



# **MySQL**

## Enterprise applications

NetApp  
January 02, 2026

# 目次

MySQL .....	1
概要 .....	1
データベース設定 .....	1
ファイル構造 .....	1
設定パラメータ .....	4
innodb_log_file_size .....	5
innodb_flush_log_at_trx_commit .....	5
innodb_doublewrite .....	6
innodb_buffer_pool_size .....	6
innodb_flush_method .....	6
innodb_io_capacity .....	7
innodb_lru_scan_depth .....	8
open_file_limits .....	8
ホストの設定 .....	8
コンテナ化 .....	8
NFSv3スロットテーブル .....	9
I/Oスケジューラ .....	9
ファイル記述子 .....	10
ストレージ構成 .....	10
NFS .....	10
SAN .....	11

# MySQL

## 概要

MariaDBやPercona MySQLを含むMySQLとその変種は、世界で最も人気のあるデータベースです。



ONTAPとMySQLデータベースに関するこのドキュメントは、以前に公開されていた [\\_TR-4722](#) : 『MySQL database on ONTAP best practices』に代わるものです。\_

ONTAPは文字通りデータベース向けに設計されているため、ONTAPはMySQLデータベースに最適なプラットフォームです。データベースワークロードのニーズに対応するために、高度なQuality of Service (QoS ; サービス品質) 機能や基本的なFlexClone機能に対するランダムI/Oレイテンシの最適化など、多数の機能が特別に開発されました。

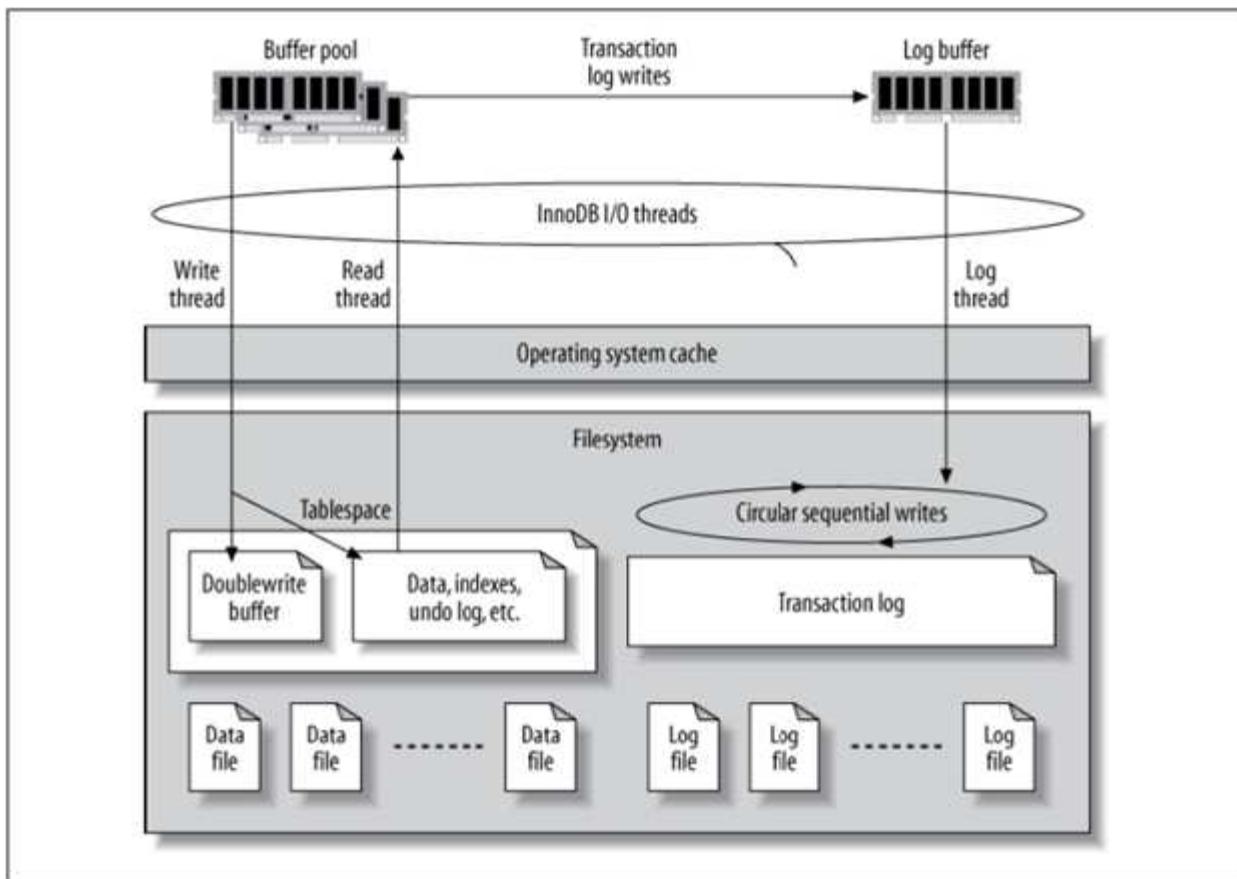
無停止アップグレード（ストレージの交換など）などの追加機能により、重要なデータベースの可用性を維持できます。また、MetroClusterを使用して大規模な環境で瞬時にディザスタリカバリを実行したり、SnapMirrorアクティブ同期を使用してデータベースを選択したりすることもできます。

