PostgreSQL

Enterprise applications

NetApp
January 02, 2026

This PDF was generated from https://docs.netapp.com/fr-fr/ontap-apps-dbs/postgres/postgres-overview.html on January 02, 2026. Always check docs.netapp.com for the latest.

Sommaire

PostgreSQL . . . 1
- Présentation . . . 1
- Configuration de la base de données . . . 1
  - Architecture . . . 1
  - Paramètres d’initialisation . . . 2
  - Paramètres . . . 3
  - Espaces de stockage . . . 5
- Configuration de stockage sous-jacente . . . 5
  - NFS . . . 5
  - SAN . . . 7
- Protection des données . . . 9
  - Protection Ddta native . . . 9
  - Snapshots . . . 10
  - Logiciels de protection des données . . . 11

PostgreSQL

Présentation

PostgreSQL est fourni avec des variantes incluant PostgreSQL, PostgreSQL plus et EDB Postgres Advanced Server (EPAS). PostgreSQL est généralement déployé en tant que base de données interne pour les applications multiniveaux. Il est pris en charge par les logiciels middleware courants (tels que PHP, Java, Python, Tcl/TK, ODBC, Et JDBC) et a toujours été un choix populaire pour les systèmes de gestion de bases de données open source. ONTAP constitue un excellent choix pour l'exécution des bases de données PostgreSQL et ses fonctionnalités de gestion des données fiables, performantes et efficaces.

Cette documentation sur ONTAP et la base de données PostgreSQL remplace la base de données *TR-4770: PostgreSQL sur les meilleures pratiques ONTAP.*

Avec la croissance exponentielle des données, la gestion des données devient de plus en plus complexe pour les entreprises. Cette complexité augmente les coûts de licence, d'exploitation, de support et de maintenance. Pour réduire le coût total de possession, envisagez de passer de bases de données commerciales à des bases de données open source grâce à un stockage interne fiable et haute performance.

ONTAP est une plateforme idéale, 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.

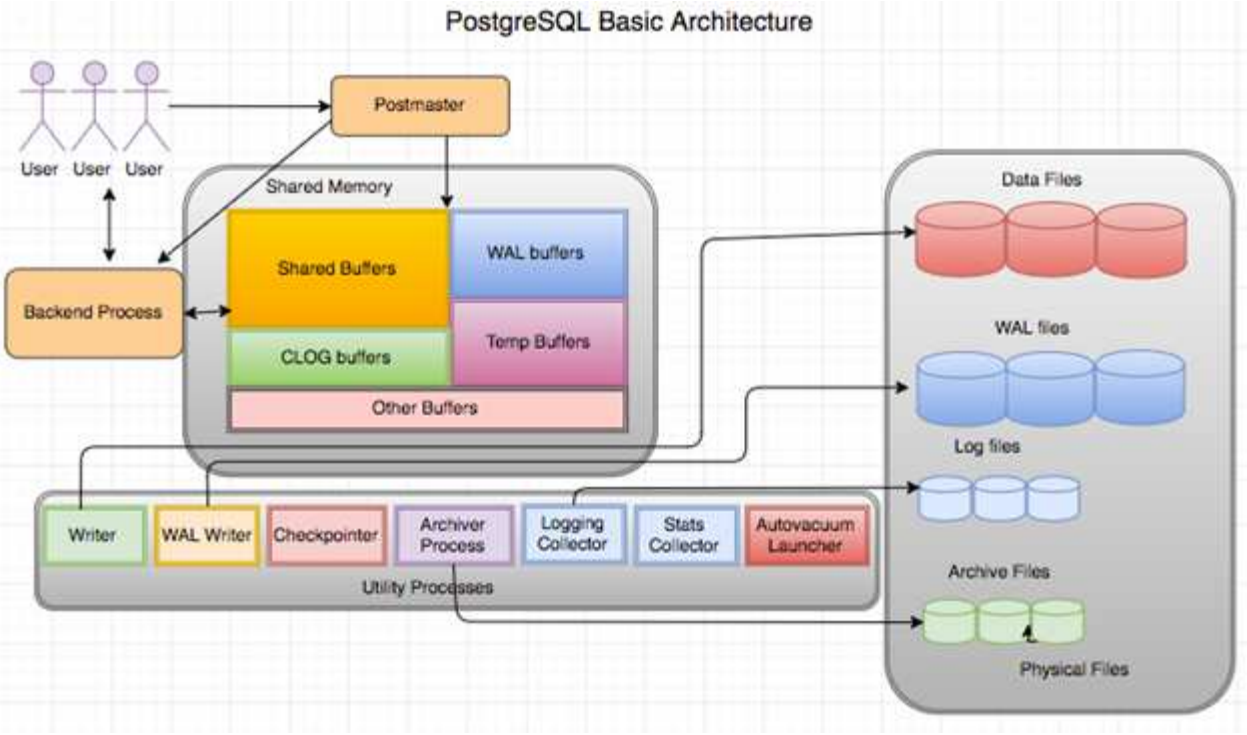
Des fonctionnalités supplémentaires, telles que les mises à niveau sans interruption (y compris le remplacement du stockage), assurent la disponibilité de vos bases de données stratégiques. Vous pouvez également bénéficier d'une reprise après incident instantanée pour les environnements volumineux via 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 ajuster la taille de votre solution de stockage à l'aide d'un contrôleur plus petit, qui exécute toujours le même système d'exploitation du stockage.

