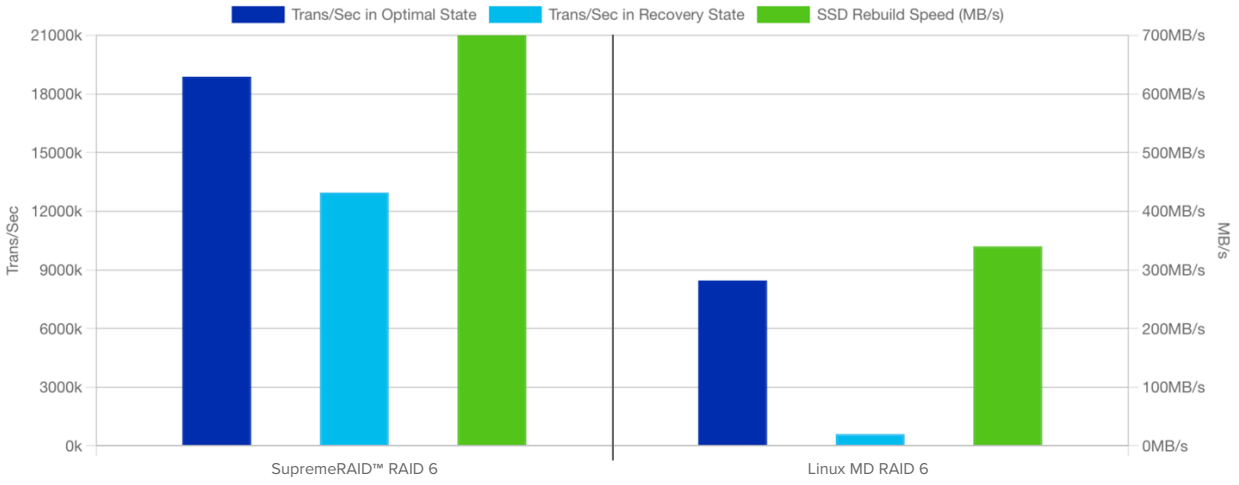
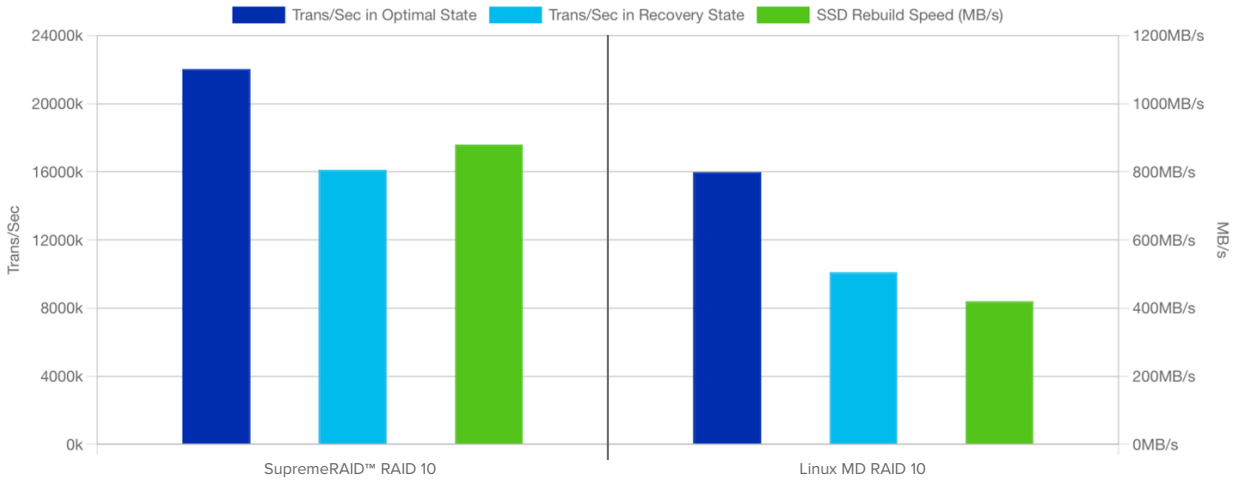
MySQL Performance Testing with RAID

