

Myrocks: What is it ... and why to use it!

Carlos Tutte
Technical Account Manager
Percona University Montevideo

PERCONA

© 2026 Percona

Agenda

De que es esta charla?

- Introduccion a MyRocks (Storage Engine)
- Que es un LSM?
- Pros/cons de MyRocks
- Comparativa vs InnoDB
- Cuando utilizar MyRocks?

Que NO es esta charla?

- Internals de Myrocks
- Como tunear Myrocks
- Una guia de migracion

PERCONA®

© 2026 Percona

MyRocks engine



PERCONA®

© 2026 Percona

Que es MyRocks?

- MySQL storage engine, basado en RocksDB
  - Almacenamiento Key-Value diseñado para un alto rendimiento.
- Desarrollado originalmente y publicado como código abierto por Facebook.
  - Percona lo adapta para Percona Server
- Diseñado para optimizar:
  - Utilización del espacio (menor ocupación del disco)
  - Eficiencia de escritura (menor amplificación de escritura -> mejor rendimiento de escritura y mayor vida útil de la unidad SSD)
  - Costes de almacenamiento de bases de datos (menos espacio + mayor vida útil = menor coste total de propiedad)
- Storage engine basado en LSM
  - Qué es esto?

PERCONA

© 2026 Percona



PERCONA®

© 2026 Percona

Log-Structured Merge-Tree (LSM-Tree)

- Hay 3 puntos clave del artículo* de 1996 que introduce los árboles LSM:
  - 1) “El árbol LSM utiliza un algoritmo que aplaza y agrupa los cambios de índice, propagando los cambios desde un componente basado en memoria a través de uno o más componentes de disco de manera eficiente, similar a la ordenación por fusión” = **Escritura optimizada**

* https://link.springer.com/article/10.1007/s002360050048

Patrick O’Neil, Edward Cheng, Dieter Gawlick & Elizabeth O’Neil

PERCONA

© 2026 Percona

Log-Structured Merge-Tree (LSM-Tree)

- Hay 3 puntos clave del artículo* de 1996 que introduce los árboles LSM:
  - 2) “El algoritmo ha reducido considerablemente los movimientos del brazo del disco en comparación con los métodos de acceso tradicionales, como los árboles B, y mejorará la relación costo-rendimiento en ámbitos donde los costos del brazo del disco para las inserciones con métodos de acceso tradicionales superan con creces los costos de los medios de almacenamiento” = **Menor costo de almacenamiento**

* https://link.springer.com/article/10.1007/s002360050048

Patrick O’Neil, Edward Cheng, Dieter Gawlick & Elizabeth O’Neil

PERCONA

© 2026 Percona

Log-Structured Merge-Tree (LSM-Tree)

- Hay 3 puntos clave del artículo* de 1996 que introduce los árboles LSM:
  - 3) Sin embargo, en algunos casos, las búsquedas indexadas que requieren una respuesta inmediata perderán eficiencia de E/S, por lo que el árbol LSM es más útil en aplicaciones donde las inserciones de índice son más comunes que las búsquedas que recuperan las entradas. = **Sobrecarga de lectura**

* https://link.springer.com/article/10.1007/s002360050048

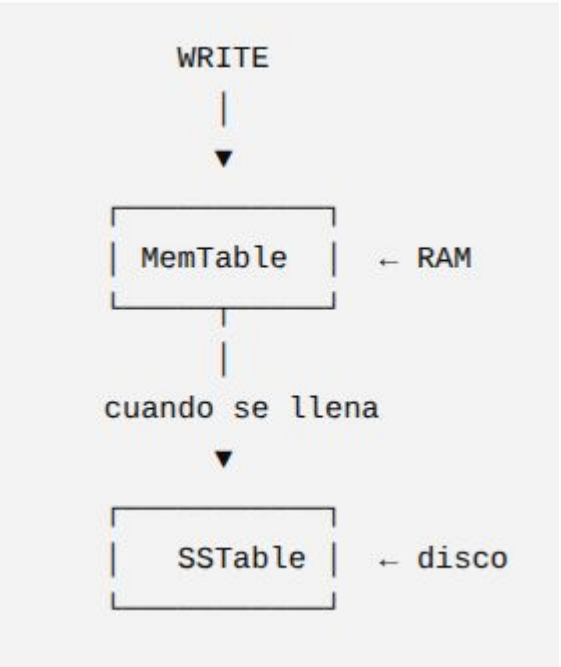
Patrick O'Neil, Edward Cheng, Dieter Gawlick & Elizabeth O'Neil

