



Datenbankkonfiguration

Enterprise applications

NetApp

February 10, 2026

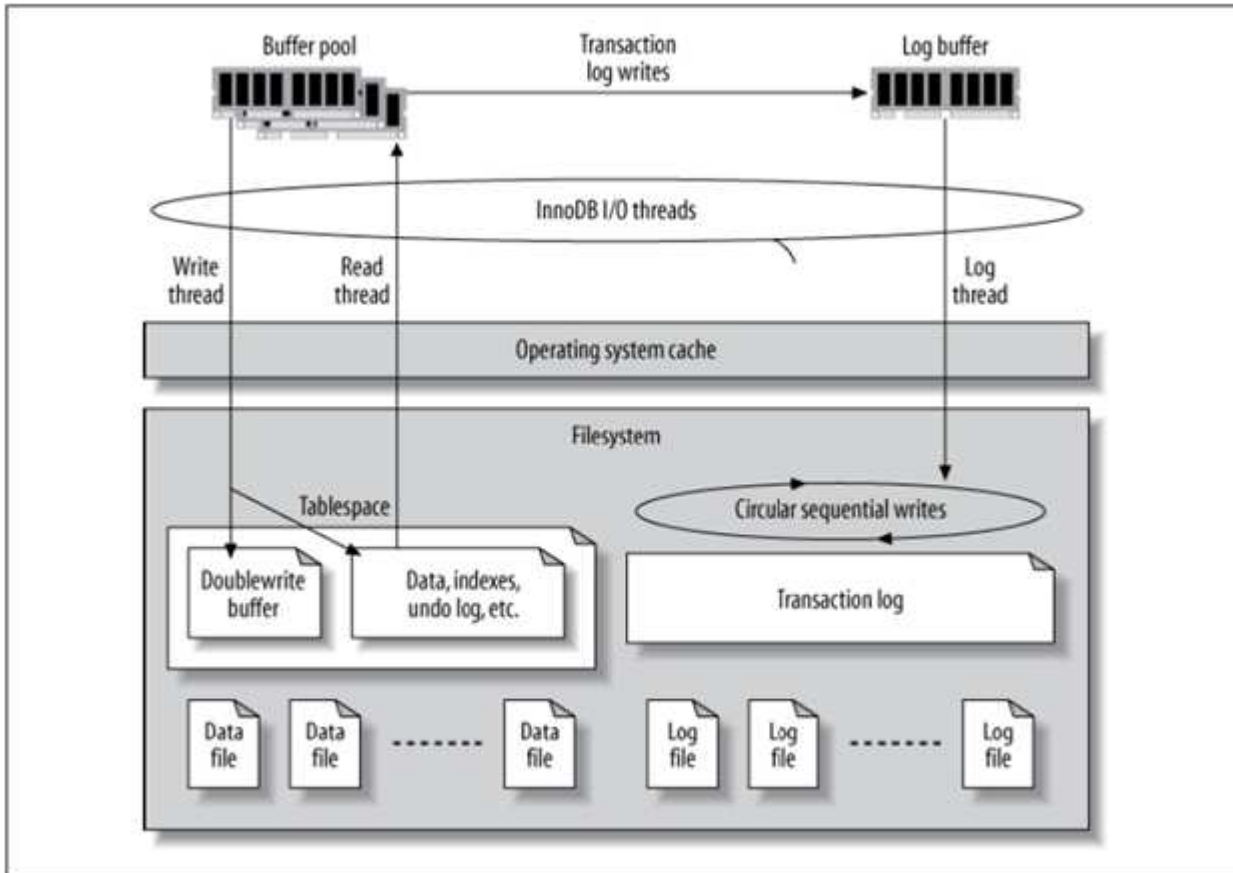
Inhalt

- Datenbankkonfiguration 1
 - Dateistruktur 1
 - Konfigurationsparameter 3
 - innodb_Log_file_size 4
 - innodb_Flush_log_at_trx_commit 4
 - innodb_doublewrite 5
 - innodb_Buffer_Pool_size 5
 - innodb_Flush_Method 6
 - Optimierungen 6
 - Beobachtung 6
 - innodb_io_Capacity 6
 - innodb_lru_Scan_depth 7
 - Open_File_Limits 7

