Oracle資料庫

Enterprise applications

NetApp
May 09, 2024

目錄

Oracle資料庫

ONTAP 上的 Oracle 資料庫

ONTAP 專為 Oracle 資料庫所設計。數十年來、 ONTAP 已針對關聯式資料庫 I/O 的獨特需求進行最佳化、並特別建立多項 ONTAP 功能、以滿足 Oracle 資料庫的需求、甚至是 Oracle Inc. 本身的要求。

本文件取代先前發佈的技術報告 _TR-3633 ： ONTAP 上的 Oracle 資料庫； TR-4591 ： Oracle 資料保護：備份、還原、複寫； TR-4592 ： MetroCluster 上的 Oracle ；以及 TR-4534 ：將 Oracle 資料庫移轉至 NetApp 儲存系統

除了 ONTAP 為您的資料庫環境帶來價值的許多可能方式之外、也有許多使用者需求、包括資料庫大小、效能需求和資料保護需求。NetApp 儲存設備的已知部署包括從虛擬化環境（約 6 、 000 個在 VMware ESX 下執行的資料庫）到目前規模為 996TB 且不斷成長的單一執行個體資料倉儲、應有盡有。因此、在 NetApp 儲存設備上設定 Oracle 資料庫的最佳實務做法很少。

在 NetApp 儲存設備上操作 Oracle 資料庫的需求有兩種解決方法。首先、當有明確的最佳實務做法時、將會特別加以說明。我們將根據 Oracle 儲存解決方案架構設計師的特定業務需求、詳細說明許多設計考量。

組態ONTAP

RAID 和 Oracle 資料庫

RAID 是指使用備援功能來保護資料、避免磁碟機遺失。

在設定用於 Oracle 資料庫和其他企業應用程式的 NetApp 儲存設備時、偶爾會出現 RAID 層級的問題。許多舊版 Oracle 儲存陣列組態的最佳實務做法都包含使用 RAID 鏡射和 / 或避免使用特定類型 RAID 的警告。雖然這些來源提出有效點、但這些來源不適用於 RAID 4 、以及 ONTAP 中使用的 NetApp RAID DP 和 RAID-TEC 技術。

RAID 4 、 RAID 5 、 RAID 6 、 RAID DP 和 RAID-TEC 都使用同位元檢查來確保磁碟機故障不會導致資料遺失。與鏡像相比、這些 RAID 選項可提供更好的儲存使用率、但大多數 RAID 實作都有影響寫入作業的缺點。在其他 RAID 實作上完成寫入作業可能需要多個磁碟機讀取才能重新產生同位元資料、這是一種通常稱為 RAID 懲罰的程序。

不過、 ONTAP 並不會因此而受到此 RAID 處罰。這是因為 NetApp WAFL （隨處寫入檔案配置）與 RAID 層整合。寫入作業會整合在 RAM 中、並準備為完整的 RAID 等量磁碟區、包括同位元檢查產生。ONTAP 不需要執行讀取即可完成寫入、這表示 ONTAP 和 WAFL 可以避免 RAID 的損失。對於延遲關鍵作業（例如重作記錄）的效能不受阻礙、隨機的資料檔案寫入不會因重新產生同位元檢查而導致任何 RAID 損失。

在統計可靠性方面、即使是 RAID DP 也能提供比 RAID 鏡射更好的保護。主要問題是 RAID 重建期間對磁碟機的需求。有了鏡射 RAID 集、當磁碟機在重建時發生故障、而在 RAID 組中重建其合作夥伴時、資料遺失的風險遠高於 RAID DP 組中發生三重磁碟機故障的風險。

Oracle 資料庫與儲存容量管理

使用可預測、可管理的高效能企業儲存設備來管理資料庫或其他企業應用程式、需要磁碟

機上的一些可用空間來進行資料和中繼資料管理。所需的可用空間量取決於使用的磁碟機類型和業務程序。

可用空間定義為任何不用於實際資料的空間、並包括集合體本身的未分配空間、以及組成磁碟區內的未使用空間。也必須考慮精簡配置。例如、某個磁碟區可能包含 1TB LUN 、其中只有 50% 被實際資料使用。在精簡佈建的環境中、這似乎是消耗 500GB 的空間。不過、在完全佈建的環境中、 1TB 的完整容量似乎正在使用中。500GB 的未分配空間會隱藏起來。實際資料未使用此空間、因此應納入總可用空間的計算。

NetApp 對於企業應用程式所使用的儲存系統建議如下：

SSD 集合體、包括 **AFF** 系統

* NetApp 建議 * 至少有 10% 的可用空間。這包括所有未使用的空間、包括集合體或磁碟區內的可用空間、以及因使用完整資源配置而分配但實際資料未使用的任何可用空間。邏輯空間並不重要、問題在於實際可用的實體空間可用於資料儲存。

