資料庫組態
Enterprise applications

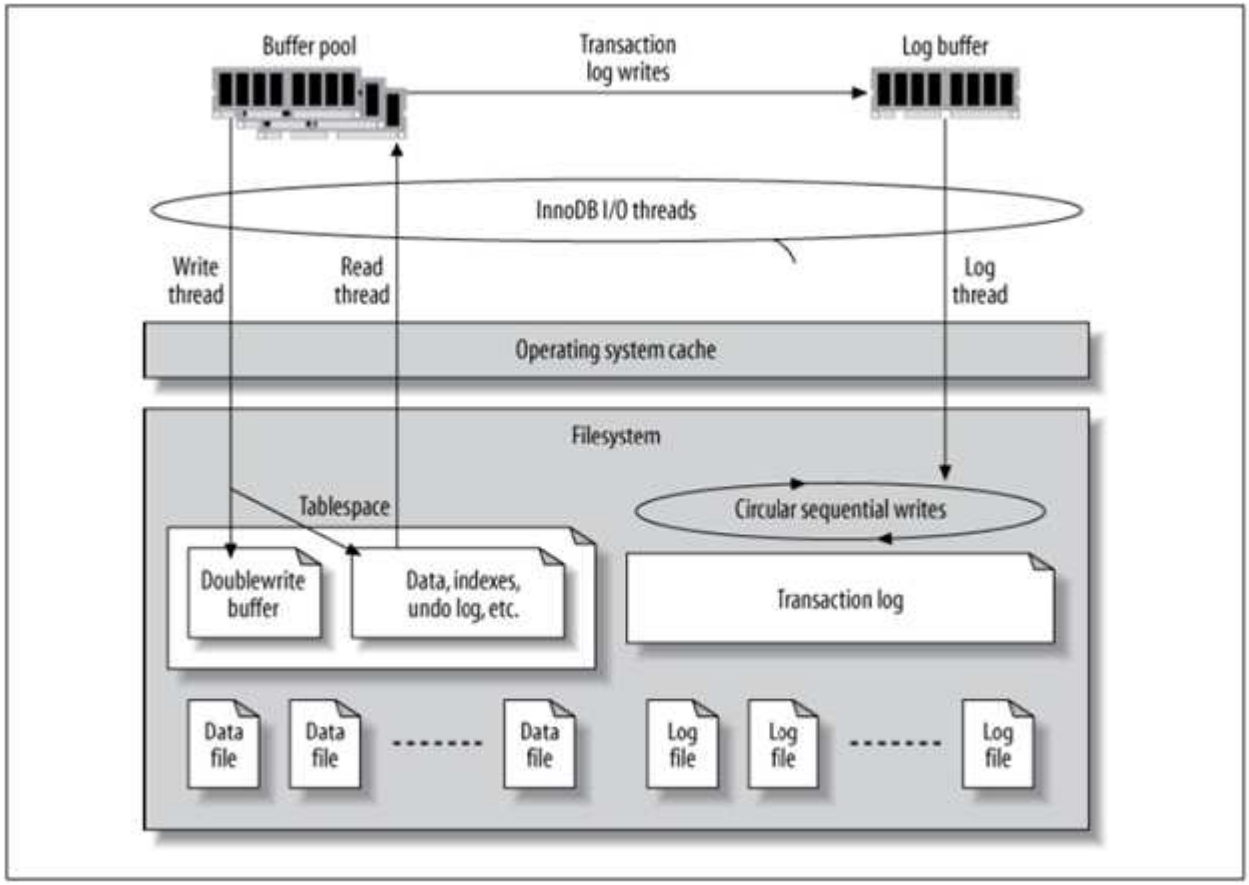
NetApp
May 09, 2024

目錄

資料庫組態

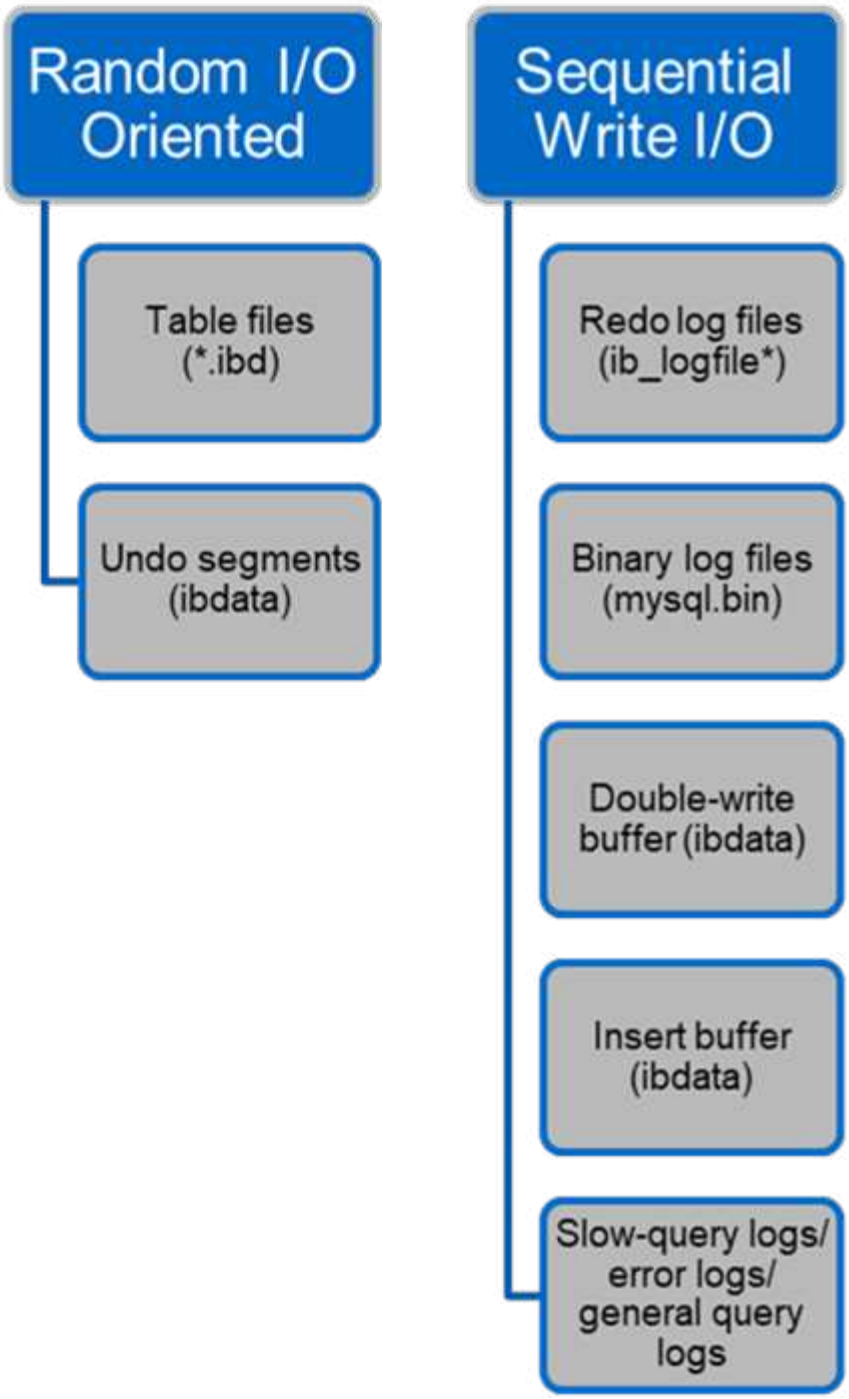
MySQL 和 InnoDB

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。如果您只執行簡短的交易或查詢、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。

MySQL 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
InnoDB_log_file_size	256M
InnoDB_Flush 記錄 _AT_TRx_Commit	2.
InnoDB_doublewrite	0%
InnoDB_Flush 方法	fsync
InnoDB_緩 衝區 _Pool_size	11g
InnoDB_IO_capAC	8192
InnoDB_緩 衝區集區執行個體	8.
InnoDB_LRU_SCAN_depth	8192
open_file_limit	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 緩衝區集區中）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` 針對效能標竿、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（包括髒頁的排清率、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、並以提早排清的方式浪費 I/O 作業、進而嚴重損害效能。

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 + `max_connections` + (`table_open_cache` x 2 ）
- `max_connections` x 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。