RECUPERACIÓN DE LA INFORMACIÓN

MEMORIA DE LA PRÁCTICA III

Autor:
Jesús Gabriel Gómez Tocino

NOVIEMBRE 2023

Índice

1. Introducción

Este documento detalla el desarrollo de un Crawler, implementado en Java, que indexa y almacena páginas de Wikipedia. El Crawler parte de una URL semilla proporcionada por el usuario y navega a través de enlaces internos hasta una profundidad máxima especificada.

2. Conceptos aplicados y diseño

La conceptualización y los principios del Crawler siguen aquellos especificados por el contenido teórico de la asignatura:

- El Crawler consta de **3 módulos**:
    1. **Módulo de descarga**: Encargado de recuperar el contenido de una determinada URL y enviarlo al módulo de almacenamiento
    2. **Módulo de almacenamiento**: Guarda una copia del contenido para su posterior indexación
    3. **Módulo de planificación**: Mantiene la cola de URLs por visitar (frontera) y las envía a los distintos, e individuales, módulos de descarga.
- Para lograr escalabilidad y tolerancia a fallos, se utilizan **múltiples hilos**
    - Debido a esta característica, muchos de los elementos utilizados en el desarrollo del Crawler son de naturaleza paralela y disponen de mecanismos para garantizar seguridad frente a la concurrencia.
- Se establece un **límite off-line** a priori: Profundidad máxima

3. Módulos

3.1. Módulo Principal

El módulo principal es el encargado de iniciar el proceso de crawling y coordinar los módulos de descarga, planificación y almacenamiento.

- `WikipediaCrawler()`: Constructor de la clase. Inicializa el planificador, el almacenamiento y el pool de hilos. También inicializa un contador atómico (adecuado para la concurrencia) para las descargas activas.
- `iniciaCrawling(URL semilla, int profundidadMaxima)`: Inicia el proceso de crawling. Crea un hilo para cada descarga y espera a que todos los hilos terminen. El proceso de crawling se detiene cuando se ha alcanzado la profundidad máxima especificada o cuando no hay más URLs en la cola del planificador.
- `URLvalida(String urlStr)`: Este método verifica si una URL es válida y pertenece a Wikipedia en español. Se utiliza para validar la URL de inicio proporcionada por el usuario.
- `main(String[] args)`: Punto de entrada del programa. Solicita al usuario que ingrese la URL de inicio y la profundidad máxima, y luego inicia el proceso de crawling. Si la URL de inicio no es válida o no pertenece a Wikipedia en español, se muestra un mensaje de error y se termina el programa.

3.2. Módulo de Planificación

El módulo de planificación gestiona la cola de URLs a visitar durante el proceso de crawling. Mantiene un registro de las URLs visitadas y su profundidad en el árbol de crawling.

- `moduloPlanificacion()`: Constructor que inicializa la cola y el mapa de URLs visitadas.
- `encolarURL(URL url, int profundidad)`: Añade una URL a la cola si no ha sido visitada o si su profundidad actual es menor que la nueva profundidad.
- `siguienteURL()`: Obtiene y elimina la siguiente URL de la cola.
- `vacia()`: Verifica si la cola está vacía.
- `visitada(URL url)`: Verifica si una URL ya ha sido visitada.
- `obtenerProfundidad(URL url)`: Obtiene la profundidad actual de una URL visitada.

3.3. Módulo de Almacenamiento

El módulo de almacenamiento maneja el almacenamiento del contenido descargado de las páginas web y gestiona un índice de URLs. Utiliza un mapa concurrente para garantizar la seguridad en entornos de múltiples hilos.

- `moduloAlmacenamiento()`: Constructor que inicializa el mapa concurrente para almacenar las páginas y crea el subdirectorio para el corpus a descargar.
- `almacenarPagina(URL url, String contenido)`: Almacena el contenido de una página web en el mapa y guarda el contenido en un archivo.
- `almacenada(URL url)`: Verifica si una página ya está almacenada.
- `guardarArchivo(URL url, String contenido)`: Guarda el contenido de una página web en un archivo.

3.4. Módulo de Descarga

El módulo de descarga implementa Runnable para permitir su ejecución en un hilo. Se encarga de la descarga y procesamiento del contenido de una URL específica.

- `moduloDescarga(URL url, int profundidadMaxima, moduloPlanificacion planificador, moduloAlmacenamiento almacenamiento)`: Constructor que inicializa los atributos del módulo de descarga.
- `run()`: Método que se ejecutará cuando el hilo comience y realiza la descarga del contenido de la URL y procesa dicho contenido.
- `procesarContenido(String contenido)`: Procesa el contenido descargado de la URL, extrayendo nuevas URLs y añadiéndolas al planificador.
- `mismoDominio(URL url)`: Verifica si una URL es interna (pertenece a Wikipedia).