SupremeRAID™ versus Linux MD RAID

Executive Summary

SupremeRAID™ by Graid Technology offers a higher-performance alternative for NVMe SSD data protection of database workloads like MySQL, especially compared to Linux MD RAID. The benefits are significant, with SupremeRAID™ RAID 6 delivering more transactions per second than Linux MD RAID 10 during optimal (non-degraded) and recovery (degraded) states. Also, SupremeRAID™ RAID 6 delivers almost 2x faster rebuild speeds than Linux MD RAID 10. The performance benefits of SupremeRAID™ are even more compelling when comparing similar RAID levels (e.g., RAID 6 with SupremeRAID™ vs. MD RAID).





Copyright © 2023 Graid Technology Inc., All Rights Reserved.

About this Test

In this test, we will deploy a MySQL 8 server on SupremeRAID™ SR-1010 RAID 6, Linux MD RAID 10, and Linux MD RAID 6. We will use sysbench, a popular database benchmark tool, to perform the OLTP Read/Write test and evaluate RAID performance.

Testing Background

Hardware Specifications

- Server: Dell PowerEdge R750 x 1
- Processor: Intel® Xeon® Gold 6338 CPU @ 2.00GHz x 2
- Memory: Samsung M393A4G43BB4-CWE 32GB DDR4 3200Mhz x 16
- SupremeRAID™: SR-1010 SR-BUN-1010-12-FD32 x 1
- SSD: Intel® SSD D7-P5510 SSDPF2KX038TZ 3.84TB x 8

Software Configurations

- OS: Ubuntu 20.04.4 LTS
- Kernel: 5.4.0-131-generic
- SupremeRAID™: Driver Version: 1.3.0-473.gbf5466fc.010
- Linux MD RAID: mdadm version v4.1 - 2018-10-01
- Filesystem: xfs 5.3.0-1ubuntu2
- MySQL version: 8.0.30-0ubuntu0.20.04.2
- Benchmark Tool: sysbench 1.1.0

Hardware Configurations

- MADT Core Enumeration: Linear
- Logical Processor: Enabled
- Device location:
    - Four Intel® SSD D7-P5510 located at CPU0
    - Four Intel® SSD D7-P5510 located at CPU1
    - One SupremeRAID™ SR-1010 located at CPU1

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Benchmark Scenario and MySQL Tuning

- Test workload: Sysbench OLTP_RW uniform
- InnoDB page size: 16K
- Concurrent users: 64, 128, 256, 512, 1024
- Dataset: 8 tables, 50M rows each, 100GB total
- InnoDB Buffer Pool (BP): 32GB (about 32% of data cached in the buffer pool)
- Worker threads:
    - innodb_buffer_pool_instances=48
    - innodb_page_cleaners=48
    - innodb_read_io_threads=32
    - innodb_write_io_threads=16
    - innodb_purge_threads=16
- Testing modes in optimal and rebuilding 1 SSD states:
    - SupremeRAID™ RAID 6 with 8 SSDs and 4k chunks
    - Linux MD RAID 6 with 8 SSDs and 4k chunks
    - Linux MD RAID 6 with 8 SSDs and 16k chunks
    - Linux MD RAID 10 with 8 SSDs and 4k chunks
    - Linux MD RAID 10 with 8 SSDs and 16k chunks

Testing Result

The results show that SR-1010 RAID 6 can provide almost twice the performance of Linux MD RAID 6. SR-1010 RAID 6 performance is competitive with Linux MD RAID 10 while providing more usable capacity and better data security.

