

ALUMNA:

210300510 Kenny Yaritza Benitez Renteral

Galera 4 Clúster con MariaDB en Linux

ASIGNATURA:

Cómputo de alto desempeño

PROGRAMA EDUCATIVO:

Ingeniería en Datos e Inteligencia Organizacional

Creación de un Clúster con Galera 4 y Pruebas de Rendimiento con Sysbench

Este documento es un reporte en donde se podrá observar el proceso de implementación de un clúster de bases de datos en Ubuntu Server utilizando Galera 4 y MariaDB, así como la ejecución de pruebas de rendimiento mediante Sysbench.

Software Utilizado:

- VirtualBox: Para la virtualización del servidor.
- <u>Ubuntu Server</u>: Sistema operativo base.
- MariaDB: Sistema de gestión de bases de datos.
- Galera 4: Herramienta para la creación de clústeres de bases de datos.
- Sysbench: Utilidad para realizar pruebas de rendimiento.
- OpenSSH: Para la conexión remota al servidor.
- <u>CMD de Windows:</u> Desde aquí se ejecutaron los comandos para interactuar con el servidor.

Pasos:

- 1. Instalación del Servidor Ubuntu Server: Se configuró una máquina virtual con un núcleo de procesador (1 core) para simular un entorno básico. En donde se la red NAT habilitó para permitir la descarga de paquetes necesarios desde internet. También se configuró el Port Forwarding para permitir conexiones SSH desde el host hacia la máquina virtual.
- **2.** Configuración Inicial del Sistema: Se instaló y habilitó el servidor SSH para permitir conexiones remotas, haciendo uso de los siguientes comandos:

sudo apt install openssh-server

sudo apt update

sudo apt -y install net-tools software-properties-common

Los cuales nos fueron de ayuda para actualizar la lista de paquetes disponibles e instalar las herramientas esenciales para la configuración del servidor.

3. Instalación y Configuración de Galera 4: Se instalaron los paquetes necesarios para el clúster, con:

sudo apt -y install mariadb-server mariadb-client galera-4 sudo apt -y install galera-arbitrator-4 mariadb-client libmariadb3

4. Configuración del Clúster: Se modificó el archivo de configuración de Galera con: '/etc/mysql/mariadb.conf.d/60-galera.cnf. para definir los parámetros del clúster, como la dirección IP de los nodos y el nombre del clúster.

También se inicio del Clúster con:

galera_new_cluster

Para asegurar que el clúster se creó correctamente, se debe validar su funcionamiento haciendo uso de:

mysql -u root -ρ -e "SHOW STATUS LIKE 'wsrep_cluster_size"

mysql -u root --execute="SHOW GLOBAL STATUS WHERE Variable_name IN ('wsrep_ready', 'wsrep_cluster_size', 'wsrep_cluster_status', 'wsrep_connected');"

Es necesario realizar una confirmación de que los puertos estuvieran abiertos: netstat -tlpn

5. Instalación de Sysbench: Se instaló sysbench la cual es una herramienta para realizar pruebas de rendimiento, usando:

sudo apt -y install sysbench

Se creó una base de datos para pruebas llamada <u>sbtest</u> para las pruebas, con:

mysql -u root -ρ -e "create database sbtest

6. Preparación del Entorno para Pruebas: Se configuró el entorno para pruebas de sólo lectura, usando los siguientes códigos: sysbench --threads=1 --db-driver=mysql --mysql-user=root --events=0 oltp_read_only prepare

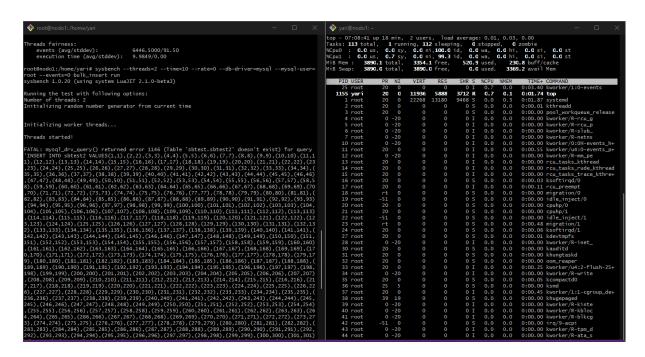
7. Ejecución de Pruebas de Rendimiento

Se realizaron pruebas de rendimiento con diferentes cargas de trabajo, tanto para un núcleo como para dos núcleos. Cada prueba tuvo una duración de 10 segundos.

