

Googol

Sistemas Distribuídos

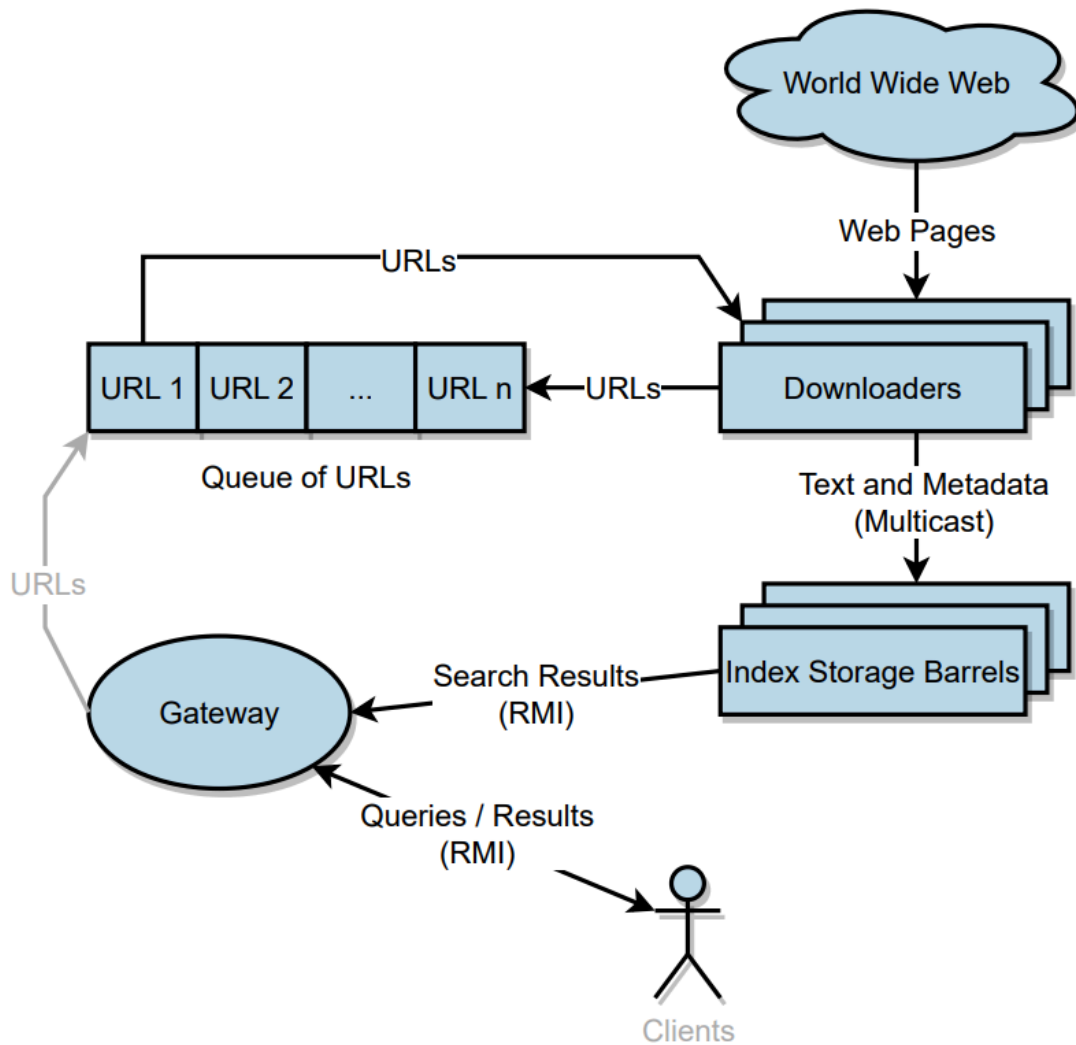
Trabalho realizado por:

- Lucas Miguel Simões Loberto nº 2021219107
- Simão Tadeu Ricacho Reis Moreira nº 2020218319

Índice

Arquitetura do Software.....	2
Funcionamento do Multicast.....	4
Mecanismo Failover.....	5
Testes feitos à Plataforma.....	7

Arquitetura do Software



Downloaders e Barrels - A comunicação entre os barrels e os downloaders é feita através de um grupo multicast, onde as threads têm acesso às mensagens enviadas. De modo a saber de onde vêm as mensagens e quem tem que as ler, criámos um protocolo que será explicado mais à frente neste relatório.

