Universidad del Caribe

2000 CANCUN, QUINTANA ROO, MÉXICO

CONOCIMIENTO Y CULTURA PARA EL DESARROLLO HUMANO

INVESTIGACIÓN/REPORTE/RESUMEN:

Tarea #998 Instalar Galera

ASIGNATURA:

Cómputo de alto desempeño

Alumna:

Liliana Jazmin Basto Euan

MATRÍCULA: **200300602**

PROGRAMA EDUCATIVO

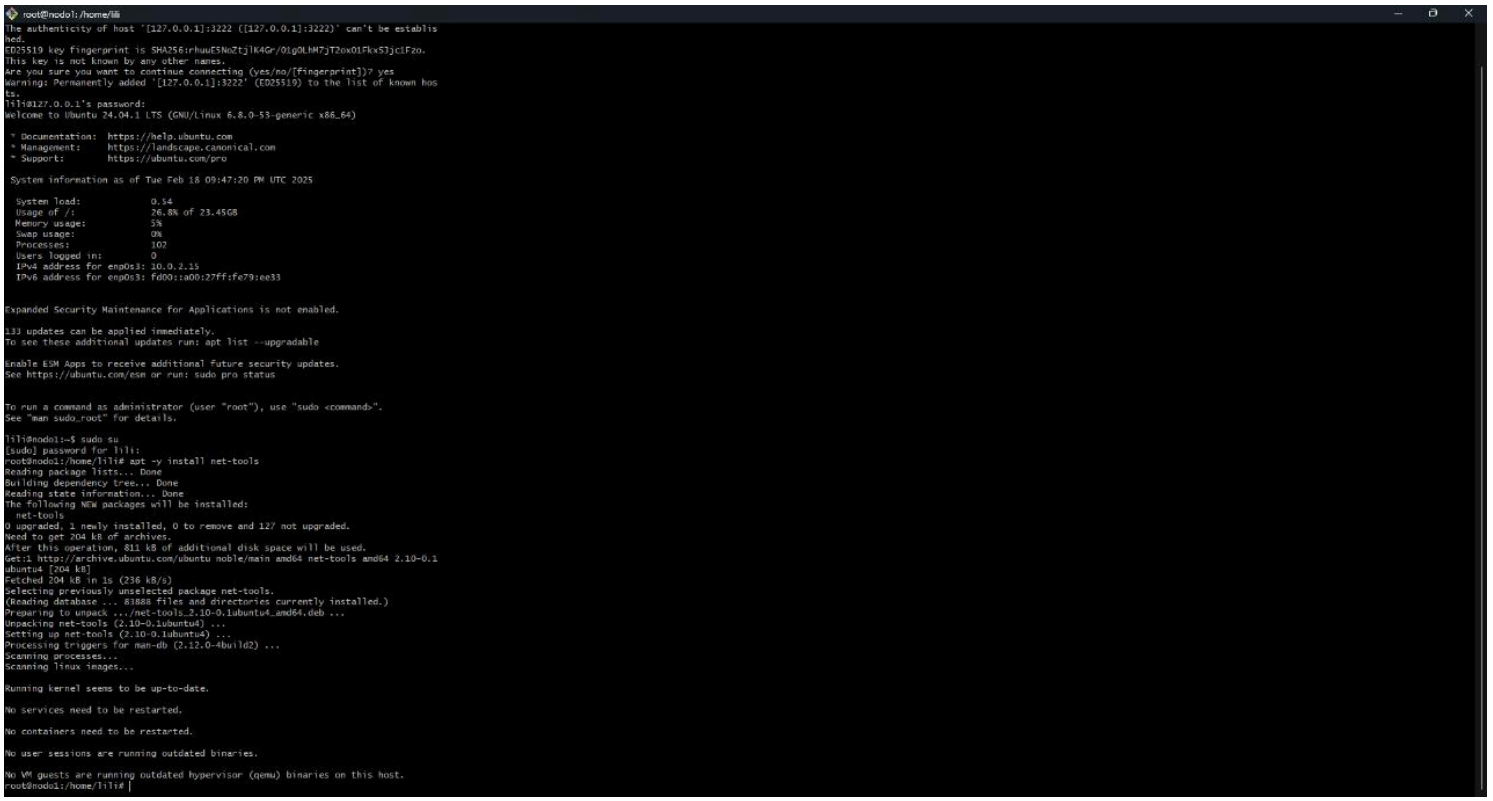
INGENIERÍA EN DATOS E INTELIGENCIA ORGANIZACIONAL

PRESENTADO A:

Ismael Jimenez Sanchez

Actividades realizadas:

- **Actualización de la lista de paquetes:** Se ejecutó el comando apt update para actualizar la lista de paquetes disponibles.
- **Actualización de paquetes:** Se encontraron 118 paquetes para actualizar (118 upgraded, 1 newly installed, 0 to remove and 117 not upgraded).
- **Instalación de paquetes:** Se instaló el paquete net-tools 2.10-0.ubuntu. Este paquete proporciona herramientas de red esenciales como ifconfig, netstat, route, etc.
- **Mantenimiento del sistema:** Se ejecutaron tareas de mantenimiento, posiblemente incluyendo la actualización del kernel (linux-image-unsigned-5.15.0-76-generic).