PERCONA

© 2026 Percona

Ejemplo LSM (1 de 2)

- Las nuevas escrituras primero van a una estructura en memoria llamada MemTable.
  - Cuando la MemTable alcanza cierto tamaño:
    - Se convierte en un archivo en disco, una SSTable.
    - La MemTable nueva sigue recibiendo escrituras.
    - Con el tiempo se generan muchas SSTables.
    - Eventualmente el sistema hace compaction para combinarlas



PERCONA

© 2026 Percona

Ejemplo LSM (2 de 2)

- Existen 3 SSTables con distintos valores
- Originalmente Key=2 tenia Value=B en SSTable1
  - Eventualmente Key=2 se actualiza a Value=X en SSTable2
- Al compactar las 3 tablas se crea una nueva SStable con la última versión de cada Key

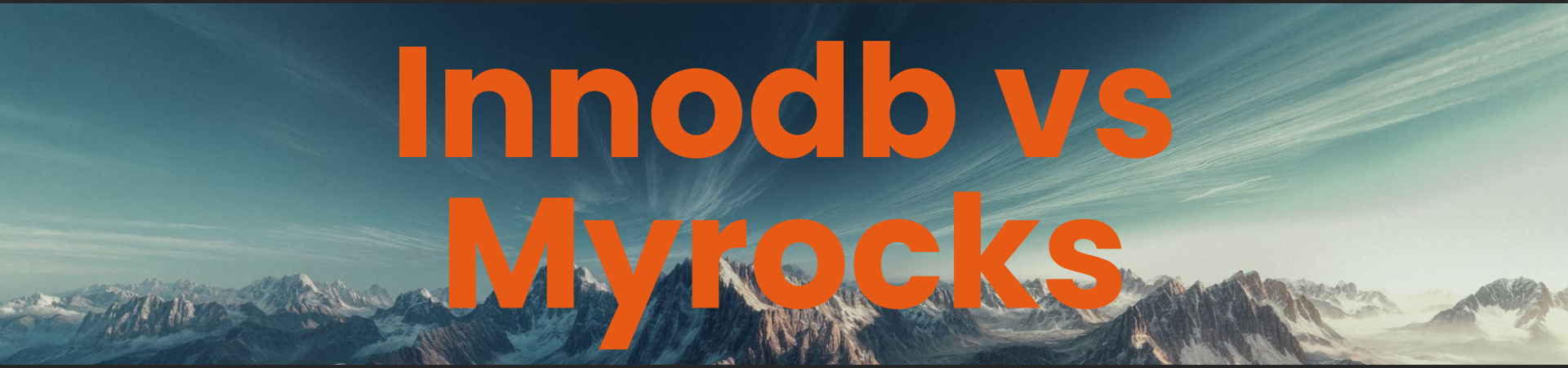
SSTable 1:	SSTable 2:	SSTable 3:
1 → A	2 → X	3 → C
2 → B	4 → D	5 → E
3 → C	6 → F	

SSTable nueva:

1 → A
2 → X
3 → C
4 → D
5 → E
6 → F

PERCONA

© 2026 Percona



Innodb vs Myrocks

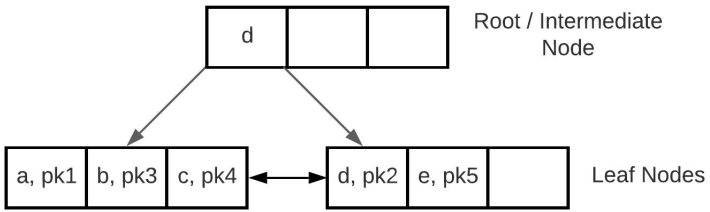
PERCONA®

© 2026 Percona

Motor basado en B+Tree vs LSM-Tree

Motor B+Tree tradicional (InnoDB)

- Optimizado para lecturas pequeñas y aleatorias.
- Las tablas son anchas y poco profundas -> se requieren menos recorridos para encontrar la mayoría de los valores.
- Las operaciones de escritura pueden ser costosas con el rebalanceo del árbol (E/S aleatoria).
- Clustered Indexing



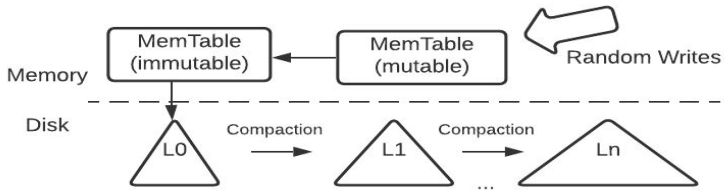
PERCONA

© 2026 Percona

B+Tree Engine vs LSM-Tree Based Engine

Motor de LSM-Tree (RocksDB)

- Optimizado para el rendimiento de escritura
- Los archivos SST intermedios se fusionan y compactan en diferentes niveles (de forma similar a MergeSort).
- Convierte escrituras aleatorias en escrituras secuenciales de solo adición a través de MemTables.