Configuration de la base de données

Architecture

PostgreSQL est un SGBDR basé sur l'architecture client et serveur. Une instance PostgreSQL est appelée cluster de base de données, qui est une collection de bases de données par opposition à une collection de serveurs.



Il existe trois éléments principaux dans une base de données PostgreSQL : le postmaster, le front end (client) et le back end Le client envoie des demandes au postmaster avec des informations telles que le protocole IP et la base de données à laquelle se connecter. Le postmaster authentifie la connexion et la transmet au processus d'arrière-plan pour une communication plus poussée. Le processus back-end exécute la requête et envoie les résultats directement au frontal (client).

Une instance PostgreSQL est basée sur un modèle multiprocessus au lieu d'un modèle multithread. Il génère plusieurs processus pour différents travaux, et chaque processus possède sa propre fonctionnalité. Les principaux processus incluent le processus client, le processus WAL writer, le processus Background writer et le processus checkpointer :

- Lorsqu'un processus client (premier plan) envoie des demandes de lecture ou d'écriture à l'instance PostgreSQL, il ne lit pas ou n'écrit pas les données directement sur le disque. Il met d'abord en mémoire tampon les données dans des tampons partagés et des tampons WAL (Write-Ahead Logging).
- Un processus WAL writer manipule le contenu des tampons partagés et des tampons WAL pour écrire dans les journaux WAL. Les journaux WAL sont généralement des journaux de transaction de PostgreSQL et sont écrits de manière séquentielle. Par conséquent, pour améliorer le temps de réponse de la base de données, PostgreSQL écrit d'abord dans les journaux de transactions et reconnaît le client.
- Pour mettre la base de données dans un état cohérent, le processus de l'enregistreur d'arrière-plan vérifie périodiquement la présence de pages sales dans le tampon partagé. Il purge ensuite les données sur les fichiers de données stockés sur des volumes NetApp ou des LUN.
- Le processus CheckPointer s'exécute également périodiquement (moins fréquemment que le processus d'arrière-plan) et empêche toute modification des tampons. Il signale au processus d'écriture WAL d'écrire et de vider l'enregistrement de point de contrôle à la fin des journaux WAL stockés sur le disque NetApp. Il signale également au processus d'écriture d'arrière-plan d'écrire et de vider toutes les pages sales sur le disque.

Paramètres d'initialisation

Vous créez un nouveau cluster de base de données à l'aide de `initdb` programme. An

`initdb` script crée les fichiers de données, les tables système et les bases de données modèles (template0 et template1) qui définissent le cluster.

La base de données de modèles représente une base de données de stock. Il contient des définitions pour les tables système, les vues standard, les fonctions et les types de données. `pgdata` sert d'argument à l' `initdb` script qui spécifie l'emplacement du cluster de base de données.

