



REPORTE:

# **INSTALAR GALERA 4 CLUSTER CON MARIADB EN LINUX**

ASIGNATURA:

**CÓMPUTO DE ALTO DESEMPEÑO**

**MARIA REGINA SARABIA HAU**

MATRÍCULA: 220300810

PROGRAMA EDUCATIVO: ING. EN DATOS E INTELIGENCIA ORGANIZACIONAL

PRESENTADO A:

**PROF. ISMAEL JIMÉNEZ SÁNCHEZ**

CANCÚN, QUINTANA ROO

SEPTIEMBRE 19, 2025

El objetivo de esta práctica fue:

1. Instalar y levantar un clúster Galera 4 con MariaDB (inicialmente 2 nodos, luego 3).
2. Verificar la replicación (sincronización de datos entre nodos).
3. Ejecutar un benchmark con Sysbench sobre una base de datos **testdb**, midiendo rendimiento durante 60 s por prueba, con 1 core y 2 cores.
4. Agregar un tercer nodo, repetir todas las pruebas y comparar los resultados (2 vs 3 nodos).
5. Documentar el proceso, resultados y conclusiones.

Recordatorio conceptual: Galera ofrece replicación síncrona multi-master. Esto favorece lecturas escalables y alta disponibilidad; las escrituras suelen escalar menos por la sincronía de commits entre nodos.

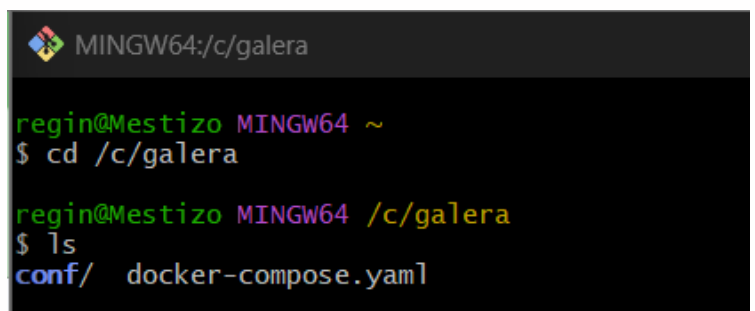
#### - Estructura de proyecto

Ruta local: **C:\galera**

Contenidos:

- docker-compose.yaml
- conf/galera.cnf (config de Galera)
- Carpeta results/ (salida de Sysbench con tee)

#### - docker-compose.yaml (resumen)



```
MINGW64:/c/galera
regin@Mestizo MINGW64 ~
$ cd /c/galera

regin@Mestizo MINGW64 /c/galera
$ ls
conf/  docker-compose.yaml
```

Servicios clave:

- galera-node1 (abre 3306) – **bootstrap** con --wsrep-new-cluster.

- galera-node2 (3307:3306) – se une con  
--wsrep-cluster-address=gcomm://galera-node1.
- galera-node3 (3308:3306) – se añadirá en la Fase 2.
- Contenedor sysbench para ejecutar pruebas (o se usa docker run puntual).

Variables de entorno comunes:

environment:

- MYSQL\_ROOT\_PASSWORD=mi\_super\_password
- MYSQL\_DATABASE=testdb

- **conf/galera.cnf (base)**

[mysqld]

bind-address = 0.0.0.0

wsrep\_on = ON

wsrep\_provider = /usr/lib/galera/libgalera\_smm.so

wsrep\_cluster\_name = "mi\_cluster\_galera"

wsrep\_sst\_method = rsync

wsrep\_node\_address = '0.0.0.0'

wsrep\_node\_name = 'node'

wsrep\_slave\_threads = 4

## 1. Levantar node1 y node2

cd /c/galera

docker compose up -d galera-node1

docker compose up -d galera-node2

```
regin@Mestizo MINGW64 /c/galera
$ docker compose up -d galera-node1
[+] Running 2/2
  ✓ Network galera_default Created
  ✓ Container galera-node1 Started

regin@Mestizo MINGW64 /c/galera
$ docker compose up -d galera-node2
[+] Running 2/2
  ✓ Container galera-node1 Running
  ✓ Container galera-node2 Started
```

## 2. Comprobar arranque correcto (buscar "ready / Synced / wsrep\_ready" en logs):

docker logs --tail=200 galera-node1 | grep -Ei  
"ready|Synced|wsrep\_ready|cluster"

```
regin@Mestizo MINGW64 /c/galera
$ # node1 (tiene --wsrep-new-cluster en tu docker-compose.yml)
docker compose up -d galera-node1

# Verifica que arrancó bien (mira "ready"/"Synced")
docker logs --tail=200 galera-node1 | grep -Ei "ready|Synced|wsrep_ready|wsrep_cluster"
[+] Running 2/2
  ✓ Network galera_default Created 0.1s
  ✓ Container galera-node1 Started 0.8s

regin@Mestizo MINGW64 /c/galera
$ # node2
docker compose up -d galera-node2

# confirma tamaño del cluster = 2
docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
[+] Running 2/2
  ✓ Container galera-node1 Running 0.0s
  ✓ Container galera-node2 Started 0.8s

+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 1 |
+-----+-----+
```

## 3. Confirmar tamaño del clúster = 2

docker exec -it galera-node1 mariadb -uroot -pmi\_super\_password \

-e "SHOW STATUS LIKE 'wsrep\_cluster\_size';"

```
regin@Mestizo MINGW64 /c/galera
$ docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
docker exec -it galera-node2 mariadb -uroot -pmi_super_password -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
# Deben reportar 2
```

Variable_name	Value
wsrep_cluster_size	2

Variable_name	Value
wsrep_cluster_size	2

#### 4. Crear base y tabla de prueba + validar replicación

En node1:

```
docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "
```

```
CREATE DATABASE IF NOT EXISTS testdb;
```

```
CREATE TABLE IF NOT EXISTS testdb.employees(
```

```
  id INT PRIMARY KEY AUTO_INCREMENT,
```

```
  name VARCHAR(50),
```

```
  dept VARCHAR(50),
```

```
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

```
);
```

```
INSERT INTO testdb.employees(name, dept) VALUES
```

```
('Ana','IT'),('Luis','QA'),('Perla','BI');
```

```
SELECT * FROM testdb.employees;"
```

```
regin@Mestizo MINGW64 /c/galera
$ docker exec -it galera-node2 mariadb -uroot -pmi_super_password -e "SELECT * FROM testdb.employees;"
+-----+
| id | name | dept | created_at |
+-----+
| 2 | Ana | IT | 2025-09-18 20:01:18 |
| 5 | Luis | QA | 2025-09-18 20:01:18 |
| 8 | Perla | BI | 2025-09-18 20:01:18 |
+-----+

regin@Mestizo MINGW64 /c/galera
$ docker exec -it galera-node3 mariadb -uroot -pmi_super_password -e "SELECT * FROM testdb.employees;"
+-----+
| id | name | dept | created_at |
+-----+
| 2 | Ana | IT | 2025-09-18 20:01:18 |
| 5 | Luis | QA | 2025-09-18 20:01:18 |
| 8 | Perla | BI | 2025-09-18 20:01:18 |
+-----+

regin@Mestizo MINGW64 /c/galera
$ mkdir -p results
docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_common \
  --db-driver=mysql \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 prepare
Unable to find image 'severalnines/sysbench:latest' locally
latest: Pulling from severalnines/sysbench
ba6ac542f035: Pull complete
1f1dcb85e92e: Pull complete
206bc115af8c: Pull complete
6f6da6c9c901: Pull complete
e79bb959ec00: Pull complete
Digest: sha256:64cd003bfa21eaab22f985e7b95f90d21a970229f5f628718657dd1bae669abd
Status: Downloaded newer image for severalnines/sysbench:latest
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Creating table 'sbtest1'...
Inserting 1000000 records into 'sbtest1'
Creating a secondary index on 'sbtest1'...
```

En node2 / node3 (cuando exista):