Al ejecutar apt update, mi sistema se ha conectado a varios "repositorios". Estos repositorios son como bibliotecas gigantescas de software para Ubuntu. Hemos visto que se ha conectado a:

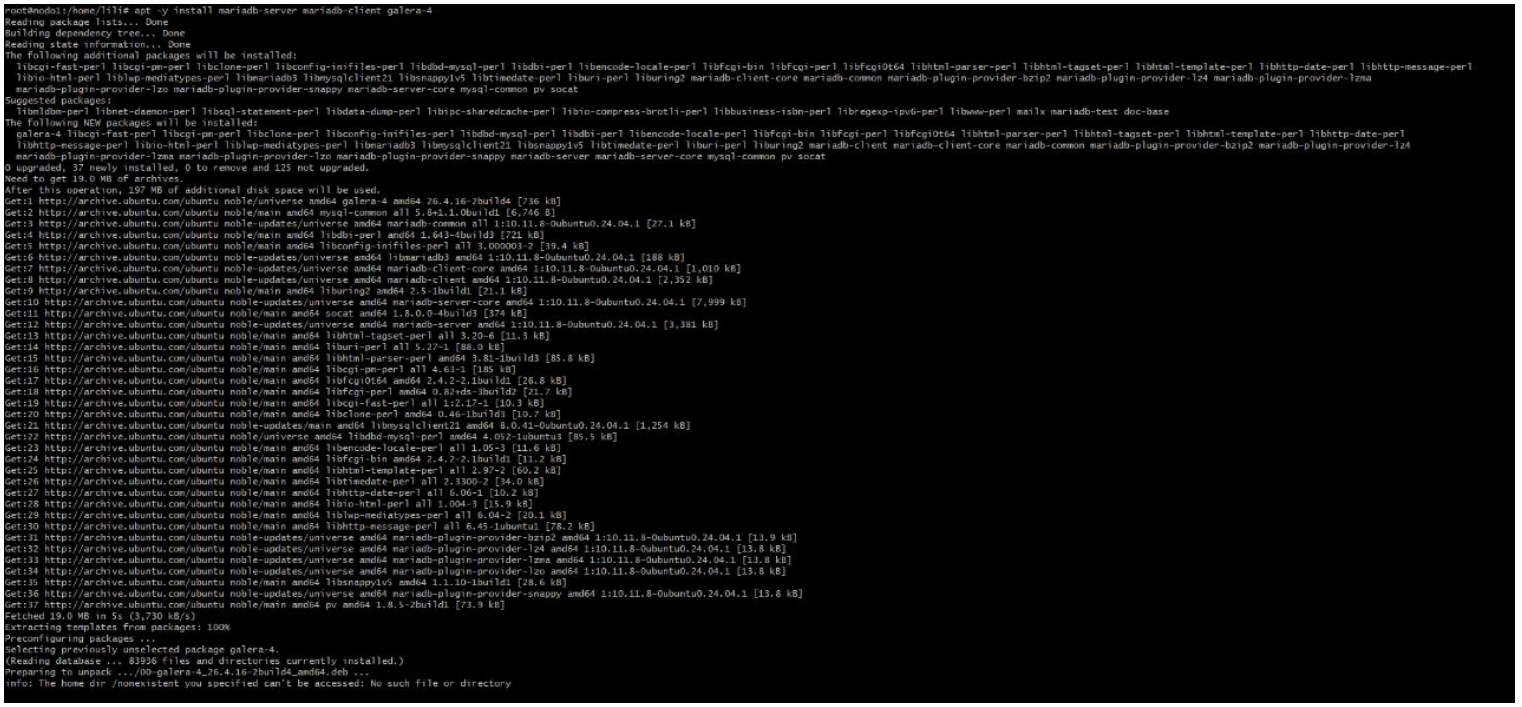
- http://security.ubuntu.com/ubuntu noble-security InRelease: Aquí es donde Ubuntu busca actualizaciones de seguridad importantes.
- http://archive.ubuntu.com/ubuntu noble InRelease: Este es el repositorio principal de Ubuntu, donde se encuentran la mayoría de los programas.
- http://archive.ubuntu.com/ubuntu noble-updates InRelease: Aquí se encuentran actualizaciones para los programas de Ubuntu.
- http://archive.ubuntu.com/ubuntu noble-backports InRelease: Este repositorio contiene programas más nuevos que han sido adaptados para funcionar en nuestra versión de Ubuntu.

Después de que apt update ha terminado de "hablar" con estos repositorios, ha creado una lista de todos los programas disponibles y sus versiones.

```
root@nodo1:/home/lili# apt update
Hit:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:3 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
125 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@nodo1:/home/lili#
```

Conectarse a servidores MariaDB: El paquete mariadb-client te da la herramienta para enviar comandos y recibir datos de un servidor MariaDB.

Comunicarse con bases de datos MariaDB: La biblioteca libmariadb3 permite que otros programas en tu sistema (como aplicaciones web o herramientas de administración) se comuniquen correctamente con bases de datos MariaDB.



sudo: Otorga privilegios de superusuario para ejecutar el comando.

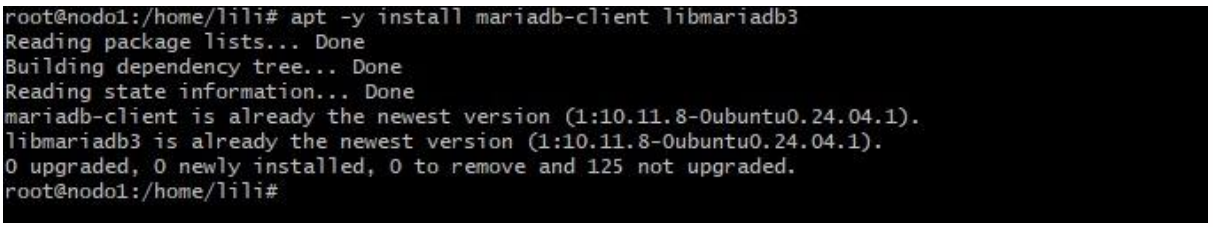
apt: Es la herramienta de administración de paquetes en sistemas Debian/Ubuntu.

-y: Responde "sí" automáticamente a todas las preguntas durante la instalación.

install: Le dice a apt que quieres instalar uno o más paquetes.

galera-arbitrator-4: El paquete que contiene el árbitro de Galera, utilizado para la gestión de clústeres Galera.