Tous les objets de base de données dans PostgreSQL sont gérés en interne par les OID respectives. Les tables et les index sont également gérés par des OID individuelles. Les relations entre les objets de base de données et leurs OID respectives sont stockées dans les tables de catalogue système appropriées, selon le type d'objet. Par exemple, les OID des bases de données et des tables de segment de mémoire sont stockées dans `pg_database` et `pg_class, respectivement. Vous pouvez déterminer les OID en émettant des requêtes sur le client PostgreSQL.

Chaque base de données a ses propres tables et fichiers d'index qui sont limités à 1 Go. Chaque table a deux fichiers associés, avec le suffixe respectivement `_fsm` et `_vm`. Ils sont appelés carte de l'espace libre et carte de visibilité. Ces fichiers stockent les informations relatives à la capacité d'espace libre et ont une visibilité sur chaque page du fichier de table. Les index ne disposent que de cartes d'espace libre individuelles et ne disposent pas de cartes de visibilité.

Le `pg_xlog/pg_wal` le répertoire contient les journaux d'écriture anticipée. Des journaux d'écriture anticipée sont utilisés pour améliorer la fiabilité et les performances de la base de données. Chaque fois que vous mettez à jour une ligne dans une table, PostgreSQL écrit d'abord la modification dans le journal d'écriture anticipée, puis écrit les modifications sur les pages de données réelles sur un disque. Le `pg_xlog` le répertoire contient généralement plusieurs fichiers, mais initdb ne crée que le premier. Des fichiers supplémentaires sont ajoutés si nécessaire. Chaque fichier xlog fait 16 Mo de long.

Paramètres

Il existe plusieurs configurations de réglage PostgreSQL qui peuvent améliorer les performances.

Les paramètres les plus utilisés sont les suivants :

- `max_connections = <num>`: Le nombre maximal de connexions de base de données à avoir en même temps. Utilisez ce paramètre pour limiter l'échange sur le disque et l'arrêt des performances. Selon les besoins de votre application, vous pouvez également régler ce paramètre pour les paramètres du pool de connexions.
- `shared_buffers = <num>`: La méthode la plus simple pour améliorer les performances de votre serveur de base de données. La valeur par défaut est faible pour la plupart des matériels modernes. Il est défini pendant le déploiement à environ 25 % de la RAM disponible sur le système. Ce paramètre varie en fonction de la façon dont il fonctionne avec des instances de base de données particulières ; vous devrez peut-être augmenter ou diminuer les valeurs par tâtonnement et erreur. Cependant, le réglage haut risque de dégrader les performances.
- `effective_cache_size = <num>`: Cette valeur indique à l'optimiseur de PostgreSQL la quantité de mémoire disponible pour la mise en cache des données et aide à déterminer si un index doit être utilisé. Une valeur plus élevée augmente la probabilité d'utiliser un index. Ce paramètre doit être défini sur la quantité de mémoire allouée à `shared_buffers` Plus la quantité de cache du système d'exploitation disponible. Cette valeur représente souvent plus de 50 % de la mémoire système totale.
- `work_mem = <num>`: Ce paramètre contrôle la quantité de mémoire à utiliser dans les opérations de tri et les tables de hachage. Si vous effectuez un tri important dans votre application, vous devrez peut-être augmenter la quantité de mémoire, mais soyez prudent. Ce n'est pas un paramètre à l'échelle du système,

mais un paramètre par opération. Si une requête complexe comporte plusieurs opérations de tri, elle utilise plusieurs unités de mémoire Work_mem et plusieurs back end peuvent le faire simultanément. Cette requête peut souvent amener votre serveur de base de données à échanger si la valeur est trop élevée. Cette option était auparavant appelée sort_mem dans les anciennes versions de PostgreSQL.