Transactions Per Second in Optimal State

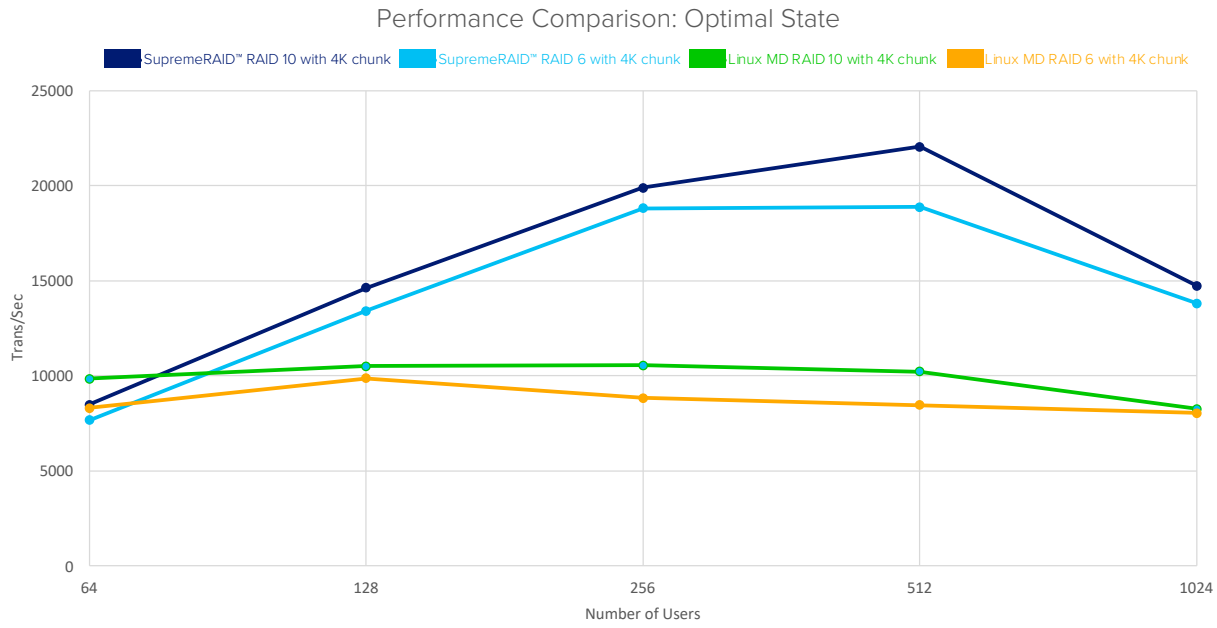
In the lower concurrent users test cases (from 64 to 256), MD RAID 10 performed well because it is not a parity-based RAID. Still, as the number of concurrent users increases, the SQL service consumes CPU resources and competes with MD RAID. This resource contention significantly drops performance to levels lower than SupremeRAID™ RAID 6.

Compared to MD RAID 6, SupremeRAID™ RAID 6 is faster in all cases and can deliver more than twice the performance with high concurrent users.

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

RAID Configuration*	64 Users	128 Users	256 Users	512 Users	1024 Users
SupremeRAID™ RAID 10	8476.39	14627.62	19889.56	22048.90	14729.23
SupremeRAID™ RAID 6	7669.51	13408.26	18813.23	18878.61	13807.17
Linux MD RAID 10	9838.45	10510.42	10556.98	10221.50	8269.07
Linux MD RAID 6	8305.70	9872.39	8826.24	8455.82	8046.76

* RAID with 4K chunks.