- **sudo**: Otorga privilegios de superusuario para ejecutar el comando.
- **apt**: Es la herramienta de administración de paquetes en sistemas Debian/Ubuntu.
- **-y**: Responde "sí" automáticamente a todas las preguntas durante la instalación.
- **install**: Le dice a apt que quieres instalar uno o más paquetes.
- **mariadb-client**: El paquete que contiene el cliente de línea de comandos mysql (o mariadb), que te permite interactuar con servidores de bases de datos MariaDB.
- **libmariadb3**: Esta biblioteca proporciona las funciones necesarias para que otros programas se conecten y trabajen con bases de datos MariaDB.

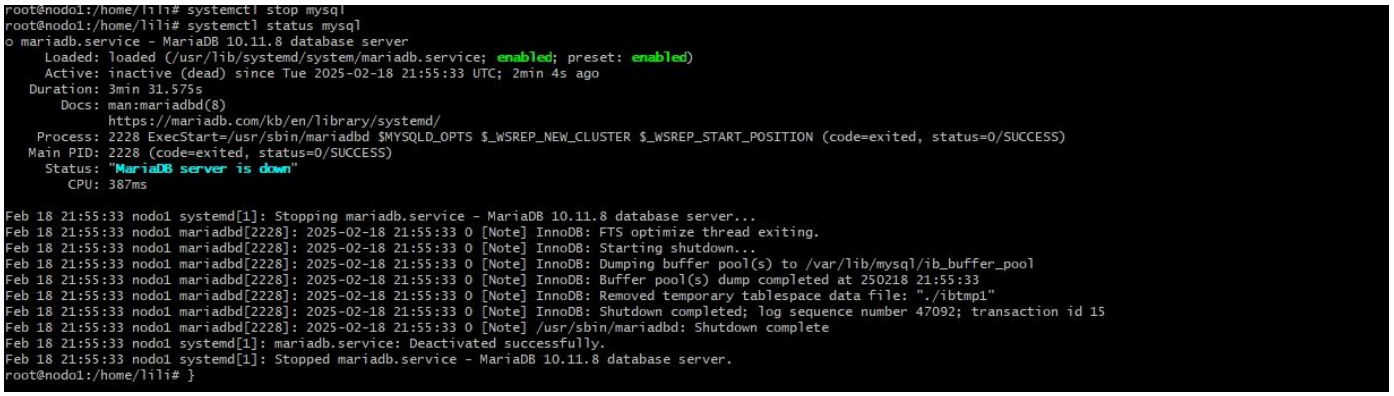


Explicación de los comandos:

1. **systemctl stop mysql**: Se intentó detener un servicio llamado mysql. Este comando falló porque el servicio mysql no estaba activo o no existía en el sistema.
2. **systemctl status mysql**: Se intentó verificar el estado de un servicio llamado mysql. Este comando falló por la misma razón que el anterior.
3. **systemctl stop mariadb**: Se detuvo el servicio MariaDB. El servicio se encontraba activo (active (running)) y se detuvo exitosamente (Deactivated successfully.).
4. **systemctl status mariadb**: Se verificó el estado del servicio MariaDB después de detenerlo. El servicio se encuentra ahora inactivo (inactive (dead)).

Resultados de la detención:

- El servicio MariaDB se detuvo exitosamente.
- El estado del servicio cambió de active (running) a inactive (dead).
- Se registraron eventos en el registro del sistema (systemd) relativos a la detención del servicio, incluyendo el PID del proceso principal (mariadbd) y el estado de finalización.
- Se realizaron tareas de limpieza por parte de MariaDB al detenerse, como el volcado del buffer pool a disco y la eliminación de archivos temporales.



sudo apt -y install mariadb-client libmariadb3:

- Resultado: Los paquetes mariadb-client y libmariadb3 ya estaban instalados en sus versiones más recientes (1:10.11.8-0ubuntu0.24.04.1).
- No se instalaron, actualizaron ni eliminaron paquetes.

systemctl stop mysql:

- Resultado: Falló porque el servicio mysql no existía.

systemctl status mysql:

- Resultado: Falló porque el servicio mysql no existía.

systemctl stop mariadb:

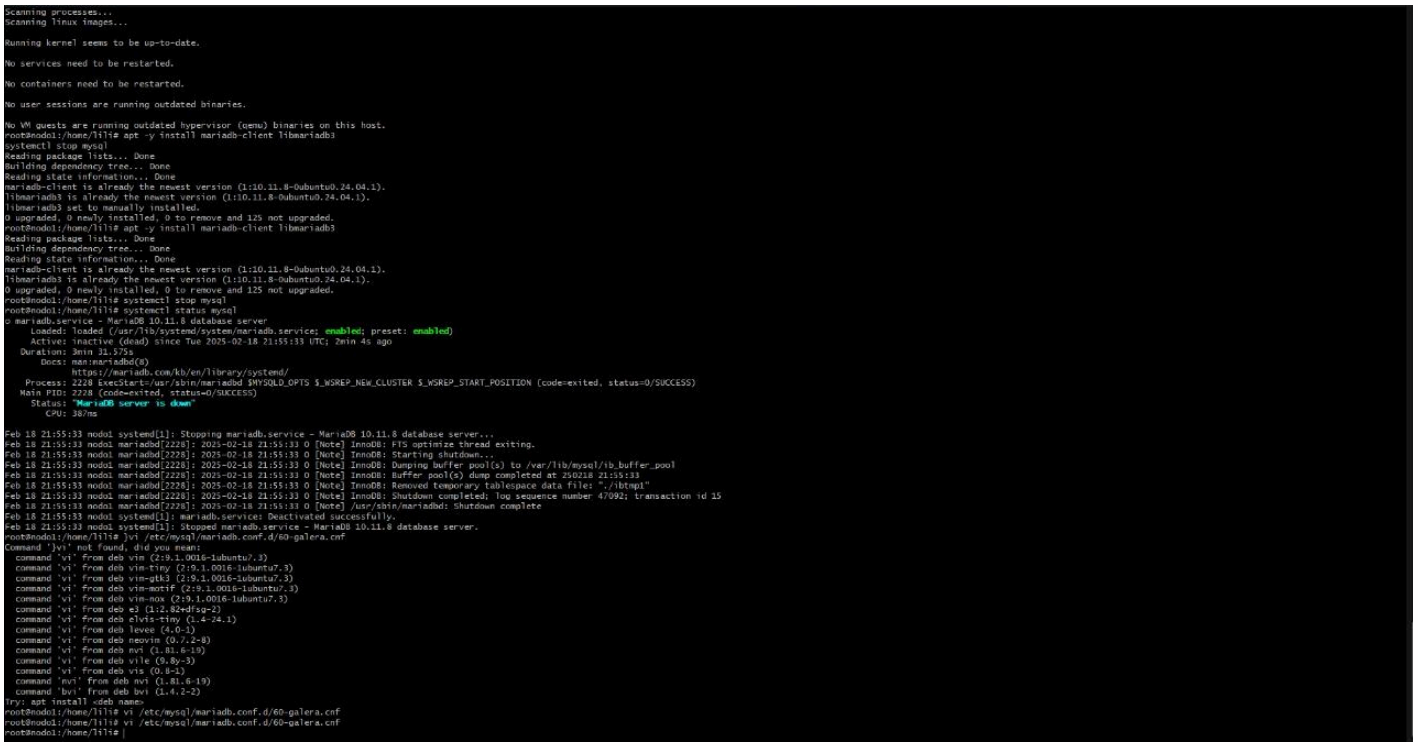
- Resultado: Se detuvo el servicio MariaDB exitosamente.
- El estado del servicio cambió de active (running) a inactive (dead).

