



# MySQL

## Enterprise applications

NetApp  
January 02, 2026

# Sommaire

MySQL .....	1
Présentation .....	1
Configuration de la base de données .....	1
Structure de fichier .....	1
Paramètres de configuration .....	4
taille_fichier_log_innodb .....	5
innodb_flush_log_at_trx_commit .....	5
innodb_doublewrite .....	6
innodb_buffer_pool_size .....	6
innodb_flush_method .....	7
innodb_io_capacity .....	7
innodb_lru_scan_depth .....	8
open_file_limits .....	8
Configuration de l'hôte .....	9
Conteneurisation .....	9
Tables d'emplacements NFSv3 .....	9
Planificateurs d'E/S .....	10
Descripteurs de fichier .....	10
Configuration de stockage sous-jacente .....	11
NFS .....	11
SAN .....	12

# MySQL

## Présentation

MySQL et ses variantes, y compris MariaDB et Percona MySQL, est la base de données la plus populaire au monde.



Cette documentation sur ONTAP et la base de données MySQL remplace la base de données *TR-4722: Base de données MySQL sur les meilleures pratiques ONTAP*.

ONTAP est une plate-forme idéale pour les bases de données MySQL car ONTAP est littéralement conçu pour les bases de données. De nombreuses fonctionnalités, telles que l'optimisation de la latence d'E/S aléatoire, pour la qualité de service (QoS) avancée et les fonctionnalités FlexClone de base, ont été spécialement conçues pour répondre aux besoins des charges de travail des bases de données.

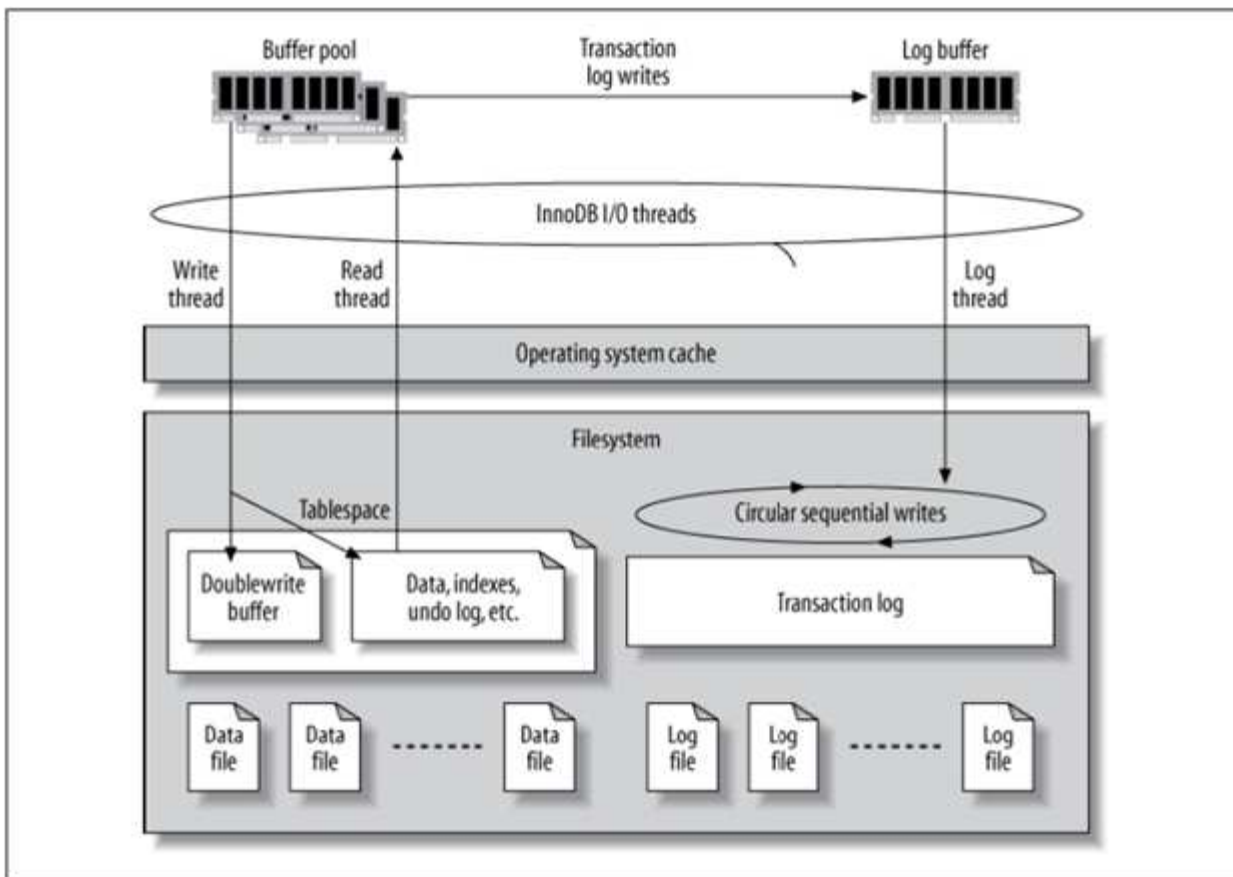
Des fonctionnalités supplémentaires, telles que les mises à niveau sans interruption (y compris le remplacement du stockage), assurent la disponibilité de vos bases de données stratégiques. Vous pouvez également bénéficier d'une reprise après incident instantanée pour les environnements volumineux via MetroCluster ou sélectionner des bases de données à l'aide de la synchronisation active SnapMirror.

