



MySQL

Enterprise applications

NetApp
January 02, 2026

Índice

MySQL	1
Visão geral	1
Configuração do banco de dados	1
Estrutura de ficheiros	1
Parâmetros de configuração	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
Configuração de host	8
Conteinerização	9
NFSv3 mesas de slot	9
Agendadores de e/S	9
Descritores de arquivo	10
Configuração de armazenamento	10
NFS	10
SAN	11

MySQL

Visão geral

MySQL e suas variantes, incluindo MariaDB e Percona MySQL, é o banco de dados mais popular do mundo.



Esta documentação sobre o ONTAP e o banco de dados MySQL substitui o banco de dados *TR-4722: MySQL publicado anteriormente sobre as melhores práticas do ONTAP*.

ONTAP é uma plataforma ideal para banco de dados MySQL porque o ONTAP é literalmente projetado para bancos de dados. Vários recursos, como otimizações aleatórias de latência de e/S para qualidade avançada de serviço (QoS) e funcionalidade básica do FlexClone, foram criados especificamente para atender às necessidades dos workloads de banco de dados.

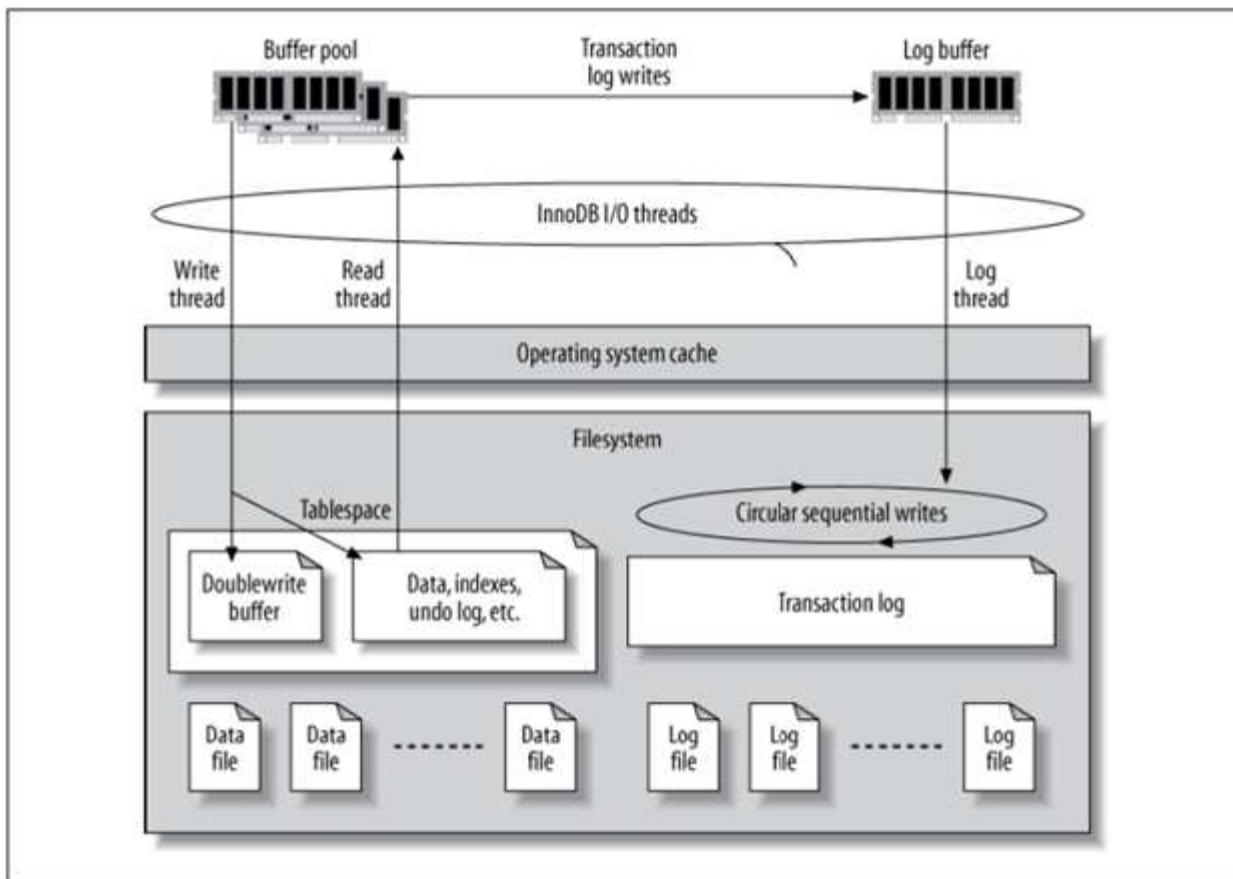
Recursos adicionais, como atualizações sem interrupções, (incluindo substituição de storage) garantem que seus bancos de dados essenciais permaneçam disponíveis. Você também pode ter recuperação instantânea de desastres para ambientes grandes por meio do MetroCluster ou selecionar bancos de dados usando o SnapMirror active Sync.