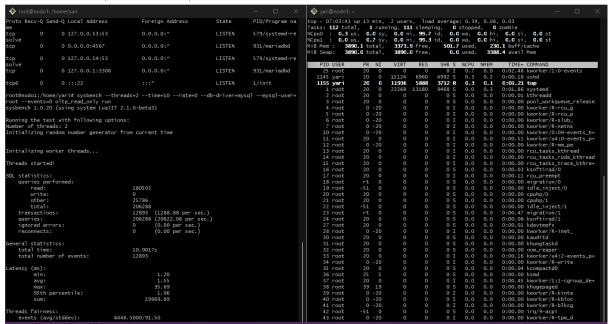
Resumen de las Pruebas de Rendimiento						
Comando	Queries Procesado s	Transaccion es	Tiempo	Latencia Máxima(ms)		
Oltp_read_only	206288	12893	10.0017s	35.69		
Oltp_delete	141139	141139	10.0007s	194.42		
Oltp_insert	5437	5437	10.0029s	14.61		

Resumen de las Pruebas de Rendimiento					
oltp_point_sele ct	145184	145184	10.0013s	17.47	
oltp_read_writ e	41578	2078	10.0072s	37.13	
oltp_update_in dex	6807	6807	10.0019s	48.90	
oltp_update_n on_index	4788	4788	10.0035s	139.04	
oltp_write_only	17328	2888	10.0052s	50.43	
oltp_update_in dex	5393	5393	10.0029s	29.88	
oltp_update_n on_index	5632	5632	10.0049s	53.79	
select_random _points	78220	78220	10.0021s	17.07	
select_random _ranges	79321	79321	10.0026s	8.68	

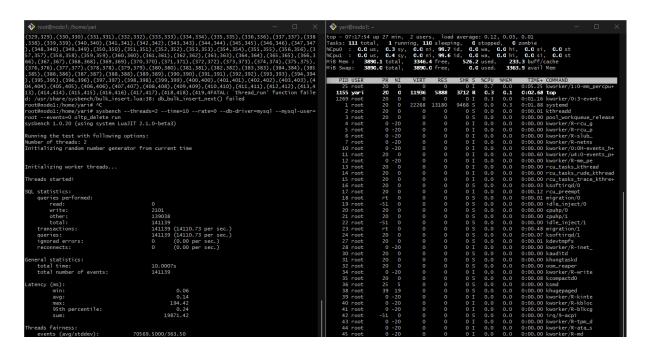
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 bulk_insert run



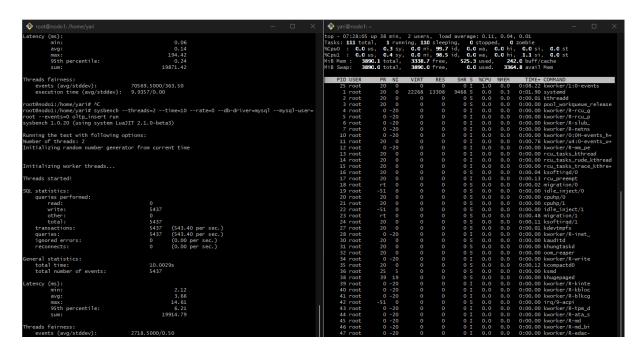
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_read_only run