Cliente e SearchModule - O Client vai enviar as pesquisas e escolhas do menu através de RMI para o Search Module dando trigger às funções criadas no server.

Downloaders e URLQueue - De forma aos downloaders terem acesso à mesma URLQueue, esta é acessada através de RMI.

SearchModule e Barrels - De modo a obter a informação que está guardada nos barrels, para as pesquisas do Client, o Search Module , através de uma conexão RMI, chama funções escritas nos barrels para receber a informação guardada nas HashMaps.

WWW e Downloaders - Para descarregar a informação de uma página, utilizamos a biblioteca jsoup, que nos permite, a partir de um link, acessar e obter toda a informação que queremos dessa página, como links associados, palavras , título e descrição.

Barrels e Files - De modo a guardar toda a informação quando o programa é desligado, escrevemos as HashMaps em ficheiros de texto sempre que toda a informação é completamente enviada pelos downloaders

Funcionamento do Multicast

O multicast é utilizado para a comunicação entre os Downloaders e os Barrels (IndexStorageBarrels) no sistema apresentado. A classe Downloader representa uma thread responsável por baixar e processar páginas web. Esta conecta-se a uma fila de URLs através de RMI e envia mensagens para os Barrels usando multicast.

No construtor da classe Downloader, é criado um MulticastSocket que se junta a um grupo multicast especificado pelo endereço e porta multicast. Isto permite que o Downloader envie mensagens para todos os membros do grupo multicast.

Durante a execução do método run() do downloader, este obtém URLs da fila de URLs através de RMI e processa-os. Para cada URL processado, o Downloader extrai informações como título, citação e palavras da página web.

Após processar um URL, o Downloader envia as informações extraídas para os Barrels usando o método enviarParaBarrels(). Esse método cria mensagens contendo os links encontrados, as palavras extraídas e o snippet de texto da página web.

O método sendMessage() é responsável por enviar as mensagens para os Barrels usando multicast. Este usa um semáforo para garantir que apenas um Downloader envie mensagens por vez, evitando colisões. Em seguida cria um DatagramPacket que contém a mensagem e envia-o através do MulticastSocket para o grupo multicast.

A classe IndexStorageBarrels representa os Barrels que recebem as mensagens enviadas pelos Downloaders. Cria um registo RMI para permitir a comunicação com os Downloaders e outros componentes do sistema.

Cada Barrel é representado por uma instância da classe Barrel, que é iniciada como uma thread separada. Os Barrels juntam-se ao mesmo grupo multicast usado pelos Downloaders para receber as mensagens.

Quando uma mensagem é recebida por um Barrel, é processada de acordo com o seu tipo (links, palavras ou snippet de texto) e as informações são armazenadas nos índices correspondentes do Barrel.

Os Barrel também fornecem métodos RMI para permitir a pesquisa de informações, como obter os links relacionados a uma determinada palavra-chave.

Mecanismo Failover

Falha durante uma pesquisa:

Caso ocorra uma 'RemoteException' durante uma pesquisa num Storage Barrel, a Gateway deve recuperar fazendo o mesmo pedido a outro Barrel disponível. Isto é implementado no método `selecionarBarrel()` da classe `IndexStorageBarrels`. Se o Barrel selecionado aleatoriamente não estiver vivo (ou seja, lançar uma `RemoteException`), ele é removido da lista de Barrels disponíveis e outro Barrel é selecionado recursivamente. Essa abordagem garante que, mesmo que um Barrel falhe durante uma pesquisa, a Gateway tentará outro Barrel disponível para concluir a operação.