PERCONA

© 2026 Percona

Entonces debemos migrar de InnoDB a Myrocks?



© 2026 Percona

Entonces debemos migrar de InnoDB a Myrocks?

- En una palabra … NO!
- Como todo, tiene sus ventajas y desventajas.
- Depende mucho de la carga de trabajo y del caso de uso.

… Veamos algunas ventajas y desventajas de MyRocks.

PERCONA®

© 2026 Percona

MyRocks Pros

Pros:

- Compresión LZ4 predeterminada + archivos SST llenos al 100% -> menor espacio en disco
- Escritura optimizada
  - El rendimiento de escritura se mantiene relativamente estable cuando los datos son mayores que la memoria
- Permite uso de discos más baratos (lentos) para cargas de trabajo con uso intensivo de escritura.
- Disponible en Percona Server 8.4 (y en 5.7 / 8.0 también)

PERCONA®

© 2026 Percona

MyRocks Cons

Cons:

- SBR (Statement based replication) no soportada
- No Gap Locking
- No Foreign Key support
  - Algunos podrían pensar esto como una ventaja …
- Rendimiento de lectura generalmente más lento (vs InnoDB)
- ORDER BY ASC o DESC tienen características de rendimiento diferentes
  - Generalmente una es rápida, la otra es lenta
- Los escaneos de rango/tabla son notablemente más lentos.
- Transportable Tablespace, Spatial Index, and Fulltext Index no soportados

PERCONA

© 2026 Percona



Performance
InnoDB vs MyRocks

PERCONA®

© 2026 Percona

Veamos algunos ejemplos comparativos

Los siguientes gráficos fueron extraídos del siguiente benchmark realizado en Abril 2026: https://www.percona.com/blog/benchmarking-myrocks-vs-innodb-in-memory-constrained-environments/

- Data set size:
  - InnoDB: 161 GB
  - MyRocks (with LZ4 compression): 125 GB

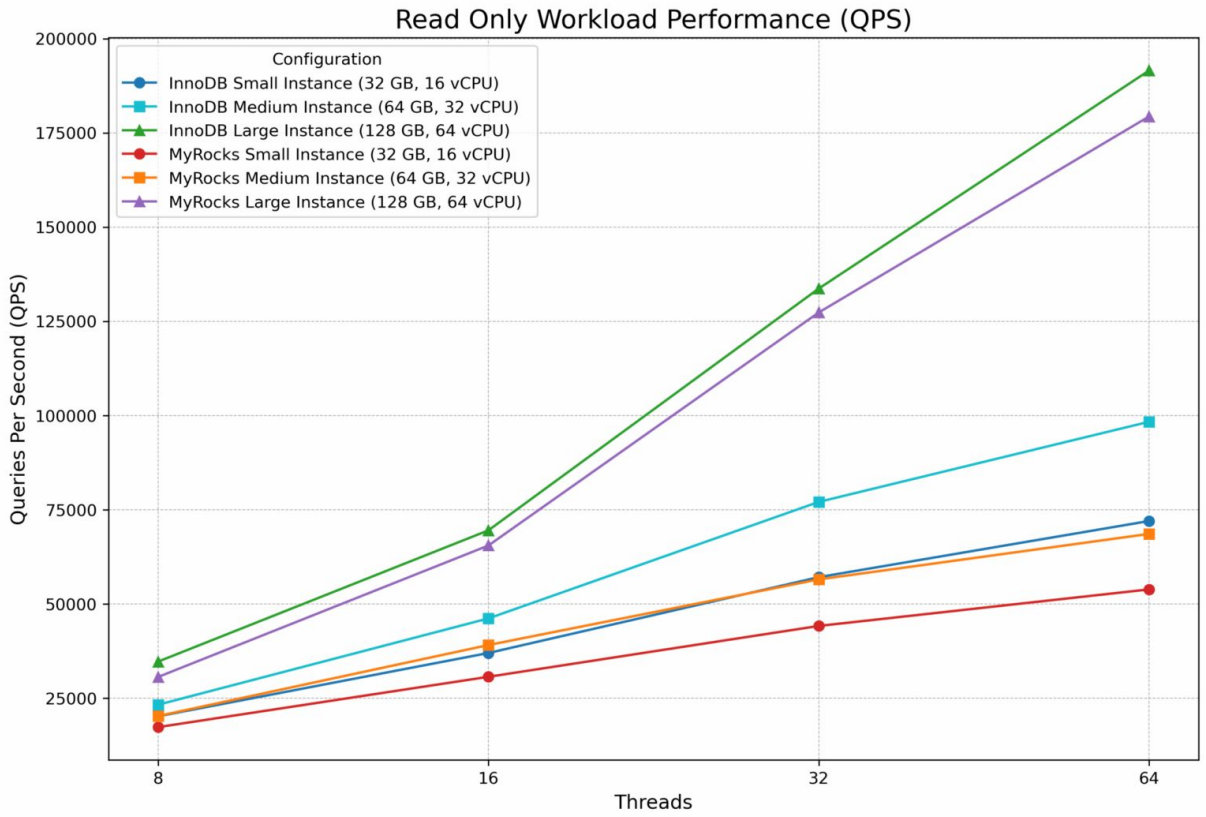
- 12 escenarios
  - 3 tipos de instancias {small, medium , large}
  - 4 niveles de concurrencia {8, 16, 32, 64 threads}