systemctl status mariadb:

- Resultado: Se verificó que el servicio MariaDB está inactivo (inactive (dead)).

sudo apt -y install mariadb-client libmariadb3:

- Resultado: Se intentó nuevamente instalar los paquetes mariadb-client y libmariadb3.
- Dado que ya estaban instalados, no se realizaron cambios.



1. Configuración del Proveedor de Galera:

- **wsrep_on=ON**: Esta línea habilita la funcionalidad de Galera. Esencial para que Galera funcione.
- **wsrep_provider=/usr/lib/galera/libgalera_smm.so**: Especifica la ruta a la biblioteca compartida del proveedor de Galera (libgalera_smm.so). Esta biblioteca es la que contiene la lógica para la comunicación y sincronización del clúster.

2. Configuración del Clúster Galera:

- **wsrep_cluster_name="test_cluster"**: Define el nombre del clúster Galera. Todos los nodos que formen parte de este clúster deben tener el mismo nombre.
- **wsrep_cluster_address="gcomm://192.168.56.101"**: Define las direcciones de los nodos que forman parte del clúster. En este caso, solo se incluye la dirección 192.168.56.101, lo que sugiere que este nodo es el único o el primero en configurarse. En un clúster real, se deben listar las direcciones de todos los nodos (usualmente separadas por comas).

3. Configuración de Sincronización de Galera:

- **wsrep_sst_method=rsync**: Define el método de transferencia de estado del sistema (SST). rsync es uno de los métodos más comunes y se utiliza para sincronizar los nodos cuando se unen al clúster o cuando se recuperan después de un fallo.

4. Configuración del Nodo Galera:

- **wsrep_node_address="192.168.56.101"**: Define la dirección de este nodo en el clúster. Debe coincidir con una de las direcciones listadas en wsrep_cluster_address.
- **wsrep_node_name="nodo1"**: Define el nombre de este nodo. Cada nodo en el clúster debe tener un nombre único.

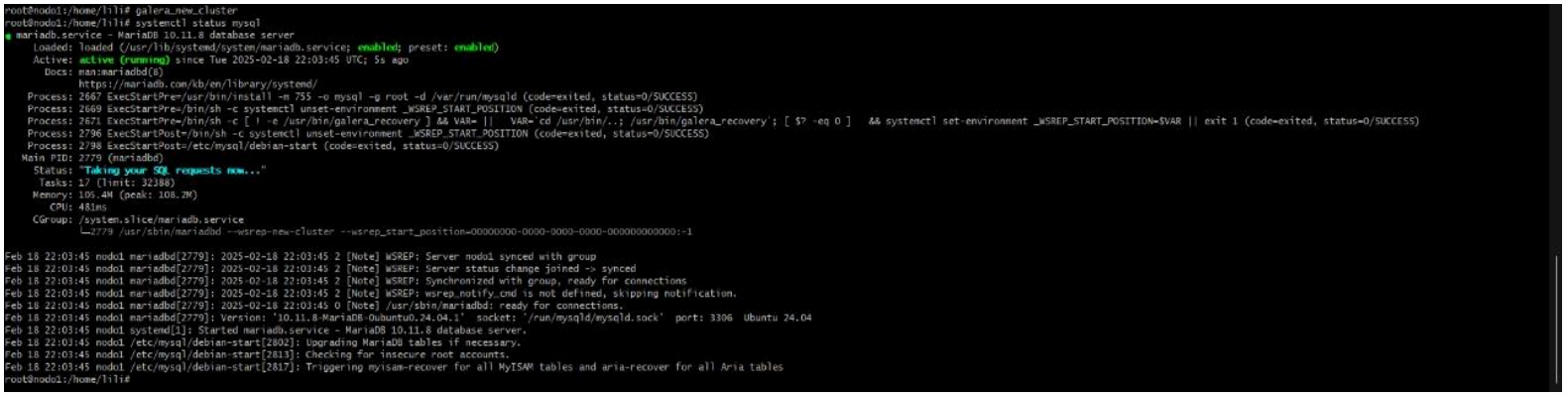
```
[mysqld]
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
bind-address=0.0.0.0

# Galera Provider Configuration
wsrep_on=ON
wsrep_provider=/usr/lib/galera/libgalera_smm.so

# Galera Cluster Configuration
wsrep_cluster_name="test_cluster"
wsrep_cluster_address="gcomm://192.168.56.101"

# Galera Synchronization Configuration
wsrep_sst_method=rsync

# Galera Node Configuration
wsrep_node_address="192.168.56.101"
wsrep_node_name="nodo1"
```

Mensajes de inicio de MariaDB: Las líneas que comienzan con "Feb 18 22:18:45" muestran el inicio de MariaDB. Esto es una *consecuencia* de galera_new_cluster. El script galera_new_cluster asegura que MariaDB se esté ejecutando con la configuración correcta de Galera.