最も重要なことは、ONTAPが卓越したパフォーマンスを提供し、お客様固有のニーズに合わせて解決策をサイジングできることです。ネットアップのハイエンドシステムは100万超のIOPSをマイクロ秒単位のレイテンシで提供できますが、必要なIOPSが10万だけの場合は、同じストレージオペレーティングシステムを実行する小型のコントローラでストレージ解決策を適切にサイジングできます。

## データベース設定

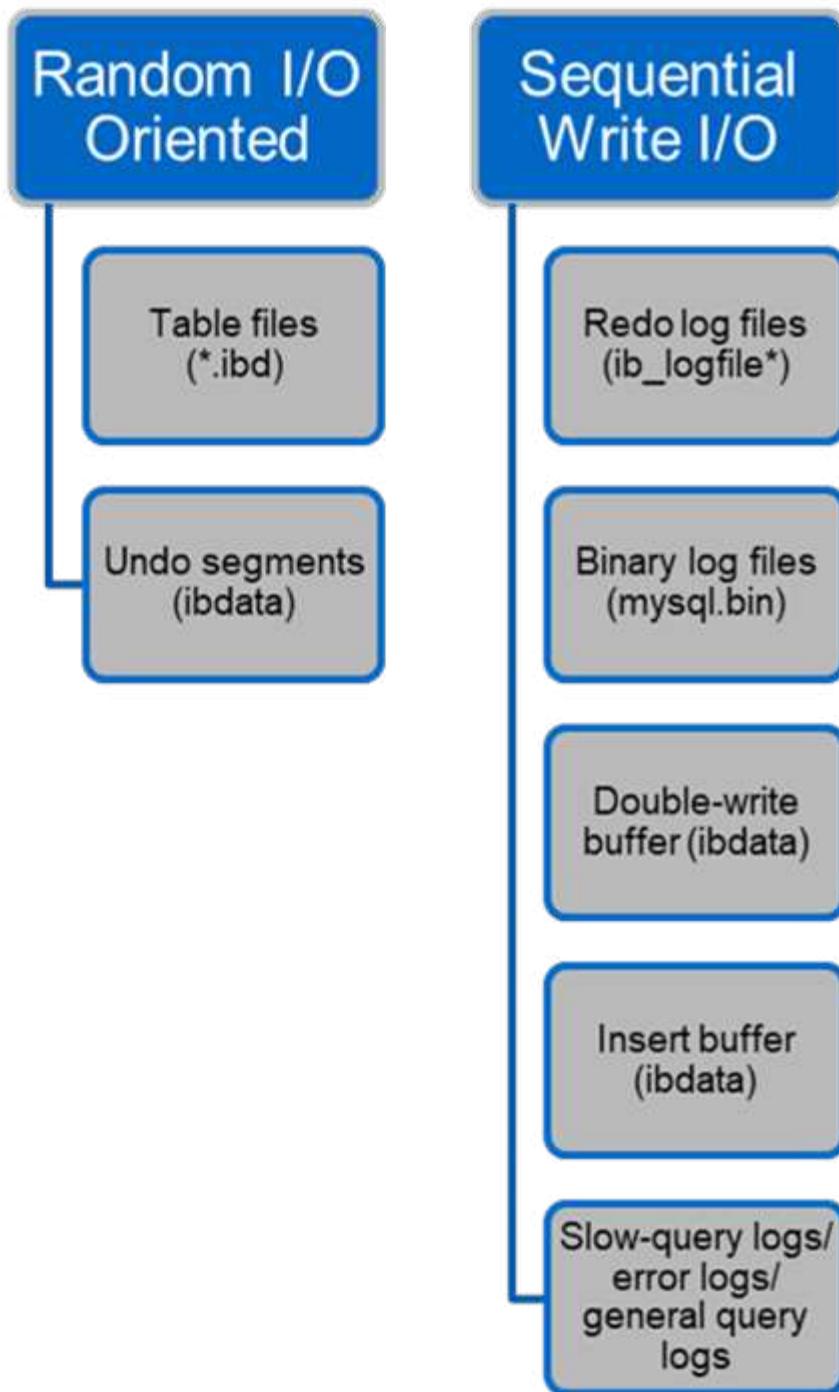
### ファイル構造

InnoDBはストレージとMySQLサーバの中間層として機能し、データをドライブに格納します。



MySQL I/Oは次の2つのタイプに分類されます。

- ・ランダムファイルI/O
- ・シーケンシャルファイルI/O



データファイルはランダムに読み取りおよび上書きされるため、IOPSが高くなります。そのため、SSDストレージを推奨します。

REDOログファイルとバイナリログファイルはトランザクションログです。それらはシーケンシャルに書き込まれるので、書き込みキャッシュを備えたHDDで優れたパフォーマンスを得ることができます。リカバリ時にシーケンシャルリードが発生しますが、ログファイルのサイズは通常データファイルより小さく、シーケンシャルリードはランダムリード（データファイルで発生）よりも高速であるため、パフォーマンスの問題が発生することはほとんどありません。

ダブル書き込みバッファはInnoDBの特別な機能です。InnoDBは、最初にフラッシュされたページをダブルライトバッファに書き込み、次にページをデータファイル上の正しい位置に書き込みます。このプロセスによ

り、ページの破損が防止されます。二重書き込みバッファがない場合、ドライブへの書き込みプロセス中に電源障害が発生すると、ページが破損する可能性があります。ダブル書き込みバッファへの書き込みはシーケンシャルであるため、HDD向けに高度に最適化されています。リカバリ時にシーケンシャルリードが発生します。

ONTAP NVRAMはすでに書き込み保護を提供しているため、ダブル書き込みバッファは必要ありません。MySQLにはパラメータがあります。skip\_innodb\_doublewrite、ダブルライトバッファをディザイブルにします。この機能により、パフォーマンスが大幅に向上します。

挿入バッファはInnoDBの特別な機能もあります。一意でないセカンダリインデックスロックがメモリ内にない場合、InnoDBはエントリを挿入バッファに挿入して、ランダムなI/O操作を回避します。定期的に、挿入バッファはデータベース内のセカンダリインデックスツリーにマージされます。挿入バッファは、I/O要求を同じブロックにマージすることでI/O処理数を削減します。ランダムI/O処理はシーケンシャルです。また、インサートバッファはHDD用に高度に最適化されています。シーケンシャルライトと読み取りは、どちらも通常運用時に発生します。