Plus important encore, ONTAP offre des performances inégalées avec la possibilité de dimensionner la solution en fonction de vos besoins spécifiques. Nos systèmes haut de gamme peuvent fournir plus de 1 million d'IOPS à des latences mesurées en microsecondes. Toutefois, si vous n'avez besoin que de 100 000 IOPS, vous pouvez dimensionner correctement votre solution de stockage avec un contrôleur plus petit, qui exécute toujours le même système d'exploitation du stockage.

## Configuration de la base de données

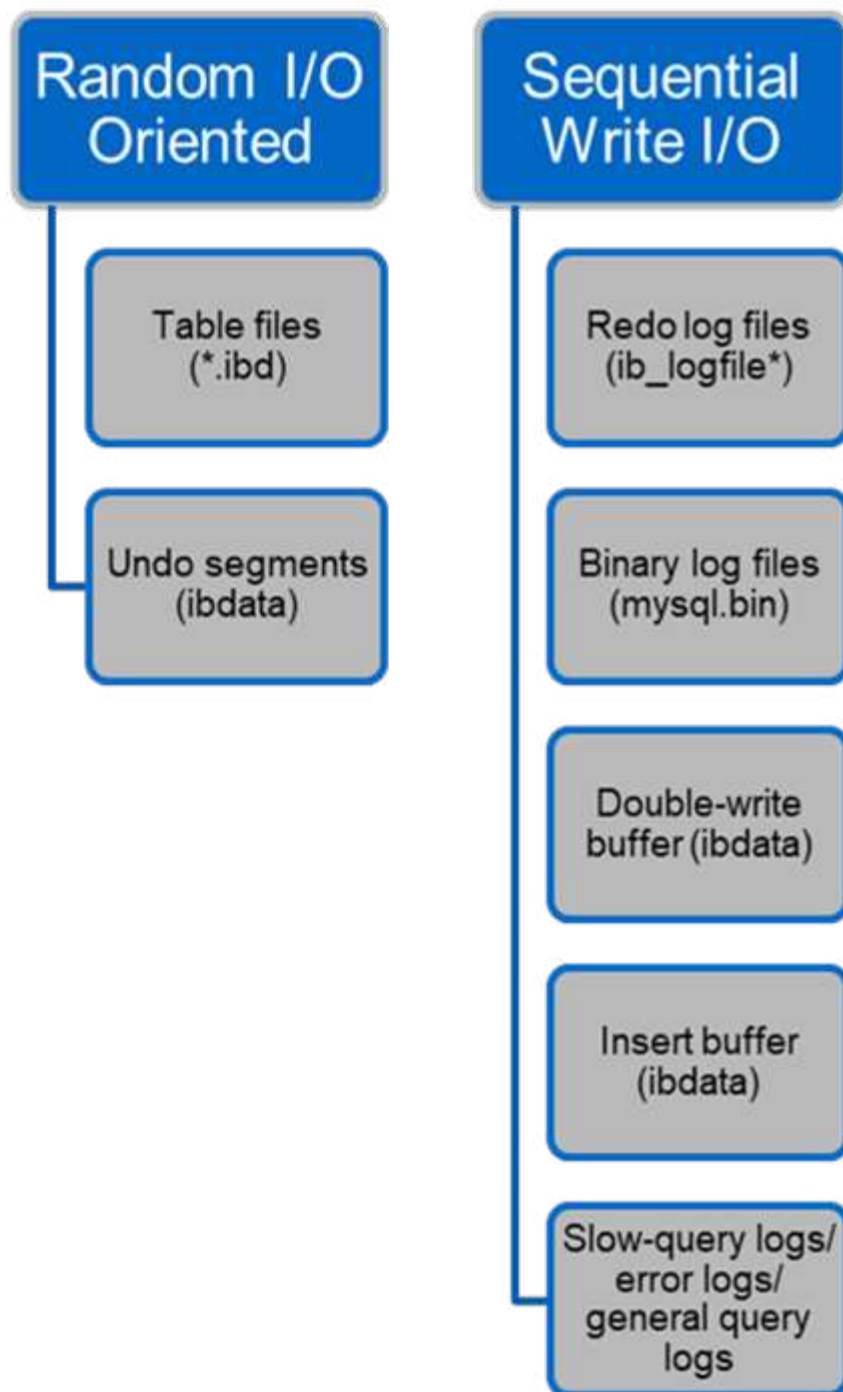
### Structure de fichier

InnoDB agit comme la couche intermédiaire entre le stockage et le serveur MySQL, il stocke les données sur les lecteurs.



Les E/S MySQL sont classées en deux types :

- E/S de fichiers aléatoires
- E/S séquentielles de fichiers



Les fichiers de données sont lus et écrasés de manière aléatoire, ce qui entraîne un nombre élevé d'IOPS. Un stockage SSD est donc recommandé.

Les fichiers redo log et les fichiers log binaires sont des journaux transactionnels. Ils sont écrits de manière séquentielle, ce qui vous permet d'obtenir de bonnes performances sur le disque dur avec le cache d'écriture. Une lecture séquentielle a lieu lors de la restauration, mais cela provoque rarement un problème de performance, car la taille du fichier journal est généralement inférieure à celle des fichiers de données et les lectures séquentielles sont plus rapides que les lectures aléatoires (se produisant sur les fichiers de données).