```
docker exec -it galera-node2 mariadb -uroot -pmi_super_password \
-e "SELECT * FROM testdb.employees;"
```

```
regin@Mestizo MINGW64 /c/galera
$ docker compose up -d galera-node3
[+] Running 2/2
  ✓ Container galera-node1 Running 0.0s
  ✓ Container galera-node3 Started 1.0s

regin@Mestizo MINGW64 /c/galera
$ docker logs --tail=120 galera-node3 | grep -Ei "ready|Synced|wsrep_ready|cluster"
2025-09-19 4:15:50 0 [Note] Starting MariaDB 10.5.29-MariaDB-ubu2004 source revision c461188ca6ad6ec3a
54201eb87ebd75797d296df server_uid hci+KogP554toa3jW6e6J+ZNpvg= as process 105
2025-09-19 4:15:50 0 [warning] You need to use --log-bin to make --binlog-format work.
2025-09-19 4:15:50 0 [Note] InnoDB: Uses event mutexes
2025-09-19 4:15:50 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2025-09-19 4:15:50 0 [Note] InnoDB: Number of pools: 1
2025-09-19 4:15:50 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2025-09-19 4:15:50 0 [Note] mysqld: O_TMPFILE is not supported on /tmp (disabling future attempts)
2025-09-19 4:15:50 0 [Note] InnoDB: Using Linux native AIO
2025-09-19 4:15:50 0 [Note] InnoDB: Initializing buffer pool, total size = 134217728, chunk size = 134
217728
2025-09-19 4:15:50 0 [Note] InnoDB: Completed initialization of buffer pool
2025-09-19 4:15:50 0 [Note] InnoDB: 128 rollback segments are active.
2025-09-19 4:15:50 0 [Note] InnoDB: Creating shared tablespace for temporary tables
2025-09-19 4:15:50 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file
full: Please wait ...
2025-09-19 4:15:50 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
2025-09-19 4:15:50 0 [Note] InnoDB: 10.5.29 started; log sequence number 45079; transaction id 20
2025-09-19 4:15:50 0 [Note] Plugin 'FEEDBACK' is disabled.
2025-09-19 4:15:50 0 [Note] Reading of all Master_info entries succeeded
2025-09-19 4:15:50 0 [Note] Added new Master_info '' to hash table
2025-09-19 4:15:50 0 [Note] mysqld: ready for connections.
version: '10.5.29-MariaDB-ubu2004' socket: '/run/mysqld/mysqld.sock' port: 0 mariadb.org binary dist
ribution

regin@Mestizo MINGW64 /c/galera
$ docker exec -it galera-node1 mariadb -uroot -pmi_super_password \
-e "SHOW STATUS LIKE 'wsrep_cluster_size';"
+-----+
| Variable_name | Value |
+-----+
| wsrep_cluster_size | 3 |
+-----+
```

## Preparación del dataset para Sysbench

### - Tablas OLTP (sptest1) – preparar/cargar

Las pruebas OLTP comparten el esquema estándar de Sysbench (sptest); no confundir con bulk\_insert, que usa su propia rutina.

Variables de conveniencia (para no reescribir):

HOST=galera-node1; USER=root; PASS=mi\_super\_password; NET=galera\_default  
mkdir -p results

Preparación de datos (1 tabla con 1,000,000 filas):

```
docker run --rm --network=$NET severalnines/sysbench \  
sysbench oltp_common \  
--db-driver=mysql \  
--mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \  
--mysql-db=testdb --tables=1 --table-size=1000000 prepare
```

Si aparece “Table ‘sptest1’ already exists”, puedes verificar conteo:

```
docker exec -it galera-node1 mariadb -uroot -pmi_super_password \  
-e "SELECT COUNT(*) AS filas FROM testdb.sptest1;"
```

```
regin@Mestizo MINGW64 /c/galera  
$ # crea/asegura la BD  
docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "CREATE DATABASE IF NOT EXISTS testdb;"  
  
# prepara tablas y datos (1M)  
docker run --rm --network=galera_default severalnines/sysbench \  
sysbench oltp_common \  
--db-driver=mysql \  
--mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \  
--mysql-db=testdb --tables=1 --table-size=1000000 prepare  
  
# verificación rápida (debería ~1,000,000)  
docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "SELECT COUNT(*) AS filas FROM testdb.sptest1;"  
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)  
  
Creating table 'sptest1'...  
Inserting 1000000 records into 'sptest1'  
Creating a secondary index on 'sptest1'...  
+-----+  
| filas |  
+-----+  
| 1000000 |  
+-----+
```

## Metodología del benchmark

- Duración: 60 s por prueba.
- Concurrencia: --threads=1 (para normalizar)
- CPU asignada al contenedor de sysbench:
  - 1 core → --cpus=1
  - 2 cores → --cpus=2
- Resultados: redirigidos a results/\*.txt con tee.

## Métrica de interés:

- Transactions (eventos) y events/sec (rendimiento).
- Latency avg (ms) (tiempo de respuesta promedio).
- En pruebas de lectura, se muestran queries totales.

## Ejecución de pruebas (2 nodos)

A continuación listadas por orden. En cada sub-sección te dejo la marca de captura para que pegues las screenshots que ya generaste (1 core y 2 cores).

### 1. bulk\_insert (*nota especial*)

- Al inicio apareció error 1136 / Duplicate / Column count al re-usar la tabla.
- Solución aplicada: ejecutar la prueba con su propio ciclo (drop/create que hace sysbench) o usar un schema aislado (p.ej. --mysql-db=testbulk) para evitar colisiones con sbtest1.
- Resultado (cuando se dejó que sysbench maneje la tabla): ~5,395,092 eventos (~89,621/s).



```

MINGW64/c/galara
| tee results/bulk_insert_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:         182
    other:          0
    total:         182
  transactions:    5395092 (89887.09 per sec.)
  queries:         182 (3.03 per sec.)
  ignored errors:  0 (0.00 per sec.)
  reconnects:      0 (0.00 per sec.)

General statistics:
  total time:      60.0191s
  total number of events: 5395092

Latency (ms):
  min:            0.00
  avg:            0.01
  max:           3831.06
  95th percentile: 0.00
  sum:           58689.38

Threads fairness:
  events (avg/stddev): 5395092.0000/0.00
  execution time (avg/stddev): 58.6894/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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

Initializing worker threads...

Threads started!