Mensajes de Galera: Las líneas que mencionan WSREP (Replicación de Conjunto de Escritura) son mensajes específicos de Galera. Indican que Galera se está inicializando y uniendo/formando el clúster. Los mensajes gwsrep están relacionados con la comunicación grupal.

Éxito de galera_new_cluster (implícito): El hecho de que estos mensajes estén apareciendo sugiere que galera_new_cluster tuvo éxito en la inicialización del clúster. Si hubiera fallado, probablemente verías mensajes de error.

"Listo para conexiones": El mensaje "listo para conexiones" indica que MariaDB (y por extensión, Galera) está listo para aceptar conexiones de clientes.



mysql: Se utiliza el cliente de línea de comandos mysql para interactuar con el servidor de base de datos.

-u root: Especifica que se va a conectar como el usuario root.

-p: Indica que se solicitará la contraseña.

-e "SHOW STATUS LIKE 'wsrep_cluster_size'": Se ejecuta la consulta SQL SHOW STATUS LIKE 'wsrep_cluster_size' en el servidor. Esta consulta muestra el valor de la variable de estado wsrep_cluster_size, que indica el número de nodos que están actualmente en el clúster Galera.

```
root@nodo1:/home/lili# mysql -u root -p -e "SHOW STATUS LIKE 'wsrep_cluster_size'"
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 1     |
+--------------------+-------+
root@nodo1:/home/lili#
```

mysql: Se utiliza el cliente de línea de comandos mysql para interactuar con el servidor de base de datos.

-u root: Especifica que se va a conectar como el usuario root.

-e "...": Se ejecuta la consulta SQL dentro de las comillas.

SHOW GLOBAL STATUS ...: Esta consulta muestra el valor de las variables de estado especificadas.

WHERE Variable_name IN: Filtra las variables de estado que se van a mostrar.

```
root@nodo1:/home/lili# mysql -u root --execute="SHOW GLOBAL STATUS WHERE Variable_name IN ('wsrep_ready', 'wsrep_cluster_size', 'wsrep_cluster_status', 'wsrep_connected');"
+----------------------+---------+
| Variable_name        | Value   |
+----------------------+---------+
| wsrep_cluster_size   | 1       |
| wsrep_cluster_status | Primary |
| wsrep_connected      | ON      |
| wsrep_ready          | ON      |
+----------------------+---------+
root@nodo1:/home/lili#
```

netstat: Es una herramienta para mostrar información sobre las conexiones de red, interfaces y tablas de enrutamiento.

-t: Muestra conexiones TCP.

-i: Muestra interfaces de red.

-p: Muestra el ID del proceso y el nombre del programa asociado a cada conexión.

-n: Muestra direcciones IP y números de puerto en lugar de nombres de host y nombres de servicio.

```
root@nodo1:/home/lili# netstat -tlpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      2779/mariadbd
tcp        0      0 127.0.0.54:53           0.0.0.0:*               LISTEN      601/systemd-resolve
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      601/systemd-resolve
tcp        0      0 0.0.0.0:4567            0.0.0.0:*               LISTEN      2779/mariadbd
tcp6       0      0 :::22                   :::*                    LISTEN      1/init
root@nodo1:/home/lili#
```

sudo: Otorga privilegios de superusuario para ejecutar el comando.

apt: Es la herramienta de administración de paquetes en sistemas Debian/Ubuntu.

-y: Responde "sí" automáticamente a todas las preguntas durante la instalación.

install: Le dice a apt que quieres instalar uno o más paquetes.

sysbench: El paquete que contiene la herramienta sysbench.

```
root@nodo1:/home/lili# apt -y install sysbench
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libluajit2-5.1-2 libluajit2-5.1-common libpq5
The following NEW packages will be installed:
  libluajit2-5.1-2 libluajit2-5.1-common libpq5 sysbench
0 upgraded, 4 newly installed, 0 to remove and 125 not upgraded.
Need to get 581 kB of archives.
After this operation, 1,641 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble/universe amd64 libluajit2-5.1-common all 2.1-20230410-1build1 [48.6 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 libluajit2-5.1-2 amd64 2.1-20230410-1build1 [277 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libpq5 amd64 16.6-0ubuntu0.24.04.1 [141 kB]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 sysbench amd64 1.0.20+ds-6build2 [114 kB]
Fetched 581 kB in 2s (292 kB/s)
Selecting previously unselected package libluajit2-5.1-common.
(Reading database ... 84906 files and directories currently installed.)
Preparing to unpack .../libluajit2-5.1-common_2.1-20230410-1build1_all.deb ...
Unpacking libluajit2-5.1-common (2.1-20230410-1build1) ...
Selecting previously unselected package libluajit2-5.1-2:amd64.
Preparing to unpack .../libluajit2-5.1-2_2.1-20230410-1build1_amd64.deb ...
Unpacking libluajit2-5.1-2:amd64 (2.1-20230410-1build1) ...
Selecting previously unselected package libpq5:amd64.
Preparing to unpack .../libpq5_16.6-0ubuntu0.24.04.1_amd64.deb ...
Unpacking libpq5:amd64 (16.6-0ubuntu0.24.04.1) ...
Selecting previously unselected package sysbench.
Preparing to unpack .../sysbench_1.0.20+ds-6build2_amd64.deb ...
Unpacking sysbench (1.0.20+ds-6build2) ...
Setting up libpq5:amd64 (16.6-0ubuntu0.24.04.1) ...
Setting up libluajit2-5.1-common (2.1-20230410-1build1) ...
Setting up libluajit2-5.1-2:amd64 (2.1-20230410-1build1) ...
Setting up sysbench (1.0.20+ds-6build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

mysql: Se utiliza el cliente de línea de comandos mysql para interactuar con el servidor de base de datos.

-uroot: Especifica que se va a conectar como el usuario root.

-p: Indica que se solicitará la contraseña.

-e "create database sbtest": Se ejecuta la instrucción SQL create database sbtest en el servidor. Esta instrucción crea una nueva base de datos llamada sbtest.

