



MySQL

Enterprise applications

NetApp
January 02, 2026

目錄

MySQL	1
總覽	1
資料庫組態	1
檔案結構	1
組態參數	4
InnoDB_log_file_size	4
InnoDB_Flush 記錄 _AT_TRx_Commit	5
InnoDB_doublewrite	5
InnoDB_緩衝區 _Pool_size	6
InnoDB_Flush 方法	6
InnoDB_IO_capAC	6
InnoDB_LRU_SCAN_depth	7
open_file_limits	7
主機組態	7
容器化	8
NFSv3 插槽表	8
I/O 排程器	8
檔案描述元	9
儲存組態	9
NFS	9
SAN	10

MySQL

總覽

MySQL 及其變種、包括 MariaDB 和 Percona MySQL 、是全球最受歡迎的資料庫。



ONTAP 和 MySQL 資料庫上的本文件取代先前發佈的 _TR-4722 : ONTAP 最佳實務做法的 MySQL 資料庫。 _

ONTAP 是適用於 MySQL 資料庫的理想平台、因為 ONTAP 確實是專為資料庫所設計。為了滿足資料庫工作負載的需求、我們特別建立了許多功能、例如隨機 IO 延遲最佳化、以提供進階服務品質 (QoS) 到基本 FlexClone 功能。

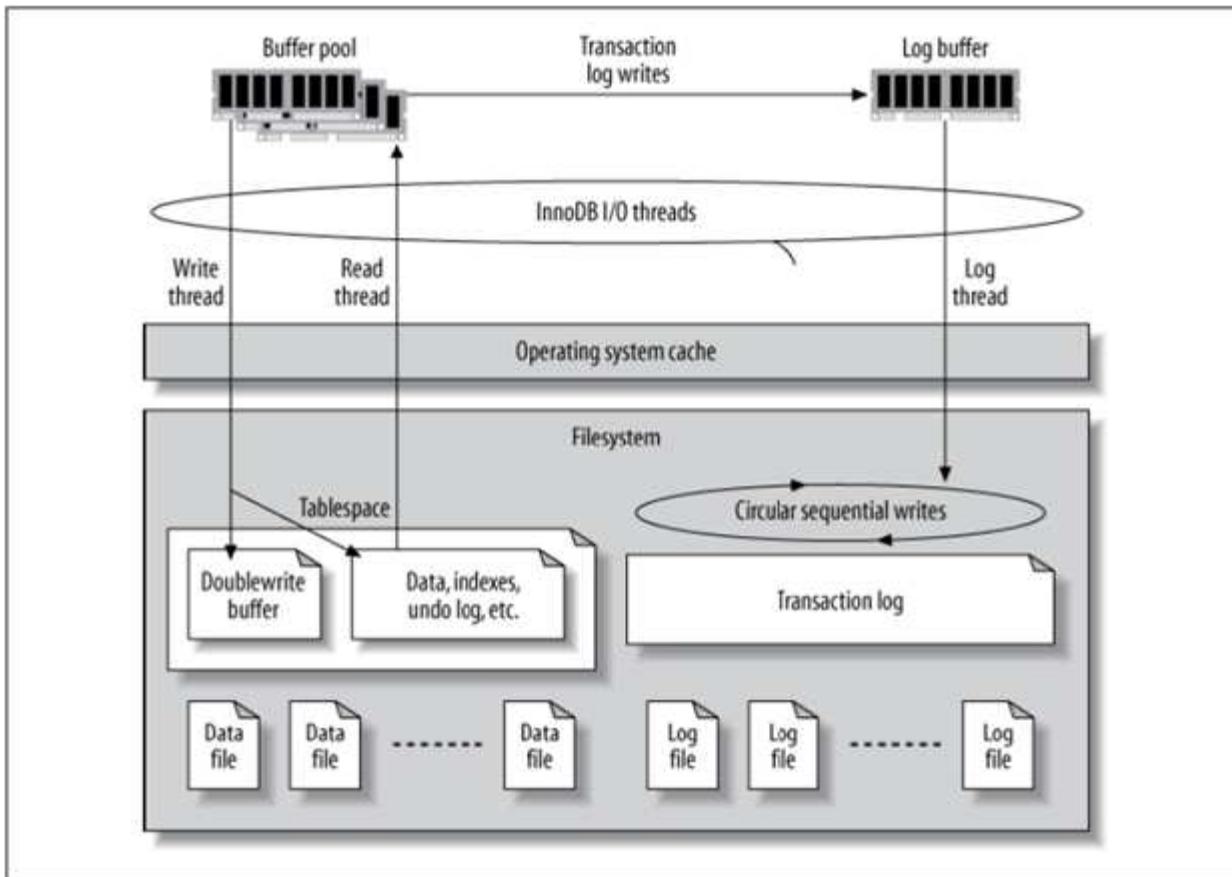
其他功能 (例如不中斷升級) (包括儲存設備更換) 、可確保關鍵資料庫仍可使用。您也可以透過 MetroCluster 為大型環境進行即時災難恢復、或是使用 SnapMirror 主動式同步來選擇資料庫。