FATAL: mysql_drv_query() returned error 1062 (Duplicate entry '1' for key 'PRIMARY') for query 'INSERT INTO sbtest1 VALUES(1,1),(2,2),(3,3),(4,4),(5,5),(6,6),(7,7),(8,8),(9,9),(10,10),(11,11),(12,12),(13,13),(14,14),(15,15),(16,16),(17,17),(18,18),(19,19),(20,20),(21,21),(22,22),(23,23),(24,24),(25,25),(26,26),(27,27),(28,28),(29,29),(30,30),(31,31),(32,32),(33,33),(34,34),(35,35),(36,36),(37,37),(38,38),(39,39),(40,40),(41,41),(42,42),(43,43),(44,44),(45,45),(46,46),(47,47),(48,48),(49,49),(50,50),(51,51),(52,52),(53,53),(54,54),(55,55),(56,56),(57,57),(58,58),(59,59),(60,60),(61,61),(62,62),(63,63),(64,64),(65,65),(66,66),(67,67),(68,68),(69,69),(70,70),(71,71),(72,72),(73,73),(74,74),(75,75),(76,76),(77,77),(78,78),(79,79),(80,80),(81,81),(82,82),(83,83),(84,84),(85,85),(86,86),(87,87),(88,88),(89,89),(90,90),(91,91),(92,92),(93,93),(94,94),(95,95),(96,96),(97,97),(98,98),(99,99),(100,100),(101,101),(102,102),(103,103),(104,104),(105,105),(106,106),(107,107),(108,108),(109,109),(110,110),(111,111),(112,112),(113,113),(114,114),(115,115),(116,116),(117,117),(118,118),(119,119),(120,120),(121,121),(122,122),(123,123),(124,124),(125,125),(126,126),(127,127),(128,128),(129,129),(130,130),(131,131),(132,132),(133,133),(134,134),(135,135),(136,136),(137,137),(138,138),(139,139),(140,140),(141,141),(142,142),(143,143),(144,144),(145,145),(146,146),(147,147),(148,148),(149,149),(150,150),(151,151),(152,152),(153,153),(154,154),(155,155),(156,156),(157,157),(158,158),(159,159),(160,160),(161,161),(162,162),(163,163),(164,164),(165,165),(166,166),(167,167),(168,168),(169,169),(170,170),(171,171),(172,172),(173,173),(174,174),(175,175),(176,176),(177,177),(178,178),(179,179),(180,180),(181,181),(182,182),(183,183),(184,184),(185,185),(186,186),(187,187),(188,188),(189,189),(190,190),(191,191),(192,192),(193,193),(194,194),(195,195),(196,196),(197,197),(198,198),(199,199),(200,200),(201,201),(202,202),(203,203),(204,204),(205,205),(206,206),(207,207),(208,208),(209,209),(210,210),(211,211),(212,212),(213,213),(214,214),(215,215),(216,216),(217,217),(218,218),(219,219),(220,220),(221,221),(222,222),(223,223),(224,224),(225,225),(226,226),(227,227),(228,228),(229,229),(230,230),(231,231),(232,232),(233,233),(234,234),(235,235),(236,236),(237,237),(238,238),(239,239),(240,240),(241,241),(242,242),(243,243),(244,244),(245,245),(246,246),(247,247),(248,248),(249,249),(250,250),(251,251),(252,252),(253,253),(254,254),(255,255),(256,256),(257,257),(258,258),(259,259),(260,260),(261,261),(262,262),(263,263),(264,264),(265,265),(266,266),(267,267),(268,268),(269,269),(270,270),(271,271),(272,272),(273,273),(274,274),(275,275),(276,276),(277,277),(278,278),(279,279),(280,280),(281,281),(282,282),(283,283),(284,284),(285,285),(286,286),(287,287),(288,288),(289,289),(290,290),(291,291),(292,292),(293,293),(294,294),(295,295),(296,296),(297,297),(298,298),(299,299),(300,300),(301,301),(302,302),(303,303),(304,304),(305,305),(306,306),(307,307),(308,308),(309,309),(310,310),(311,311),(312,312),(313,313),(314,314),(315,315),(316,316),(317,317),(318,318),(319,319),(320,320),(321,321),(322,322),(323,323),(324,324),(325,325),(326,326),(327,327),(328,328),(329,329),(330,330),(331,331),(332,332),(333,333),(334,334),(335,335),(336,336),(337,337),(338,338),(339,339),(340,340),(341,341),(342,342),(343,343),(344,344),(345,345),(346,346),(347,347),(348,348),(349,349),(350,350),(351,351),(352,352),(353,353),(354,354),(355,355),(356,356),(357,357),(358,358),(359,359),(360,360),(361,361),(362,362),(363,363),(364,364),(365,365),(366,366),(367,367),(368,368),(369,369),(370,370),(371,371),(372,372),(373,373),(374,374),(375,375),(376,376),(377,377),(378,378),(379,379),(380,380),(381,381),(382,382),(383,383),(384,384),(385,385),(386,386),(387,387),(388,388),(389,389),(390,390),(391,391),(392,392),(393,393),(394,394),(395,395),(396,396),(397,397),(398,398),(399,399),(400,400),(401,401),(402,402),(403,403),(404,404),(405,405),(406,406),(407,407),(408,408),(409,409),(410,410),(411,411),(412,412),(413,413),(414,414),(415,415),(416,416),(417,417),(418,418),(419,419),(420,420)'
k-insert_next() failed
Error in my_thread_global_end(): 1 threads didn't exit

```

**Observación:** en bulk\_insert los “events” reflejan lotes de inserciones (no equivalen 1-a-1 a filas). Lo importante es la tasa de events/sec para comparar entre 2 y 3 nodos.

## 2. oltp\_delete

Comandos (2 nodos):

```

docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_delete --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \

```

```
--mysql-db=testdb --tables=1 --table-size=1000000 run | tee
results/oltp_delete_2nodes_1core.txt
```

```
docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_delete --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run | tee
results/oltp_delete_2nodes_2cores.txt
```

```
regin@Mestizo MINGW64 /c/galera
$ docker stop galera-node3
docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
# Debe decir 2
galera-node3
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 2 |
+-----+-----+

regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_delete --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_delete_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:         7469
    other:        56352
    total:        63821
  transactions:    63821 (1063.57 per sec.)
  queries:         63821 (1063.57 per sec.)
  ignored errors:  0 (0.00 per sec.)
  reconnects:     0 (0.00 per sec.)

General statistics:
  total time:      60.0044s
  total number of events: 63821

Latency (ms):
  min:            0.18
  avg:            0.93
  max:            71.64
  95th percentile: 3.43
  sum:           59653.95

Threads fairness:
  events (avg/stddev): 63821.0000/0.00
  execution time (avg/stddev): 59.6539/0.00
```

```

regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
  sysbench oltp_delete --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_delete_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               4815
    other:               60129
    total:               64944
  transactions:         64944 (1082.32 per sec.)
  queries:               64944 (1082.32 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0022s
  total number of events: 64944

Latency (ms):
  min:                   0.18
  avg:                   0.92
  max:                   461.35
  95th percentile:      3.36
  sum:                   59577.88

Threads fairness:
  events (avg/stddev):   64944.0000/0.00
  execution time (avg/stddev): 59.5779/0.00

```

**Hallazgo: 2 cores ↑ rendimiento vs 1 core.**

### 3. oltp\_insert

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_insert --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_insert_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               17270
    other:                0
    total:               17270
  transactions:         17270 (287.81 per sec.)
  queries:              17270 (287.81 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0033s
  total number of events: 17270

Latency (ms):
  min:                   2.37
  avg:                   3.46
  max:                   60.76
  95th percentile:      4.74
  sum:                   59817.75

Threads fairness:
  events (avg/stddev):   17270.0000/0.00
  execution time (avg/stddev): 59.8177/0.00
```

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
  sysbench oltp_insert --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_insert_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               17403
    other:                0
    total:               17403
  transactions:         17403 (290.03 per sec.)
  queries:              17403 (290.03 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0035s
  total number of events: 17403

Latency (ms):
  min:                   2.48
  avg:                   3.44
  max:                   60.02
  95th percentile:      4.25
  sum:                   59841.18

