Informe Ejecutivo I - Proyecto #1

Introducción

El siguiente informe, tiene como objetivo enunciar los avances del Proyecto#1 (). En este se realizó la implementación de una base de datos key-value minimalista con único servidor. Igualmente, se buscará resaltar los aspectos más relevantes del desarrollo de esta primera entrega y plasmar las conclusiones de esta.

Desarrollo

Desde el lado del cliente, la comunicación con el servidor de base de datos se hace a través de sockets en Python, conectándose entre ellos por localhost y el puerto 7000, previamente configurados en un archivo de constantes. Soporta operaciones de CRUD que son recibidas en peticiones por el servidor de base de datos para procesarlas y devolver una respuesta.

Para la implementación de la base de datos se hace uso de la librería json, la cual nos permite almacenar de manera local (en memoria) un diccionario con las claves-valor. Haciendo uso de las operaciones de dump y de load, se hace posible guardar el diccionario donde se almacena los datos, al igual que cargar la copia del diccionario que se encuentra en local.

Como implementación del servidor de base de datos se utilizó el lenguaje de programación de Python y un archivo de configuración básico. El script server.py se encarga de manejar la lógica de negocio y con utilizando la clase database.py realiza las operaciones CRUD implementadas (SET, GET, UPDATE y DELETE). Además, haciendo uso de la librería sockets se abre el canal de entrada para las peticiones de los clientes. En pocas palabras, el servidor toma el mensaje entrante y y realiza la operación CRUD correspondiente, para finalmente retornar un mensaje de confirmación.

Por otra parte, como implementación de cliente, se realiza un CLI para recibir las consultas del usuario. Dichas consultas son posteriormente procesadas por la clase client.py, la cual se encarga de validarla para más adelante ser enviada al servidor. Como medio de comunicación se utiliza la librería sockets, con la cual se conecta a la base de datos y envía la consulta ingresada por el usuario. Una vez la clase client.py reciba respuesta del servidor de base de datos, este procede a procesarla para ser mostrar al cliente por la CLI.

Para finalizar, cada una de las clases cuenta con un archivo de configuración, el cual se utliza para cargar las configuraciones básicas para el correcto funcionamiento. Cabe destacar que, este archivo contiene información como direcciones IP, puertos y datos adicionales para el funcionamiento de cada uno de los programas.

Pasos a seguir

Como continuación del proyecto, en la segunda semana se implementará un mecanismo de particionamiento de la base de datos. Este mecanismo debe ser de forma dinámica utilizando como base un hash table. Logrando así un mejor rendimiento porque se permite el procesamiento de varias peticiones al mismo tiempo y una mejor escalabilidad. Por otro lado, se implementará un routing tier que estará encargado de recibir las peticiones y enviarlas al particionamiento indicado.

Estudiantes:

David Gómez

Sebastián Granda

Pascual Gómez

Repositorio:

- https://github.com/dgomezc1/DBDistribuida