Python + Redis Queue

Programación de cómputo en paralelo para operaciones sobre datos.

En este informe exploraremos el uso de Python más Redis Queues para implementar una simulacion de ejecucion de tareas de forma asıncrona.

El objetivo es construir una simulacion pueda validar un hash SHA y poder encontrar un elemento que, a partir del hash anterior, genere un nuevo hash. Este es capaz de recibir tres tipos de clientes, de los tipos comun, alto y VIP. El cliente de tipo alto tiene prioridad en la ejecucion sobre el tipo comun, mientras que VIP tiene prioridad sobre alto.

El codigo presentado es capaz de leer un archivo de instrucciones, realizar un encolamiento, ejecutar de forma asincrona y entregar el orden de atención de clientes y sus resultados.

Para realizar esto, es necesario levantar Redis usando

redis-server

para luego generar un rq worker con el comando

rq worker vipqueue altoqueue comunqueue

Una vez logrado esto, es posible ejecutar usando el comando

python T3.py <pathinstrucciones>.

Esto procedera a crear las colas, ejecutarlas de forma asincrona, utilizar las funciones de hash entregadas en enunciado y generar un output final.

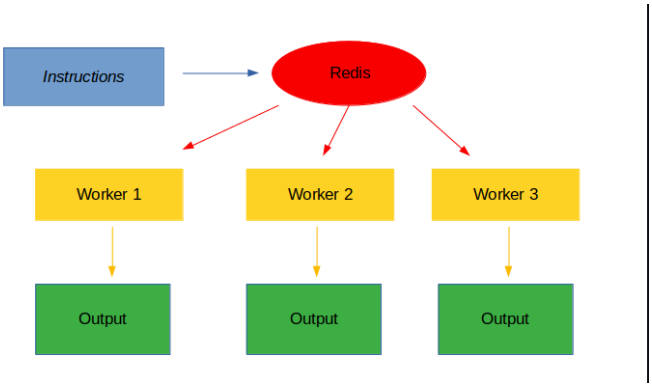

Figure 1: Proceso completo.

Y qué está pasando ?

Redis queue corresponde a una cola de mensajes. Un worker es un proceso python que se encargara de estar escuchando a la cola que se le asocie hasta encontrar una tarea a ejecutar.

Estas dos partes funcionan en conjunto, de forma que redis encola mensajes de tareas, mientras que worker escuche dicha cola y en el momento de ver que esta tenga un mensaje de tarea, la toma y ejecuta.

Todo esto funciona de forma asincrona, de tal forma que solo se ejecutara una tarea de la cola, siempre y cuando exista un worker que este escuchandola y a la vez que este disponible para realizar la tarea.