- `fsync = <boolean> (on or off)` : Ce paramètre détermine si toutes vos pages WAL doivent être synchronisées sur le disque à l'aide de fsync() avant qu'une transaction ne soit validée. Sa désactivation peut parfois améliorer les performances d'écriture et son activation renforce la protection contre le risque de corruption en cas de panne du système.
- `checkpoint_timeout`: Le processus de point de contrôle vide les données validées sur le disque. Cela implique de nombreuses opérations de lecture/écriture sur le disque. La valeur est définie en secondes et les valeurs inférieures réduisent le temps de reprise après incident et l'augmentation des valeurs peut réduire la charge sur les ressources système en réduisant les appels au point de contrôle. En fonction de la criticité de l'application, de l'utilisation et de la disponibilité de la base de données, définissez la valeur de Checkpoint_timeout.
- `commit_delay = <num>` et `commit_siblings = <num>`: Ces options sont utilisées ensemble pour aider à améliorer les performances en écrivant plusieurs transactions qui sont exécutées simultanément. Si plusieurs objets commit_frames sont actifs à l'instant où votre transaction est validée, le serveur attend les microsecondes commit_delay pour essayer de valider plusieurs transactions à la fois.
- `max_worker_processes / max_parallel_workers`: Configurer le nombre optimal de travailleurs pour les processus. Max_Parallel_workers correspond au nombre de CPU disponibles. Selon la conception de l'application, les requêtes peuvent nécessiter un nombre réduit de collaborateurs pour les opérations parallèles. Il est préférable de conserver la même valeur pour les deux paramètres, mais d'ajuster la valeur après le test.
- `random_page_cost = <num>`: Cette valeur contrôle la façon dont PostgreSQL affiche les lectures de disque non séquentielles. Une valeur plus élevée signifie que PostgreSQL est plus susceptible d'utiliser une analyse séquentielle au lieu d'une analyse d'index, indiquant que votre serveur a des disques rapides Modifier ce paramètre après avoir évalué d'autres options telles que l'optimisation basée sur un plan, l'aspiration, l'indexation pour modifier les requêtes ou le schéma.
- `effective_io_concurrency = <num>`: Ce paramètre définit le nombre d'opérations d'E/S de disque simultanées que PostgreSQL tente d'exécuter simultanément. L'augmentation de cette valeur augmente le nombre d'opérations d'E/S que toute session PostgreSQL individuelle tente d'initier en parallèle. La plage autorisée est comprise entre 1 et 1,000, ou zéro pour désactiver l'émission de demandes d'E/S asynchrones. Actuellement, ce paramètre n'affecte que les analyses de tas bitmap. Les disques SSD et les autres systèmes de stockage basés sur la mémoire (NVMe) peuvent souvent traiter un grand nombre de requêtes simultanées. Le meilleur choix peut donc se situer dans les centaines.

Consultez la documentation PostgreSQL pour obtenir une liste complète des paramètres de configuration PostgreSQL.

TOASTS

TOAST est l'acronyme de Oversized-Attribute Storage technique. PostgreSQL utilise une taille de page fixe (généralement 8 Ko) et ne permet pas aux blocs de données de couvrir plusieurs pages. Par conséquent, il n'est pas possible de stocker directement des valeurs de champ importantes. Lorsque vous essayez de stocker une ligne qui dépasse cette taille, TOAST divise les données de grandes colonnes en « morceaux » plus petits et les stocke dans une table de TOASTS.

Les grandes valeurs des attributs toastés sont extraites (si elles sont sélectionnées) uniquement au moment où le jeu de résultats est envoyé au client. La table elle-même est beaucoup plus petite et peut contenir plus de lignes dans le cache du tampon partagé qu'elle ne le pouvait sans stockage hors ligne (TOAST).

VIDE

En mode PostgreSQL normal, les blocs de données supprimés ou rendus obsolètes par une mise à jour ne sont pas physiquement supprimés de leur table ; ils restent présents jusqu'à ce que LE VIDE soit exécuté. Par conséquent, vous devez faire fonctionner le VIDE régulièrement, en particulier sur les tables fréquemment mises à jour. L'espace qu'il occupe doit ensuite être récupéré pour réutilisation par de nouvelles lignes, afin d'éviter une panne d'espace disque. Cependant, il ne renvoie pas l'espace vers le système d'exploitation.

L'espace libre dans une page n'est pas fragmenté. VIDE réécrit le bloc entier, en empaquant efficacement les lignes restantes et en laissant un seul bloc contigu d'espace libre dans une page.

