Apresentação do processo ETL e web scraping de sites e-commerce

Anderson Alan Montor

Repositório GitHub: https://github.com/andersonmontor/teste_eng_dados

Para realização do processo ETL e web scraping, foi utilizado a linguagem de programação Python 3, a biblioteca BeautifulSoup para web scraping e sqlite3 para armazenamento de dados.

O programa foi dividido entre os seguinte módulos:

- **downloader.py**: responsável por carregar os links fornecidos, baixar as páginas e guardar em um banco de dados de páginas HTML
- **extracao.py**: responsável por extrair dados relevantes das paginas HTML e guardar em um outro banco de dados com os dados extraídos
- **DAOs.py**: responsável por intermediar o acesso aos bancos de dados
- **main.py**: provê uma interface CLI para interação com o programa e consolidação dos módulos anteriores

Funcionamento:

A primeira etapa do processo consiste de baixar todas as páginas dos links do arquivo CSV e guardar num banco de dados sqlite3, nomeado paginas.db.

A forma que fiz(ver arquivo downloader.py) foi realizar um processo multithreaded, com número arbitrário de threads e usando mutexes para sincronização de acesso a lista de todos os links, que é dividida entre as threads, e também pro acesso ao banco de dados, para múltiplas threads não tentarem escrever no banco ao mesmo tempo.

Detalhe que o esquema de banco de dados não foi normalizado, então pode ser que após o processo ele ocupe mais espaço de armazenamento do que deveria.

OBS: não consegui baixar páginas dos links das Casas Bahia, aparentemente o site tem alguma proteção anti-webscraping, mesmo simulando um user agent de um navegador real.

A segunda etapa consiste de extrair informações relevantes das páginas baixadas(ver arquivo extracao.py). Há diversas funções, onde cada uma trata um tipo de link(Mercado Livre, Magazine, etc) usando métodos da biblioteca de web scraping BeautifulSoup. Cada função retorna uma tupla referente aos dados relevantes extraídos da página, que é armazenada num segundo banco de dados apenas com os dados relevantes extraídos.

A terceira etapa é realizar a consulta dos dados gerados, que é a funcionalidade fornecida por main.py, que faz uso dos DAOs para fazer o acesso ao banco e recuperar a informação requisitada.

Infelizmente, mesmo com as medidas tomadas para tentar reduzir os tempos de processamento, como uso de índices nos banco de dados e uso de múltiplas threads, o processo ainda ficou demorado tanto no download quanto na extração de dados. Então foi introduzida uma opção de reduzir o número de links a ser considerado.

Como executar o programa:

É necessário ter alguma versão do Python 3 instalado(a utilizada foi a 3.6.3), e a biblioteca externa BeautifulSoup, para instalar basta executar o comando: pip install beautifulsoup4

Após as dependências já estiverem instaladas, basta executar o arquivo main.py e escolher as opções desejadas.
O ideal seria escolher as duas primeiras opções primeiro, para baixar e extrair dados de todos os links do arquivo CSV. Mas como é um processo demorado, pode-se também escolher a opção de analisar um link específico, que caso não exista ainda nos bancos de dados, o programa baixa e analisa na hora.