```
root@nodo1:/home/lili# mysql -uroot -p -e "create database sbtest"
Enter password:
root@nodo1:/home/lili#
```

sysbench: Se utiliza la herramienta sysbench para realizar pruebas de rendimiento en bases de datos.

--threads=1: Especifica que se utilizará un solo hilo para la prueba.

--db-driver=mysql: Indica que se utilizará el controlador de base de datos MySQL (o MariaDB).

--mysql-user=root: Especifica que se conectará a la base de datos como el usuario root.

--events=0: Establece el número de eventos a 0. En el contexto del subcomando prepare, esto significa que sysbench no ejecutará ninguna transacción de prueba real, solo preparará la base de datos.

oltp_read_only prepare: Este es el subcomando que le dice a sysbench que prepare la base de datos para una prueba de tipo oltp_read_only (transacciones en línea de solo lectura). La preparación incluye la creación de tablas, la inserción de datos de prueba y la creación de índices.

```
root@nodo1:/home/lili# sysbench  --threads=1 --db-driver=mysql --mysql-user=root --events=0 oltp_read_only prepare
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Creating table 'sbtest1'...
Inserting 10000 records into 'sbtest1'
Creating a secondary index on 'sbtest1'...
root@nodo1:/home/lili#
```

1. **sysbench --threads=1 --time=60 --report-interval=1 --db-driver=mysql --mysql-user=root --mysql-password=password --test=oltp_read_only run**: Ejecución de la prueba (con 1 hilo).
2. **sysbench --threads=2 --time=60 --report-interval=1 --db-driver=mysql --mysql-user=root --mysql-password=password --test=oltp_read_only run**: Ejecución de la prueba (con 2 hilos).

Análisis Combinado y Comparativo:

- **Preparación de la prueba:**
    - Se creó la tabla sbtest1, se insertaron 10,000 registros y se creó un índice secundario. Este paso es común para ambas pruebas.
- **Ejecución de la prueba con 1 hilo:**
    - Se inició la prueba con un solo hilo, se mostró información sobre las consultas en ejecución y se produjo un error al final debido a la falta de la tabla sbtest4.

- **Ejecución de la prueba con 2 hilos:**

    - La imagen muestra un error similar al ejecutar la prueba con 2 hilos: ERROR 1146 (42S02): Table 'sbtest.sbtest4' doesn't exist.

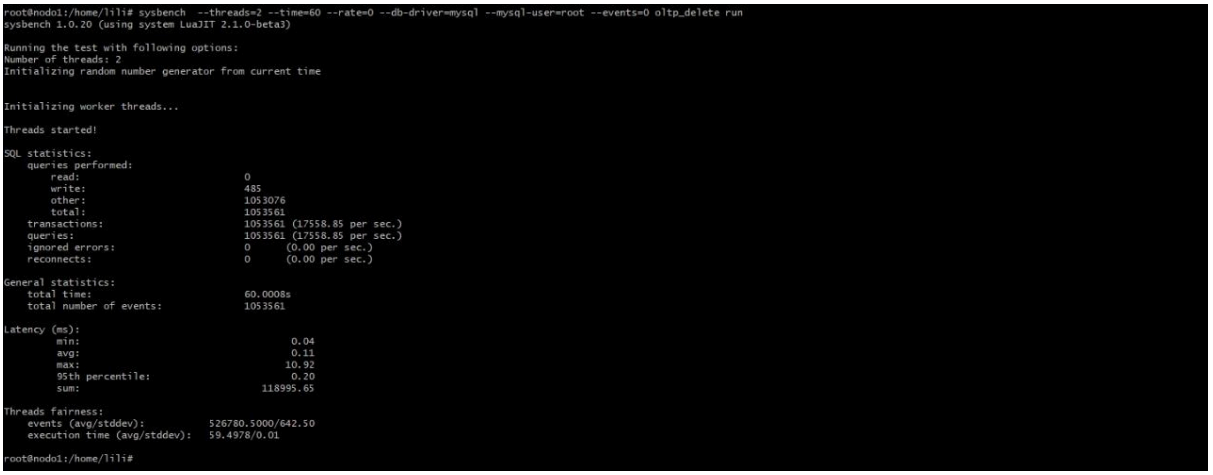
Ejecución de la prueba con 1 hilo:
sysbench --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --mysql-password=password --test=oltp_read_only run

Este comando ejecuta la prueba de lectura durante 60 segundos con un solo hilo, sin límite de velocidad de transacción (--rate=0).

Ejecución de la prueba con 2 hilos:
sysbench --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --mysql-password=password --test=oltp_read_only run

Este comando ejecuta la misma prueba durante 60 segundos, pero con dos hilos.

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_delete run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           5030
        other:                           570774
        total:                           575804
    transactions:                        575804 (9596.01 per sec.)
    queries:                             575804 (9596.01 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0005s
    total number of events:              575804

Latency (ms):
         min:                                    0.04
         avg:                                    0.10
         max:                                   18.02
         95th percentile:                        0.14
         sum:                                59605.39

Threads fairness:
    events (avg/stddev):           575804.0000/0.00
    execution time (avg/stddev):   59.6054/0.00

root@nodo1:/home/lili#
```



```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_delete run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           485
        other:                           1053076
        total:                           1053561
    transactions:                        1053561 (17558.85 per sec.)
    queries:                             1053561 (17558.85 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0008s
    total number of events:              1053561

Latency (ms):
         min:                                    0.04
         avg:                                    0.11
         max:                                   10.92
         95th percentile:                        0.20
         sum:                               118995.65

Threads fairness:
    events (avg/stddev):           526780.5000/642.50
    execution time (avg/stddev):   59.4978/0.01

root@nodo1:/home/lili#
```

Evaluar el rendimiento de la base de datos bajo diferentes cargas de trabajo OLTP y SELECT. Se probaron los siguientes tipos de prueba:

- oltp_insert
- oltp_point_select
- oltp_read_only
- oltp_read_write
- oltp_update_index
- oltp_update_non_index
- oltp_write_only
- select_random_points
- select_random_ranges

Aquí analizamos los resultados, comparando el rendimiento para cada tipo de prueba con 1 y 2 hilos. Identifica tendencias, cuellos de botella y el impacto del número de hilos en cada escenario.