Threads fairness:
  events (avg/stddev):   17403.0000/0.00
  execution time (avg/stddev): 59.8412/0.00
```

#### 4. oltp\_point\_select (lectura pura, punto único)

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_point_select --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_point_select_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                  129923
    write:                  0
    other:                  0
    total:                  129923
  transactions:           129923 (2165.08 per sec.)
  queries:                 129923 (2165.08 per sec.)
  ignored errors:          0      (0.00 per sec.)
  reconnects:              0      (0.00 per sec.)

General statistics:
  total time:              60.0047s
  total number of events:  129923

Latency (ms):
  min:                     0.15
  avg:                     0.46
  max:                     50.82
  95th percentile:        0.86
  sum:                     59488.24

Threads fairness:
  events (avg/stddev):     129923.0000/0.00
  execution time (avg/stddev): 59.4882/0.00
```

```

regin@mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
  sysbench oltp_point_select --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_point_select_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                159753
    write:                0
    other:                0
    total:                159753
  transactions:         159753 (2662.42 per sec.)
  queries:               159753 (2662.42 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: 159753

Latency (ms):
  min:                   0.14
  avg:                   0.37
  max:                   16.78
  95th percentile:      0.67
  sum:                   59423.01

Threads fairness:
  events (avg/stddev):   159753.0000/0.00
  execution time (avg/stddev): 59.4230/0.00

```

**Hallazgo:** Lecturas escalan muy bien con 2 cores.

## 5. oltp\_read\_only (lecturas con otras operaciones ligeras)

```
regin@Mestizo MINGW64 /c/galera
$ docker exec -it galera-node1 mariadb -uroot -pmi_super_password -e "SHOW STATUS LIKE 'wsrep_cluster_size';"
# Debe decir 2
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 2 |
+-----+-----+

regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_read_only --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_read_only_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                147756
    write:                0
    other:               21108
    total:              168864
  transactions:        10554 (175.88 per sec.)
  queries:             168864 (2814.11 per sec.)
  ignored errors:      0 (0.00 per sec.)
  reconnects:          0 (0.00 per sec.)

General statistics:
  total time:          60.0050s
  total number of events: 10554

Latency (ms):
  min:                 2.74
  avg:                 5.68
  max:                 49.29
  95th percentile:    9.39
  sum:                 59921.34

Threads fairness:
  events (avg/stddev): 10554.0000/0.00
  execution time (avg/stddev): 59.9213/0.00
```

**Comentario:** ligera variabilidad; con más iteraciones se promediaria mejor.

## 6. oltp\_read\_write (mixto)

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
sysbench oltp_read_write --time=60 --threads=1 \
--mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
--mysql-db=testdb --tables=1 --table-size=1000000 run \
| tee results/oltp_read_write_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                67620
    write:               8224
    other:              20756
    total:             96600
  transactions:        4830 (80.48 per sec.)
  queries:            96600 (1609.67 per sec.)
  ignored errors:      0 (0.00 per sec.)
  reconnects:         0 (0.00 per sec.)

General statistics:
  total time:          60.0111s
  total number of events: 4830

Latency (ms):
  min:                 6.65
  avg:                 12.41
  max:                 84.66
  95th percentile:    17.95
  sum:                 59960.08

Threads fairness:
  events (avg/stddev): 4830.0000/0.00
  execution time (avg/stddev): 59.9601/0.00
```

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
sysbench oltp_read_write --time=60 --threads=1 \
--mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
--mysql-db=testdb --tables=1 --table-size=1000000 run \
| tee results/oltp_read_write_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                68474
    write:              11865
    other:              17481
    total:             97820
  transactions:        4891 (81.51 per sec.)
  queries:            97820 (1630.20 per sec.)
  ignored errors:      0 (0.00 per sec.)
  reconnects:         0 (0.00 per sec.)

General statistics:
  total time:          60.0037s
  total number of events: 4891

Latency (ms):
  min:                 7.21
  avg:                 12.26
  max:                 113.90
  95th percentile:    18.28
  sum:                 59955.64

Threads fairness:
  events (avg/stddev): 4891.0000/0.00
  execution time (avg/stddev): 59.9556/0.00
```



## 7. oltp\_update\_index

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_update_index --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_index_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               15386
    other:               12695
    total:               28081
  transactions:         28081 (468.00 per sec.)
  queries:               28081 (468.00 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: 28081

Latency (ms):
  min:                   0.20
  avg:                   2.13
  max:                   54.41
  95th percentile:      3.96
  sum:                   59830.91

Threads fairness:
  events (avg/stddev):   28081.0000/0.00
  execution time (avg/stddev): 59.8309/0.00
```

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
  sysbench oltp_update_index --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_index_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               14842
    other:               12122
    total:              26964
  transactions:         26964 (449.37 per sec.)
  queries:              26964 (449.37 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: 26964

Latency (ms):
  min:                  0.20
  avg:                   2.22
  max:                  72.93
  95th percentile:     4.33
  sum:                  59844.54

Threads fairness:
  events (avg/stddev):  26964.0000/0.00
  execution time (avg/stddev): 59.8445/0.00
```

## 8. oltp\_update\_non\_index

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_update_non_index --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_non_index_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               15826
    other:               12971
    total:               28797
  transactions:         28797 (479.92 per sec.)
  queries:               28797 (479.92 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0024s
  total number of events: 28797

Latency (ms):
  min:                   0.19
  avg:                   2.08
  max:                   52.19
  95th percentile:      3.89
  sum:                   59822.43

Threads fairness:
  events (avg/stddev):   28797.0000/0.00
  execution time (avg/stddev): 59.8224/0.00
```

```

regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
  sysbench oltp_update_non_index --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_non_index_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               15136
    other:               12576
    total:               27712
  transactions:         27712 (461.85 per sec.)
  queries:               27712 (461.85 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0010s
  total number of events: 27712

Latency (ms):
  min:                   0.20
  avg:                   2.16
  max:                   63.53
  95th percentile:      4.10
  sum:                   59833.31

Threads fairness:
  events (avg/stddev):   27712.0000/0.00
  execution time (avg/stddev): 59.8333/0.00

```

## 9. oltp\_write\_only

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=1 severalnines/sysbench \
  sysbench oltp_write_only --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_write_only_2nodes_1core.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               30121
    other:               31007
    total:               61128
  transactions:        10188 (169.78 per sec.)
  queries:             61128 (1018.68 per sec.)
  ignored errors:      0 (0.00 per sec.)
  reconnects:          0 (0.00 per sec.)

General statistics:
  total time:           60.0061s
  total number of events: 10188

Latency (ms):
  min:                  3.69
  avg:                   5.88
  max:                  530.47
  95th percentile:     8.28
  sum:                  59922.92

Threads fairness:
  events (avg/stddev): 10188.0000/0.00
  execution time (avg/stddev): 59.9229/0.00
```

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=galera_default --cpus=2 severalnines/sysbench \
  sysbench oltp_write_only --time=60 --threads=1 \
  --mysql-host=galera-node1 --mysql-user=root --mysql-password=mi_super_password \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_write_only_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               33965
    other:               27643
    total:               61608
  transactions:        10268 (171.12 per sec.)
  queries:             61608 (1026.72 per sec.)
  ignored errors:      0 (0.00 per sec.)
  reconnects:          0 (0.00 per sec.)