最重要的是、ONTAP 提供無與倫比的效能、並能根據您的獨特需求調整解決方案的規模。我們的高階系統可提供超過 1M IOPS 、延遲時間以微秒為測量單位、但如果您只需要 10 萬次 IOPS 、您可以使用更小的控制器來調整儲存解決方案的大小、而該控制器仍可執行完全相同的儲存作業系統。

資料庫組態

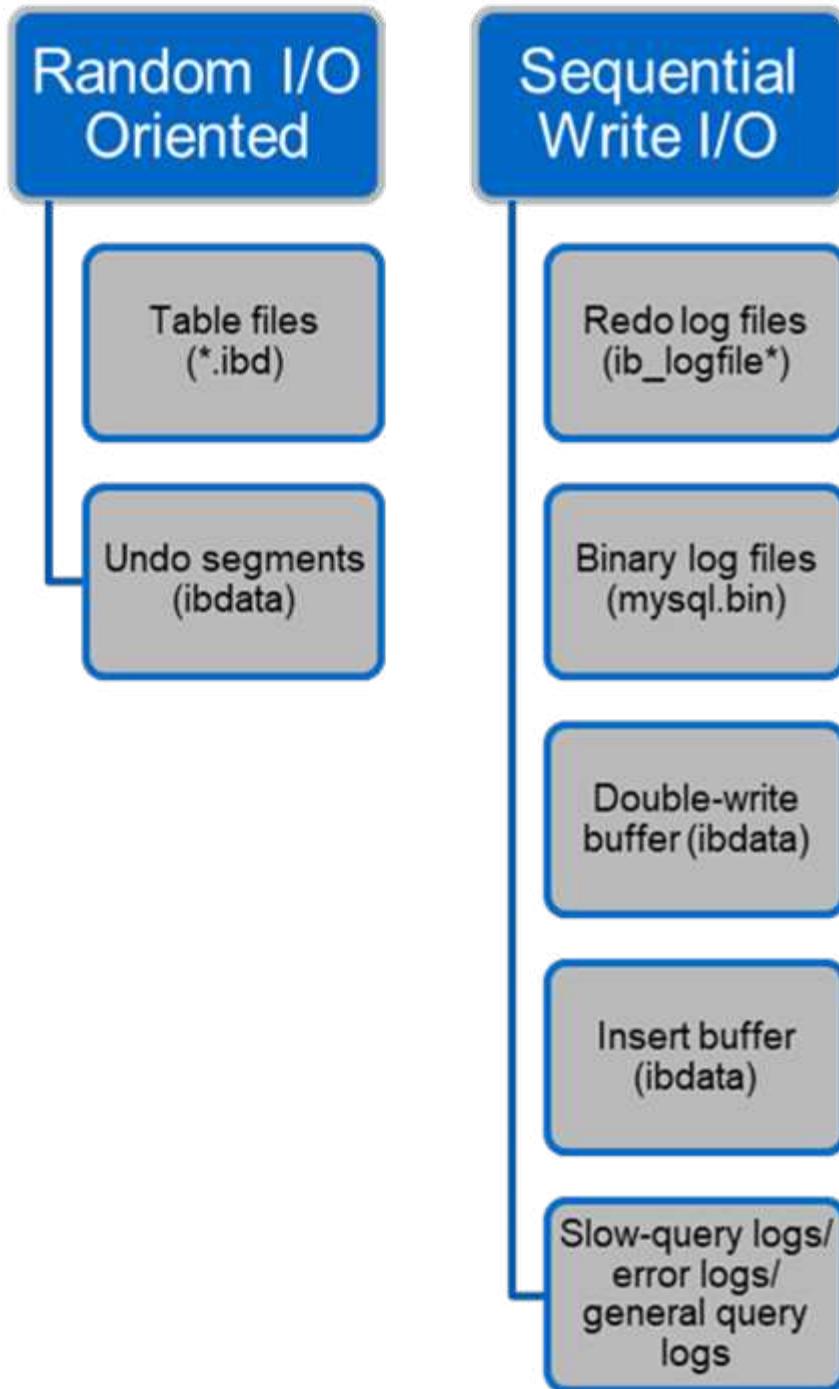
檔案結構

InnoDB 是儲存設備和 MySQL 伺服器之間的中間層、可將資料儲存到磁碟機中。



MySQL I/O 分為兩種類型：

- 隨機檔案 I/O
- 連續檔案 I/O



資料檔案會隨機讀取及覆寫、因此 IOPS 較高。因此、建議使用 SSD 儲存設備。

重做記錄檔和二進位記錄檔是交易記錄檔。它們會依序寫入、因此您可以在 HDD 上使用寫入快取獲得良好效能。恢復時會進行連續讀取、但很少會造成效能問題、因為記錄檔大小通常比資料檔案小、而連續讀取比隨機讀取快（發生在資料檔案上）。

雙寫入緩衝區是 InnoDB 的一項特殊功能。InnoDB 會先將排清的頁面寫入雙寫入緩衝區、然後將頁面寫入資料檔案的正確位置。此程序可防止頁面毀損。如果沒有雙寫入緩衝區、如果在寫入磁碟機程序期間發生電源故障、頁面可能會毀損。由於寫入雙寫入緩衝區是循序的、因此對 HDD 進行了高度最佳化。恢復時會進行連續讀取。

由於 ONTAP NVRAM 已提供寫入保護、因此不需要雙重寫入緩衝。MySQL 有一個參數、

`skip_innodb_doublewrite`，以停用雙寫入緩衝區。此功能可大幅提升效能。

插入緩衝區也是 InnoDB 的一項特殊功能。如果非唯一的次要索引區塊不在記憶體中、InnoDB 會將項目插入插入緩衝區、以避免隨機 I/O 作業。插入緩衝區會定期合併到資料庫的次要索引樹中。插入緩衝區可將 I/O 要求合併至同一個區塊、以減少 I/O 作業的數量；隨機 I/O 作業可以是連續的。插入緩衝區也針對 HDD 進行了高度最佳化。連續寫入和讀取都會在正常作業期間執行。