10% 可用空間的建議非常保守。SSD 集合體可支援使用率更高的工作負載、而不會對效能造成任何影響。不過、隨著 Aggregate 的使用率增加、如果未仔細監控使用率、則用盡空間的風險也會增加。此外、當系統的容量達到 99% 時、可能不會導致效能降低、但在訂購額外硬體時、可能需要管理人員努力避免系統完全裝滿、而且可能需要一些時間來採購和安裝額外的磁碟機。

HDD 集合體、包括 **Flash Pool** 集合體

* 使用旋轉磁碟機時、 NetApp 建議 * 至少有 15% 的可用空間。這包括所有未使用的空間、包括集合體或磁碟區內的可用空間、以及因使用完整資源配置而分配但實際資料未使用的任何可用空間。在免費語音方法中、效能將受到 10% 的影響。

Oracle 資料庫與儲存虛擬機器

Oracle 資料庫儲存管理集中在儲存虛擬機器（ SVM ）上

SVM 在 ONTAP CLI 中稱為 vserver 、是基本的儲存功能單元、比較 SVM 與 VMware ESX 伺服器上的客體是很有用的。

首次安裝時、 ESX 沒有預先設定的功能、例如代管來賓作業系統或支援終端使用者應用程式。它是空容器、直到定義虛擬機器（ VM ）為止。ONTAP 類似。第一次安裝 ONTAP 時、只有建立 SVM 、它才具備資料服務功能。定義資料服務的是 SVM 特性設定。

與儲存架構的其他層面一樣、 SVM 和邏輯介面（ LIF ）設計的最佳選項、在很大程度上取決於擴充需求和業務需求。

SVM

我們並未正式提供 ONTAP 的 SVM 資源配置最佳實務做法。正確的方法取決於管理和安全要求。

大多數客戶只需操作一部主要 SVM 、即可滿足大部分的日常需求、然後針對特殊需求建立少量 SVM 。例如、您可能想要建立：

- 由專業團隊管理的關鍵業務資料庫 SVM
- 開發群組的 SVM 已獲得完整的管理控制權、可讓他們獨立管理自己的儲存設備

- 必須限制管理團隊的 SVM 、用於處理敏感業務資料、例如人力資源或財務報告資料

在多租戶環境中、每個租戶的資料都可以獲得專用的 SVM 。每個叢集、 HA 配對和節點的 SVM 和生命量限制取決於所使用的傳輸協定、節點模型和 ONTAP 版本。 請參閱 "NetApp Hardware Universe" 針對這些限制。

ONTAP QoS 的 Oracle 資料庫效能管理

安全有效地管理多個 Oracle 資料庫、需要有效的 QoS 策略。原因在於現代儲存系統的效能功能不斷提升。

具體而言、由於採用 All Flash 儲存設備的情況增加、因此能夠整合工作負載。依賴旋轉媒體的儲存陣列往往只支援有限數量的 I/O 密集工作負載、因為舊版旋轉式磁碟機技術的 IOPS 功能有限。一或兩個高作用中的資料庫會在儲存控制器達到限制之前、使基礎磁碟機飽和。這種情況已經改變。即使是功能最強大的儲存控制器、相對少數 SSD 磁碟機的效能也能達到飽和。這意味著控制器的完整功能可以充分發揮、而不會因為旋轉媒體延遲尖峰而突然降低效能。

舉例來說、簡單的雙節點 HA AFF A800 系統能夠在延遲超過 1 毫秒之前、提供高達 100 萬次的隨機 IOPS 服務。只有很少單一工作負載會達到這類層級。充分利用此 AFF A800 系統陣列、將需要託管多個工作負載、同時確保可預測性、這需要 QoS 控制。

ONTAP 中有兩種服務品質（ QoS ）： IOPS 和頻寬。QoS 控制可套用至 SVM 、磁碟區、 LUN 和檔案。

IOPS QoS

IOPS QoS 控制顯然是以指定資源的 IOPS 總計為基礎、但 IOPS QoS 的許多層面可能並不符合直覺。剛開始有幾位客戶對於達到 IOPS 臨界值時、延遲明顯增加感到困惑。延遲增加是限制 IOPS 的自然結果。從邏輯上講、它的運作方式與權杖系統類似。例如、如果包含資料檔案的特定磁碟區有 10K IOPS 限制、則每個到達的 I/O 都必須先接收權杖才能繼續處理。只要在指定的秒數內使用的權杖不超過 10K 、就不會有延遲。如果 IO 作業必須等待接收其權杖、則此等待會顯示為額外延遲。工作負載相對於 QoS 限制的推動越大、每個 IO 在佇列中等待處理的時間就越長、使用者認為延遲越高。