Datenbankkonfiguration

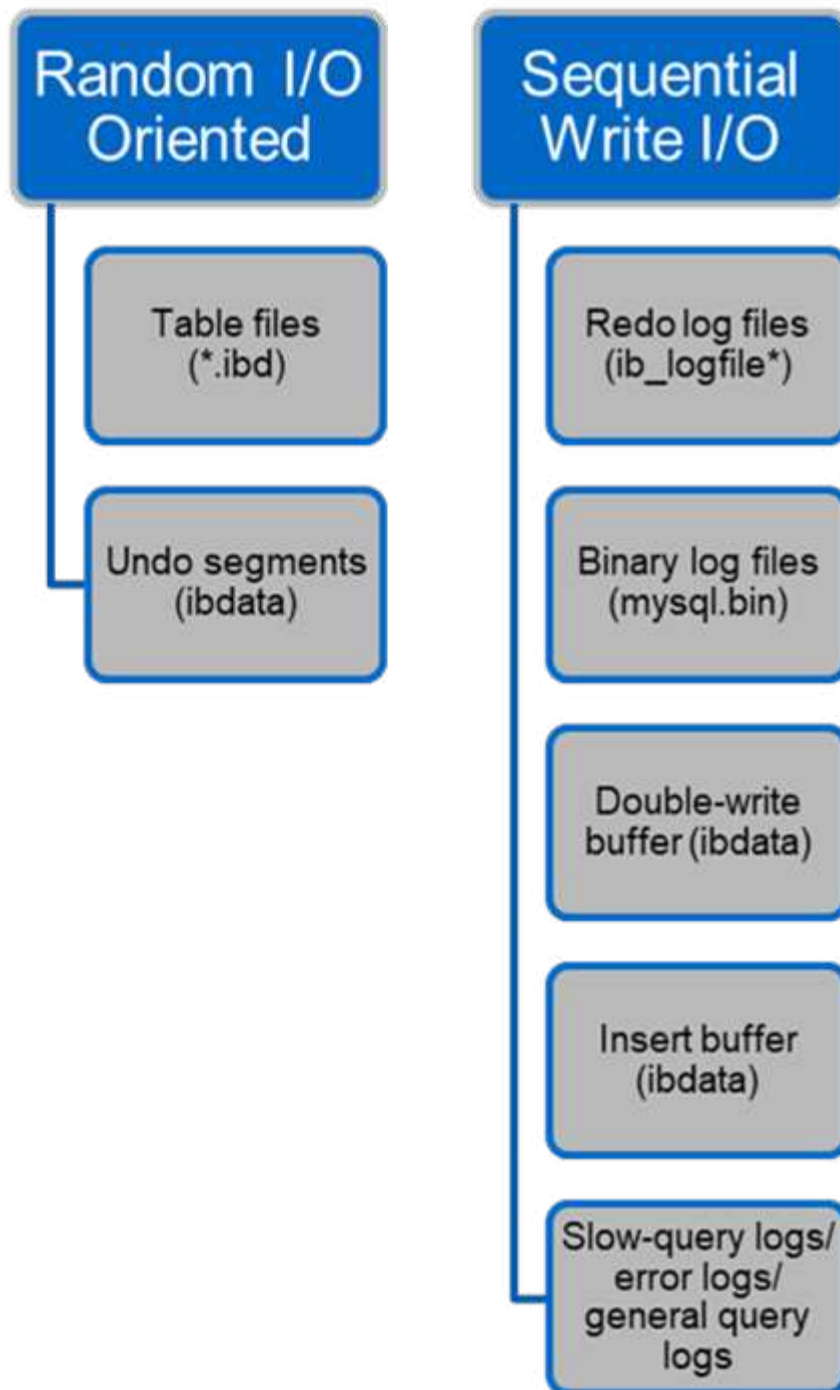
Dateistruktur

InnoDB fungiert als mittlere Schicht zwischen Speicher und MySQL-Server, es speichert die Daten auf den Laufwerken.