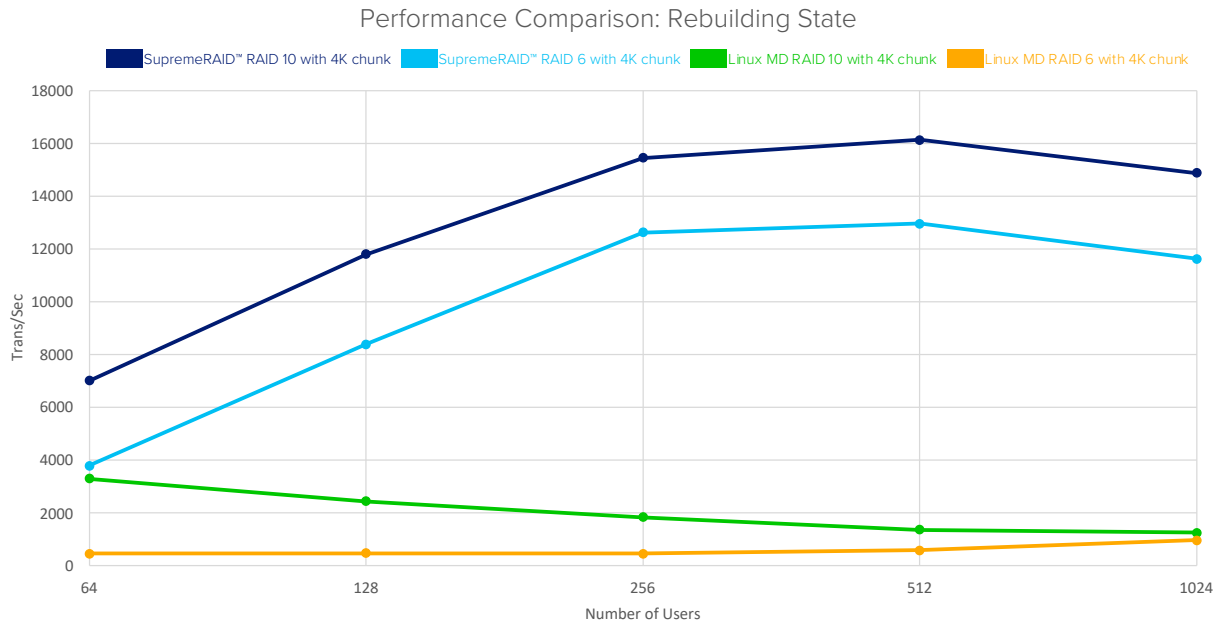
Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Transactions Per Second in Rebuilding State

RAID performance declines during rebuilds because of degraded reading and rebuilding task workloads running in the background. Testing showed performance declined up to 22% for SupremeRAID™ configured for RAID 10 and up to 87% for MD RAID configured for RAID 10. With RAID 6, performance declined up to 50% for SupremeRAID™ and up to 95% for MD RAID. However, with SupremeRAID™ configured for RAID 6 performance remained similar to 9x higher than MD RAID configured for RAID 10 for all test cases.

RAID Configuration*	64 Users	128 Users	256 Users	512 Users	1024 Users
SupremeRAID™ RAID 10	7009.97	11803.37	15445.57	16119.14	14875.35
SupremeRAID™ RAID 6	3793.36	8384.43	12623.33	12953.34	11616.73
Linux MD RAID 10	3308.86	2441.16	1851.95	1370.23	1253.52
Linux MD RAID 6	466.36	484.88	471.77	598.75	973.79

* RAID with 4K chunks.



Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Rebuilding Speed

During rebuilding, although MD RAID 10 is not affected by degraded reads, its performance still dropped due to the rebuilding traffic. As for SupremeRAID™ RAID 10, which benefits from the computing power of GPU, performance dropped less.

SupremeRAID™ RAID 6 simultaneously maintained 12,953 transactions per second and 700Mb/s rebuilding throughput (about 2.5TB per hour) at 512 users. However, MD RAID 6 at 512 users supported just 598 transactions per second and 340Mb/s rebuilding throughout.

The performance of MD RAID 6 severely declines because degraded reading and rebuilding tasks consume many CPU resources to calculate parity.

RAID Configuration*	**Transaction/Sec in Optimal State**	**Transactions/Sec in Recovery State**	**SSD Rebuild Speed (MB/s)**
SupremeRAID™ RAID 10	22048.90	16119.14	880
SupremeRAID™ RAID 6	18878.61	12953.34	700
Linux MD RAID 10	15994.38	10108.93	420
Linux MD RAID 6	8455.82	598.75	340

* RAID with 4K chunks.

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Conclusion

Databases like MySQL benefit from access to the fastest-performing storage possible, so using multiple NVMe SSDs with RAID for data protection is standard. Selecting SupremeRAID™ enables using more effective and efficient RAID 6 for data protection, delivering typically higher performance than Linux MD RAID 10.

Other benefits include:

- Data loss prevention during the simultaneous failure of two SSDs.
- 50% higher usable capacity with 8 SSDs (75% higher with 16 SSDs).
- Up to 85% faster transactions per second in RAID optimal state
- Up to 28% faster transactions per second in RAID recovery state
- Up to 945% faster transactions per second in RAID rebuilding state
- Up to 66% faster rebuild rates with a low impact on performance.