Mais importante ainda, o ONTAP oferece desempenho incomparável com a capacidade de dimensionar a solução para suas necessidades exclusivas. Nossos sistemas high-end podem fornecer mais de 1M IOPS com latências medidas em microssegundos. No entanto, se você só precisa de 100K IOPS, é possível dimensionar corretamente sua solução de storage com uma controladora menor que ainda executa o mesmo sistema operacional de storage.

Configuração do banco de dados

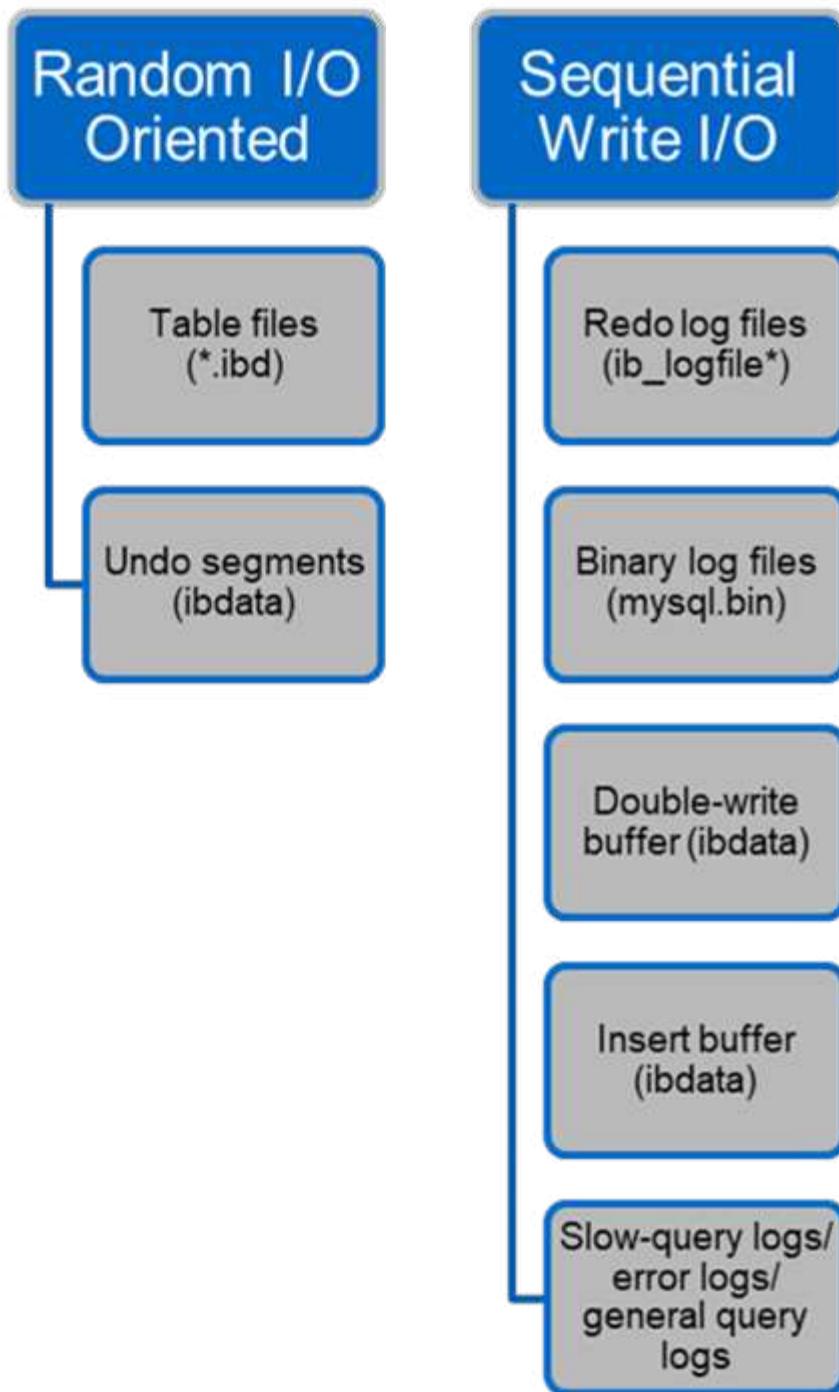
Estrutura de ficheiros

InnoDB atua como a camada intermediária entre o armazenamento e o servidor MySQL, ele armazena os dados nas unidades.



MySQL I/o é categorizado em dois tipos:

- E/S de arquivo aleatório
- E/S de arquivo sequencial



Os arquivos de dados são lidos e substituídos aleatoriamente, o que resulta em IOPS alto. Portanto, o armazenamento SSD é recomendado.

Os arquivos de log de refazer e os arquivos de log binários são logs transacionais. Eles são escritos sequencialmente, para que você possa obter um bom desempenho em HDD com o cache de gravação. Uma leitura sequencial acontece na recuperação, mas raramente causa um problema de desempenho, porque o tamanho do arquivo de log geralmente é menor do que os arquivos de dados, e as leituras sequenciais são mais rápidas do que as leituras aleatórias (que ocorrem em arquivos de dados).