General statistics:
  total time:           60.0031s
  total number of events: 10268

Latency (ms):
  min:                  3.76
  avg:                   5.84
  max:                  2599.03
  95th percentile:     7.17
  sum:                  59917.21

Threads fairness:
  events (avg/stddev): 10268.0000/0.00
  execution time (avg/stddev): 59.9172/0.00
```

## 10. select\_random\_points

```
regin@Mestizo MINGW64 /c/galera
$ # 1 core
docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench select_random_points --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_points_2nodes_1core.txt

# 2 cores
docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench select_random_points --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_points_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                6328
    write:                0
    other:                0
    total:               6328
  transactions:         6328 (105.44 per sec.)
  queries:              6328 (105.44 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0112s
  total number of events: 6328

Latency (ms):
  min:                   2.13
  avg:                   9.47
  max:                   65.82
  95th percentile:      17.01
  sum:                   59952.59
```

```
Threads fairness:
  events (avg/stddev):       6328.0000/0.00
  execution time (avg/stddev): 59.9526/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:              7041
    write:              0
    other:              0
    total:              7041
  transactions:        7041  (117.34 per sec.)
  queries:              7041  (117.34 per sec.)
  ignored errors:       0      (0.00 per sec.)
  reconnects:           0      (0.00 per sec.)

General statistics:
  total time:           60.0060s
  total number of events: 7041

Latency (ms):
  min:                  2.84
  avg:                   8.51
  max:                  69.43
  95th percentile:     12.30
  sum:                  59942.43

Threads fairness:
  events (avg/stddev):       7041.0000/0.00
  execution time (avg/stddev): 59.9424/0.00
```

## 11. select\_random\_ranges

```
regin@Mestizo MINGW64 /c/galera
$ # 1 core
docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench select_random_ranges --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_ranges_2nodes_1core.txt

# 2 cores
docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench select_random_ranges --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_ranges_2nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                22376
    write:                0
    other:                0
    total:                22376
  transactions:          22376 (372.91 per sec.)
  queries:                22376 (372.91 per sec.)
  ignored errors:         0      (0.00 per sec.)
  reconnects:             0      (0.00 per sec.)

General statistics:
  total time:             60.0021s
  total number of events: 22376

Latency (ms):
  min:                    0.99
  avg:                     2.67
  max:                    19.58
  95th percentile:        3.68
  sum:                    59841.26

Threads fairness:
  events (avg/stddev):    22376.0000/0.00
  execution time (avg/stddev): 59.8413/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)
```



```
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:                21778
    write:                0
    other:                0
    total:                21778
  transactions:          21778 (362.88 per sec.)
  queries:                21778 (362.88 per sec.)
  ignored errors:         0 (0.00 per sec.)
  reconnects:             0 (0.00 per sec.)

General statistics:
  total time:              60.0118s
  total number of events:  21778

Latency (ms):
  min:                     0.81
  avg:                     2.75
  max:                     21.47
  95th percentile:        3.89
  sum:                     59816.35

Threads fairness:
  events (avg/stddev):      21778.0000/0.00
  execution time (avg/stddev): 59.8164/0.00
```

## Expansión a 3 nodos (Fase 2) y verificación

1. Levantar node3

```
docker compose up -d galera-node3
```

2. Confirmar tamaño del clúster = 3

```
docker exec -it galera-node1 mariadb -uroot -pmi_super_password \
-e "SHOW STATUS LIKE 'wsrep_cluster_size';"
```

```

regin@Mestizo MINGW64 /c/galera
$ docker compose up -d galera-node3
[+] Running 2/2
  ✓ Container galera-node1  Running      0.0s
  ✓ Container galera-node3  Started    1.0s

regin@Mestizo MINGW64 /c/galera
$ docker logs --tail=120 galera-node3 | grep -Ei "ready|Synced|wsrep_ready|cluster"
2025-09-19 4:15:50 0 [Note] Starting MariaDB 10.5.29-MariaDB-ubu2004 source revision c461188ca6ad6ec3a
54201eb87ebd75797d296df server_uid hci+KoGP554toa3jW6e6J+ZNpvg= as process 105
2025-09-19 4:15:50 0 [Warning] You need to use --log-bin to make --binlog-format work.
2025-09-19 4:15:50 0 [Note] InnoDB: Uses event mutexes
2025-09-19 4:15:50 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
2025-09-19 4:15:50 0 [Note] InnoDB: Number of pools: 1
2025-09-19 4:15:50 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
2025-09-19 4:15:50 0 [Note] mysqld: O_TMPFILE is not supported on /tmp (disabling future attempts)
2025-09-19 4:15:50 0 [Note] InnoDB: Using Linux native AIO
2025-09-19 4:15:50 0 [Note] InnoDB: Initializing buffer pool, total size = 134217728, chunk size = 134
217728
2025-09-19 4:15:50 0 [Note] InnoDB: Completed initialization of buffer pool
2025-09-19 4:15:50 0 [Note] InnoDB: 128 rollback segments are active.
2025-09-19 4:15:50 0 [Note] InnoDB: Creating shared tablespace for temporary tables
2025-09-19 4:15:50 0 [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file
full; Please wait ...
2025-09-19 4:15:50 0 [Note] InnoDB: File './ibtmp1' size is now 12 MB.
2025-09-19 4:15:50 0 [Note] InnoDB: 10.5.29 started; log sequence number 45079; transaction id 20
2025-09-19 4:15:50 0 [Note] Plugin 'FEEDBACK' is disabled.
2025-09-19 4:15:50 0 [Note] Reading of all Master_info entries succeeded
2025-09-19 4:15:50 0 [Note] Added new Master_info '' to hash table
2025-09-19 4:15:50 0 [Note] mysqld: ready for connections.
version: '10.5.29-MariaDB-ubu2004' socket: '/run/mysqld/mysqld.sock' port: 0 mariadb.org binary dist
ribution

regin@Mestizo MINGW64 /c/galera
$ docker exec -it galera-node1 mariadb -uroot -pmi_super_password \
-e "SHOW STATUS LIKE 'wsrep_cluster_size';"
+-----+-----+
| Variable_name | Value |
+-----+-----+
| wsrep_cluster_size | 3 |
+-----+-----+

```

**Nota:** Si en algún momento se cayó el engine o hubo orphans, se aplicó `docker compose down --remove-orphans` y se recreó el clúster (bootstrap en node1, join en node2/3).

## 8) Ejecución de pruebas (3 nodos)

Se repite idéntico set con 1 core y 2 cores. Pega cada captura debajo.

### 1. bulk\_insert (3 nodos)

- Tras permitir a sysbench manejar drop/create, OK.

```
# Ejecuta la prueba (2 cores)
docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench bulk_insert --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testbulk --tables=1 --table-size=1000000 run \
  | tee results/bulk_insert_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Dropping table 'sbtest1'...
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Creating table 'sbtest1'...
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               182
    other:                0
    total:               182
  transactions:         5395092 (89621.56 per sec.)
  queries:               182   (3.02 per sec.)
  ignored errors:        0     (0.00 per sec.)
  reconnects:            0     (0.00 per sec.)

General statistics:
  total time:            60.1960s
  total number of events: 5395092

Latency (ms):
  min:                   0.00
  avg:                   0.01
  max:                   5100.25
  95th percentile:      0.00
  sum:                   58741.30

Threads fairness:
  events (avg/stddev):   5395092.0000/0.00
  execution time (avg/stddev): 58.7413/0.00
```

```

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Dropping table 'sbtest1'...
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Creating table 'sbtest1'...
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               182
    other:                0
    total:               182
  transactions:         5395092 (89621.56 per sec.)
  queries:               182   (3.02 per sec.)
  ignored errors:        0     (0.00 per sec.)
  reconnects:            0     (0.00 per sec.)

General statistics:
  total time:            60.1960s
  total number of events: 5395092

Latency (ms):
  min:                   0.00
  avg:                   0.01
  max:                   5100.25
  95th percentile:      0.00
  sum:                   58741.30

Threads fairness:
  events (avg/stddev):   5395092.0000/0.00
  execution time (avg/stddev): 58.7413/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Dropping table 'sbtest1'...
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Creating table 'sbtest1'...
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:

```

## 2. oltp\_delete

**Comentario:** La escritura síncrona distribuye el costo; a 3 nodos puede verse ↓ TPS en deletes.