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Testing Flow

SupremeRAID™ RAID 6 with 8 SSDs

1. Compose a RAID 6 group with eight physical drives and create a virtual drive with all available space.

```
$ sudo graidctl create dg raid6 0-7

$ sudo graidctl create vd 0
```

2. Format virtual drive with xfs.

```
$ sudo mkfs.xfs /dev/gvd0n1
```

3. Mount the filesystem and copy SQL data to the mount point.

```
$ sudo mount -o noatime,nodiratime /dev/gvd0n1 /mnt/graid

$ sudo rsync -av /var/lib/mysql /mnt/graid/
```

4. Start MySQL server.

```
$ sudo systemctl start mysql
```

5. Create a database called sbtest.

```
$ mysql -u root -p -e "create database sbtest;"
```

6. Prepare a dataset with eight tables, each containing 50M entries for a total size of 100GB.

```
$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \
  --mysql-storage-engine=InnoDB --tables=8 --table-size=50000000 \
  --mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \
  --mysql-db=sbtest --events=0 --threads=1 create

$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \
  --mysql-storage-engine=InnoDB --tables=8 --table-size=50000000 \
  --mysql-user=root --mysql-password=password --mysql-
  socket=/var/run/mysqld/mysqld.sock \
  --mysql-db=sbtest --events=0 --threads=32 prepare
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

7. Launch a 1-hour warm-up task with 512 threads to make SSDs enter a steady state.

```
$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \
  --tables=8 --table-size=50000000 --threads=256 --time=3600 \
  --thread-init-timeout=0 --rate=0 --rand-type=uniform --rand-seed=0  \
  --mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \
  --mysql-db=sbtest --events=0 run
```

8. Launch a 10-minute OLTP_RW uniform test with the following threads: 64, 128, 256, 512, 1024.

```
$ for threads in 64 128 256 512 1024
  do
      ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \
      --tables=8 --table-size=50000000 --threads=${threads} --time=600 \
      --thread-init-timeout=0 --rate=0 --rand-type=uniform --rand-seed=0  \
      --mysql-user=root --mysql-password=password --mysql-
     socket=/var/run/mysqld/mysqld.sock \
      --mysql-db=sbtest --events=0 run
      sleep 15
  done
```

9. Mark one physical drive offline to make RAID degraded.

```
$ sudo graidctl edit pd 0 marker offline
```

10. Mark the offline physical drive online to enter the rebuilding process.

```
$ sudo graidctl edit pd 0 marker online
```

11. Launch OLTP_RW uniform.

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Linux MD RAID 6 with 8 SSDs

1. Since Intel® D7-P5510 supports deterministic read zero after TRIM, use the discard command to rest all SSDs and skip the MD RAID initialization process.

```
$ for i in {0..7}; do sudo blkdiscard /dev/nvme"$i"n1; done
```

2. Compose a RAID 6 group with 8 physical drives and the chunk size set to 4KB to improve 16k random write performance.

```
$ sudo mdadm --create --assume-clean --verbose /dev/md6 --level=6 --raid-devices=8 --chunk=4K /dev/nvme[0-7]n1
```

3. Increase MD parity worker threads to improve overall write performance and increase the speed limit to get better rebuild speed.

```
$ echo 8 | sudo tee /sys/block/md6/md/group_thread_cnt
$ sysctl -w dev.raid.speed_limit_min=600000
$ sysctl -w dev.raid.speed_limit_max=600000
```

4. Format virtual drive with xfs.

```
$ sudo mkfs.xfs /dev/md6
```

5. Mount the filesystem and copy SQL data to the mount point.

```
$ sudo mount -o noatime,nodiratime /dev/md6 /mnt/graid
$ sudo rsync -av /var/lib/mysql /mnt/graid/
```

6. Start MySQL server.

```
$ sudo systemctl start mysql
```

7. Create a database called **sbtest.**

```
$ mysql -u root -p -e "create database sbtest;"
```

8. Prepare a dataset with eight tables, each containing 50M entries for a total size of 100GB.