O buffer de gravação dupla é um recurso especial do InnoDB. O InnoDB primeiro grava páginas apagadas no buffer de gravação dupla e, em seguida, grava as páginas em suas posições corretas nos arquivos de dados.

Este processo impede a corrupção de página. Sem o buffer de gravação dupla, a página pode ficar corrompida se ocorrer uma falha de energia durante o processo de gravação em unidades. Como a gravação no buffer de gravação dupla é sequencial, ele é altamente otimizado para HDDs. Leituras sequenciais ocorrem na recuperação.

Como o ONTAP NVRAM já fornece proteção contra gravação, o buffer de gravação dupla não é necessário. O MySQL tem um parâmetro, `skip_innodb_doublewrite`, para desativar o buffer de gravação dupla. Esse recurso pode melhorar substancialmente o desempenho.

O buffer de inserção também é um recurso especial do InnoDB. Se blocos de índice secundários não exclusivos não estiverem na memória, o InnoDB insere entradas no buffer de inserção para evitar operações de e/S aleatórias. Periodicamente, o buffer de inserção é mesclado nas árvores de índice secundárias no banco de dados. O buffer de inserção reduz o número de operações de e/S mesclando solicitações de e/S para o mesmo bloco; operações de e/S aleatórias podem ser sequenciais. O buffer de inserção também é altamente otimizado para HDDs. As gravações e leituras sequenciais ocorrem durante operações normais.

Os segmentos de desfazer são orientados por e/S aleatórias. Para garantir a simultaneidade de múltiplas versões (MVCC), o InnoDB deve registrar imagens antigas nos segmentos de desfazer. A leitura de imagens anteriores dos segmentos de desfazer requer leituras aleatórias. Se você executar uma transação longa com leituras repetíveis (como mysqldump - transação única) ou executar uma consulta longa, leituras aleatórias podem ocorrer. Portanto, armazenar segmentos de desfazer em SSDs é melhor nessa instância. Se você executar apenas transações curtas ou consultas, as leituras aleatórias não são um problema.

A NetApp recomenda o seguinte layout de design de armazenamento devido às características de e/S do InnoDB.



- Um volume para armazenar arquivos aleatórios e sequenciais orientados por e/S do MySQL
- Outro volume para armazenar arquivos puramente sequenciais orientados por e/S do MySQL

Esse layout também ajuda a projetar políticas e estratégias de proteção de dados.

Parâmetros de configuração

O NetApp recomenda alguns parâmetros de configuração importantes do MySQL para obter um desempenho ideal.

Parâmetros	Valores
<code>innodb_log_file_size</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
<code>open_file_limit</code>	65535

Para definir os parâmetros descritos nesta seção, você deve alterá-los no arquivo de configuração do MySQL (my.cnf). As melhores práticas da NetApp são resultado de testes realizados internamente.

innodb_log_file_size

Selecionar o tamanho certo para o tamanho do arquivo de log InnoDB é importante para as operações de gravação e para ter um tempo de recuperação decente após uma falha do servidor.

Como muitas transações são registradas no arquivo, o tamanho do arquivo de log é importante para operações de gravação. Quando os Registros são modificados, a alteração não é imediatamente gravada de volta para o espaço da tabela. Em vez disso, a alteração é gravada no final do arquivo de log e a página é marcada como suja. InnoDB usa seu log para converter a e/S aleatória em e/S sequencial

Quando o log está cheio, a página suja é escrita para a tablespace em sequência para liberar espaço no arquivo de log. Por exemplo, suponha que um servidor trava no meio de uma transação, e as operações de gravação são registradas apenas no arquivo de log. Antes que o servidor possa voltar ao vivo, ele deve passar por uma fase de recuperação na qual as alterações registradas no arquivo de log são reproduzidas. Quanto mais entradas estiverem no arquivo de log, mais tempo leva para que o servidor se recupere.

Neste exemplo, o tamanho do arquivo afeta tanto o tempo de recuperação quanto o desempenho de gravação. Ao escolher o número certo para o tamanho do arquivo de log, equilibre o tempo de recuperação em relação ao desempenho de gravação. Normalmente, qualquer coisa entre 128M e 512M é um bom valor.

innodb_flush_log_at_trx_commit

Quando há uma alteração nos dados, a alteração não é imediatamente gravada no armazenamento.

Em vez disso, os dados são gravados em um buffer de log, que é uma parte da memória que o InnoDB atribui a alterações de buffer que são registradas no arquivo de log. O InnoDB limpa o buffer para o arquivo de log quando uma transação é confirmada, quando o buffer fica cheio, ou uma vez por segundo, qualquer evento acontece primeiro. A variável de configuração que controla este processo é `innodb_flush_log_at_trx_commit`. As opções de valor incluem:

