Документация за Python проект: Попълване на таблица с много нишки

Този Python проект демонстрира разликата в производителността между еднонишковия и многоннишковия подход за попълване на 2D таблица. Проектът включва сървър-клиент архитектура, при която сървърът може да обработва множество клиентски заявки едновременно, а всеки клиент може да поиска таблица(чиито размери сам определя) да бъде попълнена както с една нишка, така и с множество нишки. Сървърът измерва времето, необходимо за двата подхода и връща резултатите на клиента.

Основни характеристики:

- **Сървър-клиент архитектура**: Сървърът може да обработва множество клиенти едновременно чрез нишки.
- **Попълване на таблица с една и с множество нишки**: Проектът сравнява времето, необходимо за попълване на таблица с една нишка спрямо множество нишки.
- **Измерване на производителността**: Изчислява се разликата във времето между двата подхода и тя се връща на клиента.
- **Паралелизация**: Многонишковият подход разделя таблицата на части, като всяка нишка обработва определен участък от таблицата.

Компоненти

1. client.py

Клиентският скрипт се използва за изпращане на заявки към сървъра, като потребителят задава броя редове, колони и брой нишки за многонишковия подход. Клиентът получава попълнената таблица и времето за изпълнение за двата метода от сървъра.

Функции:

- `receive_all(sock)`: Прочита всички данни от отговор на сървъра.
- `client_request()`: Изпраща заявка към сървъра, съдържаща размерите на таблицата и броя на нишките. След това получава резултатите (попълнената таблица и времето за изпълнение) и ги показва.

Работен процес:

1. Клиентът изисква от потребителя да въведе размерите на таблицата (редове и колони) и броя на нишките.
2. Клиентът изпраща тези данни към сървъра във формат JSON.
3. Клиентът чака отговор от сървъра, който съдържа попълнената таблица и времето за изпълнение на двата метода.
4. Клиентът отпечатва резултатите: попълнената таблица и времето за изпълнение.

2. server.py

Сървърният скрипт слуша за входящи клиентски заявки и ги обработва с помощта на нишки, за да позволи на множество клиенти да се свързват едновременно. Сървърът изчислява времето, необходимо за попълване на таблицата както с еднонишков, така и с многонишков метод.

Функции:

- `fill_table(table, start_row, end_row, value)`: Помощна функция, която попълва част от таблицата с дадена стойност.
- `single_thread_fill_table(rows, cols)`: Попълва таблицата с една нишка и измерва времето за изпълнение.
- `parallel_fill_table(rows, cols, num_threads)`: Попълва таблицата с множество нишки. Таблицата се разделя на части въз основа на броя редове, колони и желания брой нишки, за да се определи начинът на разпределение. Всяка нишка попълва своята част от таблицата.
- `handle_client(conn)`: Обработва свързан клиент, обработва заявката и изпраща отговор на клиента.
- `start_server(host, port)`: Инициализира сървъра, слуша за клиентски връзки и стартира нова нишка за обработка на всяка връзка.

Работен процес:

1. Сървърът чака за връзки от клиенти.
2. Когато клиент се свърже, сървърът получава размерите на таблицата и броя нишки.
3. След това сървърът попълва таблицата с еднонишков и многонишков метод, като измерва времето за изпълнение за всеки.
4. Сървърът изпраща попълнената таблица, времената за изпълнение и разликата във времето между двата метода на клиента.

Комуникация:

- Комуникацията между клиента и сървъра е базирана на JSON-кодирани данни.
- Клиентът изпраща JSON обект, съдържащ редовете, колоните и броя на нишките.
- Сървърът отговаря с JSON обект, който съдържа:
    - Попълнената таблица (списък от списъци).

- Времето за изпълнение на еднонишковото попълване (`single_thread_time`).
- Времето за изпълнение на многонишковото попълване (`multi_thread_time`).
- Разликата във времето между двата метода (`time_difference`).