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_insert run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           60147
        other:                           0
        total:                           60147
    transactions:                        60147  (1002.41 per sec.)
    queries:                             60147  (1002.41 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0012s
    total number of events:              60147

Latency (ms):
         min:                                    0.38
         avg:                                    0.99
         max:                                   39.80
         95th percentile:                        1.86
         sum:                                59767.84

Threads fairness:
    events (avg/stddev):           60147.0000/0.00
    execution time (avg/stddev):   59.7678/0.00

root@nodo1:/home/lili#
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_insert run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           92111
        other:                           0
        total:                           92111
    transactions:                        92111  (1535.11 per sec.)
    queries:                             92111  (1535.11 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0017s
    total number of events:              92111

Latency (ms):
         min:                                    0.50
         avg:                                    1.30
         max:                                  123.55
         95th percentile:                        2.22
         sum:                               119626.05

Threads fairness:
    events (avg/stddev):           46055.5000/3.50
    execution time (avg/stddev):   59.8130/0.00

root@nodo1:/home/lili#
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_point_select run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            298524
        write:                           0
        other:                           0
        total:                           298524
    transactions:                        298524 (4975.22 per sec.)
    queries:                             298524 (4975.22 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0006s
    total number of events:              298524

Latency (ms):
         min:                                    0.08
         avg:                                    0.20
         max:                                    2.99
         95th percentile:                        0.30
         sum:                                59657.92

Threads fairness:
    events (avg/stddev):           298524.0000/0.00
    execution time (avg/stddev):   59.6579/0.00

root@nodo1:/home/lili#
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_point_select run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            1022205
        write:                           0
        other:                           0
        total:                           1022205
    transactions:                        1022205 (17036.22 per sec.)
    queries:                             1022205 (17036.22 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0009s
    total number of events:              1022205

Latency (ms):
         min:                                    0.04
         avg:                                    0.12
         max:                                    4.38
         95th percentile:                        0.20
         sum:                               119194.70

Threads fairness:
    events (avg/stddev):           511102.5000/240.50
    execution time (avg/stddev):   59.5973/0.01

root@nodo1:/home/lili#
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_read_only run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            272860
        write:                           0
        other:                           38980
        total:                           311840
    transactions:                        19490  (324.82 per sec.)
    queries:                             311840 (5197.10 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0015s
    total number of events:              19490

Latency (ms):
         min:                                    1.33
         avg:                                    3.08
         max:                                    8.24
         95th percentile:                        3.75
         sum:                                59943.23

Threads fairness:
    events (avg/stddev):           19490.0000/0.00
    execution time (avg/stddev):   59.9432/0.00

root@nodo1:/home/lili#
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_read_only run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            894866
        write:                           0
        other:                           127838
        total:                           1022704
    transactions:                        63919  (1065.26 per sec.)
    queries:                             1022704 (17044.24 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0018s
    total number of events:              63919

Latency (ms):
         min:                                    0.87
         avg:                                    1.88
         max:                                    5.09
         95th percentile:                        2.07
         sum:                               119848.43

Threads fairness:
    events (avg/stddev):           31959.5000/24.50
    execution time (avg/stddev):   59.9242/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_read_write run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            139356
        write:                           34739
        other:                           24985
        total:                           199080
    transactions:                        9954   (165.89 per sec.)
    queries:                             199080 (3317.75 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0036s
    total number of events:              9954

Latency (ms):
         min:                                    3.69
         avg:                                    6.02
         max:                                   19.95
         95th percentile:                        7.43
         sum:                                59966.63

Threads fairness:
    events (avg/stddev):           9954.0000/0.00
    execution time (avg/stddev):   59.9666/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_read_write run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            291480
        write:                           79362
        other:                           45552
        total:                           416394
    transactions:                        20818  (346.94 per sec.)
    queries:                             416394 (6939.41 per sec.)
    ignored errors:                      2      (0.03 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0034s
    total number of events:              20818

Latency (ms):
         min:                                    2.30
         avg:                                    5.76
         max:                                   38.48
         95th percentile:                        8.43
         sum:                               119924.26

Threads fairness:
    events (avg/stddev):           10409.0000/1.00
    execution time (avg/stddev):   59.9621/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_index run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           66605
        other:                           2355
        total:                           68960
    transactions:                        68960  (1149.25 per sec.)
    queries:                             68960  (1149.25 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0032s
    total number of events:              68960

Latency (ms):
         min:                                    0.09
         avg:                                    0.87
         max:                                   24.45
         95th percentile:                        1.21
         sum:                                59904.86

Threads fairness:
    events (avg/stddev):           68960.0000/0.00
    execution time (avg/stddev):   59.9049/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_index run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           109875
        other:                           3847
        total:                           113722
    transactions:                        113722 (1895.29 per sec.)
    queries:                             113722 (1895.29 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0014s
    total number of events:              113722

Latency (ms):
         min:                                    0.06
         avg:                                    1.05
         max:                                   45.63
         95th percentile:                        1.44
         sum:                               119839.76

Threads fairness:
    events (avg/stddev):           56861.0000/8.00
    execution time (avg/stddev):   59.9199/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_non_index run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           67246
        other:                           2261
        total:                           69507
    transactions:                        69507  (1158.42 per sec.)
    queries:                             69507  (1158.42 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0006s
    total number of events:              69507