MySQL I/O wird in zwei Typen unterteilt:

- Zufällige Datei-I/O
- Sequenzielle Datei-I/O



Datendateien werden zufällig gelesen und überschrieben, was zu einem hohen IOPS-Wert führt. Daher wird SSD-Speicher empfohlen.

Redo-Log-Dateien und binäre Log-Dateien sind Transaktionsprotokolle. Sie werden sequenziell geschrieben, sodass Sie mit dem Schreib-Cache eine gute Performance auf der Festplatte erzielen können. Ein sequenzieller Lesevorgang findet bei der Wiederherstellung statt, jedoch verursacht er selten ein Performance-Problem, da die Größe der Protokolldatei normalerweise kleiner ist als die von Datendateien und sequenzielle Lesevorgänge schneller sind als zufällige Lesevorgänge (die bei Datendateien auftreten).

Der Double-Write-Puffer ist eine Besonderheit von InnoDB. InnoDB schreibt zunächst gerötete Seiten in den Double-Write-Puffer und schreibt dann die Seiten an die richtigen Positionen in den Datendateien. Dieser

Prozess verhindert eine Beschädigung der Seite. Ohne den Double-Write-Puffer kann die Seite beschädigt werden, wenn während des Write-to-Drive-Prozesses ein Stromausfall auftritt. Da das Schreiben in den Doppelschreibpuffer sequenziell ist, wird es für HDDs stark optimiert. Bei der Wiederherstellung werden sequenzielle Lesevorgänge durchgeführt.

Da ONTAP NVRAM bereits einen Schreibschutz bietet, ist keine doppelte Schreibpufferung erforderlich. MySQL hat einen Parameter, `skip_innodb_doublewrite`, um den Double-Write-Puffer zu deaktivieren. Diese Funktion kann die Leistung erheblich verbessern.

Der Insert-Puffer ist ebenfalls eine Besonderheit von InnoDB. Wenn sich nicht eindeutige sekundäre Indexblöcke nicht im Speicher befinden, fügt InnoDB Einträge in den Insert-Puffer ein, um zufällige I/O-Vorgänge zu vermeiden. Der Insert-Puffer wird in regelmäßigen Abständen in die sekundären Indexbäume der Datenbank eingebunden. Der Insert-Puffer reduziert die Anzahl der I/O-Vorgänge, indem I/O-Anfragen an denselben Block zusammengeführt werden. Zufällige I/O-Vorgänge können sequenziell sein. Der Einfügepuffer ist auch für HDDs stark optimiert. Sowohl sequenzielle Schreibvorgänge als auch Lesevorgänge erfolgen im normalen Betrieb.

Rückgängig-Segmente sind wahlfrei E/A-orientiert. Um die Multiversionenparallelität (MVCC) zu gewährleisten, muss InnoDB alte Bilder in den Undo-Segmenten registrieren. Beim Lesen vorheriger Bilder aus den Rückgängigmachungssegmenten sind zufällige Lesevorgänge erforderlich. Wenn Sie eine lange Transaktion mit wiederholbaren Lesevorgängen ausführen (wie z. B. `mysqldump` – einzelne Transaktion) oder eine lange Abfrage ausführen, können zufällige Lesevorgänge auftreten. Daher ist das Speichern von undo-Segmenten auf SSDs in dieser Instanz besser. Wenn Sie nur kurze Transaktionen oder Abfragen ausführen, stellen die zufälligen Lesevorgänge kein Problem dar.

NetApp empfiehlt aufgrund der InnoDB I/O-Eigenschaften das folgende Speicherdesign-Layout.