復原區段以隨機 I/O 為導向。為了保證多重版本併發（MVCC）、InnoDB 必須在復原區段中登錄舊影像。從復原區段讀取先前影像需要隨機讀取。如果您執行具有可重複讀取的長交易（例如 mysqldump —單一交易）或執行長查詢、可能會發生隨機讀取。因此、在這種情況下、將復原區段儲存在 SSD 上會更好。如果您只執行短的交易或查詢、隨機讀取並不是問題。

- 由於 InnoDB I/O 特性、NetApp 建議 * 下列儲存設計配置。
- 一個用於儲存 MySQL 的隨機和連續 I/O 導向檔案的磁碟區
- 另一個用於儲存 MySQL 純粹循序 I/O 導向檔案的磁碟區

此配置也能協助您設計資料保護原則與策略。

組態參數

NetApp 建議使用一些重要的 MySQL 組態參數、以獲得最佳效能。

參數	價值
<code>InnoDB_log_file_size</code>	256M
<code>InnoDB_Flush 記錄_AT_TRx_Commit</code>	2.
<code>InnoDB_doublewrite</code>	0%
<code>InnoDB_Flush 方法</code>	<code>fsync</code>
<code>InnoDB_緩衝區_Pool_size</code>	11g
<code>InnoDB_IO_capAC</code>	8192
<code>InnoDB_緩衝區集區執行個體</code>	8.
<code>InnoDB_LRU_SCAN_depth</code>	8192
<code>open_file_limit</code>	65535