元に戻すセグメントは、ランダムI/Oです。Multi-Version Concurrency (MVCC) を保証するために、InnoDBは元に戻すセグメントに古いイメージを登録する必要があります。元に戻すセグメントから以前の画像を読み取るには、ランダムな読み取りが必要です。繰り返し実行可能な読み取りで長時間のトランザクション (mysqldump—単一トランザクションなど) を実行したり、長時間のクエリを実行したりすると、ランダムリードが発生する可能性があります。したがって、この場合、元に戻すセグメントをSSDに保存する方が適しています。短いトランザクションまたはクエリのみを実行する場合、ランダムリードは問題ではありません。

- NetAppでは、InnoDBのI/O特性を考慮して、以下のストレージ設計レイアウトを推奨しています。
- MySQLのランダムI/OおよびシーケンシャルI/O指向ファイルを1つのボリュームに格納
- 純粋にシーケンシャルなI/O指向のMySQLファイルを格納するための別のボリューム

このレイアウトは、データ保護のポリシーと戦略の設計にも役立ちます。

## 設定パラメータ

NetAppでは、最適なパフォーマンスを得るために、いくつかの重要なMySQL構成パラメータを推奨しています。

パラメータ	値
innodb_log_file_size	2億5、600万
innodb_flush_log_at_trx_commit	2.
innodb_doublewrite	0
innodb_flush_method	fsync
innodb_buffer_pool_size	11g
innodb_io_capacity	8、192です
innodb_buffer_pool_instances	8
innodb_lru_scan_depth	8、192です
open_file_limit	65535

このセクションで説明するパラメータを設定するには、MySQL構成ファイル（my.cnf）でパラメータを変更する必要があります。NetAppのベストプラクティスは、社内で実施したテストの結果です。

## **innodb\_log\_file\_size**

InnoDBログファイルのサイズに適したサイズを選択することは、書き込み処理およびサーバクラッシュ後の適切なリカバリ時間を確保するために重要です。

ファイルに記録されるトランザクションの数が非常に多いため、書き込み処理ではログファイルのサイズが重要になります。レコードが変更されても、変更はすぐにはテーブルスペースに書き込まれません。代わりに、変更はログファイルの最後に記録され、ページはダーティとしてマークされます。InnoDBはログを使用してランダムI/OをシーケンシャルI/Oに変換します。