```
$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \
  --mysql-storage-engine=InnoDB --tables=8 --table-size=50000000 \
  --mysql-user=root --mysql-password=password --mysql-socket=/var/run/mysqld/mysqld.sock \
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

```
  --mysql-db=sbtest --events=0 --threads=1 create

$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

  --mysql-storage-engine=InnoDB --tables=8 --table-size=50000000 \

  --mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \

  --mysql-db=sbtest --events=0 --threads=32 prepare
```

9. Launch a 1-hour warm-up task with 512 threads to make SSDs enter a steady state.

```
$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

--tables=8 --table-size=50000000 --threads=256 --time=3600 \

--thread-init-timeout=0 --rate=0 --rand-type=uniform --rand-seed=0  \

--mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \

--mysql-db=sbtest --events=0 run
```

10. Launch a 10-minute test with the following threads: 64, 128, 256, 512, 1024.

```
$ for threads in 64 128 256 512 1024

  do

      ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

      --tables=8 --table-size=50000000 --threads=${threads} --time=600 \

      --thread-init-timeout=0 --rate=0 --rand-type=uniform --rand-seed=0  \

      --mysql-user=root --mysql-password=password --mysql-
     socket=/var/run/mysqld/mysqld.sock \

      --mysql-db=sbtest --events=0 run

      sleep 15

  done
```

11. Mark one SSD offline to make RAID degraded.

```
$ sudo mdadm --manage --set-faulty /dev/md6 /dev/nvme0n1

$ sudo mdadm --manage /dev/md6 -r /dev/nvme0n1

$ sudo mdadm --zero-superblock /dev/nvme0n1

Add removed SSD back to enter the rebuilding process.
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

```
$ sudo mdadm --manage /dev/md6 -a /dev/nvme0n1

Launch OLTP_RW uniform again
```

Linux MD RAID 10 with 8 SSDs

1. Since Intel® D7-P5510 supports deterministic read zero after TRIM, use the discard command to rest all SSDs and skip the MD RAID initialization process.

```
$ for i in {0..7}; do sudo blkdiscard /dev/nvme"$i"n1; done
```

2. Compose a RAID 6 group with 8 physical drives.

```
$ sudo mdadm --create --assume-clean --verbose /dev/md10 --level=10 --raid-devices=8 --chunk=16K /dev/nvme[0-7]n1
```

3. Increase the speed limit to get better rebuild speed.

```
$ sysctl -w dev.raid.speed_limit_min=600000

$ sysctl -w dev.raid.speed_limit_max=600000
```

4. Format virtual drive with xfs.

```
$ sudo mkfs.xfs /dev/md10
```

5. Mount the filesystem and copy SQL data to the mount point.

```
$ sudo mount -o noatime,nodiratime /dev/md10 /mnt/graid

$ sudo rsync -av /var/lib/mysql /mnt/graid/
```

6. Start MySQL server.

```
$ sudo systemctl start mysql
```

7. Create a database called sbtest.

```
$ mysql -u root -p -e "create database sbtest;"
```

8. Prepare a dataset with eight tables, each containing 50M entries for a total size of 100GB.

```
$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

  --mysql-storage-engine=InnoDB --tables=8 --table-size=50000000 \

  --mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

```
  --mysql-db=sbtest --events=0 --threads=1 create

$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

  --mysql-storage-engine=InnoDB --tables=8 --table-size=50000000 \

  --mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \

  --mysql-db=sbtest --events=0 --threads=32 prepare
```

9. Launch a 1-hour warm-up task with 512 threads to make SSDs enter a steady state.

```
$ ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

--tables=8 --table-size=50000000 --threads=256 --time=3600 \

--thread-init-timeout=0 --rate=0 --rand-type=uniform --rand-seed=0  \

--mysql-user=root --mysql-password=password --mysql-
 socket=/var/run/mysqld/mysqld.sock \

--mysql-db=sbtest --events=0 run
```