若要設定本節所述的參數、您必須在 MySQL 組態檔（my.cnf）中變更這些參數。NetApp 最佳實務做法是在內部執行測試的結果。

InnoDB_log_file_size

為 InnoDB 記錄檔大小選取適當的大小、對於寫入作業以及在伺服器當機後擁有適當的還原時間都很重要。

由於有這麼多交易登入檔案、因此記錄檔大小對於寫入作業非常重要。修改記錄時、變更不會立即回寫到表格區。相反地、變更會記錄在記錄檔的結尾、而且頁面會標示為「髒污」。InnoDB 使用其記錄檔將隨機 I/O 轉換成連續 I/O

當記錄檔已滿時、會依序將不完整頁面寫入資料表空間、以釋放記錄檔中的空間。例如、假設某個伺服器在交易過程中當機、而且寫入作業只會記錄在記錄檔中。伺服器必須經過復原階段、記錄檔中記錄的變更會重新播放、才能重新上線。記錄檔中的項目越多、伺服器恢復所需的時間就越長。

在此範例中、記錄檔大小會同時影響還原時間和寫入效能。為記錄檔大小選擇正確的數字時、請平衡恢復時間與寫入效能。一般而言、128M 和 512M 之間的任何項目都是物超所值的。

InnoDB_Flush 記錄 _AT_TRx_Commit

當資料發生變更時、變更不會立即寫入儲存設備。

而是記錄在記錄緩衝區中、這是 InnoDB 分配給記錄在記錄檔中的緩衝區變更的記憶體部分。InnoDB 會在交易提交、緩衝區滿時、或每秒一次（以先發生的事件為準）、將緩衝區排清至記錄檔。控制此程序的組態變數是 InnoDB_Flush 記錄 _AT_TRx_Commit。價值選項包括：

- 當您設定時 `innodb_flush_log_trx_at_commit=0`、InnoDB 會將修改過的資料（在 InnoDB 緩衝區集中）寫入記錄檔（`IB_logfile`）並每秒清除記錄檔（寫入儲存區）。不過、提交交易時、它不會執行任何動作。如果發生電源故障或系統當機、則沒有任何未排清的資料可恢復、因為它不會寫入記錄檔或磁碟機。
- 當您設定時 `innodb_flush_log_trx_commit=1`、InnoDB 會將記錄緩衝區寫入交易記錄檔、並在每筆交易中將記錄檔排清至持久儲存區。例如、對於所有交易認可、InnoDB 會寫入記錄、然後寫入儲存設備。儲存速度變慢會對效能造成負面影響、例如每秒 InnoDB 交易數會減少。
- 當您設定時 `innodb_flush_log_trx_commit=2`、InnoDB 會在每次提交時將記錄緩衝區寫入記錄檔、但不會將資料寫入儲存區。InnoDB 每秒會清除一次資料。即使發生電源故障或系統當機、記錄檔中也有選項 2 資料可供使用、而且可恢復。

如果效能是主要目標、請將值設為 2。由於 InnoDB 每秒一次寫入磁碟機、而非每次提交交易時、效能大幅提升。如果發生停電或當機、可從交易記錄中恢復資料。

如果資料安全是主要目標、請將值設為 1、以便每次提交交易時、InnoDB 都會將資料清出磁碟機。不過、效能可能會受到影響。



* NetApp 建議 * 將 InnoDB_Flush 日誌 _TRx_Commit 值設為 2、以獲得更好的效能。

InnoDB_doublewrite

何時 `innodb_doublewrite` 啟用（預設）時、InnoDB 會將所有資料儲存兩次：先儲存至雙寫入緩衝區、然後儲存至實際資料檔案。

您可以使用關閉此參數 `--skip-innodb_doublewrite` 針對效能標準、或是當您比較在意最高效能、而非資料完整性或可能的故障時。InnoDB 使用稱為雙寫入的檔案清除技術。InnoDB 在將頁面寫入資料檔案之前、會將其寫入稱為雙寫入緩衝區的鄰近區域。寫入和清除雙寫入緩衝區完成後、InnoDB 會將頁面寫入資料檔案中的適當位置。如果作業系統或 `mysqld` 程序在頁面寫入期間當機、InnoDB 之後可以在當機恢復期間、從雙寫入緩衝區找到良好的頁面複本。