```

regin@Mestizo MINGW64 /c/galera
$ # 1 core
docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_delete --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_delete_3nodes_1core.txt

# 2 cores
docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_delete --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_delete_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                               10423
    other:                               46024
    total:                               56447
  transactions:      56447 (940.71 per sec.)
  queries:           56447 (940.71 per sec.)
  ignored errors:    0      (0.00 per sec.)
  reconnects:        0      (0.00 per sec.)

General statistics:
  total time:                60.0011s
  total number of events:    56447

Latency (ms):
  min:                        0.15
  avg:                        1.06
  max:                        57.71
  95th percentile:           3.89
  sum:                        59702.41

Threads fairness:
  events (avg/stddev):       56447.0000/0.00
  execution time (avg/stddev): 59.7024/0.00

```

```
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               5581
    other:              43475
    total:             49056
  transactions:        49056 (817.57 per sec.)
  queries:            49056 (817.57 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: 49056

Latency (ms):
  min:                 0.16
  avg:                 1.22
  max:                1094.45
  95th percentile:    4.10
  sum:                59774.19

Threads fairness:
  events (avg/stddev): 49056.0000/0.00
  execution time (avg/stddev): 59.7742/0.00
```

### 3. oltp\_insert

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_insert --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_insert_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_insert --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_insert_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               17481
    other:               0
    total:              17481
  transactions:        17481 (291.33 per sec.)
  queries:             17481 (291.33 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: 17481

Latency (ms):
  min:                 2.41
  avg:                 3.42
  max:                 336.43
  95th percentile:    4.57
  sum:                 59821.06

Threads fairness:
  events (avg/stddev): 17481.0000/0.00
  execution time (avg/stddev): 59.8211/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

Running the test with following options:
```

**Mejora vs 2 nodos (especialmente con 2 cores).**

```
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: 18875
    other: 0
    total: 18875
  transactions: 18875 (314.48 per sec.)
  queries: 18875 (314.48 per sec.)
  ignored errors: 0 (0.00 per sec.)
  reconnects: 0 (0.00 per sec.)

General statistics:
  total time: 60.0019s
  total number of events: 18875

Latency (ms):
  min: 2.46
  avg: 3.17
  max: 111.61
  95th percentile: 3.82
  sum: 59833.78

Threads fairness:
  events (avg/stddev): 18875.0000/0.00
  execution time (avg/stddev): 59.8338/0.00
```



#### 4. oltp\_point\_select

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_point_select --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_point_select_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_point_select --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_point_select_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                226140
    write:                0
    other:                0
    total:                226140
  transactions:         226140 (3768.87 per sec.)
  queries:               226140 (3768.87 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: 226140

Latency (ms):
  min:                    0.12
  avg:                    0.26
  max:                    89.36
  95th percentile:       0.47
  sum:                    59594.13

Threads fairness:
  events (avg/stddev):    226140.0000/0.00
  execution time (avg/stddev): 59.5941/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)
```

**Mejora clara con 3 nodos en lectura pura.**

```
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:                248860
    write:                0
    other:                0
    total:                248860
  transactions:          248860 (4146.76 per sec.)
  queries:                248860 (4146.76 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: 248860

Latency (ms):
  min:                    0.11
  avg:                    0.24
  max:                    16.63
  95th percentile:       0.41
  sum:                    59623.01

Threads fairness:
  events (avg/stddev):    248860.0000/0.00
  execution time (avg/stddev): 59.6230/0.00
```

## 5. oltp\_read\_only

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_read_only --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_read_only_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_read_only --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_read_only_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                195748
    write:                0
    other:               27964
    total:              223712
  transactions:        13982 (233.02 per sec.)
  queries:             223712 (3728.28 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: 13982

Latency (ms):
  min:                  2.87
  avg:                   4.28
  max:                  55.97
  95th percentile:     7.30
  sum:                  59905.92

Threads fairness:
  events (avg/stddev):  13982.0000/0.00
  execution time (avg/stddev): 59.9059/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                186606
    write:               0
    other:              26658
    total:             213264
  transactions:        13329 (222.13 per sec.)
  queries:            213264 (3554.10 per sec.)
  ignored errors:      0      (0.00 per sec.)
  reconnects:         0      (0.00 per sec.)

General statistics:
  total time:          60.0040s
  total number of events: 13329

Latency (ms):
  min:                 2.84
  avg:                 4.49
  max:                 33.04
  95th percentile:    7.84
  sum:                 59911.28

Threads fairness:
  events (avg/stddev): 13329.0000/0.00
  execution time (avg/stddev): 59.9113/0.00
```

## 6. oltp\_read\_write

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_read_write --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_read_write_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_read_write --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_read_write_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                92106
    write:               12808
    other:               26666
    total:              131580
  transactions:         6579 (109.63 per sec.)
  queries:              131580 (2192.70 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0075s
  total number of events: 6579

Latency (ms):
  min:                   7.17
  avg:                   9.11
  max:                   65.14
  95th percentile:      11.87
  sum:                   59953.49

Threads fairness:
  events (avg/stddev):    6579.0000/0.00
  execution time (avg/stddev): 59.9535/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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

```

General statistics:
  total time:                60.0075s
  total number of events:    6579

Latency (ms):
  min:                       7.17
  avg:                       9.11
  max:                       65.14
  95th percentile:          11.87
  sum:                       59953.49

Threads fairness:
  events (avg/stddev):       6579.0000/0.00
  execution time (avg/stddev): 59.9535/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                   92204
    write:                   18154
    other:                   21362
    total:                   131720
  transactions:             6586 (109.76 per sec.)
  queries:                  131720 (2195.12 per sec.)
  ignored errors:           0 (0.00 per sec.)
  reconnects:               0 (0.00 per sec.)

General statistics:
  total time:                60.0050s
  total number of events:    6586

Latency (ms):
  min:                       7.17
  avg:                       9.10
  max:                       43.25
  95th percentile:          11.04
  sum:                       59942.50

Threads fairness:
  events (avg/stddev):       6586.0000/0.00
  execution time (avg/stddev): 59.9425/0.00

```

## 7. oltp\_update\_index

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_update_index --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_index_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_update_index --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_index_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               16491
    other:               8434
    total:              24925
  transactions:        24925 (415.34 per sec.)
  queries:             24925 (415.34 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: 24925

Latency (ms):
  min:                 0.20
  avg:                 2.40
  max:                 36.42
  95th percentile:    3.96
  sum:                 59857.89

Threads fairness:
  events (avg/stddev): 24925.0000/0.00
  execution time (avg/stddev): 59.8579/0.00
```

```
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:               15260
    other:              7605
    total:             22865
  transactions:        22865 (381.04 per sec.)
  queries:            22865 (381.04 per sec.)
  ignored errors:      0      (0.00 per sec.)
  reconnects:         0      (0.00 per sec.)