En revanche, LE VIDE COMPLET composera activement les tables en écrivant une version complètement nouvelle du fichier table sans espace mort. Cette action réduit la taille de la table mais peut prendre un certain temps. Elle nécessite également de l'espace disque supplémentaire pour la nouvelle copie de la table jusqu'à ce que l'opération soit terminée. L'objectif du VIDE DE routine est d'éviter toute activité de VIDE COMPLET. Ce processus permet non seulement de conserver les tables à leur taille minimale, mais également de conserver une utilisation régulière de l'espace disque.

Espaces de stockage

Deux tablespaces sont créés automatiquement lorsque le cluster de base de données est initialisé.

Le `pg_global` l'espace table est utilisé pour les catalogues système partagés. Le `pg_default` tablespace est l'espace table par défaut des bases de données templage1 et template0. Si la partition ou le volume sur lequel le cluster a été initialisé est à court d'espace et ne peut pas être étendu, un espace table peut être créé sur une partition différente et utilisé jusqu'à ce que le système puisse être reconfiguré.

Un index très utilisé peut être placé sur un disque rapide et hautement disponible, comme un périphérique SSD. Par ailleurs, une table qui stocke des données archivées rarement utilisées ou non critiques pour les performances peut être stockée sur un système sur disque moins onéreux et plus lent, tel que des disques SAS ou SATA.

Les tablespaces font partie du cluster de base de données et ne peuvent pas être traités comme un ensemble autonome de fichiers de données. Elles dépendent des métadonnées contenues dans le répertoire de données principal et ne peuvent donc pas être reliées à un autre cluster de base de données ou sauvegardées individuellement. De même, si vous perdez un espace de table (suite à la suppression d'un fichier, à une panne de disque, etc.), le cluster de base de données peut devenir illisible ou ne pas démarrer. Le fait de placer un tablespace sur un système de fichiers temporaire, tel qu'un disque RAM, risque de nuire à la fiabilité de l'ensemble du cluster.

Une fois créé, un espace table peut être utilisé à partir de n'importe quelle base de données si l'utilisateur demandeur dispose de privilèges suffisants. PostgreSQL utilise des liens symboliques pour simplifier l'implémentation des tablespaces. PostgreSQL ajoute une ligne au `pg_tablespace` Tableau (table à l'échelle du cluster) et attribue un nouvel identifiant d'objet (OID) à cette ligne. Enfin, le serveur utilise l'OID pour créer un lien symbolique entre votre cluster et le répertoire donné. Le répertoire `$PGDATA/pg_tblspc` contient des liens symboliques pointant vers chacun des tablespaces non intégrés définis dans le cluster.

Configuration de stockage sous-jacente

NFS

Les bases de données PostgreSQL peuvent être hébergées sur les systèmes de fichiers

NFSv3 ou NFSv4. La meilleure option dépend de facteurs extérieurs à la base de données.

Par exemple, le comportement de verrouillage NFSv4 peut être préférable dans certains environnements en cluster. (Voir "ici" pour plus d'informations)

Dans le cas contraire, les fonctionnalités de la base de données doivent être proches des mêmes, y compris les performances. La seule exigence est l'utilisation du `hard` option de montage. Ceci est nécessaire pour garantir que les délais d'expiration ne produisent pas d'erreurs d'E/S irrécupérables.

Si NFSv4 est choisi en tant que protocole, NetApp recommande d'utiliser NFSv4.1. Certaines améliorations fonctionnelles du protocole NFSv4 dans NFSv4.1 améliorent la résilience sur NFSv4.0.

Utilisez les options de montage suivantes pour les charges de travail de base de données générales :

```
rw,hard,nointr,bg,vers=[3|4],proto=tcp,rsize=65536,wsize=65536
```

Si des E/S séquentielles lourdes sont attendues, la taille du transfert NFS peut être augmentée comme décrit dans la section suivante.

Tailles de transfert NFS

Par défaut, ONTAP limite la taille des E/S NFS à 64 Ko.

Les E/S aléatoires utilisent la plupart des applications et bases de données une taille de bloc bien inférieure à la taille maximale de 64 Ko. Les E/S de blocs volumineux sont généralement parallélisées de sorte que le maximum de 64 Ko ne limite pas non plus l'obtention d'une bande passante maximale.