La mémoire tampon en double écriture est une fonction spéciale d'InnoDB. InnoDB écrit d'abord les pages vidées dans le tampon de double écriture, puis écrit les pages à leur position correcte sur les fichiers de

données. Ce processus empêche la corruption de la page. Sans le tampon de double écriture, la page peut être corrompue si une panne de courant se produit pendant le processus d'écriture sur les lecteurs. L'écriture sur la mémoire tampon en double écriture étant séquentielle, elle est optimisée pour les disques durs. Les lectures séquentielles ont lieu lors de la restauration.

Comme la mémoire NVRAM ONTAP fournit déjà une protection en écriture, la mise en mémoire tampon en double écriture n'est pas nécessaire. MySQL a un paramètre, `skip_innodb_doublewrite`, pour désactiver le tampon de double écriture. Cette fonction peut améliorer considérablement les performances.

Le tampon d'insertion est également une fonction spéciale d'InnoDB. Si des blocs d'index secondaires non uniques ne sont pas en mémoire, InnoDB insère des entrées dans le tampon d'insertion pour éviter les opérations d'E/S aléatoires. Périodiquement, le tampon d'insertion est fusionné dans les arborescences d'index secondaires de la base de données. La mémoire tampon d'insertion réduit le nombre d'opérations d'E/S en fusionnant les demandes d'E/S vers le même bloc ; les opérations d'E/S aléatoires peuvent être séquentielles. Le tampon d'insertion est également hautement optimisé pour les disques durs. Les écritures et les lectures séquentielles ont lieu pendant les opérations normales.

Les segments d'annulation sont orientés E/S aléatoires. Pour garantir la simultanéité multiversion (MVCC), InnoDB doit enregistrer les anciennes images dans les segments d'annulation. La lecture des images précédentes à partir des segments d'annulation nécessite des lectures aléatoires. Si vous exécutez une longue transaction avec des lectures reproductibles (comme `mysqldump`—single transaction) ou exécutez une longue requête, les lectures aléatoires peuvent se produire. Par conséquent, le stockage des segments d'annulation sur des disques SSD est préférable dans ce cas. Si vous exécutez uniquement des transactions ou des requêtes courtes, les lectures aléatoires ne sont pas un problème.

**NetApp recommande** la disposition de conception de stockage suivante en raison des caractéristiques d'E/S InnoDB.



- Un volume pour stocker des fichiers MySQL orientés E/S aléatoires et séquentiels
- Un autre volume pour stocker des fichiers MySQL orientés E/S purement séquentiels

Cette disposition vous aide également à concevoir des stratégies et des règles de protection des données.

### Paramètres de configuration

NetApp recommande quelques paramètres de configuration MySQL importants pour obtenir des performances optimales.

Paramètres	Valeurs
<code>taille_fichier_log_innodb</code>	256M
<code>innodb_flush_log_at_trx_commit</code>	2
<code>innodb_doublewrite</code>	0
<code>innodb_flush_method</code>	<code>fsync</code>
<code>innodb_buffer_pool_size</code>	11G
<code>innodb_io_capacity</code>	8192
<code>innodb_buffer_pool_instances</code>	8
<code>innodb_lru_scan_depth</code>	8192

open_file_limit	65535
-----------------	-------

Pour définir les paramètres décrits dans cette section, vous devez les modifier dans le fichier de configuration MySQL (my.cnf). Les meilleures pratiques NetApp sont le résultat de tests réalisés en interne.

## taille\_fichier\_log\_innodb

Il est important de sélectionner la bonne taille pour le fichier journal InnoDB pour les opérations d'écriture et pour avoir un temps de récupération décent après une panne du serveur.

Étant donné que tant de transactions sont connectées au fichier, la taille du fichier journal est importante pour les opérations d'écriture. Lorsque les enregistrements sont modifiés, la modification n'est pas immédiatement réécrite dans l'espace de table. Au lieu de cela, la modification est enregistrée à la fin du fichier journal et la page est marquée comme sale. InnoDB utilise son journal pour convertir les E/S aléatoires en E/S séquentielles

Lorsque le journal est plein, la page sale est écrite dans l'espace table en séquence pour libérer de l'espace dans le fichier journal. Par exemple, supposons qu'un serveur se bloque au milieu d'une transaction et que les opérations d'écriture ne sont enregistrées que dans le fichier journal. Avant que le serveur puisse de nouveau être mis en service, il doit passer par une phase de récupération dans laquelle les modifications enregistrées dans le fichier journal sont relus. Plus le nombre d'entrées dans le fichier journal est important, plus la restauration du serveur prend de temps.

Dans cet exemple, la taille du fichier journal affecte à la fois le temps de restauration et les performances d'écriture. Lorsque vous choisissez le bon nombre pour la taille du fichier journal, équilibrez le délai de restauration par rapport aux performances d'écriture. En général, tout ce qui se trouve entre 128M et 512M est d'une bonne valeur.

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

En cas de modification des données, celles-ci ne sont pas immédiatement écrites sur le support de stockage.