- Cierto tuneo y config para ambos engines
  - Mas info en el blog

PERCONA

© 2026 Percona

Round 1: Read Only Workload



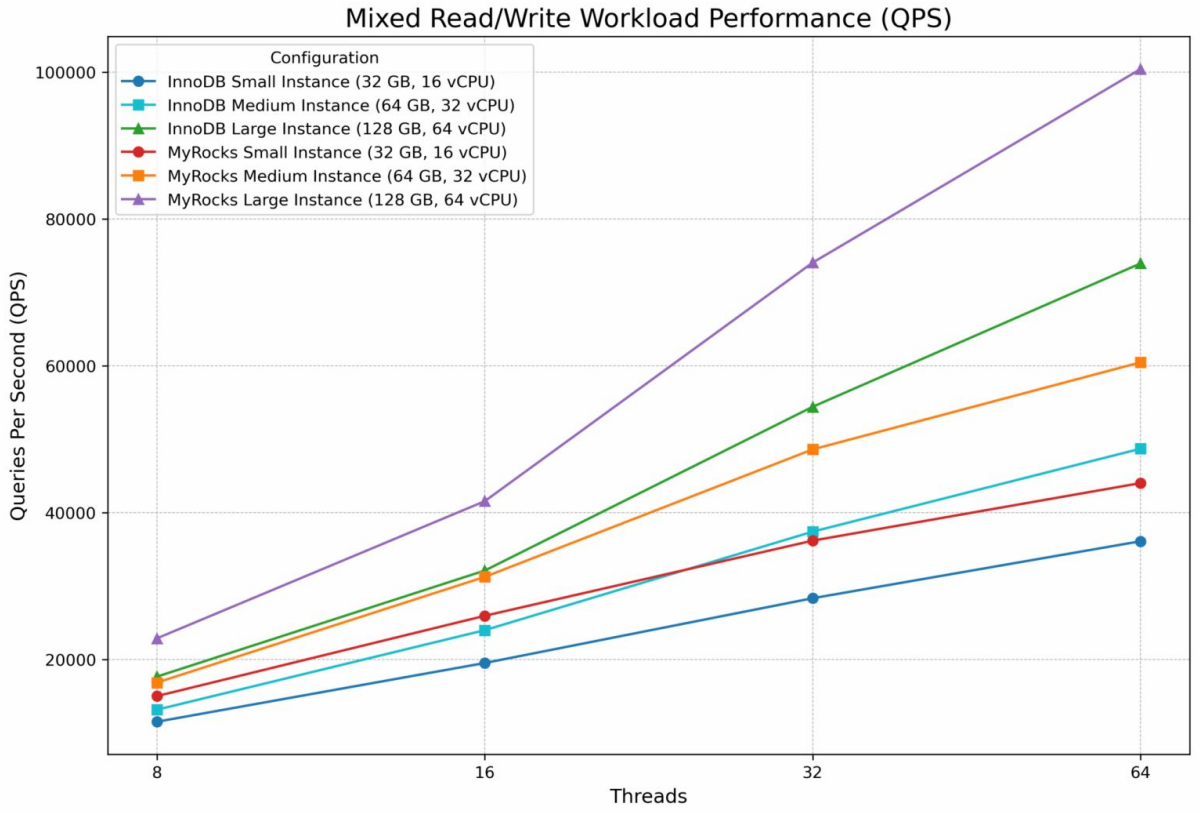
PERCONA

© 2026 Percona

Configuration for Read Only Workload	8 thds	16 thds	32 thds	64 thds
InnoDB Small Instance (32 GB, 16 vCPU)	20,157	36,956	57,088	71,989
InnoDB Medium Instance (64 GB, 32 vCPU)	23,229	46,134	77,052	98,325
InnoDB Large Instance (128 GB, 64 vCPU)	34,670	69,508	133,670	191,537
MyRocks Small Instance (32 GB, 16 vCPU)	17,285	30,644	44,141	53,836
MyRocks Medium Instance (64 GB, 32 vCPU)	20,228	39,069	56,491	68,572
MyRocks Large Instance (128 GB, 64 vCPU)	30,595	65,481	127,327	179,350