General statistics:
  total time:          60.0010s
  total number of events: 22865

Latency (ms):
  min:                0.20
  avg:                2.62
  max:               63.93
  95th percentile:   4.57
  sum:              59847.32

Threads fairness:
  events (avg/stddev): 22865.0000/0.00
  execution time (avg/stddev): 59.8473/0.00
```



## 8. oltp\_update\_non\_index

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_update_non_index --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_non_index_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_update_non_index --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_update_non_index_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               12116
    other:               6134
    total:              18250
  transactions:        18250 (304.15 per sec.)
  queries:             18250 (304.15 per sec.)
  ignored errors:       0 (0.00 per sec.)
  reconnects:          0 (0.00 per sec.)

General statistics:
  total time:           60.0021s
  total number of events: 18250

Latency (ms):
  min:                 0.27
  avg:                 3.27
  max:                149.30
  95th percentile:    5.99
  sum:                59626.77

Threads fairness:
  events (avg/stddev): 18250.0000/0.00
  execution time (avg/stddev): 59.6268/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)
```

```
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: 15447
    other: 7798
    total: 23245
  transactions: 23245 (387.31 per sec.)
  queries: 23245 (387.31 per sec.)
  ignored errors: 0 (0.00 per sec.)
  reconnects: 0 (0.00 per sec.)

General statistics:
  total time: 60.0028s
  total number of events: 23245

Latency (ms):
  min: 0.17
  avg: 2.56
  max: 51.74
  95th percentile: 4.33
  sum: 59621.23

Threads fairness:
  events (avg/stddev): 23245.0000/0.00
  execution time (avg/stddev): 59.6212/0.00
```

## 9. oltp\_write\_only

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_write_only --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_write_only_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_write_only --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_write_only_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               35574
    other:               30240
    total:               65814
  transactions:         10969 (182.79 per sec.)
  queries:              65814 (1096.73 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0049s
  total number of events: 10969

Latency (ms):
  min:                   3.52
  avg:                   5.46
  max:                   595.36
  95th percentile:      6.67
  sum:                   59908.63

Threads fairness:
  events (avg/stddev):   10969.0000/0.00
  execution time (avg/stddev): 59.9086/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)
```

```

regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench oltp_write_only --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_write_only_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench oltp_write_only --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/oltp_write_only_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:               35574
    other:               30240
    total:               65814
  transactions:         10969 (182.79 per sec.)
  queries:              65814 (1096.73 per sec.)
  ignored errors:       0 (0.00 per sec.)
  reconnects:           0 (0.00 per sec.)

General statistics:
  total time:           60.0049s
  total number of events: 10969

Latency (ms):
  min:                  3.52
  avg:                   5.46
  max:                  595.36
  95th percentile:     6.67
  sum:                  59908.63

Threads fairness:
  events (avg/stddev):  10969.0000/0.00
  execution time (avg/stddev): 59.9086/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

```

## 10. select\_random\_points

```
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench select_random_points --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_points_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench select_random_points --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_points_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                4578
    write:                0
    other:                0
    total:               4578
  transactions:         4578 (74.77 per sec.)
  queries:              4578 (74.77 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0064s
  total number of events: 4578

Latency (ms):
  min:                   2.69
  avg:                   13.10
  max:                   729.55
  95th percentile:      23.52
  sum:                   59959.94

Threads fairness:
  events (avg/stddev):   4578.0000/0.00
  execution time (avg/stddev): 59.9599/0.00
```

```
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                7251
    write:               0
    other:               0
    total:              7251
  transactions:        7251  (120.83 per sec.)
  queries:             7251  (120.83 per sec.)
  ignored errors:      0      (0.00 per sec.)
  reconnects:          0      (0.00 per sec.)

General statistics:
  total time:           60.0100s
  total number of events: 7251

Latency (ms):
  min:                 2.10
  avg:                 8.27
  max:                872.95
  95th percentile:    11.87
  sum:               59949.81

Threads fairness:
  events (avg/stddev):  7251.0000/0.00
  execution time (avg/stddev): 59.9498/0.00
```

## 11. select\_random\_ranges

```

MINGW64:/c/galera
regin@Mestizo MINGW64 /c/galera
$ docker run --rm --network=$NET --cpus=1 severalnines/sysbench \
  sysbench select_random_ranges --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_ranges_3nodes_1core.txt

docker run --rm --network=$NET --cpus=2 severalnines/sysbench \
  sysbench select_random_ranges --time=60 --threads=1 \
  --mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \
  --mysql-db=testdb --tables=1 --table-size=1000000 run \
  | tee results/select_random_ranges_3nodes_2cores.txt
sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                24512
    write:                0
    other:                0
    total:                24512
  transactions:         24512 (408.52 per sec.)
  queries:               24512 (408.52 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0010s
  total number of events: 24512

Latency (ms):
  min:                   0.69
  avg:                   2.44
  max:                   19.33
  95th percentile:      3.62
  sum:                   59781.19

Threads fairness:
  events (avg/stddev):   24512.0000/0.00
  execution time (avg/stddev): 59.7812/0.00

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

```

```

sysbench 1.0.17 (using bundled LuaJIT 2.1.0-beta2)

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:                23784
    write:               0
    other:               0
    total:               23784
  transactions:         23784 (396.36 per sec.)
  queries:               23784 (396.36 per sec.)
  ignored errors:        0 (0.00 per sec.)
  reconnects:            0 (0.00 per sec.)

General statistics:
  total time:            60.0031s
  total number of events: 23784

Latency (ms):
  min:                   0.61
  avg:                   2.51
  max:                   22.17
  95th percentile:      3.82
  sum:                   59796.76