À la place, les données sont enregistrées dans une mémoire tampon, qui est une partie de la mémoire allouée par InnoDB aux modifications de mémoire tampon enregistrées dans le fichier journal. InnoDB vide le tampon dans le fichier journal lorsqu'une transaction est validée, lorsque le tampon est plein ou une fois par seconde, quel que soit l'événement qui se produit en premier. La variable de configuration qui contrôle ce processus est `innodb_flush_log_at_trx_commit`. Les options de valeur comprennent :

- Lorsque vous réglez `innodb_flush_log_trx_at_commit=0`, InnoDB écrit les données modifiées (dans le pool de mémoire tampon InnoDB) dans le fichier journal (`ib_logfile`) et purge le fichier journal (écriture dans le stockage) toutes les secondes. Cependant, elle ne fait rien lorsque la transaction est validée. En cas de panne de courant ou de panne du système, aucune des données non rincées n'est récupérable car elles ne sont pas écrites sur le fichier journal ou les lecteurs.
- Lorsque vous réglez `innodb_flush_log_trx_commit=1`, InnoDB écrit la mémoire tampon du journal dans le journal de transactions et vide jusqu'à un stockage durable pour chaque transaction. Par exemple, pour toutes les validations de transactions, InnoDB écrit dans le journal, puis écrit dans le stockage. Un stockage plus lent affecte négativement les performances. Par exemple, le nombre de transactions InnoDB par seconde est réduit.
- Lorsque vous réglez `innodb_flush_log_trx_commit=2`, InnoDB écrit la mémoire tampon du journal

dans le fichier journal à chaque validation ; cependant, il n'écrit pas de données dans le stockage. InnoDB vide les données une fois par seconde. Même en cas de panne de courant ou de panne du système, les données de l'option 2 sont disponibles dans le fichier journal et peuvent être récupérées.

Si la performance est l'objectif principal, définissez la valeur sur 2. Comme InnoDB écrit sur les disques une fois par seconde, pas pour chaque validation de transaction, les performances s'améliorent considérablement. En cas de panne de courant ou de panne de courant, les données peuvent être récupérées à partir du journal de transactions.

Si la sécurité des données est l'objectif principal, définissez la valeur sur 1 afin que, pour chaque validation de transaction, InnoDB vide les lecteurs. Cependant, les performances peuvent être affectées.



**NetApp recommande** de définir la valeur `innodb_flush_log_trx_commit` sur 2 pour de meilleures performances.

## **innodb\_doublewrite**

Quand `innodb_doublewrite` Est activé (valeur par défaut), InnoDB stocke toutes les données deux fois : d'abord dans le tampon de double écriture, puis dans les fichiers de données réels.

Vous pouvez désactiver ce paramètre avec `--skip-innodb_doublewrite` pour les bancs d'essai ou lorsque vous êtes davantage préoccupé par les performances que par l'intégrité des données ou par les défaillances possibles. InnoDB utilise une technique de vidage de fichier appelée double écriture. Avant d'écrire des pages dans les fichiers de données, InnoDB les écrit dans une zone contiguë appelée tampon de double écriture. Une fois l'écriture et le vidage de la mémoire tampon en double écriture terminés, InnoDB écrit les pages dans leur position correcte dans le fichier de données. Si le système d'exploitation ou un processus mysqld se bloque lors d'une écriture de page, InnoDB peut plus tard trouver une bonne copie de la page à partir du tampon de double écriture pendant la récupération après panne.



**NetApp recommande** de désactiver le tampon en double écriture. La mémoire NVRAM de ONTAP remplit la même fonction. La double mise en mémoire tampon endommagera inutilement les performances.

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

Le pool de mémoire tampon InnoDB est la partie la plus importante de toute activité de réglage.

InnoDB s'appuie fortement sur le pool de mémoire tampon pour mettre en cache les index et ramener les données, l'index de hachage adaptatif, le tampon d'insertion et de nombreuses autres structures de données utilisées en interne. Le pool de mémoire tampon met également en mémoire tampon les modifications apportées aux données afin que les opérations d'écriture n'aient pas à être exécutées immédiatement sur le stockage, ce qui améliore les performances. Le pool de mémoire tampon fait partie intégrante d'InnoDB et sa taille doit être ajustée en conséquence. Tenez compte des facteurs suivants lors de la définition de la taille du pool de mémoire tampon :

- Pour une machine dédiée uniquement InnoDB, définissez la taille du pool de mémoire tampon sur 80 % ou plus de la mémoire RAM disponible.
- S'il ne s'agit pas d'un serveur dédié MySQL, définissez la taille sur 50 % de RAM.



## innodb\_flush\_method

Le paramètre `innodb_flush_method` indique comment InnoDB ouvre et vide les fichiers journaux et de données.

### Optimisations

Dans l'optimisation InnoDB, la définition de ce paramètre permet de régler les performances de la base de données, le cas échéant.

Les options suivantes permettent de vider les fichiers via InnoDB :

- `fsync`. InnoDB utilise le `fsync()` appel système pour vider les fichiers de données et les fichiers journaux. Cette option est le paramètre par défaut.
- `O_DSYNC`. InnoDB utilise le `O_DSYNC` option permettant d'ouvrir et de vider les fichiers journaux et `fsync()` pour vider les fichiers de données. InnoDB n'utilise pas `O_DSYNC` Directement, parce qu'il y a eu des problèmes avec elle sur de nombreuses variétés d'UNIX.
- `O_DIRECT`. InnoDB utilise le `O_DIRECT` option (ou `directio()` Sous Solaris) pour ouvrir les fichiers de données et les utilise `fsync()` pour vider les fichiers de données et les fichiers journaux. Cette option est disponible sur certaines versions de GNU/Linux, FreeBSD et Solaris.
- `O_DIRECT_NO_FSYNC`. InnoDB utilise le `O_DIRECT` Option pendant le vidage des E/S ; cependant, il ignore le `fsync()` appel système par la suite. Cette option n'est pas adaptée à certains types de systèmes de fichiers (par exemple, XFS). Si vous n'êtes pas sûr que votre système de fichiers nécessite un `fsync()` l'appel système, par exemple pour conserver toutes les métadonnées de fichier, utilise le `O_DIRECT` à la place.