- Quando você define `innodb_flush_log_trx_at_commit=0`, o InnoDB grava os dados modificados (no pool de buffers do InnoDB) no arquivo de log (ib_logfile) e limpa o arquivo de log (gravação no armazenamento) a cada segundo. No entanto, não faz nada quando a transação é cometida. Se houver uma falha de energia ou falha do sistema, nenhum dos dados não lavados é recuperável porque não é gravado no arquivo de log ou nas unidades.
- Quando você define `'innodb_flush_log_trx_commit=1'`, o InnoDB grava o buffer de log no log de transações e passa para o armazenamento durável para cada transação. Por exemplo, para todos os commits de transação, o InnoDB grava no log e depois grava no armazenamento. O armazenamento mais lento afeta negativamente o desempenho; por exemplo, o número de transações InnoDB por segundo é reduzido.
- Quando você define `innodb_flush_log_trx_commit=2`, o InnoDB grava o buffer de log no arquivo de log em cada commit; no entanto, ele não grava dados no armazenamento. O InnoDB limpa os dados uma vez a cada segundo. Mesmo que haja uma falha de energia ou falha do sistema, os dados da opção 2 estão disponíveis no arquivo de log e são recuperáveis.

Se o desempenho for o objetivo principal, defina o valor como 2. Como o InnoDB grava nas unidades uma vez por segundo, não para cada commit de transação, o desempenho melhora drasticamente. Se ocorrer uma

falta de energia ou falha, os dados podem ser recuperados do log de transações.

Se a segurança de dados for o objetivo principal, defina o valor como 1 para que, para cada commit de transação, o InnoDB seja enviado para as unidades. No entanto, o desempenho pode ser afetado.



NetApp recomenda defina o valor innodb_flush_log_trx_commit como 2 para um melhor desempenho.

innodb_doublewrite

`'innodb_doublewrite'` Quando está ativado (o padrão), o InnoDB armazena todos os dados duas vezes: Primeiro para o buffer de gravação dupla e depois para os arquivos de dados reais.

Você pode desativar esse parâmetro com `--skip-innodb_doublewrite` para benchmarks ou quando estiver mais preocupado com o desempenho superior do que com a integridade de dados ou possíveis falhas. InnoDB usa uma técnica de flush de arquivo chamada double-write. Antes de gravar páginas nos arquivos de dados, o InnoDB as grava em uma área contígua chamada buffer de gravação dupla. Depois que a gravação e o flush para o buffer de gravação dupla estiverem completos, o InnoDB grava as páginas em suas posições apropriadas no arquivo de dados. Se o sistema operacional ou um processo mysqld falhar durante uma gravação de página, o InnoDB poderá encontrar uma boa cópia da página do buffer de gravação dupla durante a recuperação de falhas.



A NetApp recomenda a desativação do buffer de gravação dupla. O ONTAP NVRAM serve a mesma função. O buffer duplo danificará desnecessariamente o desempenho.

innodb_buffer_pool_size

O pool de buffers InnoDB é a parte mais importante de qualquer atividade de ajuste.

O InnoDB depende muito do pool de buffers para armazenar em cache índices e remendar os dados, o índice de hash adaptativo, o buffer de inserção e muitas outras estruturas de dados usadas internamente. O pool de buffers também armazena alterações nos dados para que as operações de gravação não precisem ser executadas imediatamente no armazenamento, melhorando assim o desempenho. O buffer pool é parte integrante do InnoDB e seu tamanho deve ser ajustado em conformidade. Considere os seguintes fatores ao definir o tamanho do pool de buffers:

- Para uma máquina exclusiva InnoDB dedicada, defina o tamanho do pool de buffers para 80% ou mais da RAM disponível.
- Se não for um servidor dedicado MySQL, defina o tamanho para 50% da RAM.

innodb_flush_method

O parâmetro innodb_flush_method especifica como o InnoDB abre e limpa os arquivos de log e dados.

Otimizações

Na otimização InnoDB, a configuração desse parâmetro ajusta o desempenho do banco de dados quando aplicável.

As seguintes opções são para limpar os arquivos através do InnoDB:

- `fsync`. O InnoDB usa a `fsync()` chamada do sistema para liberar os arquivos de dados e log. Esta opção é a predefinição.
- `O_DSYNC`. InnoDB usa a `O_DSYNC` opção para abrir e lavar os arquivos de log e `fsync()` para lavar os arquivos de dados. InnoDB não usa `O_DSYNC` diretamente, porque houve problemas com ele em muitas variedades de UNIX.
- `O_DIRECT`. O InnoDB usa a `O_DIRECT` opção (ou `directio()` no Solaris) para abrir os arquivos de dados e usa `fsync()` para limpar os arquivos de dados e log. Esta opção está disponível em algumas versões GNU/Linux, FreeBSD e Solaris.
- `O_DIRECT_NO_FSYNC`. O InnoDB usa a `O_DIRECT` opção durante a lavagem de e/S; no entanto, ele ignora a `fsync()` chamada do sistema depois. Essa opção não é adequada para alguns tipos de sistemas de arquivos (por exemplo, XFS). Se você não tiver certeza se o sistema de arquivos requer uma `fsync()` chamada de sistema, por exemplo, para preservar todos os metadados do arquivo, use a `O_DIRECT` opção.