© 2026 Percona

Round 2: Mixed Read/Write Workload



PERCONA

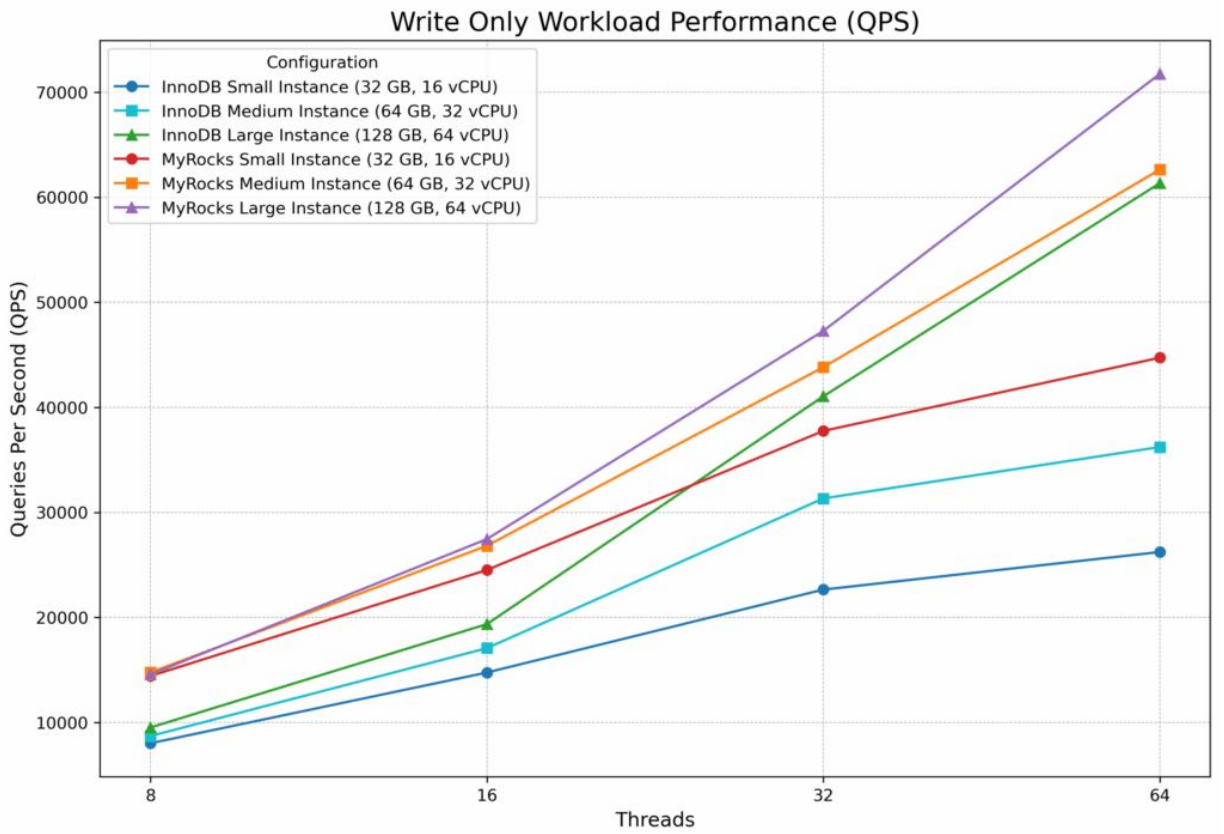
© 2026 Percona

Configuration for Mixed Read/Write	8 thds	16 thds	32 thds	64 thds
InnoDB Small Instance (32 GB, 16 vCPU)	11,521	19,520	28,346	36,110
InnoDB Medium Instance (64 GB, 32 vCPU)	13,156	23,991	37,414	48,714
InnoDB Large Instance (128 GB, 64 vCPU)	17,664	32,113	54,424	73,957
MyRocks Small Instance (32 GB, 16 vCPU)	15,008	25,950	36,192	44,031
MyRocks Medium Instance (64 GB, 32 vCPU)	16,848	31,251	48,617	60,480
MyRocks Large Instance (128 GB, 64 vCPU)	22,878	41,565	74,066	100,436

PERCONA®

© 2026 Percona

Round 3: Write Only Workload



PERCONA

© 2026 Percona

Configuration for Write Only Workload	8 thds	16 thds	32 thds	64 thds
InnoDB Small Instance (32 GB, 16 vCPU)	8,019	14,752	22,649	26,225
InnoDB Medium Instance (64 GB, 32 vCPU)	8,704	17,072	31,326	36,219
InnoDB Large Instance (128 GB, 64 vCPU)	9,520	19,358	41,045	61,334
MyRocks Small Instance (32 GB, 16 vCPU)	14,421	24,515	37,751	44,721
MyRocks Medium Instance (64 GB, 32 vCPU)	14,760	26,814	43,819	62,622
MyRocks Large Instance (128 GB, 64 vCPU)	14,572	27,455	47,276	71,736

PERCONA

© 2026 Percona

Algunos números

- Round 1: InnoDB gana
  - Small instances: 17-33%
  - Medium instances: 15-43%
  - Large instances 6-13%
    - Mas memoria da ventaja a InnoDB
- Round 2: MyRocks gana
  - Small and medium instances: 28-33%
  - Large instances 29-36%
- Round 3: MyRocks gana
  - Small instances: 67-80%
  - Medium instances: 40-73%
  - Large instances 15-53%
    - Mas memoria da ventaja a InnoDB

PERCONA®

© 2026 Percona

Advertencias



PERCONA®

© 2026 Percona

Cuando utilizar MyRocks?