### Observation

Dans les tests de laboratoire NetApp, le `fsync` L'option par défaut a été utilisée sur NFS et SAN, et il s'agissait d'un outil d'amélioration des performances par rapport à `O_DIRECT`. Lors de l'utilisation de la méthode de rinçage comme `O_DIRECT` Avec ONTAP, nous avons observé que le client écrit beaucoup d'écritures sur un seul octet à la frontière du bloc 4096 en série. Ces écritures ont augmenté la latence sur le réseau et dégradé les performances.

## innodb\_io\_capacity

Dans le plug-in InnoDB, un nouveau paramètre appelé `innodb_io_Capacity` a été ajouté à partir de MySQL 5.7.

Il contrôle le nombre maximal d'IOPS qu'InnoDB exécute (qui inclut la vitesse de vidage des pages sales ainsi que la taille de lot du tampon d'insertion [ibuf]). Le paramètre `innodb_io_Capacity` définit une limite supérieure sur les IOPS par les tâches d'arrière-plan InnoDB, telles que le vidage des pages du pool de mémoire tampon et la fusion des données à partir du tampon de changement.

Définissez le paramètre `innodb_io_Capacity` sur le nombre approximatif d'opérations d'E/S que le système peut effectuer par seconde. Idéalement, maintenez le paramètre aussi bas que possible, mais pas si bas que les activités en arrière-plan ralentissent. Si le paramètre est trop élevé, les données sont supprimées du pool de mémoire tampon et la mémoire tampon est insérée trop rapidement pour que la mise en cache offre un avantage significatif.



**NetApp recommande** que si vous utilisez ce paramètre sur NFS, analysez le résultat du test d'IOPS (SysBench/FiO) et définissez le paramètre en conséquence. Utilisez la plus petite valeur possible pour le vidage et la purge pour continuer à fonctionner, sauf si vous voyez plus de pages modifiées ou sales que vous le souhaitez dans le pool de mémoire tampon InnoDB.



N'utilisez pas de valeurs extrêmes telles que 20,000 ou plus, sauf si vous avez prouvé que des valeurs inférieures ne suffisent pas à votre charge de travail.

Le paramètre `InnoDB_IO_Capacity` régule les débits de rinçage et les E/S associées



Vous pouvez sérieusement nuire aux performances en définissant ce paramètre ou le paramètre `innodb_io_Capacity_max` trop élevé et gaspillé

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

Le `innodb_lru_scan_depth` Le paramètre influence les algorithmes et les heuristiques de l'opération de vidage pour le pool de mémoire tampon InnoDB.

Ce paramètre intéresse principalement les experts en performances qui s'intéressent au réglage des charges de travail exigeantes en E/S. Pour chaque instance de pool de mémoire tampon, ce paramètre indique la distance vers le bas dans la liste de pages LRU (least recently used) que le thread de nettoyage de page doit poursuivre la numérisation, en recherchant les pages sales à vider. Cette opération d'arrière-plan est effectuée une fois par seconde.

Vous pouvez régler la valeur vers le haut ou vers le bas pour réduire le nombre de pages libres. Ne définissez pas la valeur beaucoup plus haut que nécessaire, car les analyses peuvent avoir un coût de performance important. Pensez également à ajuster ce paramètre lors de la modification du nombre d'instances de pool de mémoire tampon, car `innodb_lru_scan_depth * innodb_buffer_pool_instances` définit la quantité de travail effectuée par le thread de nettoyage de page chaque seconde.

Un paramètre inférieur à celui par défaut convient à la plupart des workloads. Envisagez d'augmenter la valeur uniquement si vous disposez d'une capacité d'E/S disponible pour une charge de travail classique. Inversement, si une charge de travail exigeante en écriture sature votre capacité d'E/S, diminuez la valeur, en particulier si vous disposez d'un pool de mémoire tampon important.

## **open\_file\_limits**

Le `open_file_limits` paramètre détermine le nombre de fichiers que le système d'exploitation autorise à ouvrir mysqld.

La valeur de ce paramètre au moment de l'exécution est la valeur réelle autorisée par le système et peut être différente de la valeur spécifiée au démarrage du serveur. La valeur est 0 sur les systèmes où MySQL ne peut pas modifier le nombre de fichiers ouverts. L'efficace `open_files_limit` la valeur est basée sur la valeur spécifiée au démarrage du système (le cas échéant) et sur les valeurs de `max_connections` et `table_open_cache` en utilisant ces formules :

- $10 + \text{max\_connections} + (\text{table\_open\_cache} \times 2)$
- $\text{max\_connections} \times 5$
- Limite du système d'exploitation si positif

- Si la limite du système d'exploitation est infinie : `open_files_limit` la valeur est spécifiée au démarrage ; 5,000 si aucune

Le serveur tente d'obtenir le nombre de descripteurs de fichier en utilisant le maximum de ces quatre valeurs. Si ce nombre de descripteurs ne peut pas être obtenu, le serveur tente d'obtenir autant que le système le permet.

## Configuration de l'hôte

### Conteneurisation

La conteneurisation des bases de données MySQL est de plus en plus répandue.

La gestion des conteneurs de faible niveau est presque toujours effectuée via Docker. Les plateformes de gestion de conteneurs comme OpenShift et Kubernetes simplifient encore la gestion des grands environnements de conteneurs. La conteneurisation présente des avantages à moindre coût, car il n'est pas nécessaire de posséder une licence pour un hyperviseur. De plus, les conteneurs permettent à plusieurs bases de données de s'exécuter isolées les unes des autres tout en partageant le même noyau et le même système d'exploitation sous-jacents. Les conteneurs sont provisionnés en quelques microsecondes.