* NetApp 建議 * 停用雙寫入緩衝區。ONTAP NVRAM 的功能相同。雙重緩衝會不必要地損害效能。

InnoDB_緩衝區 _Pool_size

InnoDB 緩衝資源池是任何調校活動中最重要的部分。

InnoDB 在很大程度上仰賴緩衝區集區來快取索引和資料、調適性雜湊索引、插入緩衝區、以及許多內部使用的其他資料結構。緩衝區集區也會緩衝資料的變更、這樣就不需要立即對儲存設備執行寫入作業、進而改善效能。緩衝區集區是 InnoDB 的一部分、必須據此調整其大小。設定緩衝區集區大小時、請考量下列因素：

- 若為僅 InnoDB 的專用機器、請將緩衝區集區大小設為 80% 以上的可用 RAM 。
- 如果不是 MySQL 專用伺服器、請將 RAM 大小設為 50% 。

InnoDB_Flush 方法

InnoDB_flush_method 參數指定 InnoDB 如何開啟及排清記錄檔和資料檔。

最佳化

在 InnoDB 最佳化中、如果適用、設定此參數會調整資料庫效能。

下列選項用於透過 InnoDB 排清檔案：

- `fsync`。InnoDB 使用 `fsync()` 系統呼叫以清除資料和記錄檔。此選項為預設設定。
- `O_DSYNC`。InnoDB 使用 `O_DSYNC` 用於打開和刷新日誌文件和 `fsync()` 以刷新數據文件的選項。InnoDB 不使用 `O_DSYNC` 直接來說、因為 UNIX 的許多種類都有問題。
- `O_DIRECT`。InnoDB 使用 `O_DIRECT` 選項（或 `directio()` 在 Solaris 上）開啟資料檔案及使用 `fsync()` 清除資料和記錄檔。此選項可在某些版本的 GNU/Linux、FreeBSD 和 Solaris 上使用。
- `O_DIRECT_NO_FSYNC`。InnoDB 使用 `O_DIRECT` 排清 I/O 時的選項；不過、它會跳過 `fsync()` 之後進行系統通話。此選項不適用於某些類型的檔案系統（例如 XFS）。如果您不確定檔案系統是否需要 `fsync()` 系統呼叫（例如為了保留所有檔案中繼資料）使用 `O_DIRECT` 選項。

觀察

在 NetApp 實驗室測試中、`fsync` 預設選項用於 NFS 和 SAN、與相較之下、這是一項很棒的效能改進工具 `O_DIRECT`。使用「齊平」方法時為 `O_DIRECT` 使用 ONTAP 時、我們觀察到用戶端會以序列方式、在 4096 區塊的邊界寫入大量的單位位元組寫入資料。這些寫入會增加網路延遲並降低效能。

InnoDB_IO_capAC

在 InnoDB 外掛程式中、從 MySQL 5.7 新增名為 InnoDB_IO_capACure.

它可控制 InnoDB 執行的 IOPS 上限（包括麟頁的排清率、以及插入緩衝區 [ibuf] 批次大小）。InnoDB_IO_capAC容量 參數會根據 InnoDB 背景工作來設定 IOPS 上限、例如從緩衝區排清頁面、以及從變更緩衝區合併資料。

將 InnoDB_IO_capAC容量 參數設定為系統每秒可執行的 I/O 作業大約數目。理想情況下、請盡可能將設定保持在低的位置、但不要太低、讓背景活動變慢。如果設定太高、資料會從緩衝區移除、並太快插入緩衝區以供快取、以提供顯著效益。