ログがいっぱいになると、ダーティページが順番にテーブルスペースに書き込まれ、ログファイルのスペースが解放されます。たとえば、トランザクションの途中でサーバがクラッシュし、書き込み処理のみがログファイルに記録されるとします。サーバを再び稼働させるには、ログファイルに記録された変更が再生されるリカバリフェーズを実行する必要があります。ログファイル内のエントリ数が多いほど、サーバのリカバリにかかる時間が長くなります。

この例では、ログファイルのサイズがリカバリ時間と書き込みパフォーマンスの両方に影響します。ログファイルのサイズに適切な数を選択する場合は、リカバリ時間と書き込みパフォーマンスのバランスを取ります。通常、128Mから512Mの間の値は良い値です。

## **innodb\_flush\_log\_at\_trx\_commit**

データに変更があっても、変更はすぐにはストレージに書き込まれません。

代わりに、データはログバッファに記録されます。これは、InnoDBがログファイルに記録されるバッファ変更に割り当てるメモリの一部です。InnoDBは、トランザクションがコミットされたとき、バッファがいっぱいになったとき、または1秒に1回のイベントが発生したときに、バッファをログファイルにフラッシュします。このプロセスを制御する構成変数は、innodb\_flush\_log\_at\_trx\_commitです。値オプションには次のものがあります。

- 設定時 `innodb\_flush\_log\_trx\_at\_commit=0` では、（InnoDBバッファプール内の）変更されたデータがログファイル（ib\_logfile）に書き込まれ、ログファイルが1秒ごとにフラッシュされます（ストレージへの書き込み）。ただし、トランザクションがコミットされても何も実行されません。停電またはシステムクラッシュが発生した場合、フラッシュされていないデータはログファイルまたはドライブに書き込まれないため、リカバリできません。
- 設定時 `innodb\_flush\_log\_trx\_commit=1` InnoDBはログバッファをトランザクションログに書き込み、トランザクションごとに永続的なストレージにフラッシュします。たとえば、すべてのトランザクションコミットについて、InnoDBはログに書き込み、その後ストレージに書き込みます。ストレージの速度が遅いと、パフォーマンスが低下します。たとえば、1秒あたりのInnoDBトランザクション数が減少します。
- 設定時 `innodb\_flush\_log\_trx\_commit=2` InnoDBはコミットのたびにログバッファをログファイルに書き込みますが、ストレージにデータを書き込むことはありません。InnoDBは、1秒に1回データをフラッシュします。停電やシステムクラッシュが発生した場合でも、オプション2のデータはログファイルに保存され、リカバリ可能です。

パフォーマンスが主な目標である場合は、値を2に設定します。InnoDBは、トランザクションコミットごとではなく、1秒に1回ドライブに書き込みを行うため、パフォーマンスが大幅に向上します。停電やクラッシュが発生した場合は、トランザクションログからデータをリカバリできます。

データの安全性が主な目的である場合は、トランザクションコミットごとにInnoDBがドライブにフラッシュされるように、値を1に設定します。ただし、パフォーマンスに影響する可能性があります。



\* NetAppでは、パフォーマンスを向上させるために、`innodb_flush_log_trx_commit`値を2に設定することを推奨しています。

## **innodb\_doublewrite**

いつ `innodb_doublewrite` イネーブル(デフォルト)では、InnoDBはすべてのデータを2回格納します。最初にダブルライトバッファに格納し、次に実際のデータファイルに格納します。

このパラメータは、次のコマンドで無効になります。`--skip-innodb_doublewrite` ベンチマークの場合や、データの整合性や障害の可能性よりも最高のパフォーマンスに関心がある場合。InnoDBでは、ダブルライトと呼ばれるファイルフラッシュ技術が使用されています。InnoDBは、ページをデータファイルに書き込む前に、ダブル書き込みバッファと呼ばれる連続領域にページを書き込みます。書き込みとダブル書き込みバッファへのフラッシュが完了すると、InnoDBはページをデータファイル内の適切な位置に書き込みます。ページの書き込み中にオペレーティングシステムまたはmysqldプロセスがクラッシュした場合、InnoDBは後でクラッシュリカバリ中にダブルライトバッファからページの適切なコピーを見つけることができます。



\* NetAppでは、ダブル書き込みバッファを無効にすることを推奨しています。ONTAP NVRAMは同じ機能を果たします。ダブルバッファリングは不必要にパフォーマンスを低下させます。

## **innodb\_buffer\_pool\_size**

InnoDBバッファプールは、チューニングアクティビティの中で最も重要な部分です。

InnoDBは、インデックスのキャッシュとデータのローリング、アダプティブハッシュインデックス、挿入バッファ、および内部で使用される他の多くのデータ構造をバッファプールに大きく依存しています。また、データへの変更もバッファに格納されるため、書き込み処理をストレージに対してすぐに実行する必要はありません。これにより、パフォーマンスが向上します。バッファプールはInnoDBの不可欠な部分であり、それに応じてサイズを調整する必要があります。バッファプールサイズを設定するときは、次の点を考慮してください。