- Datasets muy grandes
  - Working dataset mas grande que la memoria
- Cargas de trabajo intensivas en escritura
  - I/O bound, not CPU bound
- Storage hardware limitado
  - O queriendo bajar los costos de storage
- Range scan queries limitadas
  - Read queries deberían ser por tuplas puntuales
  - No es ideal para generar informes de consultas.

PERCONA®

© 2026 Percona

Cuando utilizar MyRocks?

- Cuidado con ONLINE DDL
  - Herramientas como pt-online-schema-change o gh-ost pueden ayudar.
  - INSTANT desde PS 8.4.5
- SBR no soportado
  - Herramientas como pt-table-checksum requiere SBR
- Que no sea necesario:
  - FKs, Fulltext Indexes, Unique Keys (soportadas, pero fuertemente penalizadas), o Spatial Indexes

PERCONA®

© 2026 Percona

Si a todo!

- Es posible tener algunas tablas en InnoDB y otras en Myrocks
- Es posible que una misma tabla este en InnoDB en primary y Myrocks en replica
  - Tables equivales, soportadas y RBR
- A Myrocks tambien hay que tunearlo!
  - Algunos parametros a ajustar
  - No usar toda la memoria para InnoDB Buffer Pool!
- Myrocks activamente soportado y desarrollado por Percona

PERCONA®

© 2026 Percona



Conclusiones

PERCONA®

© 2026 Percona

Conclusiones

- InnoDB es buena para proposito general pero Myrocks para un caso específico
  - Ideal para instancias grandes que sean escritura-intensivas
- Myrocks es muy eficiente para escrituras
  - Pero sufre para lecturas (sequential scans)
- Antes de migrar a Myrocks
  - Revisar las restricciones
  - Hacer benchmark para el tipo de carga esperado
    - No olvidar tunear los parámetros de Myrocks!
-

PERCONA®

© 2026 Percona



PERCONA®

© 2026 Percona



© 2026 Percona