Scrapper Distribuido 💻

Autores ✒️

Nombre(s) y Apellidos	Grupo	Correo	GitHub
Reinaldo Barrera Travieso	C-411	r.barrera@estudiantes.matcom.uh.cu	@ArielXL
Ariel Plasencia Díaz	C-412	a.plasencia@estudiantes.matcom.uh.cu	@Reinaldo14

Empezando 🔧

Implementación

El proyecto está implementado en python 3.6.8. Para una mejor y mayor comprensión del código fuente propuesto entendemos que hay que tener profundos conocimientos acerca de python como lenguaje de programación. Nos apoyamos fundamentalmente en las librerías threading, socket, BeautifulSoup y urllib para su implementación.

Para la instalación de las dependencias ejecutamos el siguiente comando:

```
pip install -r requirements.txt
```

Ejecución

Para ejecutar nuestro proyecto de manera más sencilla proveemos un makefile con varias opciones. A continuación mostramos la ayuda.

```
cd src/
make help
info                           Display project description
version                        Show the project version
server                         Run a server with default parameters
client                         Run a client with default parameters
install                        Install the project dependencies
clean                          Remove temporary files
help                           Show this help
```

Para personalizar los parámetros de entrada consulte la documentación del fichero a ejecutar escribiendo las siguientes líneas:

```
cd src/
python server.py --help
python client.py --help
```

Sobre el Scrapper Distribuido 🕸

Web Scrapping

Web scrapping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.

El web scrapping está muy relacionado con la indexación de la web, la cual indexa la información de la web utilizando un robot y es una técnica universal adoptada por la mayoría de los motores de búsqueda. Sin embargo, el web scrapping se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento. Alguno de los usos del web scrapping son la comparación de precios en tiendas, la monitorización de datos relacionados con el clima de cierta región y la detección de cambios y la integración de datos en sitios webs.

En los últimos años el web scrapping se ha convertido en una técnica muy utilizada dentro del sector del posicionamiento web gracias a su capacidad de generar grandes cantidades de datos para crear contenidos de calidad.

Sistema Distribuido

El sistema estará constituido por nodos los cuales tendrán una responsabilidad específica (role). La existencia de un único role para cada nodo no impide que dos nodos no puedan estar en un mismo host. El sistema debe estar disponible siempre que halla algún nodo disponible por cada role (eso no implica que deba dar una respuesta rápida ante los pedidos). El sistema no puede perder datos en caso de que falle un nodo de almacenamiento. En caso de que el sistema sufra una partición (se divida en dos o más subsistemas producto de la pérdida de nodos o enlaces entre nodos), debe ser capaz de reconectarse y funcionar como un solo sistema cuando exista la posibilidad de comunicación entre los nodos de los subsistemas.

En nuestro proyecto cada nodo tiene las siguientes funcionalidades:

1. *Join network*: Se une a otro nodo según IP y PORT especificados.
2. *Leave network*: Deja la red de manera informada y replica los archivos que contiene el nodo.
3. *Print finger table*: Imprime la finger table del nodo.
4. *Print my predecessor and successor*: Imprime tanto el ID del nodo actual como su predecesor y sucesor.
5. *Print IP, PORT and LEVEL*: Imprime tanto el ID, IP, PORT y LEVEL (nivel de profundidad) del nodo actual.
6. *Print files on the network*: Imprime los archivos contenidos en el nodo.
7. *Make web scrapping*: Dada una url y un nivel de profundidad, se descargan todos los ficheros correspondientes a la url en la carpeta *src/downloads/urls/* con la profundidad especificada y se sube el html principal hacia la red.
8. *Upload file*: Se sube un archivo determinado y replicado a varios nodos en la red.
9. *Download file*: Se descarga el archivo especificado en la dirección src/downloads/files/.

License 📄