Observação

Nos testes de laboratório do NetApp, a `fsync` opção padrão foi usada em NFS e SAN, e foi um ótimo improvisador de desempenho em comparação com `O_DIRECT`. Ao usar o método flush como `O_DIRECT` no ONTAP, observamos que o cliente escreve muitas gravações de byte único na borda do bloco 4096 em modo serial. Essas gravações aumentam a latência na rede e degradam o desempenho.

innodb_io_capacity

No plug-in InnoDB, um novo parâmetro chamado `innodb_io_Capacity` foi adicionado a partir do MySQL 5.7.

Ele controla o número máximo de IOPS que o InnoDB executa (que inclui a taxa de limpeza de páginas sujas, bem como o tamanho do lote do buffer de inserção [ibuf]). O parâmetro `innodb_io_Capacity` define um limite superior em IOPS por tarefas de fundo do InnoDB, como limpar páginas do pool de buffers e mesclar dados do buffer de mudança.

Defina o parâmetro `innodb_io_Capacity` para o número aproximado de operações de e/S que o sistema pode executar por segundo. Idealmente, mantenha a configuração o mais baixa possível, mas não tão baixa que as atividades de fundo desacelerem. Se a configuração for muito alta, os dados serão removidos do pool de buffers e inserirão o buffer muito rapidamente para que o armazenamento em cache forneça um benefício significativo.

 **A NetApp recomenda** que, se estiver usando essa configuração em NFS, analise o resultado do teste de IOPS (SysBench/fio) e defina o parâmetro de acordo. Use o menor valor possível para flushing e purging para manter-se a menos que você veja mais páginas modificadas ou sujas do que você deseja no pool de buffers InnoDB.

 Não use valores extremos como 20.000 ou mais, a menos que você tenha provado que valores mais baixos não são suficientes para sua carga de trabalho.

O parâmetro `InnoDB_io_Capacity` regula as taxas de lavagem e e/S relacionadas



Você pode prejudicar seriamente o desempenho definindo este parâmetro ou o parâmetro innodb_io_Capacity_Max muito alto e wastin

innodb_lru_scan_depth

O `innodb_lru_scan_depth` parâmetro influencia os algoritmos e heurísticas da operação de flush para o pool de buffers InnoDB.

Esse parâmetro é do interesse principal dos especialistas em performance que ajustam workloads com uso intenso de e/S. Para cada instância de pool de buffer, este parâmetro especifica o quanto abaixo na lista de páginas menos usadas recentemente (LRU) o thread de limpeza de páginas deve continuar a digitalizar, procurando páginas sujas para serem flush. Esta operação em segundo plano é executada uma vez por segundo.

Você pode ajustar o valor para cima ou para baixo para minimizar o número de páginas livres. Não defina o valor muito maior do que o necessário, porque as digitalizações podem ter um custo de desempenho significativo. Além disso, considere ajustar esse parâmetro ao alterar o número de instâncias do pool de buffers, `innodb_lru_scan_depth * innodb_buffer_pool_instances` porque define a quantidade de trabalho executada pelo thread de limpeza de páginas a cada segundo.

Uma configuração menor do que o padrão é adequada para a maioria das cargas de trabalho. Considere aumentar o valor somente se você tiver capacidade de e/S sobressalente em uma carga de trabalho típica. Por outro lado, se uma carga de trabalho com uso intenso de gravação saturar a capacidade de e/S, diminua o valor, especialmente se você tiver um grande pool de buffers.

open_file_limits

O `open_file_limits` parâmetro determina o número de arquivos que o sistema operacional permite que o mysqld abra.

O valor deste parâmetro em tempo de execução é o valor real permitido pelo sistema e pode ser diferente do valor especificado na inicialização do servidor. O valor é 0 em sistemas onde o MySQL não pode alterar o número de arquivos abertos. O valor efetivo `open_files_limit` é baseado no valor especificado na inicialização do sistema (se houver) e nos valores de `max_connections` e `table_open_cache` usando estas fórmulas:

- $10 \text{ max_connections} (\text{'table_open_cache}' \times 2)$
- $\text{max_connections} \times 5$
- Limite do sistema operacional se positivo
- Se o limite do sistema operacional for infinito: `open_files_limit` O valor é especificado na inicialização; 5.000 se nenhum

O servidor tenta obter o número de descritores de arquivo usando o máximo desses quatro valores. Se muitos descritores não puderem ser obtidos, o servidor tentará obter o máximo que o sistema permitir.

Configuração de host

Conteinerização

Conteinerização de bancos de dados MySQL está se tornando mais prevalente.

O gerenciamento de contentores de baixo nível é quase sempre realizado através do Docker. As plataformas de gerenciamento de contêineres, como OpenShift e Kubernetes, simplificam ainda mais o gerenciamento de ambientes de contêineres grandes. Os benefícios da Conteinerização incluem custos mais baixos, porque não há necessidade de licenciar um hypervisor. Além disso, os containers permitem que vários bancos de dados sejam executados isolados uns dos outros enquanto compartilham o mesmo kernel subjacente e sistema operacional. Os contêineres podem ser provisionados em microsegundos.