Falhas do lado do cliente:

O método `serverErrorHandler()` é responsável por lidar com falhas na conexão com o servidor. Se ocorrer uma exceção ao tentar se conectar ao servidor, o cliente entra num loop onde vai tentando uma reconexão periodicamente até que a conexão seja restabelecida.

Além disso, o código do `RMIClient` faz uso extensivo de blocos try-catch para lidar com exceções que podem ocorrer durante a execução, como erros ao ler o input do usuário ou ao comunicar com o servidor. Esses blocos try-catch ajudam a evitar que o cliente interrompa abruptamente em caso de exceções e permitem que ele continue a execução ou tome ações melhoradas, como exibir mensagens de erro para o usuário.

Essas abordagens garantem que, mesmo que ocorram falhas do lado do cliente, como perda de conexão com o servidor, o cliente tentará recuperar e continuar a operação.

Falha de um Storage Barrel:

Na classe `RMIServer`, o método `indexar()` adiciona a URL à fila usando o método `inserirLink()` da interface `URLQueueInterface`. Se ocorrer uma exceção ao adicionar a URL à fila, a mensagem de erro é retornada.

Uma possível implementação de failover seria envolver a chamada ao método `inserirLink()` em um bloco try-catch. Se ocorrer uma exceção, o servidor poderia tentar novamente a indexação após um intervalo de tempo, possivelmente selecionando outro Storage Barrel disponível usando o método `selecionarBarrel()` da interface `RMIServerInterface`.

Além disso, a classe URLQueue poderia implementar um mecanismo de persistência para armazenar as URLs pendentes em caso de falha, garantindo que elas não sejam perdidas e possam ser processadas posteriormente.

Recuperação da Gateway:

Se ocorrer uma RemoteException ao criar o registro RMI na Gateway, será tentado novamente após um intervalo de 1 segundo.

Isto é feito no método tentarNovamente() da classe IndexStorageBarrels. Se a conexão com o servidor RMI for perdida, a Gateway tentará se reconectar periodicamente até que a conexão seja restabelecida.

Essa abordagem garante que, mesmo que a Gateway falhe e se recupere, ela tentará se reconectar ao servidor RMI para continuar operando normalmente.

Testes feitos à Plataforma

No final da elaboração da estrutura e do código, dedicamo-nos a realizar uma série de testes abrangentes. O objetivo era compreender as capacidades e limitações do produto desenvolvido, resolver eventuais problemas que surgissem e aprimorar áreas específicas da plataforma conforme necessário.

Testes de registo:

- ☒ Utilizador não se consegue registar se as credenciais forem inválidas
- ☒ Utilizador não se consegue registar se, quando este se tentar registar, já existir um utilizador com o mesmo username na base de dados
- ☒ Utilizador consegue-se registar se as credenciais forem válidas, guardando assim as credenciais num ficheiro de texto

Testes de login:

- ☒ Utilizador não consegue fazer login se pelo menos uma das credenciais não corresponder aos valores guardados na base de dados
- ☒ Utilizador consegue fazer login se as ambas as credenciais corresponderem aos valores guardados na base de dados

Testes de logout:

- ☒ Após o registo/tentativa de registo ou login/tentativa de login, se o utilizador digitar “6” (opção visível no menu de comandos), será disponibilizado o menu inicial de login, registo, ou sair.

Testes de encerramento do programa:

- ☒ Após o registo/tentativa de registo ou login/tentativa de login, se o utilizador digitar “s” (opção visível no menu de comandos), a sua conexão com o servidor é corretamente terminada

Testes de indexamento de URL:

- ☒ Quando o utilizador introduzir um URL válido, se o URL não tiver “http://” ou “https://” ou “www.”, adiciona estas ‘strings’ em falta

Testes de listagem:

- ☒ Quando o utilizador digita “5”, será disponibilizada uma lista com top 10 pesquisas

Testes do Menu:

- ☒ Utilizador introduz um comando inválido e é devolvido ao menu, com mensagem de erro

Testes de conexão:

- ☒ O utilizador manda uma instrução ao servidor, contudo este encontra-se offline, o que leva o cliente a apresentar uma mensagem de perda de conexão e a tentar conectar-se ao servidor