- 専用のInnoDB専用マシンの場合は、バッファプールサイズを使用可能なRAMの80%以上に設定します。
- MySQL専用サーバーでない場合は、サイズをRAMの50%に設定します。

## **innodb\_flush\_method**

`innodb_flush_method`パラメータは、InnoDBがログファイルとデータファイルを開いてフラッシュする方法を指定します。

### **最適化**

InnoDB最適化では、このパラメータを設定すると、必要に応じてデータベースのパフォーマンスが調整されます。

次のオプションは、InnoDBを使用してファイルをフラッシュするためのものです。

- `fsync`。InnoDBでは、`fsync()` データファイルとログファイルの両方をフラッシュするシステムコール。このオプションはデフォルト設定です。
- `O_DSYNC`。InnoDBでは、`O_DSYNC` ログファイルを開いてフラッシュし、データファイルをフラッシュするには`fsync()`を使用します。InnoDBが使用しない `O_DSYNC` 直接、なぜなら、多くの種類のUNIXで問題があったからです。
- `O_DIRECT`。InnoDBでは、`O_DIRECT` オプション（または `directio()`（Solarisの場合）データファイルを開くには、次のコマンドを使用します。`fsync()` データファイルとログファイルの両方をフラッシュします。このオプションは、一部のGNU/Linuxバージョン、FreeBSD、Solarisで利用できます。
- `O_DIRECT_NO_FSYNC`。InnoDBでは、`O_DIRECT` オプションはI/Oのフラッシュ時に使用されますが、`fsync()` 後でシステムコール。このオプションは、一部のタイプのファイルシステム（XFSなど）には適していません。ファイルシステムに`fsync()` システムコール（たとえば、すべてのファイルメタデータを保持する場合）は、`O_DIRECT` 代わりにオプション。

## 観察

NetAppラボテストでは、`fsync` NFSとSANではdefaultオプションが使用されており、`O_DIRECT` flushメソッドを使用している場合 `O_DIRECT` ONTAPでは、クライアントが4096ブロックの境界で大量のシングルバイト書き込みをシリアル方式で書き込みました。この書き込みにより、ネットワーク経由のレイテンシが増加し、パフォーマンスが低下します。

## innodb\_io\_capacity

InnoDBプラグインでは、MySQL 5.7から`InnoDB_io_capacity`という新しいパラメータが追加されました。

InnoDBが実行する最大IOPS（ダーティページのフラッシュレートと挿入バッファ[ibuf]バッチサイズを含む）を制御します。`innodb_io_capacity`パラメータは、バッファプールからのページのフラッシュや変更バッファからのデータのマージなど、InnoDBバックグラウンドタスクによるIOPSの上限を設定します。

`innodb_io_capacity`パラメータに、システムが1秒あたりに実行できるI/O処理のおおよその数を設定します。理想的には、設定ができるだけ低くしてくださいが、バックグラウンドアクティビティが遅くなるように低くしないでください。この設定が高すぎると、データがバッファプールから削除され、キャッシングするにはバッファの挿入が早すぎて大きなメリットが得られません。



\* NetAppでは\*この設定をNFSで使用する場合は、IOPSのテスト結果（SysBench / Fio）を分析し、それに応じてパラメータを設定することを推奨します。InnoDBバッファプールで必要以上に変更されたページやダーティなページが表示されない場合は、フラッシュとページにできるだけ小さい値を使用してください。



ワークロードに対して低い値では不十分であることを証明した場合を除き、20、000以上などの極端な値は使用しないでください。

`innodb_io_capacity`パラメータは、フラッシュ速度と関連するI/Oを調整します。



このパラメータまたは`innodb_io_capacity_max`パラメータを高すぎて無駄にすると、パフォーマンスに深刻な影響を与える可能性があります。

## **innodb\_lru\_scan\_depth**

。 `innodb_lru_scan_depth` パラメータは、InnoDBバッファプールのフラッシュ操作のアルゴリズムとヒューリスティックに影響を与えます。

このパラメータは、I/O負荷の高いワークロードを調整するパフォーマンスのエキスパートが主に関心を持っています。このパラメータは、バッファプールインスタンスごとに、Least Recently Used (LRU) ページリスト内でページクリーナースレッドがスキャンを続行し、フラッシュするデータページを探す距離を指定します。このバックグラウンド処理は1秒に1回実行されます。

値を上下に調整して、空きページ数を最小限に抑えることができます。必要以上の値を設定しないでください。スキャンを実行すると、パフォーマンスが大幅に低下する可能性があります。また、バッファプールインスタンスの数を変更する場合は、このパラメータを調整することを検討してください。

