Jaime Arana, Javier Álvarez

Práctica Redis

Implementar diferentes versiones de una base de datos Redis y realiza benchmarks

Práctica redis

Sistemas Distribuidos

# Práctica Redis 1.1 – Instalar REDIS stand-alone sobre Docker

A lo largo de la práctica se han ido haciendo capturas de los apartados donde hay resultados que analizar y no de los procesos de instalación o configuración ya que los comandos están dados en la práctica.

**Benchmark desde contenedor redis (redis-cli)**

Texto

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Imagen de la pantalla de una computadora

Descripción generada automáticamente con confianza baja

**Benchmark desde el terminal (host)**

Texto

Descripción generada automáticamente

Como se puede ver en los resultados al hacer el benchmark directamente desde el contenedor de Redis es capaz de hacer el mismo número de peticiones en menor tiempo. Este resultado tiene sentido ya que al estar directamente dentro del contenedor no tiene que hacer el proceso de entrada en el contenedor para cada petición.

|  |  |  |
| --- | --- | --- |
| nº peticiones | **Tiempo** dentro del contenedor | **Tiempo** desde el host |
| 100000 | 1.49 seg | 7.16 seg |

# Práctica Redis 1.2 – Instalar REDIS stand-alone sobre kubernetes

**¿Qué tipo de despliegue se está haciendo?**

En este caso se trata de un tipo de despliegue de clúster de solo una máquina, tratándose de una instalación en minikube.

**Benchmark kubernetes**

Texto

Descripción generada automáticamente

En este caso el resultado no concuerda con lo esperado, ya que el tiempo empleado para hacer las 100000 requests es demasiado bajo. No obstante, tras haber comprobado todos los parámetros y haber repetido el benchmark varias veces, obtenemos el mismo resultado. Así que el resultado, aunque no esperado, si parece correcto.

# Práctica Redis 1.4 – Instalar clúster dockerizados de REDIS

En las gráficas inferiores se presentan los resultados tras haber ejecutado los comandos para ver los nodos del clúster y los slots.

Texto

Descripción generada automáticamente

Es importante ver o comprender que para que el clúster sea útil y sirva, los nodos esclavos y maestros no pueden pertenecer a la misma máquina.

En este caso la configuración es correcta ya que, por ejemplo, el nodo maestro de la máquina Ubun1 está conectado al esclavo de la máquina Ubun2.

**Benchmark desde el host a Ubun1**

Texto

Descripción generada automáticamente

**Benchmark desde Ubun1 a propia instancia**

Texto

Descripción generada automáticamente

**Benchmark desde Ubun1 a Ubun3**

Texto

Descripción generada automáticamente

# Práctica Redis 1.5 – Tipos de datos en REDIS

Este último apartado se ha solucionado en el fichero adjunto **redis.ipynb** y para la sección de Publish/Subscribe hay, además, otro archivo .pynb que hace de consumidor de los channels creados.