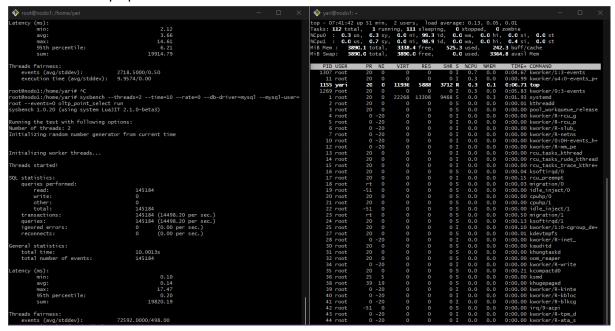
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_delete run



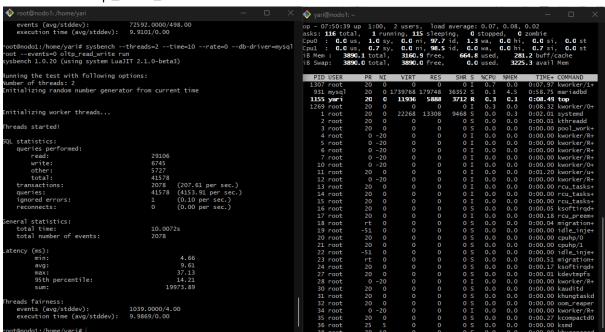
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_insert run



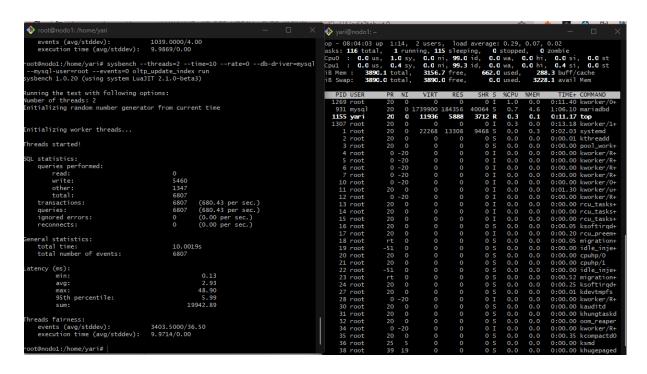
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_point_select run



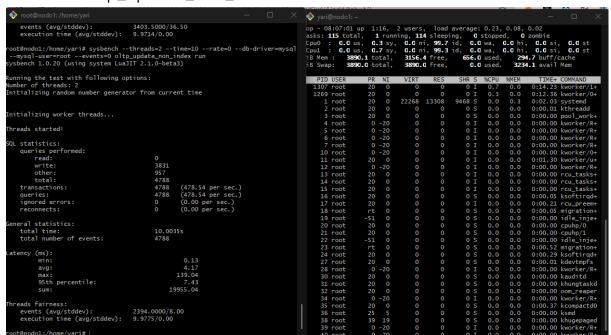
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_read_write run



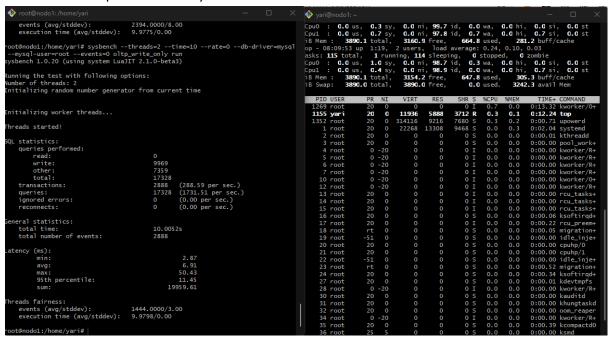
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_index run



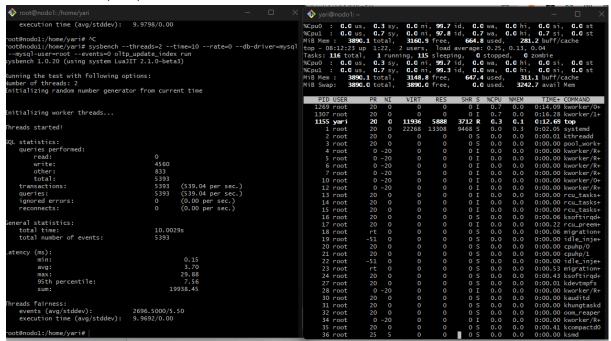
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_non_index run