Latency (ms):
         min:                                    0.09
         avg:                                    0.86
         max:                                   35.46
         95th percentile:                        1.18
         sum:                                59904.41

Threads fairness:
    events (avg/stddev):           69507.0000/0.00
    execution time (avg/stddev):   59.9044/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_update_non_index run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           110338
        other:                           3787
        total:                           114125
    transactions:                        114125 (1902.01 per sec.)
    queries:                             114125 (1902.01 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0012s
    total number of events:              114125

Latency (ms):
         min:                                    0.07
         avg:                                    1.05
         max:                                   26.45
         95th percentile:                        1.42
         sum:                               119830.78

Threads fairness:
    events (avg/stddev):           57062.5000/51.50
    execution time (avg/stddev):   59.9154/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_write_only run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           121246
        other:                           63542
        total:                           184788
    transactions:                        30798  (513.28 per sec.)
    queries:                             184788 (3079.71 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0008s
    total number of events:              30798

Latency (ms):
         min:                                    0.96
         avg:                                    1.95
         max:                                   45.37
         95th percentile:                        2.43
         sum:                                59913.71

Threads fairness:
    events (avg/stddev):           30798.0000/0.00
    execution time (avg/stddev):   59.9137/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 oltp_write_only run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            0
        write:                           245612
        other:                           125064
        total:                           370676
    transactions:                        61778  (1029.59 per sec.)
    queries:                             370676 (6177.66 per sec.)
    ignored errors:                      3      (0.05 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0016s
    total number of events:              61778

Latency (ms):
         min:                                    0.78
         avg:                                    1.94
         max:                                   50.54
         95th percentile:                        2.86
         sum:                               119827.26

Threads fairness:
    events (avg/stddev):           30889.0000/29.00
    execution time (avg/stddev):   59.9136/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 select_random_points run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            1156
        write:                           0
        other:                           0
        total:                           1156
    transactions:                        1156   (19.25 per sec.)
    queries:                             1156   (19.25 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0603s
    total number of events:              1156

Latency (ms):
         min:                                   15.82
         avg:                                   51.95
         max:                                   89.65
         95th percentile:                       69.29
         sum:                                60050.06

Threads fairness:
    events (avg/stddev):           1156.0000/0.00
    execution time (avg/stddev):   60.0501/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 select_random_points run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            448462
        write:                           0
        other:                           0
        total:                           448462
    transactions:                        448462 (7474.08 per sec.)
    queries:                             448462 (7474.08 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0014s
    total number of events:              448462

Latency (ms):
         min:                                    0.09
         avg:                                    0.27
         max:                                    7.82
         95th percentile:                        0.39
         sum:                               119123.06

Threads fairness:
    events (avg/stddev):           224231.0000/1790.00
    execution time (avg/stddev):   59.5615/0.01
```

```
root@nodo1:/home/lili# sysbench  --threads=1 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 select_random_ranges run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 1
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            837
        write:                           0
        other:                           0
        total:                           837
    transactions:                        837    (13.95 per sec.)
    queries:                             837    (13.95 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0088s
    total number of events:              837

Latency (ms):
         min:                                   16.43
         avg:                                   71.68
         max:                                  250.08
         95th percentile:                       97.55
         sum:                                59997.83

Threads fairness:
    events (avg/stddev):           837.0000/0.00
    execution time (avg/stddev):   59.9978/0.00
```

```
root@nodo1:/home/lili# sysbench  --threads=2 --time=60 --rate=0 --db-driver=mysql --mysql-user=root --events=0 select_random_ranges run
sysbench 1.0.20 (using system LuaJIT 2.1.0-beta3)

Running the test with following options:
Number of threads: 2
Initializing random number generator from current time


Initializing worker threads...

Threads started!

SQL statistics:
    queries performed:
        read:                            427414
        write:                           0
        other:                           0
        total:                           427414
    transactions:                        427414 (7123.27 per sec.)
    queries:                             427414 (7123.27 per sec.)
    ignored errors:                      0      (0.00 per sec.)
    reconnects:                          0      (0.00 per sec.)

General statistics:
    total time:                          60.0016s
    total number of events:              427414

Latency (ms):
         min:                                    0.10
         avg:                                    0.28
         max:                                    4.31
         95th percentile:                        0.40
         sum:                               119165.82

Threads fairness:
    events (avg/stddev):           213707.0000/984.00
    execution time (avg/stddev):   59.5829/0.01
```

Conclusión

Rendimiento Variable: El rendimiento de la base de datos varió significativamente dependiendo del tipo de prueba ejecutada. Las pruebas que involucran principalmente operaciones de lectura (oltp_read_only, select_random_points, select_random_ranges) mostraron un rendimiento generalmente alto, mientras que las pruebas con operaciones de escritura intensivas (oltp_write_only, oltp_update_index, oltp_update_non_index) presentaron un rendimiento más bajo.

Impacto Limitado del Número de Hilos: En algunas pruebas, el aumento del número de hilos de 1 a 2 no produjo una mejora significativa en el rendimiento. Esto sugiere la presencia de cuellos de botella en otros recursos del sistema (E/S de disco, CPU, memoria) o limitaciones en la configuración de la base de datos.

Latencia: La latencia promedio se mantuvo relativamente baja en la mayoría de las pruebas, pero se observaron picos de latencia en algunas ejecuciones. Esto indica posibles problemas de concurrencia o de bloqueo en la base de datos.

Error de Tabla Inexistente: El error ERROR 1146 (42S02): Table 'sbtest.sbtest4' doesn't exist se presentó en varias pruebas, lo que sugiere un problema con la preparación de la base de datos o con la configuración de sysbench. Es crucial resolver este error para obtener resultados precisos.

Equidad de Hilos: El análisis de la equidad de hilos mostró que, en general, los hilos trabajaron de manera uniforme, aunque se observaron algunas variaciones en el número de eventos y el tiempo de ejecución.