A NetApp oferece o Astra Trident para fornecer funcionalidades avançadas de gerenciamento de storage. Por exemplo, o Astra Trident permite que um contêiner criado no Kubernetes provisione automaticamente seu storage na camada apropriada, aplique políticas de exportação, defina políticas de snapshot e até clone um contêiner para outro. Para obter informações adicionais, consulte o "["Documentação do Astra Trident"](#)".

NFSv3 mesas de slot

O desempenho do NFSv3 no Linux depende de um parâmetro `tcp_max_slot_table_entries` chamado .

As tabelas de slot TCP são equivalentes a NFSv3 mm de profundidade de fila do adaptador de barramento do host (HBA). Essas tabelas controlam o número de operações NFS que podem ficar pendentes de uma só vez. O valor padrão é geralmente 16, o que é muito baixo para um desempenho ideal. O problema oposto ocorre em kernels Linux mais recentes, que podem aumentar automaticamente o limite da tabela de slots TCP para um nível que satura o servidor NFS com solicitações.

Para um desempenho ideal e para evitar problemas de desempenho, ajuste os parâmetros do kernel que controlam as tabelas de slots TCP.

Executar o `sysctl -a | grep tcp.*.slot_table` comando e respeitar os seguintes parâmetros:

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

Todos os sistemas Linux devem incluir `sunrpc.tcp_slot_table_entries`, mas apenas alguns incluem `sunrpc.tcp_max_slot_table_entries`. Ambos devem ser definidos para 128.



A falha em definir esses parâmetros pode ter efeitos significativos no desempenho. Em alguns casos, o desempenho é limitado porque o sistema operacional linux não está emitindo e/S suficiente. Em outros casos, as latências de e/S aumentam à medida que o sistema operacional linux tenta emitir mais e/S do que pode ser reparado.

Agendadores de e/S.

O kernel Linux permite um controle de baixo nível sobre a maneira como e/S para bloquear dispositivos é agendada.

Os padrões em várias distribuições do Linux variam consideravelmente. O MySQL recomenda que você use

NOOP ou um deadline agendador de e/S com e/S assíncrona nativa (AIO) no Linux. Em geral, os clientes da NetApp e os testes internos mostram melhores resultados com NoOps.

O mecanismo de armazenamento InnoDB do MySQL usa o subsistema de e/S assíncrono (AIO nativo) no Linux para executar solicitações de leitura e gravação para páginas de arquivos de dados. Esse comportamento é controlado pela `innodb_use_native_aio` opção de configuração, que é ativada por padrão. Com AIO nativo, o tipo de agendador de e/S tem maior influência no desempenho de e/S. Realize benchmarks para determinar qual agendador de e/S fornece os melhores resultados para sua carga de trabalho e ambiente.

Consulte a documentação relevante do Linux e MySQL para obter instruções sobre como configurar o agendador de e/S.

Descriptores de arquivo

Para executar, o servidor MySQL precisa de descriptores de arquivo e os valores padrão não são suficientes.

Ele os usa para abrir novas conexões, armazenar tabelas no cache, criar tabelas temporárias para resolver consultas complicadas e acessar as persistentes. Se o mysqld não for capaz de abrir novos arquivos quando necessário, ele pode parar de funcionar corretamente. Um sintoma comum deste problema é o erro 24, "muitos arquivos abertos". O número de descriptores de arquivo que o mysqld pode abrir simultaneamente é definido pela `open_files_limit` opção definida no arquivo de configuração (`/etc/my.cnf`). Mas `open_files_limit` também depende dos limites do sistema operacional. Essa dependência torna a configuração da variável mais complicada.

O MySQL não pode definir sua `open_files_limit` opção mais alta do que a especificada em `ulimit 'open files'`. Portanto, você precisa definir explicitamente esses limites no nível do sistema operacional para permitir que o MySQL abra arquivos conforme necessário. Existem duas maneiras de verificar o limite de arquivos no Linux:

- O `ulimit` comando fornece rapidamente uma descrição detalhada dos parâmetros que estão sendo permitidos ou bloqueados. As alterações feitas ao executar este comando não são permanentes e serão apagadas após uma reinicialização do sistema.
- As alterações ao `/etc/security/limit.conf` ficheiro são permanentes e não são afetadas pela reinicialização do sistema.

Certifique-se de alterar os limites físicos e macios para o usuário mysql. Os seguintes trechos são da configuração:

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

Em paralelo, atualize a mesma configuração no `my.cnf` para usar totalmente os limites de arquivos abertos.

Configuração de armazenamento

NFS

A documentação do MySQL recomenda que você use o NFSv4 para implantações nas.

Tamanhos de transferência de NFS do ONTAP

Por padrão, o ONTAP limitará os tamanhos de e/S NFS a 64K. Random IO com um banco de dados MySQL usa um tamanho de bloco muito menor que está bem abaixo do máximo de 64KMB. O IO de bloco grande geralmente é paralelizado, portanto o máximo de 64K também não é uma limitação.