* NetApp 建議 * 如果在 NFS 上使用此設定、請分析 IOPS (Sys 台 / Fio) 的測試結果、並據此設定參數。除非您在 InnoDB 緩衝資源池中看到比您想要的更多修改或不乾淨頁面、否則請使用最小的值來進行排清和清除。



除非您證明較低的值不足以應付工作負載、否則請勿使用極端值、例如 20,000 或更多。

InnoDB_IO_capACure. 參數可規範排清率和相關 I/O



您可以將此參數或 InnoDB_IO_capACure_max 參數設定得太高且浪費，嚴重損害效能

InnoDB_LRU_SCAN_depth

- innodb_lru_scan_depth 參數會影響 InnoDB 緩衝區集區之排清作業的演算法和啟發性。

此參數主要是效能專家調校 I/O 密集工作負載的興趣所在。對於每個緩衝區集區執行個體、此參數會指定最少使用 (LRU) 頁面清單中、頁面清理程式執行緒應繼續掃描的程度、以尋找要清除的髒頁面。此背景作業每秒執行一次。

您可以上下調整值、將可用頁數降至最低。請勿將此值設定得比所需值高得多、因為掃描可能會產生重大的效能成本。此外、請考慮在變更緩衝區集區執行個體數目時調整此參數、因為 innodb_lru_scan_depth * innodb_buffer_pool_instances 定義頁面清理程式執行緒每秒執行的工作量。

小於預設值的設定適用於大部分的工作負載。只有在典型工作負載下有備用 I/O 容量時、才考慮增加此值。相反地、如果寫入密集的工作負載使 I/O 容量飽和、請降低該值、尤其是當您擁有大型緩衝區集區時。

open_file_limits

- open_file_limits 參數決定作業系統允許 mysqld 開啟的檔案數目。

此參數在執行階段的值是系統允許的實際值、可能與您在伺服器啟動時指定的值不同。在 MySQL 無法變更開啟檔案數量的系統上、此值為 0。有效 open_files_limit 此值是根據系統啟動時指定的值 (如果有) 和的值而定 max_connections 和 table_open_cache 使用這些公式：

- $10 + \text{max_connections} + (\text{table_open_cache} \times 2)$
- $\text{max_connections} \times 5$.
- 如果為正、則作業系統限制
- 如果作業系統限制為無限： open_files_limit 在啟動時指定值；若無、則指定值為 5,000

伺服器會嘗試使用這四個值的最大值來取得檔案描述元數目。如果無法取得這麼多描述元、伺服器會嘗試取得系統允許的數量。

主機組態

容器化

MySQL 資料庫的容器化日漸普及。

低層級的容器管理幾乎總是透過 Docker 來執行。OpenShift 和 Kubernetes 等容器管理平台讓大型容器環境的管理變得更簡單。容器化的優點包括成本較低、因為不需要授權 Hypervisor。此外、容器可讓多個資料庫彼此隔離執行、同時共用相同的基礎核心和作業系統。容器可在微秒內完成佈建。

NetApp 提供 Astra Trident、可提供進階的儲存管理功能。例如、Astra Trident 可讓在 Kubernetes 中建立的容器自動在適當的層級上配置儲存設備、套用匯出原則、設定快照原則、甚至將一個容器複製到另一個容器。如需其他資訊，請參閱["Astra Trident文件"](#)。

NFSv3 插槽表

Linux 上的 NFSv3 效能取決於所呼叫的參數 `tcp_max_slot_table_entries`。

TCP 插槽表是與主機匯流排介面卡（HBA）併列深度相當的 NFSv3。這些表格可控制任何時間都可以處理的 NFS 作業數量。預設值通常為 16、這對於最佳效能而言太低。相反的問題發生在較新的 Linux 核心上、這會自動將 TCP 插槽表格限制增加到要求使 NFS 伺服器飽和的層級。

為了達到最佳效能並避免效能問題、請調整控制 TCP 插槽表的核心參數。

執行 `sysctl -a | grep tcp.*.slot_table` 並觀察下列參數：

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

所有 Linux 系統都應該包括在內 `sunrpc.tcp_slot_table_entries`、但只有部分包含在內 `sunrpc.tcp_max_slot_table_entries`。兩者都應設為 128。