- Ein Volume zur Speicherung zufälliger und sequenzieller I/O-orientierter MySQL-Dateien
- Ein weiteres Volume zur Speicherung rein sequenzieller I/O-orientierter Dateien von MySQL

Dieses Layout hilft Ihnen auch bei der Entwicklung von Datensicherungsrichtlinien und -Strategien.

Konfigurationsparameter

NetApp empfiehlt ein paar wichtige MySQL-Konfigurationsparameter, um eine optimale Performance zu erzielen.

Parameter	Werte
<code>innodb_Log_file_size</code>	256 MIO.
<code>innodb_Flush_log_at_trx_commit</code>	2
<code>innodb_doublewrite</code>	0
<code>innodb_Flush_Method</code>	Fsync
<code>innodb_Buffer_Pool_size</code>	11 G
<code>innodb_io_Capacity</code>	8192
<code>innodb_Buffer_Pool_Instances</code>	8

innodb_lru_scan_depth	8192
Open_File_Limit	65535

Um die in diesem Abschnitt beschriebenen Parameter einzustellen, müssen Sie sie in der MySQL-Konfigurationsdatei (my.cnf) ändern. Die Best Practices von NetApp wurden in internen Tests durchgeführt.

innodb_log_file_size

Die Auswahl der richtigen Größe für die InnoDB-Protokolldateigröße ist wichtig für die Schreibvorgänge und für eine anständige Wiederherstellungszeit nach einem Serverabsturz.

Da so viele Transaktionen in der Datei angemeldet sind, ist die Größe der Protokolldatei für Schreibvorgänge wichtig. Wenn Datensätze geändert werden, wird die Änderung nicht sofort in den Tablespace zurückgeschrieben. Stattdessen wird die Änderung am Ende der Protokolldatei aufgezeichnet und die Seite als verschmutzt markiert. InnoDB verwendet sein Protokoll, um zufällige I/O in sequenzielle I/O-Vorgänge zu konvertieren

Wenn das Protokoll voll ist, wird die fehlerhafte Seite nacheinander in den Tablespace geschrieben, um Speicherplatz in der Protokolldatei freizugeben. Angenommen, ein Server stürzt mitten in einer Transaktion ab, und die Schreibvorgänge werden nur in der Protokolldatei aufgezeichnet. Bevor der Server wieder live gehen kann, muss er eine Wiederherstellungsphase durchlaufen, in der die in der Protokolldatei aufgezeichneten Änderungen wiedergegeben werden. Je mehr Einträge in der Protokolldatei vorhanden sind, desto länger dauert es, bis der Server wiederhergestellt ist.

In diesem Beispiel wirkt sich die Größe der Protokolldatei sowohl auf die Wiederherstellungszeit als auch auf die Schreib-Performance aus. Wenn Sie die richtige Zahl für die Größe der Protokolldatei wählen, gleichen Sie die Recovery-Zeit mit der Schreib-Performance ab. Normalerweise ist alles zwischen 128M und 512M ein gutes Preis-Leistungs-Verhältnis.

innodb_flush_log_at_trx_commit

Wenn eine Datenänderung vorgenommen wird, wird nicht sofort in den Storage geschrieben.

Stattdessen werden die Daten in einem Protokollpuffer aufgezeichnet, einem Teil des Speichers, den InnoDB Pufferänderungen zuweist, die in der Protokolldatei aufgezeichnet werden. InnoDB leert den Puffer in die Protokolldatei, wenn eine Transaktion durchgeführt wird, wenn der Puffer voll wird, oder einmal pro Sekunde, je nachdem, welches Ereignis zuerst eintritt. Die Konfigurationsvariable, die diesen Prozess steuert, ist `innodb_flush_log_at_trx_commit`. Die Wertoptionen umfassen:

- Wenn Sie es einstellen `innodb_flush_log_trx_at_commit=0`, InnoDB schreibt die geänderten Daten (im InnoDB-Pufferpool) in die Log-Datei (`ib_logfile`) und leert die Log-Datei (Write to Storage) jede Sekunde. Es tut jedoch nichts, wenn die Transaktion durchgeführt wird. Bei einem Stromausfall oder Systemabsturz können die nicht gespeicherten Daten nicht wiederhergestellt werden, da sie weder auf die Protokolldatei noch auf die Laufwerke geschrieben werden.
- Wenn Sie es einstellen `innodb_flush_log_trx_commit=1`, InnoDB schreibt den Protokollpuffer in das Transaktionsprotokoll und leert für jede Transaktion auf eine dauerhafte Speicherung. Beispielsweise schreibt InnoDB für alle Transaction Commits in das Protokoll und schreibt anschließend in den Speicher. Langsamer Speicher beeinträchtigt die Performance, beispielsweise wird die Anzahl der InnoDB-

Transaktionen pro Sekunde reduziert.

- Wenn Sie es einstellen `innodb_flush_log_trx_commit=2`, InnoDB schreibt den Protokollpuffer bei jedem Commit in die Protokolldatei, schreibt aber keine Daten in den Speicher. InnoDB leert die Daten einmal pro Sekunde. Selbst bei einem Stromausfall oder Systemabsturz sind die Daten von Option 2 in der Protokolldatei verfügbar und können wiederhergestellt werden.

Wenn die Leistung das Hauptziel ist, setzen Sie den Wert auf 2. Da InnoDB einmal pro Sekunde auf die Laufwerke schreibt, nicht für jede Transaktion, verbessert sich die Performance erheblich. Bei einem Stromausfall oder Absturz können die Daten aus dem Transaktions-Log wiederhergestellt werden.

Wenn die Datensicherheit das Hauptziel ist, setzen Sie den Wert auf 1, sodass InnoDB für jeden Transaktionscommit zu den Laufwerken leert. Möglicherweise ist die Performance jedoch beeinträchtigt.



NetApp empfiehlt Setzen Sie den `innodb_flush_log_trx_commit` Wert auf 2, um eine bessere Leistung zu erzielen.

innodb_doublewrite

Wenn `innodb_doublewrite` Ist aktiviert (Standard), speichert InnoDB alle Daten zweimal: Zuerst in den Double-Write-Puffer und dann in die eigentlichen Datendateien.

Sie können diesen Parameter mit deaktivieren `--skip-innodb_doublewrite` Für Benchmarks oder wenn Sie sich mehr um Top-Performance als um Datenintegrität oder mögliche Ausfälle sorgen. InnoDB verwendet eine Datei-Flush-Technik namens Double-Write. Bevor die Seiten in die Datendateien geschrieben werden, schreibt InnoDB sie in einen zusammenhängenden Bereich, den sogenannten Double-Write-Puffer. Nachdem das Schreiben und der Flush in den Double-Write-Puffer abgeschlossen sind, schreibt InnoDB die Seiten an die richtigen Positionen in der Datendatei. Falls das Betriebssystem oder ein mysqld-Prozess während eines Page Write abstürzt, kann InnoDB später während der Crash-Wiederherstellung eine gute Kopie der Seite aus dem Double-Write-Puffer finden.



NetApp empfiehlt den Double-Write-Puffer zu deaktivieren. ONTAP NVRAM dient dieselbe Funktion. Die doppelte Pufferung beeinträchtigt die Leistung unnötig.

innodb_Buffer_Pool_size

Der InnoDB Pufferpool ist der wichtigste Teil jeder Tuning-Aktivität.

