:

Utilizador sem o login feito:

type|||search;;clientNo|||*clientNo*;;word|||*words*

Utilizador com o login feito:

type|||search;;clientNo|||*clientNo*;;word|||*words*;;username|||*username*

Respostas do Multicast Server:

type|||searchResult;;clientNo|||*clientNo*;;urlCount|||*numero de urls*;;title_0|||*1º título*;;url_0|||*1º url*;;text_0|||*1º citação de texto;;title_1|||*2º título*;;url_1|||*2º url*;;text_1|||*2º citação de texto* etc.

Search history

Pedido do servidor RMI:

type|||searchHistory;;clientNo|||*clientNo*;;username|||*username*

Respostas do Multicast Server:

type|||searchHistoryResult;;clientNo|||*clientNo*;;searchCount|||*numero de urls*;;search_0|||*1º url*;;search_1|||*2º url*;; etc.

URL's a apontar para um dado URL (Links Pointing)

Pedido do servidor RMI:

type|||linksPointing;;clientNo|||*clientNo*;;url|||*url*

Respostas do Multicast Server:

type|||linksPointingResult;;clientNo|||*clientNo*;;linkCount|||*numero de links*;;link_0|||*1º url*;;link_1|||*2º url*;; etc.

Promoção a administrador

Pedido do servidor RMI:

type|||promote;;clientNo|||*clientNo*;;username|||*username*

Respostas do Multicast Server:

Resposta em que o novo admin não está online:

type|||promoteResult;;clientNo|||*clientNo*;;status|||valid

Resposta em que o novo admin está online

type|||promoteResult;;clientNo|||*clientNo*;;status|||valid
;;newAdminNo|||*newAdminNo*

Resposta em que o user já é admin

type|||promoteResult;;clientNo|||*clientNo*;;status|||invalid;;message|||User is already admin

Reposta em que o user não existe

type|||promoteResult;;clientNo|||*clientNo*;;status|||invalid;;message|||That user doesn't exist

Real Time Statistics

Pedido do servidor RMI:

type|||rts;;clientNo|||*clientNo*

Respostas do Multicast Server:

type|||rtsResult;;clientNo|||*clientNo*;;url_0|||*1º url*;;...;;url_9|||*9º url*;;search_0|||*1ª word no top*;;...;;search_9|||*Última word no top*;;address|||*1º endereço*;;port|||*1º porto*;; etc.

Update em tempo real

type|||rtsUpdate;;clientNo|||0;;url_0|||*1º url*;;...;;url_9|||*9º url*;;search_0|||*1ª word no top*;;...;;search_9|||*Última word no top*;;address|||*1º endereço*;;port|||*1º porto*;; etc.

Logout

RMI server manda mensagem a informar que utilizador deu logout:

type|||logout;;clientNo|||*clientNo*;;username|||*username*

Multicast Server confirma a operação

type|||logoutResult;;clientNo|||*clientNo*;;status|||valid

Multicast Server Start

O multicast server avisa o rmi server que se ligou:

type|||multicastServerStarter;;ipAddress|||*endereço ip do pc onde está a correr o multicast server*;;porto|||*porto definido*

O RMI responde com o numero atribuído a este multicast server e os endereços e portos dos outros servidores Multicast:

type|||multicastServerStarterResult;;serverNo|||*o numero do multicast server*;;serverCount|||*numero de multicast servers existentes*;;serverNo_0|||*nº do 1º*;;ip_0|||*ip do 1º*;;porto_0|||*porto do 1º*

Mensagens trocadas entre multicast servers para verificar que estão vivos

Multicast server manda mensagem ao restantes para responderem

type|||checkStatus

Multicast servers respondem a confirmar que estão vivos

type|||checkStatusConfirm;;serverNo|||*serverNo*

Mensagem enviada para o RMI server se detetar que um multicast server foi down

type|||multicastServerDown;;serverNo|||*serverNo*

Mensagem trocadas entre Multicast servers e RMI server quando RMI server começa

RMI server avisa que ligou

type|||rmiServerStarter

Resposta do Multicast server com endereços, portos e números dos Multicast Servers

type||| rmiServerStarterResult;;clientNo|||0;;serverCount|||*número de servidores*;;serverNo_0|||*número do 1º servidor*;;address_0|||*endereço do 1º servidor*;;port_0|||*porto do 1º servidor*;; etc.

Funcionamento do Servidor RMI

O servidor RMI contém uma série de métodos remotos impementados numa interface (ServerInterface) que são chamados pelo Client RMI. Estes métodos, por norma, preparam a mensagem com os parâmetros a enviar e enviam uma mensagem *multicast* para todos os servidores multicast. Após o envio, fica à espera de receber uma mensagem de resposta, (ex. para login a resposta é loginResult) sendo esta depois devolvida ao cliente RMI que por sua vez mostra o resultado ao utilizador através da linha de comandos.

Funcionamento do Cliente RMI

O cliente RMI contém uma interface através de linha de comandos para interação com o utilizador. De forma geral, sempre que uma opção é selecionada é chamado um método remoto do servidor RMI sendo que a resposta a essa chamada é a mensagem de resultado. O cliente RMI depois processa a resposta para interagir e reproduzir feedback da melhor forma possível para o utilizador.

Distribuição de tarefas

- Alexandre Serra → cliente e servidor RMI
- João Fernandes → servidor Multicast

Testes realizados

Registar novo utilizador	✓
Acesso protegido com password (exceto pesquisas)	✓
Indexar novo URL introduzido por administrador	✓
Indexar iterativamente ou recursivamente todos os URLs encontrados	✓
Pesquisar páginas que contenham um conjunto de palavras	✓
Resultados ordenados por número de ligações para cada página	✓
Consultar lista de páginas com ligações para uma página específica	✓
Consultar lista de pesquisas feitas pelo próprio utilizador	✓
Dar privilégios de administrador a um utilizador	✓
Página de administração atualizada em tempo real	✓
Notificação imediata de privilégios de administrador (online users)	✓
Entrega posterior de notificações (offline users)	✓
Avaria de um servidor RMI não tem efeito visível nos clientes	✓
Servidor RMI secundário testa e substitui o primário em caso de avaria longa	✓
Em caso de avaria longa os clientes RMI ligam ao secundário (sessão mantida)	✓
Avarias temporárias (<30s) dos servidores multicast são invisíveis para clientes	✓ (1)
Pedidos são garantidamente processados por N>=1 servidores multicast	✓
Pedidos de indexação são respondidos apenas por um servidor multicast	✓
O serviço funciona se houver pelo menos um servidor multicast disponível	✓
Os servidores multicast recuperam de disco o seu estado se avariarem	✓
Cada servidor multicast replica a sua parte do índice por outros servidores (TCP)	✓
Cada servidor distribui URLs para serem indexados por outros servidores	✗
O servidor RMI original, quando recupera, torna-se secundário	✓
[Extra] Balanceamento da carga dos servidores multicast	✓

(1) O client tem de efetuar novamente o pedido