 若未設定這些參數、可能會對效能造成重大影響。在某些情況下、效能會受到限制、因為 Linux 作業系統沒有發出足夠的 I/O 在其他情況下、隨著 Linux 作業系統嘗試發出的 I/O 數量超過可服務的數量、I/O 延遲也會增加。

I/O 排程器

Linux 核心可讓您以低層級控制 I/O 排程封鎖裝置的方式。

Linux 各版本的預設值差異極大。MySQL 建議您使用 NOOP 或是 deadline 在 Linux 上具有原生非同步 I/O (AIO) 的 I/O 排程器。一般而言、NetApp 客戶和內部測試都能透過 NoOps 獲得更好的結果。

MySQL 的 InnoDB 儲存引擎使用 Linux 上的非同步 I/O 子系統（原生 AIO）來執行預先讀取和寫入資料檔案頁面的要求。此行為由控制 `innodb_use_native_aio` 組態選項、預設為啟用。有了原生的整合式全功能電腦、I/O 排程器的類型對 I/O 效能有更大的影響。執行效能標竿、判斷哪一個 I/O 排程器可為您的工作負載和環境提供最佳結果。

如需設定 I/O 排程器的指示、請參閱相關的 Linux 和 MySQL 文件。

檔案描述元

若要執行、MySQL 伺服器需要檔案描述元、而且預設值不足。

它會使用它們來開啟新的連線、將資料表儲存在快取中、建立暫時資料表來解決複雜的查詢、以及存取持續性的查詢。如果 mysqld 在需要時無法開啟新檔案、它就能停止正常運作。此問題的常見症狀是錯誤 24 「開啟的檔案太多」。mysqld 可以同時開啟的檔案描述元數量是由定義的 open_files_limit 設定在組態檔案中的選項 (/etc/my.cnf)。但是 open_files_limit 也取決於作業系統的限制。這種相依性會使設定變數變得更複雜。

MySQL 無法設定 open_files_limit 選項高於下所指定的選項 ulimit 'open files'。因此、您必須在作業系統層級明確設定這些限制、才能讓 MySQL 視需要開啟檔案。檢查 Linux 檔案限制的方法有兩種：

- ulimit 命令會快速為您提供所允許或鎖定參數的詳細說明。執行此命令所做的變更並非永久性變更、將會在系統重新開機後清除。
- 變更為 /etc/security/limit.conf 檔案是永久性的、不受系統重新開機影響。

請務必同時變更使用者 mysql 的硬限制和軟限制。以下摘錄來自組態：

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

同時、請在中更新相同的組態 my.cnf 以完全使用開放式檔案限制。

儲存組態

NFS

MySQL 文件建議您將 NFSv4 用於 NAS 部署。

ONTAP NFS 傳輸大小

根據預設、ONTAP 將 NFS IO 大小限制為 64K。使用 MySQL 資料庫的隨機 IO 使用的區塊大小要小得多、遠低於 64K 上限。大型區塊 IO 通常是平行處理的、因此 64K 最大值也不是限制。

有些工作負載的上限為 64K、因此會造成限制。尤其是、如果資料庫執行的 IO 數量較少、但容量較大、例如完整表格掃描備份作業等單執行緒作業、將會更快、更有效率地執行。ONTAP 搭配資料庫工作負載的最佳 IO 處理大小為 256k。針對下列特定作業系統所列出的 NFS 裝載選項已相應從 64K 更新至 256k。

指定 ONTAP SVM 的最大傳輸大小可變更如下：

```
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
```

 切勿將 ONTAP 上允許的傳輸大小上限降至低於目前掛載之 NFS 檔案系統的 rsize/wsize 值。這可能會在某些作業系統中造成當機或甚至資料毀損。例如、如果 NFS 用戶端目前設定為 rsize/wsize 65536、則 ONTAP 最大傳輸大小可在 65536 到 1048576 之間調整、因為用戶端本身受到限制、因此沒有任何影響。將傳輸大小上限降至 65536 以下可能會損害可用度或資料。