NetApp propose Astra Trident pour fournir des fonctionnalités de gestion avancées du stockage. Par exemple, Astra Trident permet à un conteneur créé dans Kubernetes de provisionner automatiquement son stockage sur le Tier approprié, d'appliquer des règles d'exportation, de définir des règles Snapshot et même de cloner un conteneur vers un autre. Pour plus d'informations, reportez-vous à la ["Documentation Astra Trident"](#).

### Tables d'emplacements NFSv3

Les performances de NFSv3 sous Linux dépendent d'un paramètre appelé

`tcp_max_slot_table_entries`.

Les tables d'emplacements TCP sont l'équivalent NFSv3 de la profondeur de file d'attente de l'adaptateur de bus hôte (HBA). Ces tableaux contrôlent le nombre d'opérations NFS qui peuvent être en attente à la fois. La valeur par défaut est généralement 16, un chiffre bien trop faible pour assurer des performances optimales. Le problème inverse se produit sur les noyaux Linux plus récents : la limite de la table des emplacements TCP augmente automatiquement par envoi de demandes, jusqu'à atteindre le niveau de saturation du serveur NFS.

Pour des performances optimales et pour éviter les problèmes de performances, ajustez les paramètres du noyau qui contrôlent les tables d'emplacements TCP.

Exécutez le `sysctl -a | grep tcp.*.slot_table` et observez les paramètres suivants :

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

Tous les systèmes Linux doivent inclure `sunrpc.tcp_slot_table_entries`, mais seulement certains incluent `sunrpc.tcp_max_slot_table_entries`. Ils doivent tous deux être réglés sur 128.



Si vous ne définissez pas ces paramètres, vous risquez d'avoir des effets importants sur les performances. Dans certains cas, les performances sont limitées car le système d'exploitation linux n'émet pas suffisamment d'E/S. Dans d'autres cas, les latences d'E/S augmentent à mesure que le système d'exploitation linux tente d'émettre plus d'E/S que ce qui peut être traité.

## Planificateurs d'E/S.

Le noyau Linux permet un contrôle de bas niveau sur la façon dont les E/S sont planifiées pour bloquer les périphériques.

Les valeurs par défaut sur les différentes distributions de Linux varient considérablement. MySQL vous recommande d'utiliser `NOOP` ou un `deadline` Planificateur d'E/S avec E/S asynchrones natives (AIO) sous Linux. De manière générale, les clients NetApp et les tests internes montrent de meilleurs résultats avec NoOps.

Le moteur de stockage InnoDB de MySQL utilise le sous-système d'E/S asynchrone (AIO natif) sur Linux pour effectuer des demandes de lecture et d'écriture pour les pages de fichiers de données. Ce comportement est contrôlé par le `innodb_use_native_aio` option de configuration, activée par défaut. Avec le tout-en-un natif, le type de planificateur d'E/S a une plus grande influence sur les performances E/S. Menez des bancs d'essai pour déterminer quel planificateur d'E/S offre les meilleurs résultats pour votre charge de travail et votre environnement.

Consultez la documentation Linux et MySQL appropriée pour obtenir des instructions sur la configuration du planificateur d'E/S.

## Descripteurs de fichier

Pour s'exécuter, le serveur MySQL a besoin de descripteurs de fichier et les valeurs par défaut ne sont pas suffisantes.

Il les utilise pour ouvrir de nouvelles connexions, stocker des tables dans le cache, créer des tables temporaires pour résoudre des requêtes complexes et accéder à des requêtes persistantes. Si `mysqld` n'est pas en mesure d'ouvrir de nouveaux fichiers lorsque cela est nécessaire, il peut arrêter de fonctionner correctement. Un symptôme courant de ce problème est l'erreur 24, "trop de fichiers ouverts". Le nombre de descripteurs de fichier que `mysqld` peut ouvrir simultanément est défini par le `open_files_limit` option définie dans le fichier de configuration (`/etc/my.cnf`). Mais `open_files_limit` dépend également des limites du système d'exploitation. Cette dépendance complique la définition de la variable.

MySQL ne peut pas définir son `open_files_limit` option supérieure à celle spécifiée sous `ulimit 'open files'`. Par conséquent, vous devez définir explicitement ces limites au niveau du système d'exploitation pour permettre à MySQL d'ouvrir des fichiers si nécessaire. Il existe deux façons de vérifier la limite de fichiers sous Linux :

- Le `ulimit` commande vous donne rapidement une description détaillée des paramètres autorisés ou verrouillés. Les modifications apportées par l'exécution de cette commande ne sont pas permanentes et seront effacées après un redémarrage du système.
- Modifications apportées au `/etc/security/limit.conf` les fichiers sont permanents et ne sont pas affectés par un redémarrage du système.

Assurez-vous de modifier les limites matérielles et logicielles de l'utilisateur `mysql`. Les extraits suivants sont issus de la configuration :

```
mysql hard nfile 65535
mysql soft nfile 65353
```

En parallèle, mettez à jour la même configuration dans `my.cnf` pour utiliser pleinement les limites de fichiers ouverts.

## Configuration de stockage sous-jacente

### NFS

La documentation MySQL recommande d'utiliser NFSv4 pour les déploiements NAS.