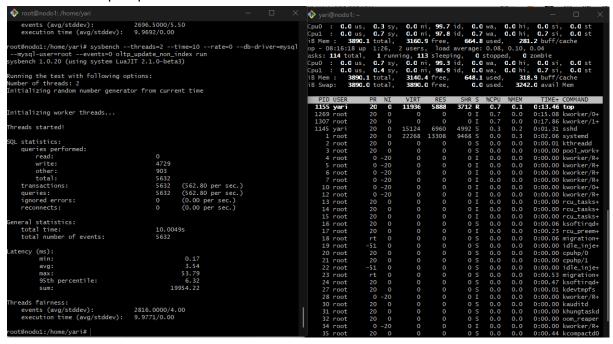
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_write_only run



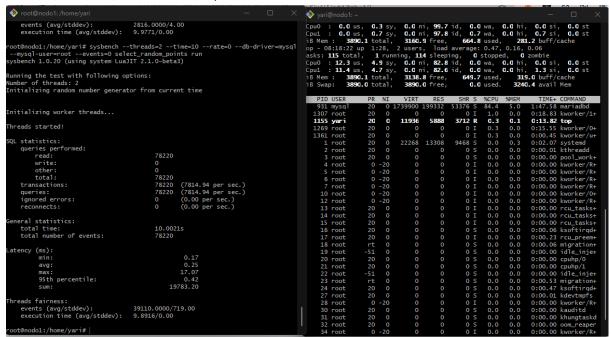
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_index run



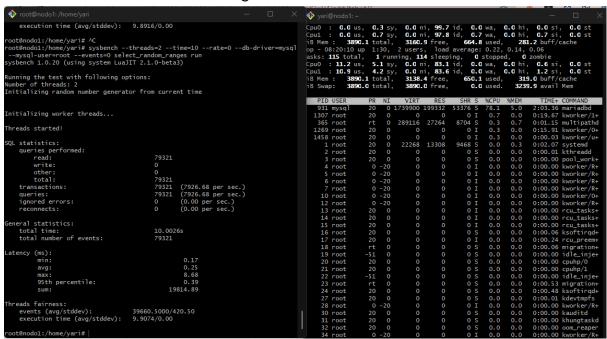
sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_non_index run



sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 select_random_points run



sysbench --threads=2 --time=10 --rate=0 --db-driver=mysql --mysql-user=root --events=0 select_random_ranges run



Al implementar un clúster de bases de datos con Galera 4 y MariaDB logramos analizar qué factores afectan su desempeño y estabilidad. A lo largo de las pruebas realizadas con Sysbench, se observó que la variabilidad en los resultados está directamente influenciada por la configuración del clúster, el tipo de carga de trabajo y los parámetros de ejecución de las pruebas.

Uno de los elementos clave en la variabilidad del rendimiento es la cantidad de threads utilizados en las pruebas. A medida que el número de hilos de ejecución aumenta, la carga de procesamiento sobre la CPU y la concurrencia en el acceso a la base de datos también se incrementan, lo que puede mejorar el rendimiento hasta cierto punto, pero eventualmente generar saturación y latencias más altas debido a la competencia por recursos del sistema.

Por otro lado, el tiempo de ejecución de cada prueba influye en la estabilidad de los resultados. Debido a que en las pruebas de corta duración, la variabilidad es mayor debido a la influencia de procesos externos, caché y la inicialización del sistema. A diferencia de las pruebas más prolongadas, las cuales permiten obtener métricas más representativas del desempeño real del clúster, evidenciando patrones de optimización o degradación a lo largo del tiempo.

Debido a ello es importante diseñar pruebas de rendimiento las cuales reflejan condiciones reales de operación, lo cual incluye el considerar tanto los recursos computacionales disponibles como la arquitectura del clúster al interpretar los resultados.