Existem algumas cargas de trabalho em que o máximo 64K cria uma limitação. Em particular, operações de um único processo, como operações de backup de verificação de tabela completa, serão executadas de forma mais rápida e eficiente se o banco de dados puder executar menos, mas maiores. O tamanho ideal de manuseio de e/S para ONTAP com cargas de trabalho de banco de dados é 256K. As opções de montagem NFS listadas para sistemas operacionais específicos abaixo foram atualizadas de 64K para 256K em conformidade.

O tamanho máximo de transferência para um determinado SVM do ONTAP pode ser alterado da seguinte forma:

```
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
```

 Nunca diminua o tamanho máximo de transferência permitido no ONTAP abaixo do valor de rsize/wsize dos sistemas de arquivos NFS atualmente instalados. Isso pode criar pendências ou até mesmo corrupção de dados com alguns sistemas operacionais. Por exemplo, se os clientes NFS estiverem atualmente definidos em um rsize/wsize de 65536, o tamanho máximo de transferência do ONTAP poderá ser ajustado entre 65536 e 1048576 sem efeito, porque os próprios clientes são limitados. Reduzir o tamanho máximo de transferência abaixo de 65536 pode danificar a disponibilidade ou os dados.

A NetApp recomenda

 Configurando a seguinte configuração NFSv4 fstab (/etc/fstab):

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

 Um problema comum com o NFSv3 foi os arquivos de log do InnoDB bloqueados após uma falha de energia. Usar o tempo ou trocar arquivos de log resolveu esse problema. No entanto, o NFSv4 tem operações de bloqueio e mantém o controle de arquivos abertos e delegações.

SAN

Bancos de dados menores podem ser colocados em um par de LUNs padrão, desde que as demandas de e/S e capacidade estejam dentro dos limites de um único sistema de arquivos LUN. Por exemplo, um banco de dados que requer aproximadamente 2K IOPS aleatório pode ser hospedado em um único sistema de arquivos

em um único LUN. Da mesma forma, um banco de dados com apenas 100GB MB de tamanho caberia em um único LUN sem criar um problema de gerenciamento.

Bancos de dados maiores exigem vários LUNs. Por exemplo, um banco de dados que requer 100K IOPS provavelmente precisará de pelo menos oito LUNs. Um único LUN se tornaria um gargalo devido ao número inadequado de canais SCSI para unidades. Um banco de dados 10TB seria igualmente difícil de gerenciar em um único LUN 10TB. Os gerenciadores lógicos de volume são projetados para unir os recursos de desempenho e capacidade de vários LUNs para melhorar o desempenho e a capacidade de gerenciamento.

Em ambos os casos, um par de volumes ONTAP deve ser suficiente. Com uma configuração simples, o LUN do arquivo de dados seria colocado em um volume dedicado, assim como o LUN de log. Com uma configuração lógica do gerenciador de volumes, todos os LUNs no grupo de volumes de arquivos de dados estariam em um volume dedicado e os LUNs do grupo de volumes de log estariam em um segundo volume dedicado.

A NetApp recomenda usando dois sistemas de arquivos para implantações MySQL na SAN:

- O primeiro sistema de arquivos armazena todos os dados MySQL, incluindo tablespace, dados e índice.
- O segundo sistema de arquivos armazena todos os logs (logs binários, logs lentos e logs de transações).

Existem várias razões para separar dados dessa maneira, incluindo:

- Os padrões de e/S de arquivos de dados e arquivos de log diferem. Separá-los permitiria mais opções com controles de QoS.
- O uso ideal da tecnologia Snapshot requer a capacidade de restaurar os arquivos de dados de forma independente. Commingling arquivos de dados com arquivos de log interfere com a restauração de arquivos de dados.
- A tecnologia NetApp SnapMirror pode ser usada para fornecer uma funcionalidade de recuperação de desastres simples e de baixo RPO para um banco de dados. No entanto, ela requer diferentes programações de replicação para arquivos e logs de dados.

i Use esse layout básico de dois volumes para preparar a solução para o futuro, de modo que todos os recursos do ONTAP possam ser usados, se necessário.

A NetApp recomenda a formatação da sua unidade com o sistema de arquivos ext4 devido aos seguintes recursos:

- Abordagem estendida aos recursos de gerenciamento de blocos usados no sistema de arquivos de journaling (JFS) e recursos de alocação atrasada do sistema de arquivos estendido (XFS).
- EXT4 permite sistemas de arquivos de até 1 exbibyte (2⁶⁰ bytes) e arquivos de até 16 tebibytes (16 * 2⁴⁰ bytes). Em contraste, o sistema de arquivos ext3 suporta apenas um tamanho máximo de sistema de arquivos de 16TB e um tamanho máximo de arquivo de 2TB.
- Em sistemas de arquivos ext4, a alocação de vários blocos (mballoc) aloca vários blocos para um arquivo em uma única operação, em vez de alocá-los um por um, como em ext3. Essa configuração reduz a sobrecarga de chamar o alocador de bloco várias vezes e otimiza a alocação de memória.
- Embora o XFS seja o padrão para muitas distribuições Linux, ele gerencia metadados de forma diferente e não é adequado para algumas configurações do MySQL.