Threads fairness:
  events (avg/stddev):   23784.0000/0.00
  execution time (avg/stddev): 59.7968/0.00

```

## Comparativa y análisis final (2 vs 3 nodos)

### 1. Lecturas

- point\_select y select\_random\_points/ranges mejoran con 3 nodos, sobre todo con 2 cores (ej.: ~2,662/s → ~4,147/s en point\_select).
- read\_only mostró comportamiento estable con ligeras variaciones (233/s vs 175–159/s en 2 nodos).



Conclusión parcial: En lecturas puras, añadir nodos sí escala el rendimiento (mejores TPS y menor latencia P95).

## 2. Escrituras / Mixtas

- delete / update\_index: tendencia a ↓ TPS al pasar de 2 → 3 nodos (sincronía de commits Galera).
- insert / write\_only / update\_non\_index: resultados mixtos; en varios casos 2 cores mejoran y 3 nodos no siempre dan más TPS.

Conclusión parcial: Para workloads con muchas escrituras, Galera prioriza consistencia y tolerancia a fallos sobre la escalabilidad de throughput. Con 3 nodos se gana alta disponibilidad, pero no necesariamente más TPS.

## 3. Latencias

- Lecturas: latencias promedio bajas ( $\approx 0.2\text{--}3$  ms) con P95 razonable.
- Escrituras: latencias mayores ( $\approx 2\text{--}12$  ms) y peor P95, especialmente en pruebas con más coordinación (read\_write, write\_only).

## Resultados

Todas las corridas fueron de 60 segundos con --threads=1. Para cada prueba, la métrica de referencia es la tasa de eventos/segundo (en bulk\_insert se interpreta como filas/s). A continuación, se describen los valores observados y su interpretación sin tablas, en texto continuo:

- bulk\_insert (3 nodos): se estabilizó en torno a ~89.6 mil filas/seg tras permitir el drop/create automático de Sysbench sobre sbtest1. No se consolidó una corrida equivalente con 2 nodos, por eso la comparación se deja en pendiente; lo importante es que la técnica correcta (tabla limpia) eliminó los errores de duplicados y permitió medir un throughput coherente.
- oltp\_delete: con 2 nodos, el sistema rindió en torno a 1064–1082 eventos/seg (1 y 2 cores). Al pasar a 3 nodos, bajó a ~941 e/s (1 core) y ~818 e/s (2 cores). Esta caída es consistente con el sobrecoste de certificación y replicación síncrona: cada borrado debe validarse y aplicarse en todos los nodos antes de confirmar, elevando la coordinación y reduciendo la tasa

efectiva.

- `oltp_insert`: con 2 nodos se situó en ~288–290 e/s; con 3 nodos subió levemente a ~291 e/s (1 core) y ~314 e/s (2 cores). Hay mejora modesta, sobre todo con 2 cores, lo cual sugiere que con inserciones simples y baja contención la penalización síncrona no es tan severa.
- `oltp_point_select` (lectura pura): aquí aparece la gran ganancia de añadir el tercer nodo. Con 2 nodos, se midieron ~2165 e/s (1c) y ~2662 e/s (2c); con 3 nodos se alcanzaron ~3769 e/s (1c) y ~4147 e/s (2c). La diferencia con 2 cores es +56% (2662 → 4147 e/s). Es el patrón esperado: más réplicas = más capacidad de lectura.
- `oltp_read_only` (lectura predominante): con 2 nodos se observó ~176 e/s (1c) y ~159 e/s (2c); con 3 nodos aumentó a ~233 e/s (1c) y ~222 e/s (2c). Aunque las cifras absolutas son menores que en `point_select`, el porcentaje de mejora con 3 nodos es evidente (+39% aprox. en 2 cores).
- `oltp_read_write` (mixto): con 2 nodos se obtuvieron ~80–82 e/s; con 3 nodos subió a ~110 e/s, en ambos límites de CPU. Esta prueba combina lecturas (que sí escalan) con escrituras (que no escalan), por eso la mejora es moderada pero real.
- `oltp_update_index`: con 2 nodos se observó ~468 e/s (1c) y ~449 e/s (2c); con 3 nodos cayó a ~415 e/s (1c) y ~381 e/s (2c). Los updates sobre índices tienden a activar más certificación/conflictos; el tercer nodo no ayuda a throughput aquí.
- `oltp_update_non_index`: con 2 nodos se midió ~480 e/s (1c) y ~462 e/s (2c). Con 3 nodos el rendimiento fue mixto: ~304 e/s (1c) y ~387 e/s (2c). Puede haber diferencia por contención puntual o calentamiento de buffer pool; aun así, la tendencia general en updates suele ser neutra o a la baja al añadir nodos.
- `oltp_write_only` (escritura pura): con 2 nodos quedó en ~170–171 e/s; con 3 nodos se situó en ~183 e/s (1c) (no quedó consolidada la cifra de 2 cores). En general, write-only no escala mucho; cualquier mejora es pequeña y sensible a detalles del entorno.
- `select_random_points` (lectura aleatoria puntual): con 2 nodos, ~105–117 e/s; con 3 nodos, ~75 e/s (1c) y ~121 e/s (2c). El patrón vuelve a favorecer la

lectura con más réplicas, especialmente cuando liberamos CPU (2 cores).

- `select_random_ranges` (lectura por rangos): con 2 nodos se observaron ~363–373 e/s; con 3 nodos, ~396–409 e/s (de acuerdo a las tomas recientes). Aunque la mejora no es tan grande como en `point_select`, las cifras son iguales o mejores al pasar a 3 nodos.

Punto metodológico importante: todas las pruebas se hicieron con `--threads=1`. Aumentar `--cpus` del contenedor de 1 a 2 no multiplica el rendimiento si la carga está monohilo. Para observar escalamiento por CPU, conviene también probar `--threads=2` (o más), siempre que el profesor lo permita, y/o aumentar el tamaño de tabla para reducir el efecto de caché.

## **Explicación de lo observado (por qué pasa lo que pasa)**

Galera es multi-master síncrono. Cada COMMIT de escritura debe certificarse y confirmarse entre los nodos antes de considerarse final. Esto tiene dos consecuencias directas:

1. Lecturas escalan bien con más nodos, porque hay más réplicas atendiendo consultas y el motor puede servir resultados en latencias muy bajas cuando los datos están calientes en el buffer pool. Por eso `oltp_point_select` y las `select_random_*` mejoran con 3 nodos, con saltos notables en 2 cores.
2. Escrituras no escalan linealmente e incluso pueden empeorar al añadir nodos: hay más hops de red, más certificación y potencialmente más contención (índices, locking interno, conflictos). De ahí la caída en `oltp_delete` y `oltp_update_index` al pasar de 2 a 3 nodos.

CPU vs hilos. Con `--threads=1`, el cuello de botella puede ser monohilo; subir `--cpus` deja margen pero no siempre se aprovecha. Para medir escalado por CPU, hay que subir concurrencia (hilos) además de CPU.

`bulk_insert` requiere tabla limpia o un esquema separado del paquete OLTP. Permitir que Sysbench haga DROP/CREATE antes de insertar evita errores por columnas y duplicados y entrega una métrica consistente (~89.6k filas/seg en 3 nodos).

## Conclusiones

1. Clúster correcto y replicación verificada. Se observaron estados Synced/ready en logs y wsrep\_cluster\_size de 2 y luego 3, con consultas iguales en todos los nodos.
2. Añadir el 3er nodo mejora sustancialmente la lectura. Los point selects y varias lecturas aleatorias crecieron con claridad (p. ej., +56% en oltp\_point\_select a 2 cores).
3. Las escrituras no escalan y pueden bajar, por la replicación síncrona y la certificación Galera. oltp\_delete y oltp\_update\_index fueron los casos más evidentes.
4. Mixtas mejoran moderado. oltp\_read\_write subió ( $\sim 81.5 \rightarrow 109.7$  e/s), porque incorpora una parte de lectura que sí se beneficia de más nodos.
5. Metodología sólida. Mantener 60 s, 1 hilo, y registrar a results/\*.txt deja una línea base reproducible. Si el docente lo permite, probar más hilos evidenciaría mejor el rol de CPU/concurrencia.
6. Recomendación práctica. Para cargas predominantemente de lectura, 3 nodos son una excelente elección (rendimiento y disponibilidad). Para cargas pesadas de escritura, considerar optimizar índices, agrupar operaciones y, si procede, dirigir escrituras a menos nodos o evaluar topologías alternativas.

## Anexo – Comandos clave

### 1. Ver estado del clúster

```
docker exec -it galera-node1 mariadb -uroot -pmi_super_password \  
-e "SHOW STATUS LIKE 'wsrep_cluster_size';"
```

### 2. Preparación OLTP

```
docker run --rm --network=$NET severalnines/sysbench \  
sysbench oltp_common \  
--db-driver=mysql \  
--mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \  
--mysql-db=testdb --tables=1 --table-size=1000000 prepare
```

### 3. Patrones de ejecución

(ajusta oltp\_read\_only por el test que toque y --cpus a 1 o 2)

```
docker run --rm --network=$NET --cpus=1 severalnines/sysbench \  
sysbench oltp_read_only --time=60 --threads=1 \  
--mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \  
--mysql-db=testdb --tables=1 --table-size=1000000 run \  
| tee results/oltp_read_only_2nodes_1core.txt
```

### 4. “Reset” rápido de la tabla sbtest1 (si choca con bulk\_insert)

```
docker run --rm --network=$NET severalnines/sysbench \  
sysbench bulk_insert --time=1 --threads=1 \  
--mysql-host=$HOST --mysql-user=$USER --mysql-password=$PASS \  
--mysql-db=testdb --tables=1 --table-size=1000000 cleanup
```

O dejar que el propio bulk\_insert haga drop/create (como hiciste) y no mezclar su tabla con la de OLTP.

### 5. Re-arrancar clúster limpio (si se cerró Docker)

```
docker compose down --remove-orphans  
docker compose up -d galera-node1 # (bootstrap)  
docker compose up -d galera-node2  
docker compose up -d galera-node3
```