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 ou de restauration, ou encore les analyses de table complète de base de données s'exécutent plus rapidement et plus efficacement si la base de données peut exécuter moins d'E/S, mais plus volumineuses. La taille optimale de gestion des E/S pour ONTAP est de 256 Ko.

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
Cluster01::*>
```

Ne réduisez 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.

Une fois la taille de transfert augmentée au niveau ONTAP, les options de montage suivantes sont utilisées :

```
rw,hard,nointr,bg,vers=[3|4],proto=tcp,rsize=262144,wsize=262144
```

Tables d'emplacements TCP NFSv3

Si NFSv3 est utilisé avec Linux, il est essentiel de définir correctement les tables d'emplacements TCP.

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.

SAN

Les bases de données PostgreSQL avec SAN sont généralement hébergées sur des systèmes de fichiers xfs, mais d'autres peuvent être utilisées si elles sont prises en charge par le fournisseur du système d'exploitation

Même si un seul LUN peut généralement prendre en charge jusqu'à 100 000 IOPS, les bases de données exigeantes en E/S nécessitent généralement l'utilisation de LVM avec répartition.

Marquage LVM

Avant l'ère des disques Flash, la répartition était utilisée pour surmonter les limites de performances des disques rotatifs. Par exemple, si un système d'exploitation doit effectuer une opération de lecture de 1 Mo, la lecture de ce 1 Mo de données à partir d'un seul disque demande beaucoup de tête de lecture lorsque le transfert des 1 Mo est lent. Si ce 1 Mo de données a été réparti sur 8 LUN, le système d'exploitation pourrait exécuter huit opérations de lecture de 128 K en parallèle et réduire le temps nécessaire au transfert de 1 Mo.

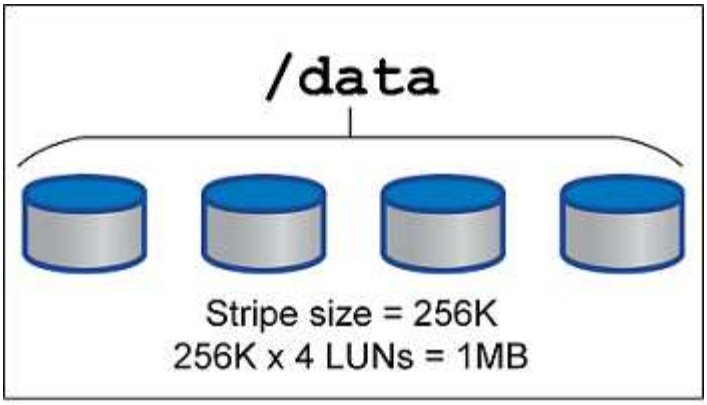
Le striping avec des disques rotatifs était plus difficile, car le modèle d'E/S devait être connu à l'avance. Si la répartition n'a pas été correctement réglée pour les véritables modèles d'E/S, les configurations à bandes risquent d'endommager les performances. Avec les bases de données Oracle, et en particulier les configurations 100 % Flash, le striping est beaucoup plus facile à configurer et a fait ses preuves pour améliorer considérablement les performances.

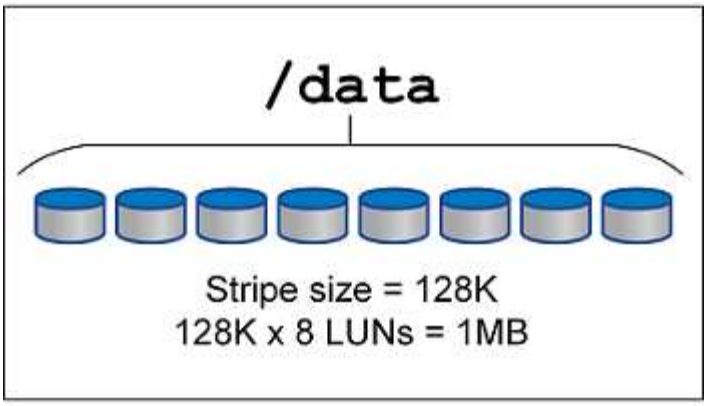
Par défaut, les gestionnaires de volumes logiques, tels que la bande Oracle ASM, ne le font pas pour le système d'exploitation natif LVM. Certaines lient plusieurs LUN ensemble en tant que périphérique concaténé. Résultat : des fichiers de données existent sur un seul périphérique LUN. Ceci provoque des points chauds. Les autres implémentations LVM prennent par défaut en charge les extensions distribuées. Cette méthode est similaire à la répartition, mais elle est plus grossière. Les LUN du groupe de volumes sont tranchées en grandes parties, appelées extensions et généralement mesurées en plusieurs mégaoctets. Ensuite, les volumes logiques sont distribués sur ces extensions. Il en résulte des E/S aléatoires sur un fichier qui doit être bien réparti entre les LUN, mais les opérations d'E/S séquentielles ne sont pas aussi efficaces qu'elles pourraient l'être.

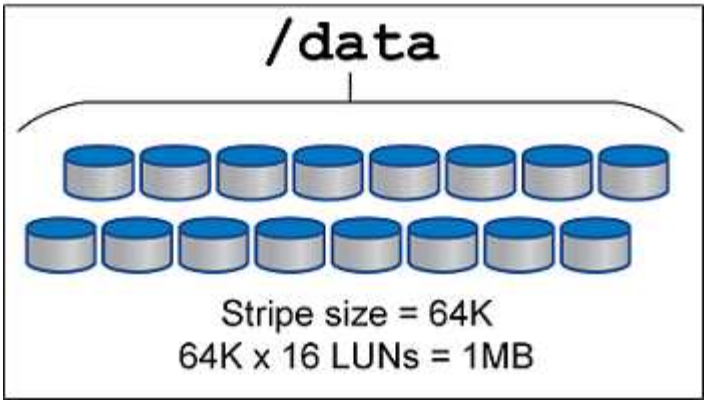