10. Launch a 10-minute test with the following threads: 64, 128, 256, 512, 1024.

```
$ for threads in 64 128 256 512 1024

  do

      ./sysbench-1.1-new2020 lua/OLTP_RW-trx.lua --db-driver=mysql \

      --tables=8 --table-size=50000000 --threads=${threads} --time=600 \

      --thread-init-timeout=0 --rate=0 --rand-type=uniform --rand-seed=0  \

      --mysql-user=root --mysql-password=password --mysql-
     socket=/var/run/mysqld/mysqld.sock \

      --mysql-db=sbtest --events=0 run

      sleep 15

  done
```

11. Mark one SSD offline to make RAID degraded.

```
$ sudo mdadm --manage --set-faulty /dev/md10 /dev/nvme0n1

$ sudo mdadm --manage /dev/md10 -r /dev/nvme0n1

$ sudo mdadm --zero-superblock /dev/nvme0n1
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

12. Add the removed SSD back to enter the rebuilding process.

```
$ sudo mdadm --manage /dev/md10 -a /dev/nvme0n1
```

13. Launch OLTP_RW uniform.

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

Appendix

MySQL Server Configuration

```
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
 user=mysql
# pid-file=/var/run/mysqld/mysqld.pid
 socket=/mnt/graid/mysql/mysqld.sock
 port= 3306
 datadir=/mnt/graid/mysql

# general
 max_connections=4000
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

```
back_log=4000
ssl=0
table_open_cache=8000
table_open_cache_instances=16
default_authentication_plugin=mysql_native_password
default_password_lifetime=0
max_prepared_stmt_count=512000
skip_log_bin=1
character_set_server=latin1
collation_server=latin1_swedish_ci
skip-character-set-client-handshake
transaction_isolation=REPEATABLE-READ

# files
innodb_file_per_table
innodb_log_file_size=1024M
innodb_log_files_in_group=16
innodb_open_files=4000

# buffers
innodb_buffer_pool_size=32000M
innodb_buffer_pool_instances=48
innodb_log_buffer_size=64M
innodb_numa_interleave=on

# tune
innodb_doublewrite=1
innodb_thread_concurrency=0
innodb_flush_log_at_trx_commit=1
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

```
innodb_max_dirty_pages_pct=90
innodb_max_dirty_pages_pct_lwm=10

join_buffer_size=32K
sort_buffer_size=32K
innodb_use_native_aio=1
innodb_stats_persistent=1
innodb_spin_wait_delay=6

innodb_max_purge_lag_delay=300000
innodb_max_purge_lag=0
innodb_flush_method=O_DIRECT
innodb_checksum_algorithm=crc32
innodb_io_capacity=20000
innodb_io_capacity_max=40000
innodb_lru_scan_depth=1000
innodb_change_buffering=none
innodb_read_only=0
innodb_page_cleaners=48
innodb_undo_log_truncate=off

# perf special
innodb_adaptive_flushing=1
innodb_flush_neighbors=0
innodb_read_io_threads=32
innodb_write_io_threads=16
innodb_purge_threads=16
innodb_adaptive_hash_index=0
```

Copyright © 2023 Graid Technology Inc., All Rights Reserved.

```
# monitoring
 innodb_monitor_enable='%'
 performance_schema=ON


# etc.
 loose_log_error_verbosity=3
 secure_file_priv=
 core_file
 innodb_buffer_pool_in_core_file=off
```

Reference

MySQL Performance: Benchmark kit (BMK-kit)

About Graid Technology

Graid Technology, creators of SupremeRAID™ next-generation GPU-based RAID, is led by a team of experts in the storage industry and is headquartered in Silicon Valley, California with an R&D center in Taipei, Taiwan. Designed for performance-demanding workloads, SupremeRAID™ is the fastest NVMe and NVMeoF RAID solution for PCIe Gen 3, 4, and 5 servers. A single SupremeRAID™ card delivers up to 19M IOPS and 110GB/s and supports up to 32 native NVMe drives, delivering superior NVMe/NVMeoF performance while increasing scalability, improving flexibility, and lowering TCO. For more information on Graid Technology, visit graidtech.com or connect with us on Twitter or LinkedIn.

Copyright © 2023 Graid Technology Inc., All Rights Reserved.