InnoDB setzt stark auf den Pufferpool für das Caching von Indizes und Rudern der Daten, den adaptiven Hash-Index, den Insert-Puffer und viele andere interne Datenstrukturen. Der Puffer-Pool puffert Änderungen an Daten, damit Schreibzugriffe nicht sofort in den Storage übernommen werden müssen, was die Performance verbessert. Der Pufferpool ist integraler Bestandteil von InnoDB und muss entsprechend angepasst werden. Berücksichtigen Sie beim Festlegen der Größe des Puffer-Pools die folgenden Faktoren:

- Stellen Sie für eine dedizierte InnoDB-Maschine die Pufferpoolgröße auf 80 % oder mehr verfügbaren RAM ein.
- Wenn es sich nicht um einen dedizierten MySQL-Server handelt, stellen Sie die Größe auf 50 % des RAM ein.

innodb_Flush_Method

Der Parameter `innodb_flush_method` gibt an, wie InnoDB die Protokoll- und Datendateien öffnet und löscht.

Optimierungen

In der InnoDB-Optimierung wird durch die Einstellung dieses Parameters die Datenbankleistung ggf. optimiert.

Die folgenden Optionen sind für das Spülen der Dateien über InnoDB:

- `fsync`. InnoDB verwendet die `fsync()` Systemaufruf, um sowohl die Daten- als auch die Protokolldateien zu leeren. Diese Option ist die Standardeinstellung.
- `O_DSYNC`. InnoDB verwendet die `O_DSYNC` Option zum Öffnen und Leeren der Protokolldateien und `fsync()` zum Leeren der Datendateien. InnoDB wird nicht verwendet `O_DSYNC` Direkt, weil es Probleme mit ihm auf vielen Sorten von UNIX.
- `O_DIRECT`. InnoDB verwendet die `O_DIRECT` Option (oder `directio()` Unter Solaris), um die Datendateien zu öffnen und zu verwenden `fsync()` Um sowohl die Daten als auch die Protokolldateien zu löschen. Diese Option ist auf einigen GNU/Linux-Versionen, FreeBSD und Solaris verfügbar.
- `O_DIRECT_NO_FSYNC`. InnoDB verwendet die `O_DIRECT` Option beim Spülen von E/A, wird jedoch übersprungen `fsync()` Systemaufruf danach. Diese Option ist für einige Arten von Dateisystemen ungeeignet (z. B. XFS). Wenn Sie sich nicht sicher sind, ob Ihr Dateisystem einen erfordert `fsync()` Systemaufruf – zum Beispiel zum Beibehalten aller Dateimetadaten – verwendet den `O_DIRECT` Wählen Sie stattdessen.

Beobachtung

In den NetApp-Labortests ist der `fsync` Auf NFS und SAN kam die Standardoption zum Einsatz. Im Vergleich dazu war sie ein großartiger Performance-Improvisator `O_DIRECT`. Bei Verwendung der Spülmethode als `O_DIRECT` Bei ONTAP konnten wir beobachten, dass der Client viele Single-Byte-Schreibvorgänge am Rand des 4096. Blocks in serieller Form schreibt. Diese Schreibvorgänge haben die Latenz über das Netzwerk erhöht und die Performance beeinträchtigt.

innodb_io_Capacity

Im InnoDB Plug-in wurde ein neuer Parameter namens `innodb_io_Capacity` aus MySQL 5.7 hinzugefügt.

Er steuert die maximale Anzahl von IOPS, die InnoDB durchführt (einschließlich der Spülrate von schmutzigen Seiten sowie der Batch-Größe des Insert Buffer [ibuf]). Der `innodb_io_capacity` Parameter setzt eine Obergrenze für IOPS durch InnoDB-Hintergrundaufgaben, wie das Leeren von Seiten aus dem Pufferpool und das Zusammenführen von Daten aus dem Änderungspuffer.

Legen Sie den Parameter `innodb_io_Capacity` auf die ungefähre Anzahl von E/A-Vorgängen fest, die das System pro Sekunde durchführen kann. Halten Sie die Einstellung idealerweise so niedrig wie möglich, aber nicht so niedrig, dass Hintergrundaktivitäten langsamer werden. Ist die Einstellung zu hoch, werden die Daten aus dem Puffer-Pool entfernt und Puffer für das Caching zu schnell eingefügt, um einen wesentlichen Vorteil zu erzielen.