A NetApp recomenda usar opções de tamanho de bloco 4K com o utilitário mkfs para alinhar com o tamanho de LUN de bloco existente.

`mkfs.ext4 -b 4096`

Os LUNs NetApp armazenam dados em 4KB blocos físicos, o que rende oito blocos lógicos de 512 bytes.

Se você não configurar o mesmo tamanho de bloco, a e/S não será alinhada com os blocos físicos corretamente e poderá gravar em duas unidades diferentes em um grupo RAID, resultando em latência.

i É importante alinhar a e/S para operações de leitura/gravação suaves. No entanto, quando a e/S começa em um bloco lógico que não está no início de um bloco físico, a e/S está desalinhada. As operações de e/S são alinhadas somente quando começam em um bloco lógico, o primeiro bloco lógico em um bloco físico.

Informações sobre direitos autorais

Copyright © 2026 NetApp, Inc. Todos os direitos reservados. Impresso nos EUA. Nenhuma parte deste documento protegida por direitos autorais pode ser reproduzida de qualquer forma ou por qualquer meio — gráfico, eletrônico ou mecânico, incluindo fotocópia, gravação, gravação em fita ou storage em um sistema de recuperação eletrônica — sem permissão prévia, por escrito, do proprietário dos direitos autorais.

O software derivado do material da NetApp protegido por direitos autorais está sujeito à seguinte licença e isenção de responsabilidade:

ESTE SOFTWARE É FORNECIDO PELA NETAPP "NO PRESENTE ESTADO" E SEM QUAISQUER GARANTIAS EXPRESSAS OU IMPLÍCITAS, INCLUINDO, SEM LIMITAÇÕES, GARANTIAS IMPLÍCITAS DE COMERCIALIZAÇÃO E ADEQUAÇÃO A UM DETERMINADO PROPÓSITO, CONFORME A ISENÇÃO DE RESPONSABILIDADE DESTE DOCUMENTO. EM HIPÓTESE ALGUMA A NETAPP SERÁ RESPONSÁVEL POR QUALQUER DANO DIRETO, INDIRETO, INCIDENTAL, ESPECIAL, EXEMPLAR OU CONSEQUENCIAL (INCLUINDO, SEM LIMITAÇÕES, AQUISIÇÃO DE PRODUTOS OU SERVIÇOS SOBRESSALENTES; PERDA DE USO, DADOS OU LUCROS; OU INTERRUPÇÃO DOS NEGÓCIOS), INDEPENDENTEMENTE DA CAUSA E DO PRINCÍPIO DE RESPONSABILIDADE, SEJA EM CONTRATO, POR RESPONSABILIDADE OBJETIVA OU PREJUÍZO (INCLUINDO NEGLIGÊNCIA OU DE OUTRO MODO), RESULTANTE DO USO DESTE SOFTWARE, MESMO SE ADVERTIDA DA RESPONSABILIDADE DE TAL DANO.

A NetApp reserva-se o direito de alterar quaisquer produtos descritos neste documento, a qualquer momento e sem aviso. A NetApp não assume nenhuma responsabilidade nem obrigação decorrentes do uso dos produtos descritos neste documento, exceto conforme expressamente acordado por escrito pela NetApp. O uso ou a compra deste produto não representam uma licença sob quaisquer direitos de patente, direitos de marca comercial ou quaisquer outros direitos de propriedade intelectual da NetApp.

O produto descrito neste manual pode estar protegido por uma ou mais patentes dos EUA, patentes estrangeiras ou pedidos pendentes.

LEGENDA DE DIREITOS LIMITADOS: o uso, a duplicação ou a divulgação pelo governo estão sujeitos a restrições conforme estabelecido no subparágrafo (b)(3) dos Direitos em Dados Técnicos - Itens Não Comerciais no DFARS 252.227-7013 (fevereiro de 2014) e no FAR 52.227- 19 (dezembro de 2007).

Os dados aqui contidos pertencem a um produto comercial e/ou serviço comercial (conforme definido no FAR 2.101) e são de propriedade da NetApp, Inc. Todos os dados técnicos e software de computador da NetApp fornecidos sob este Contrato são de natureza comercial e desenvolvidos exclusivamente com despesas privadas. O Governo dos EUA tem uma licença mundial limitada, irrevogável, não exclusiva, intransferível e não sublicenciável para usar os Dados que estão relacionados apenas com o suporte e para cumprir os contratos governamentais desse país que determinam o fornecimento de tais Dados. Salvo disposição em contrário no presente documento, não é permitido usar, divulgar, reproduzir, modificar, executar ou exibir os dados sem a aprovação prévia por escrito da NetApp, Inc. Os direitos de licença pertencentes ao governo dos Estados Unidos para o Departamento de Defesa estão limitados aos direitos identificados na cláusula 252.227-7015(b) (fevereiro de 2014) do DFARS.

Informações sobre marcas comerciais

NETAPP, o logotipo NETAPP e as marcas listadas em <http://www.netapp.com/TM> são marcas comerciais da NetApp, Inc. Outros nomes de produtos e empresas podem ser marcas comerciais de seus respectivos proprietários.