#### Tailles de transfert NFS ONTAP

Par défaut, ONTAP limite les tailles d'E/S NFS à 64 Ko. Les E/S aléatoires avec une base de données MySQL utilisent une taille de bloc bien inférieure à la taille maximale de 64 Ko. Les E/S de bloc volumineux sont généralement parallélisées de sorte que le maximum de 64 000 ne constitue pas non plus une limitation.

Dans certains cas, le maximum de 64 000 charges de travail entraîne une limitation. En particulier, les opérations à thread unique, telles que les opérations de sauvegarde d'analyse de table complète, s'exécuteront plus rapidement et plus efficacement si la base de données peut exécuter moins d'E/S, mais de plus grande taille. La taille optimale de gestion des E/S pour ONTAP avec charges de travail de base de données est de 256 Ko. Les options de montage NFS répertoriées pour les systèmes d'exploitation spécifiques ci-dessous ont été mises à jour de 64 Ko à 256 Ko en conséquence.

La taille maximale de transfert pour un SVM ONTAP donné peut être modifiée comme suit :

```
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
```



Ne diminuez jamais la taille de transfert maximale autorisée sur ONTAP en dessous de la valeur de `rsize/wsize` des systèmes de fichiers NFS actuellement montés. Cela peut provoquer des blocages ou même une corruption des données avec certains systèmes d'exploitation. Par exemple, si les clients NFS sont actuellement définis sur une taille `rsize/wsize` de 65536, la taille maximale du transfert ONTAP peut être ajustée entre 65536 et 1048576 sans effet car les clients eux-mêmes sont limités. Réduire la taille de transfert maximale en dessous de 65536 peut endommager la disponibilité ou les données.

## NetApp recommande



Définition du paramètre NFSv4 fstab (/etc/fstab) suivant :

```
nfs4 rw,  
hard,nointr,bg,vers=4,proto=tcp,noatime,rsiz=262144,wsiz=262144
```



NFSv3 présentait fréquemment un problème de verrouillage des fichiers journaux InnoDB après une panne de courant. L'utilisation du temps ou la commutation des fichiers journaux a résolu ce problème. Cependant, NFSv4 dispose d'opérations de verrouillage et assure le suivi des fichiers ouverts et des délégations.

## SAN

Les bases de données plus petites peuvent être placées sur une paire de LUN standard tant que les besoins en E/S et en capacité se situent dans les limites d'un seul système de fichiers LUN. Par exemple, une base de données qui nécessite environ 2 000 IOPS aléatoires peut être hébergée sur un système de fichiers unique sur une seule LUN. De même, une base de données de 100 Go seulement serait prise en charge sur une seule LUN sans problème de gestion.

Les bases de données plus grandes nécessitent plusieurs LUN. Par exemple, une base de données qui nécessite 100 000 IOPS aurait probablement besoin d'au moins huit LUN. Un seul LUN deviendrait un goulot d'étranglement en raison du nombre insuffisant de canaux SCSI vers les disques. Une base de données de 10 To serait tout aussi difficile à gérer sur un seul LUN de 10 To. Les gestionnaires de volumes logiques sont conçus pour lier les performances et les capacités de plusieurs LUN afin d'améliorer les performances et la gestion.

Dans les deux cas, une paire de volumes ONTAP doit suffire. Dans une configuration simple, le LUN du fichier de données serait placé dans un volume dédié, tout comme le LUN du journal. Avec une configuration de gestionnaire de volumes logique, toutes les LUN du groupe de volumes de fichiers de données se trouvent dans un volume dédié, et les LUN du groupe de volumes de logs se trouvent dans un second volume dédié.

**NetApp recommande** l'utilisation de deux systèmes de fichiers pour les déploiements MySQL sur SAN :

- Le premier système de fichiers stocke toutes les données MySQL, y compris l'espace table, les données et l'index.
- Le second système de fichiers stocke tous les journaux (journaux binaires, journaux lents et journaux des transactions).



Il existe plusieurs raisons de séparer les données de cette manière :

- Les modèles d'E/S des fichiers de données et des fichiers journaux diffèrent. De les séparer, on disposerait d'un plus grand nombre d'options avec les contrôles de QoS.
- Pour optimiser l'utilisation de la technologie Snapshot, vous devez pouvoir restaurer les fichiers de données de manière indépendante. La connexion de fichiers de données avec des fichiers journaux interfère avec la restauration des fichiers de données.
- La technologie SnapMirror de NetApp peut être utilisée pour fournir une fonctionnalité de reprise d'activité simple à faible RPO pour une base de données. Toutefois, le planning de réplication des fichiers de données et des journaux doit être différent.



Utilisez cette disposition de base à deux volumes pour pérenniser votre solution et utiliser toutes les fonctionnalités ONTAP si nécessaire.

**NetApp recommande** de formater votre lecteur avec le système de fichiers ext4 en raison des fonctionnalités suivantes :



- Approche étendue des fonctions de gestion des blocs utilisées dans le système de fichiers de journalisation (JFS) et fonctions d'allocation différée du système de fichiers étendu (XFS).
- Ext4 autorise des systèmes de fichiers allant jusqu'à 1 exbioctet ( $2^{60}$  octets) et des fichiers allant jusqu'à 16 tébioctets ( $16 * 2^{40}$  octets). En revanche, le système de fichiers ext3 ne prend en charge qu'une taille de système de fichiers maximale de 16 To et une taille de fichier maximale de 2 To.
- Dans les systèmes de fichiers ext4, l'allocation multi-blocs (mballoc) alloue plusieurs blocs pour un fichier en une seule opération, au lieu de les allouer un par un, comme dans ext3. Cette configuration réduit la surcharge liée à l'appel de l'ALlocator de bloc plusieurs fois et optimise l'allocation de mémoire.
- Bien que XFS soit la valeur par défaut pour de nombreuses distributions Linux, il gère les métadonnées différemment et ne convient pas à certaines configurations MySQL.