NetApp empfiehlt, wenn Sie diese Einstellung über NFS verwenden, das Testergebnis von IOPS (SysBench/FiO) analysieren und den Parameter entsprechend einstellen. Verwenden Sie den kleinstmöglichen Wert zum Spülen und Spülen, um mit dem Schritt zu bleiben, es sei denn, Sie sehen mehr geänderte oder schmutzige Seiten als Sie im InnoDB-Puffer-Pool möchten.



Verwenden Sie keine Extremwerte wie 20,000 oder mehr, es sei denn, Sie haben bewiesen, dass niedrigere Werte für Ihre Arbeitsbelastung nicht ausreichen.

Der `InnoDB_IO_Capacity` Parameter regelt Spülraten und zugehörige I/O.



Sie können die Leistung ernsthaft beeinträchtigen, indem Sie diesen Parameter oder den `innodb_io_Capacity_max`-Parameter zu hoch und zu verschwenderisch einstellen

innodb_lru_scan_depth

Der `innodb_lru_scan_depth` Parameter beeinflusst die Algorithmen und Heuristiken des Flush-Vorgangs für den InnoDB Pufferpool.

Dieser Parameter ist in erster Linie an Performance-Experten interessiert, die I/O-intensive Workloads optimieren. Dieser Parameter gibt für jede Pufferpoolinstanz an, wie weit der Seitenaufbau des Seitenreinigers in der LRU-Seitenliste (Least Recently Used) weiter scannen soll, und sucht nach verschmutzten Seiten, die bereinigt werden sollen. Dieser Hintergrundvorgang wird einmal pro Sekunde durchgeführt.

Sie können den Wert nach oben oder unten anpassen, um die Anzahl der freien Seiten zu minimieren. Stellen Sie den Wert nicht wesentlich höher ein als erforderlich, da die Scans erhebliche Performancekosten verursachen können. Ziehen Sie außerdem in Betracht, diesen Parameter anzupassen, wenn Sie die Anzahl der Pufferpool-Instanzen ändern, da `innodb_lru_scan_depth * innodb_buffer_pool_instances` Definiert den Umfang der Arbeit, die durch den Seitenreinigungsfaden jede Sekunde durchgeführt wird.

Eine Einstellung, die kleiner als die Standardeinstellung ist, eignet sich für die meisten Workloads. Ziehen Sie in Betracht, diesen Wert nur zu erhöhen, wenn Sie freie I/O-Kapazität bei einem typischen Workload haben. Wenn ein schreibintensiver Workload die I/O-Kapazität ausschüttet, verringern Sie den Wert umgekehrt, insbesondere wenn ein großer Puffer-Pool vorhanden ist.

Open_File_Limits

Der `open_file_limits` Parameter bestimmt die Anzahl der Dateien, die das Betriebssystem `mysqld` zum Öffnen zulässt.

Der Wert dieses Parameters zur Laufzeit ist der vom System zulässige Realwert und kann sich von dem Wert unterscheiden, den Sie beim Serverstart angeben. Der Wert ist 0 auf Systemen, bei denen MySQL die Anzahl der geöffneten Dateien nicht ändern kann. Die effektive `open_files_limit` Der Wert basiert auf dem Wert, der beim Systemstart (falls vorhanden) angegeben wurde, und den Werten von `max_connections` Und `table_open_cache` Mit diesen Formeln:

- $10 + \text{max_connections} + (\text{table_open_cache} \times 2)$
- $\text{max_connections} \times 5$
- Betriebssystemgrenze, wenn positiv

- Wenn die Betriebssystemgrenze unendlich ist: `open_files_limit` Wert wird beim Start angegeben; 5,000 wenn keine

Der Server versucht, die Anzahl der Dateideskriptoren mit dem Maximum dieser vier Werte zu ermitteln. Wenn so viele Deskriptoren nicht abgerufen werden können, versucht der Server, so viele zu erhalten, wie das System zulässt.

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFT SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.