`innodb_lru_scan_depth * innodb_buffer_pool_instances` ページクリーナースレッドが毎秒実行する作業量を定義します。

デフォルトよりも小さい設定は、ほとんどのワークロードに適しています。一般的なワークロードでスペアのI/O容量がある場合にのみ、値を増やすことを検討してください。逆に、大量の書き込みが発生するワークロードでI/O容量が飽和状態になった場合は、特にバッファプールが大きい場合は値を小さくしてください。

## **open\_file\_limits**

。 `open_file_limits` パラメータは、オペレーティングシステムがmysqldを開くことを許可するファイルの数を決定します。

実行時のこのパラメータの値は、システムで許可されている実際の値であり、サーバの起動時に指定した値とは異なる場合があります。MySQLが開いているファイルの数を変更できないシステムでは、値は0です。効果的な `open_files_limit` 値は、システムの起動時に指定された値（存在する場合）と `max_connections` および `table_open_cache` 次の式を使用します。

- 10以上 `max_connections` [+] (`table_open_cache` ×2)
- `max_connections` × 5
- オペレーティングシステムの制限（正の場合）
- オペレーティングシステムの制限が無限大の場合：`open_files_limit` 起動時に値が指定されます。指定されていない場合は5,000

サーバは、これら4つの値の最大値を使用してファイル記述子の数を取得しようとします。これだけ多くのディスクリプタを取得できない場合、サーバはシステムが許可する数だけのディスクリプタを取得しようとします。

## **ホストの設定**

### **コンテナ化**

MySQLデータベースのコンテナ化はますます普及しています。

低レベルのコンテナ管理は、ほとんどの場合Dockerを使用して実行されます。OpenShiftやKubernetesなどのコンテナ管理プラットフォームを使用すると、大規模なコンテナ環境の管理がさらに簡単になります。コンテナ化のメリットとしては、ハイパーバイザーのライセンスが不要なため、コストの削減が挙げられます。ま

た、コンテナを使用すると、基盤となる同じカーネルとオペレーティングシステムを共有しながら、複数のデータベースを互いに分離して実行できます。コンテナはマイクロ秒単位でプロビジョニングできます。

NetAppは、ストレージの高度な管理機能を提供するAstra Tridentを提供しています。たとえば、Astra Tridentを使用すると、Kubernetesで作成されたコンテナは、適切な階層にストレージを自動的にプロビジョニングしたり、エクスポートポリシーを適用したり、Snapshotポリシーを設定したり、コンテナを別の階層にクローニングしたりできます。詳細については、["Astra Trident のドキュメント"](#)ください。

## NFSv3スロットテーブル

LinuxでのNFSv3のパフォーマンスは、`tcp_max_slot_table_entries`。

TCPスロットテーブルは、NFSv3でホストバスアダプタ（HBA）のキュー深度に相当します。一度に未処理となることのできるNFS処理の数を制御します。デフォルト値は通常16ですが、最適なパフォーマンスを得るには小さすぎます。逆に、新しいLinuxカーネルでTCPスロットテーブルの上限をNFSサーバが要求でいっぱいになるレベルに自動的に引き上げることができます。

パフォーマンスを最適化し、パフォーマンスの問題を回避するには、TCPスロットテーブルを制御するカーネルパラメータを調整します。

を実行します `sysctl -a | grep tcp.*.slot_table` コマンドを実行し、次のパラメータを確認します。

```
# sysctl -a | grep tcp.*.slot_table
sunrpc.tcp_max_slot_table_entries = 128
sunrpc.tcp_slot_table_entries = 128
```

すべてのLinuxシステムに`sunrpc.tcp_slot_table_entries`、ただし、次のようなものがあります。  
`sunrpc.tcp_max_slot_table_entries`。どちらも128に設定する必要があります。

これらのパラメータを設定しないと、パフォーマンスに大きく影響する可能性があります。Linux OSが十分ないI/Oを発行していないためにパフォーマンスが制限される場合もあります。一方では、Linux OSが問題で処理できる以上のI/Oを試行すると、I/Oレイテンシが増加します。

## I/Oスケジューラ

Linuxカーネルでは、ブロックデバイスへのI/Oのスケジュール方法を低レベルで制御できます。

Linuxのさまざまなディストリビューションでのデフォルト設定は大きく異なります。MySQLでは、NOOPまたはdeadline Linuxでネイティブの非同期I/O（AIO）を使用するI/Oスケジューラ。一般に、NetAppのお客様や社内テストでは、NoOpsの方が効果的です。