**NetApp recommande** d'utiliser les options de taille de bloc de 4 ko avec l'utilitaire mkfs pour l'aligner avec la taille de LUN de bloc existante.

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

Les LUN NetApp stockent les données dans des blocs physiques de 4 Ko, ce qui produit huit blocs logiques de 512 octets.

Si vous ne configurez pas la même taille de bloc, les E/S ne seront pas alignées avec les blocs physiques correctement et pourraient écrire sur deux disques différents dans un groupe RAID, ce qui entraîne une latence.



Il est important d'aligner les E/S pour que les opérations de lecture/écriture soient fluides. Cependant, lorsque les E/S commencent au niveau d'un bloc logique qui n'est pas au début d'un bloc physique, les E/S sont mal alignées. Les opérations d'E/S ne sont alignées que lorsqu'elles commencent au niveau d'un bloc logique, le premier bloc logique d'un bloc physique.

## Informations sur le copyright

Copyright © 2026 NetApp, Inc. Tous droits réservés. Imprimé aux États-Unis. Aucune partie de ce document protégé par copyright ne peut être reproduite sous quelque forme que ce soit ou selon quelque méthode que ce soit (graphique, électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système de récupération électronique) sans l'autorisation écrite préalable du détenteur du droit de copyright.

Les logiciels dérivés des éléments NetApp protégés par copyright sont soumis à la licence et à l'avis de non-responsabilité suivants :

CE LOGICIEL EST FOURNI PAR NETAPP « EN L'ÉTAT » ET SANS GARANTIES EXPRESSES OU TACITES, Y COMPRIS LES GARANTIES TACITES DE QUALITÉ MARCHANDE ET D'ADÉQUATION À UN USAGE PARTICULIER, QUI SONT EXCLUES PAR LES PRÉSENTES. EN AUCUN CAS NETAPP NE SERA TENU POUR RESPONSABLE DE DOMMAGES DIRECTS, INDIRECTS, ACCESSOIRES, PARTICULIERS OU EXEMPLAIRES (Y COMPRIS L'ACHAT DE BIENS ET DE SERVICES DE SUBSTITUTION, LA PERTE DE JOUISSANCE, DE DONNÉES OU DE PROFITS, OU L'INTERRUPTION D'ACTIVITÉ), QUELLES QU'EN SOIENT LA CAUSE ET LA DOCTRINE DE RESPONSABILITÉ, QU'IL S'AGISSE DE RESPONSABILITÉ CONTRACTUELLE, STRICTE OU DÉLICTELLE (Y COMPRIS LA NÉGLIGENCE OU AUTRE) DÉCOULANT DE L'UTILISATION DE CE LOGICIEL, MÊME SI LA SOCIÉTÉ A ÉTÉ INFORMÉE DE LA POSSIBILITÉ DE TELS DOMMAGES.

NetApp se réserve le droit de modifier les produits décrits dans le présent document à tout moment et sans préavis. NetApp décline toute responsabilité découlant de l'utilisation des produits décrits dans le présent document, sauf accord explicite écrit de NetApp. L'utilisation ou l'achat de ce produit ne concède pas de licence dans le cadre de droits de brevet, de droits de marque commerciale ou de tout autre droit de propriété intellectuelle de NetApp.

Le produit décrit dans ce manuel peut être protégé par un ou plusieurs brevets américains, étrangers ou par une demande en attente.

**LÉGENDE DE RESTRICTION DES DROITS :** L'utilisation, la duplication ou la divulgation par le gouvernement sont sujettes aux restrictions énoncées dans le sous-paragraphe (b)(3) de la clause Rights in Technical Data-Noncommercial Items du DFARS 252.227-7013 (février 2014) et du FAR 52.227-19 (décembre 2007).

Les données contenues dans les présentes se rapportent à un produit et/ou service commercial (tel que défini par la clause FAR 2.101). Il s'agit de données propriétaires de NetApp, Inc. Toutes les données techniques et tous les logiciels fournis par NetApp en vertu du présent Accord sont à caractère commercial et ont été exclusivement développés à l'aide de fonds privés. Le gouvernement des États-Unis dispose d'une licence limitée irrévocable, non exclusive, non cessible, non transférable et mondiale. Cette licence lui permet d'utiliser uniquement les données relatives au contrat du gouvernement des États-Unis d'après lequel les données lui ont été fournies ou celles qui sont nécessaires à son exécution. Sauf dispositions contraires énoncées dans les présentes, l'utilisation, la divulgation, la reproduction, la modification, l'exécution, l'affichage des données sont interdits sans avoir obtenu le consentement écrit préalable de NetApp, Inc. Les droits de licences du Département de la Défense du gouvernement des États-Unis se limitent aux droits identifiés par la clause 252.227-7015(b) du DFARS (février 2014).

## Informations sur les marques commerciales

NETAPP, le logo NETAPP et les marques citées sur le site <http://www.netapp.com/TM> sont des marques déposées ou des marques commerciales de NetApp, Inc. Les autres noms de marques et de produits sont des marques commerciales de leurs propriétaires respectifs.