Les E/S des applications exigeantes en performances sont presque toujours de (a) en unités de taille de bloc de base ou (b) d'un mégaoctet.

L'objectif principal d'une configuration à bandes est de s'assurer que les E/S de fichier unique peuvent être exécutées comme une seule unité, et que les E/S de plusieurs blocs, d'une taille de 1 Mo, peuvent être parallélisées de façon homogène sur toutes les LUN du volume réparti. Cela signifie que la taille de bande ne doit pas être inférieure à la taille du bloc de base de données, et que la taille de bande multipliée par le nombre de LUN doit être de 1 Mo.

La figure suivante présente trois options possibles pour le réglage de la taille et de la largeur des bandes. Le nombre de LUN est sélectionné pour répondre aux exigences de performances comme décrit ci-dessus, mais dans tous les cas, le total des données dans une seule bande est de 1 Mo.







Protection des données

Protection Ddta native

L'un des principaux aspects de la conception du stockage est l'activation de la protection pour les volumes PostgreSQL. Les clients peuvent protéger leurs bases de données PostgreSQL en utilisant l'approche dump ou en utilisant des sauvegardes de système de fichiers. Cette section décrit les différentes approches de sauvegarde de bases de données individuelles ou de l'ensemble du cluster.

Il existe trois approches de sauvegarde des données PostgreSQL :

- Dump SQL Server
- Sauvegarde au niveau du système de fichiers

- Archivage continu

L'idée derrière la méthode de vidage de SQL Server est de générer un fichier avec des commandes SQL Server qui, une fois renvoyées au serveur, peuvent recréer la base de données telle qu'elle était au moment de la sauvegarde. PostgreSQL fournit les programmes utilitaires `pg_dump` et `pg_dump_all` pour la création de sauvegardes individuelles et au niveau du cluster. Ces vidages sont logiques et ne contiennent pas suffisamment d'informations pour être utilisés par la relecture WAL.

Une autre stratégie de sauvegarde consiste à utiliser une sauvegarde au niveau du système de fichiers, dans laquelle les administrateurs copient directement les fichiers utilisés par PostgreSQL pour stocker les données dans la base de données. Cette méthode s'effectue en mode hors ligne : la base de données ou le cluster doit être arrêté. Une autre alternative est d'utiliser `pg_basebackup` Pour exécuter une sauvegarde de diffusion à chaud de la base de données PostgreSQL.

Snapshots

Les sauvegardes basées sur des copies Snapshot avec PostgreSQL requièrent la configuration de snapshots pour les fichiers de données, les fichiers WAL et les fichiers WAL archivés afin d'assurer une restauration complète ou instantanée.

Pour les bases de données PostgreSQL, la durée moyenne de sauvegarde avec des copies Snapshot est comprise entre quelques secondes et quelques minutes. Cette vitesse de sauvegarde est 60 à 100 fois plus rapide que `pg_basebackup` et d'autres approches de sauvegarde basées sur le système de fichiers.

Les copies Snapshot situées sur un système de stockage NetApp peuvent être à la fois cohérentes après panne et cohérentes au niveau des applications. Un snapshot cohérent après panne est créé sur un système de stockage sans interrompre la base de données, tandis qu'un Snapshot cohérent avec les applications est créé lorsque la base de données est en mode de sauvegarde. NetApp garantit également que les copies Snapshot suivantes sont des sauvegardes incrémentielles à l'infini pour promouvoir les économies de stockage et l'efficacité réseau.