• NetApp 推薦 *

 設定下列 NFSv4 Fstab (/etc/fstab) 設定：

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

 NFSv3 的常見問題是停電後鎖定的 InnoDB 記錄檔。使用時間或切換記錄檔可解決此問題。不過、NFSv4 有鎖定作業、並追蹤開啟的檔案和委派。

SAN

只要 I/O 和容量需求在單一 LUN 檔案系統的限制範圍內、就可以將較小的資料庫放在一對標準 LUN 上。例如、需要大約 2K 隨機 IOPS 的資料庫、可以裝載在單一 LUN 上的單一檔案系統上。同樣地、只有 100GB 大小的資料庫也能放在單一 LUN 上、而不會造成管理問題。

大型資料庫需要多個 LUN。例如、需要 10 萬 IOPS 的資料庫最可能需要至少八個 LUN。由於磁碟機的 SCSI 通道數量不足、單一 LUN 將成為瓶頸。10TB 資料庫同樣難以在單一 10TB LUN 上進行管理。邏輯磁碟區管理程式的設計旨在將多個 LUN 的效能和容量功能結合在一起、以改善效能和管理能力。

在這兩種情況下、一對 ONTAP 磁碟區應該足夠。透過簡單的組態、資料檔案 LUN 會與記錄 LUN 一樣置於專用磁碟區中。在邏輯 Volume Manager 組態下、資料檔案 Volume 群組中的所有 LUN 都將位於專用磁碟區、而記錄 Volume 群組的 LUN 則位於第二個專用磁碟區。

- NetApp 建議 * 在 SAN 上使用兩個檔案系統進行 MySQL 部署：
- 第一個檔案系統會儲存所有 MySQL 資料、包括資料表空間、資料和索引。
- 第二個檔案系統會儲存所有記錄（二進位記錄、慢速記錄和交易記錄）。

以這種方式分隔資料有多種原因、包括：

- 資料檔案和記錄檔的 I/O 模式各不相同。將它們區隔、就能透過 QoS 控制提供更多選項。
- 最佳化使用 Snapshot 技術需要能夠個別還原資料檔案。將資料檔案與記錄檔混合會干擾資料檔案還原。
- NetApp SnapMirror 技術可用於為資料庫提供簡單、低 RPO 的災難恢復功能、但是資料檔案和記錄檔需要不同的複寫排程。

使用這種基本的兩個磁碟區配置、以符合未來需求的解決方案、以便在需要時使用所有 ONTAP 功能。

- NetApp 建議 * 使用 ext4 檔案系統格式化磁碟機、因為有下列功能：
- 延伸的區塊管理方法、用於日誌檔案系統（JFS）中的區塊管理功能、以及延伸檔案系統（XFS）的延遲分配功能。
- ext4 允許檔案系統最多 1 個 exbibyte（ 2^{60} 位元組）、檔案最多 16 個 tebibytes（ $16 \cdot 2^{40}$ 位元組）。相反地、ext3 檔案系統僅支援 16TB 的最大檔案系統大小、最大檔案大小為 2TB。
- 在 ext4 檔案系統中、多區塊分配（mballoc）會在單一作業中為檔案分配多個區塊、而非如 ext3 中逐一分配。此組態可減少多次呼叫區塊分配器的成本、並最佳化記憶體配置。
- 雖然 XFS 是許多 Linux 套裝作業系統的預設值、但它以不同方式管理中繼資料、不適合某些 MySQL 組態。

- NetApp 建議 * 將 4K 區塊大小選項與 mkfs 公用程式搭配使用、以符合現有區塊 LUN 大小。

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

NetApp LUN 以 4KB 實體區塊儲存資料、產生八個 512 位元組的邏輯區塊。

如果您未設定相同的區塊大小、I/O 將無法正確對齊實體區塊、而且可能會在 RAID 群組中的兩個不同磁碟機中寫入、導致延遲。

請務必調整 I/O、以順利進行讀寫作業。但是、當 I/O 從不在實體區塊開頭的邏輯區塊開始時、I/O 會未對齊。I/O 作業只有在邏輯區塊（實體區塊中的第一個邏輯區塊）開始時才會對齊。

版權資訊

Copyright © 2026 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP 「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變更本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。