MySQLのInnoDBストレージエンジンは、Linux上の非同期I/Oサブシステム（ネイティブAIO）を使用して、データファイルページの先読み要求と書き込み要求を実行します。この動作は、`innodb_use_native_aio`設定オプション。デフォルトで有効になっています。ネイティブの一体型I/Oでは、I/OスケジューラのタイプがI/Oパフォーマンスに大きく影響します。ベンチマークを実施して、ワークロードと環境に最適な結果を提供するI/Oスケジューラを特定します。

I/Oスケジューラの設定手順については、該当するLinuxおよびMySQLのドキュメントを参照してください。

## ファイル記述子

MySQLサーバを実行するにはファイル記述子が必要であり、デフォルト値では不十分です。

これらを使用して、新しい接続を開いたり、テーブルをキャッシュに格納したり、複雑なクエリを解決するための一時テーブルを作成したり、永続的なテーブルにアクセスしたりします。mysqldが必要なときに新しいファイルを開くことができない場合、正常に機能しなくなる可能性があります。この問題の一般的な現象は、エラー24「開いているファイルが多すぎます」です。mysqldが同時に開くことができるファイル記述子の数は、`open_files_limit`構成ファイルに設定されたオプション(`/etc/my.cnf`)。でも…。

`open_files_limit`オペレーティングシステムの制限によっても異なります。この依存関係により、変数の設定がより複雑になります。

MySQLカセットティテキナイ `open_files_limit` オプションがで指定されている値よりも大きい `ulimit 'open files'`。したがって、必要に応じてMySQLがファイルを開くことができるよう、これらの制限をオペレーティングシステムレベルで明示的に設定する必要があります。Linuxでファイル制限を確認するには、次の2つの方法があります。

- `ulimit`コマンドを実行すると、許可またはロックされているパラメータの詳細な概要がすばやく表示されます。このコマンドを実行して行った変更は永続的ではなく、システムのリブート後に消去されます。
- に対する変更 `/etc/security/limit.conf` ファイルは永続的であり、システムの再起動の影響を受けません。

ユーザmysqlのハードリミットとソフトリミットの両方を変更してください。構成からの抜粋を次に示します。

```
mysql hard nofile 65535
mysql soft nofile 65353
```

並行して、`my.cnf` 開いているファイル制限を完全に使用するには。

## ストレージ構成

### NFS

MySQLのドキュメントでは、NAS環境にはNFSv4を使用することを推奨しています。

#### ONTAP NFS転送サイズ

ONTAPでは、デフォルトでNFS IOサイズが64Kに制限されます。MySQLデータベースでランダムIOを使用する場合、ブロックサイズは64Kの最大値よりもはるかに小さくなります。ラージブロックIOは通常並列化されるため、64Kの最大値も制限ではありません。

一部のワークロードでは、最大64Kに制限があります。特に、フルテーブルスキャンバックアップ処理などのシングルスレッド処理は、データベースで実行するI/Oが少なくても大容量であれば、より高速かつ効率的に実行されます。データベースワークロードを使用するONTAPに最適なIO処理サイズは256Kです。以下のオペ

レーティングシステムのNFSマウントオプションは、それに応じて64Kから256Kに更新されています。

特定のONTAP SVMの最大転送サイズは、次のように変更できます。

```
Cluster01::> set advanced
```

```
Warning: These advanced commands are potentially dangerous; use them only  
when directed to do so by NetApp personnel.
```

```
Do you want to continue? {y|n}: y
```

```
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size  
262144
```

 ONTAPで許容される最大転送サイズを、現在マウントされているNFSファイルシステムのrsize/wsizeの値より小さくしないでください。これにより、一部のオペレーティングシステムでハングしたり、データが破損したりする可能性があります。たとえば、NFSクライアントのrsize / wsizeが65536に設定されている場合は、クライアント自体が制限されているため、ONTAPの最大転送サイズを65536~1048576の間で調整しても効果はありません。最大転送サイズを65536未満に縮小すると、可用性やデータが損傷する可能性があります。

- NetAppの推奨事項\*

 次のNFSv4 fstab (/etc/fstab) 設定を設定します。

```
nfs4 rw,  
hard,nointr,bg,vers=4,proto=tcp,noatime,rsize=262144,wsize=262144
```

 NFSv3を使用する一般的な問題は、停電後にロックされたInnoDBログファイルでした。時間または切り替えログファイルを使用して、この問題を解決しました。ただし、NFSv4にはロック処理があり、開いているファイルや委譲が追跡されます。

## SAN

I/Oと容量の要件が单一のLUNファイルシステムの制限内であれば、小規模なデータベースを標準LUNのペアに配置できます。たとえば、約2,000 IOPSのランダムIOPSを必要とするデータベースを、1つのLUN上の単一のファイルシステムでホストできます。同様に、サイズが100GBしかないデータベースでも、1つのLUNに収まります。管理上の問題は発生しません。