Comme les snapshots sont rapides et n'affectent pas les performances du système, vous pouvez planifier plusieurs copies Snapshot chaque jour au lieu de créer une sauvegarde quotidienne comme avec les autres technologies de sauvegarde en streaming. Lorsqu'une opération de restauration et de restauration est nécessaire, le temps d'interruption du système est réduit grâce à deux fonctionnalités clés :

- Avec la technologie de restauration des données NetApp SnapRestore, la restauration s'exécute en quelques secondes.
- Les objectifs de point de restauration (RPO) agressifs signifient qu'il faut moins de journaux de base de données et que la restauration par progression est également accélérée.

Pour sauvegarder PostgreSQL, vous devez vous assurer que les volumes de données sont protégés simultanément avec WAL (groupe de cohérence) et les journaux archivés. Lorsque vous utilisez la technologie Snapshot pour copier des fichiers WAL, assurez-vous de les exécuter `pg_stop` Pour vider toutes les entrées WAL qui doivent être archivées. Si vous videz les entrées WAL pendant la restauration, il vous suffit d'arrêter la base de données, de démonter ou de supprimer le répertoire de données existant et d'effectuer une opération SnapRestore sur le stockage. Une fois la restauration terminée, vous pouvez monter le système et le ramener à son état actuel. Pour la restauration instantanée, vous pouvez également restaurer les journaux WAL et d'archivage ; PostgreSQL décide alors du point le plus cohérent et le récupère automatiquement.

Les groupes de cohérence sont une fonctionnalité de ONTAP recommandée lorsque plusieurs volumes sont montés sur une seule instance ou une base de données avec plusieurs tablespaces. Une copie Snapshot de groupe de cohérence garantit que tous les volumes sont regroupés et protégés. Pour gérer efficacement un

groupe de cohérence, ONTAP vous pouvez même le cloner et créer une copie d'instance d'une base de données à des fins de test ou de développement.

Logiciels de protection des données

Le plug-in NetApp SnapCenter pour les bases de données PostgreSQL, associé aux technologies Snapshot et NetApp FlexClone, vous offre les avantages suivants :

- Sauvegarde et restauration rapides.
- Clones compacts.
- La possibilité de mettre en place un système de reprise d'activité rapide et efficace.

> Vous pouvez choisir les partenaires de sauvegarde premium de NetApp, tels que Veeam Software et CommVault dans les circonstances suivantes :
>
> - Gestion des workloads dans un environnement hétérogène
> - Stocker les sauvegardes dans le cloud ou sur bande pour les conserver à long terme
> - Prise en charge d'un large éventail de versions et de types de systèmes d'exploitation

Le plug-in SnapCenter pour PostgreSQL est un plug-in pris en charge par la communauté et la configuration et la documentation sont disponibles sur le magasin d'automatisation NetApp. Grâce à SnapCenter, l'utilisateur peut sauvegarder la base de données, cloner et restaurer les données à distance.

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 DÉLICTUELLE (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.