大規模なデータベースには複数のLUNが必要です。たとえば、10万IOPSを必要とするデータベースには、少なくとも8つのLUNが必要です。ドライブへのSCSIチャネルの数が不十分なため、1つのLUNがボトルネックになります。同じように、10TBのデータベースを1つの10TB LUNで管理するのは困難です。論理ボリュームマネージャは、複数のLUNのパフォーマンス機能と容量機能を結合して、パフォーマンスと管理性を向上させるように設計されています。

どちらの場合も、ONTAPボリュームのペアで十分です。単純な設定では、ログLUNと同様に、データファイルLUNも専用ボリュームに配置されます。論理ボリュームマネージャ構成の場合、データファイルボリュームグループ内のすべてのLUNは専用ボリュームに配置され、ログボリュームグループのLUNは2つ目の専用ボリュームに配置されます。

- NetAppでは\* SANへのMySQL導入には2つのファイルシステムを使用することを推奨しています。
- 最初のファイルシステムには、表領域、データ、インデックスを含むすべてのMySQLデータが格納されます。
- 2番目のファイルシステムには、すべてのログ（バイナリログ、低速ログ、トランザクションログ）が格納されます。

この方法でデータを分離する理由には、次のようなものがあります。

- データファイルとログファイルのI/Oパターンは異なります。これらを分離すると、QoS制御でより多くのオプションを使用できるようになります。
- Snapshotテクノロジを最適に使用するには、データファイルを個別にリストアできる必要があります。データファイルとログファイルを混在させると、データファイルのリストアが妨げられます。
- NetApp SnapMirrorテクノロジを使用すると、シンプルでRPOの低いディザスタリカバリ機能をデータベースに提供できますが、データファイルとログには異なるレプリケーションスケジュールが必要です。

ONTAPのすべての機能を必要に応じて使用解決策できるように、この基本的な2ボリュームレイアウトを使用して、将来のニーズにも対応します。

- NetAppでは、次の機能により、ドライブをext4ファイルシステムでフォーマットすることを推奨しています。
  - JFS（ジャーナルファイルシステム）で使用されるブロック管理機能の拡張アプローチと、XFS（拡張ファイルシステム）の遅延割り当て機能。
  - ext4は、最大1エクスビバイト ( $2^{60}$ バイト) のファイルシステムと最大16テビバイト ( $16 \times 2^{40}$ バイト) のファイルシステムを許可します。一方、ext3ファイルシステムでサポートされる最大ファイルシステムサイズは16TB、最大ファイルサイズは2TBです。
  - ext4ファイルシステムでは、複数ブロック割り当て(malloc)は、ext3のようにファイルを1つずつ割り当てるのではなく、1回の操作で複数のブロックを割り当てます。この構成により、ブロックアロケータを数回呼び出すオーバーヘッドが削減され、メモリの割り当てが最適化されます。
  - XFSは多くのLinuxディストリビューションではデフォルトですが、メタデータの管理方法が異なり、一部のMySQL構成には適していません。
- 
- NetAppでは、mkfsユーティリティで4kブロックサイズオプションを使用して、既存のブロックLUNサイズに合わせることを推奨\*しています。

```
mkfs.ext4 -b 4096
```

NetApp LUNは4KBの物理ブロックにデータを格納するため、512バイトの論理ブロックが8個生成されます。

同じブロックサイズを設定しないと、I/Oは物理ブロックと正しくアライメントされず、RAIDグループ内の2つの異なるドライブに書き込みが行われてレイテンシが発生する可能性があります。



スムーズな読み取り/書き込み処理を実現するためには、I/Oのアライメントが重要です。ただし、物理ブロックの先頭以外の論理ブロックからI/Oが開始されると、I/Oはミスマライメントされます。I/O処理がアライメントされるのは、I/O処理が論理ブロック（物理ブロック内の最初の論理ブロック）で開始されたときだけです。

## 著作権に関する情報

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S.このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発生理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を隨時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

権利の制限について：政府による使用、複製、開示は、DFARS 252.227-7013（2014年2月）およびFAR 5225.227-19（2007年12月）のRights in Technical Data -Noncommercial Items（技術データ - 非商用品目に関する諸権利）条項の(b)(3)項、に規定された制限が適用されます。

本書に含まれるデータは商用製品および / または商用サービス（FAR 2.101の定義に基づく）に関係し、データの所有権はNetApp, Inc.にあります。本契約に基づき提供されるすべてのネットアップの技術データおよびコンピュータソフトウェアは、商用目的であり、私費のみで開発されたものです。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc.の書面による許可を事前に得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b)項（2014年2月）で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetAppのロゴ、<http://www.netapp.com/TM>に記載されているマークは、NetApp, Inc.の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。