> 將 QoS 控制套用至資料庫交易 / 重做記錄資料時、請務必謹慎。雖然重做記錄的效能需求通常比資料檔案低很多、但重做記錄活動卻很繁瑣。IO 會以簡短的脈衝形式發生、而顯示適合平均重做 IO 層級的 QoS 限制、對於實際需求而言可能太低。結果可能會造成嚴重的效能限制、因為每次重做記錄突增時、 QoS 都會啟動。一般而言、重作和歸檔記錄不應受到 QoS 的限制。

頻寬 QoS

並非所有 I/O 大小都相同。例如、資料庫可能會執行大量的小區塊讀取、導致達到 IOPS 臨界值、 但資料庫也可能執行完整的資料表掃描作業、這項作業會由極少數的大量區塊讀取所組成、佔用大量頻寬、但 IOPS 相對較少。

同樣地、 VMware 環境在開機期間可能會產生極高的隨機 IOPS 、但在外部備份期間執行的 IO 會較少、但會較大。

有時有效管理效能需要 IOPS 或頻寬 QoS 限制、甚至兩者都需要。

最低 / 保證的 QoS

許多客戶尋求的解決方案都包含保證的 QoS 、這比看起來更難達成、而且可能相當浪費。例如、如果要放置 10 個具有 10K IOPS 保證的資料庫、就必須針對所有 10 個資料庫同時以 10K IOPS 執行的情況來調整系統規模、總共需要 10 萬個。

最適合用於最低 QoS 控制的是保護關鍵工作負載。例如、假設 ONTAP 控制器的 IOPS 最高可達 50 萬、同時混合了生產與開發工作負載。您應該將 QoS 原則上限套用至開發工作負載、以防止任何指定的資料庫壟斷控制器。然後、您可以將最低 QoS 原則套用至正式作業工作負載、以確保它們在需要時隨時都能使用所需的 IOPS 。

調適性**QoS**

調適性 QoS 是指 ONTAP 功能、其中 QoS 限制是根據儲存物件的容量而定。它很少用於資料庫、因為資料庫的大小與其效能需求之間通常沒有任何連結。大型資料庫可能幾乎無法運作、而較小的資料庫則可能是 IOPS 最密集的資料庫。

Adaptive QoS 對於虛擬化資料存放區非常有用、因為這類資料集的 IOPS 需求往往與資料庫的總大小相關。較新的資料存放區包含 1TB 的 VMDK 檔案、可能需要的效能約為 2TB 資料存放區的一半。Adaptive QoS 可讓您在資料存放區填入資料時、自動增加 QoS 限制。

Oracle 資料庫與 **ONTAP** 效率功能

ONTAP 空間效率功能已針對 Oracle 資料庫進行最佳化。在幾乎所有情況下、最佳方法是在啟用所有效率功能的情況下、保留預設值。

空間效率功能（例如壓縮、壓縮和重複資料刪除）的設計、是為了增加符合特定實體儲存量的邏輯資料量。結果是降低成本和管理成本。

在高層級、壓縮是一種數學程序、可偵測及編碼資料模式、以減少空間需求。相反地、重複資料刪除功能會偵測實際重複的資料區塊、並移除額外的複本。資料實作可讓多個邏輯區塊在媒體上共用相同的實體區塊。

請參閱以下關於精簡配置的章節、以瞭解儲存效率與部分保留之間互動的說明。

壓縮

在提供 All Flash 儲存系統之前、以陣列為基礎的壓縮價值有限、因為大多數 I/O 密集的工作負載都需要大量磁碟來提供可接受的效能。儲存系統的容量總是比所需的容量大得多、這是大量磁碟機的副作用。固態儲存設備的興起、改變了這種情況。不再需要純粹為了獲得良好效能而大幅過度配置磁碟機。儲存系統中的磁碟機空間可與實際容量需求相符。

固態硬碟機（ SSD ）的 IOPS 容量增加、幾乎總是比旋轉硬碟機節省成本、但壓縮技術可以增加固態媒體的有效容量、進而進一步節省成本。

壓縮資料的方法有好幾種。許多資料庫都包含自己的壓縮功能、但在客戶環境中很少會發現這種情況。其原因通常是對壓縮資料的 * 變更 * 效能會受到影響、而對於某些應用程式而言、資料庫層級壓縮的授權成本較高。最後、對資料庫作業的整體效能影響。對於執行資料壓縮與解壓縮的 CPU 、而非實際的資料庫工作、支付高昂的每 CPU 授權成本是不合理的。更好的選擇是將壓縮工作卸載到儲存系統。

自適應壓縮

即使在以微秒為單位測量延遲的 All Flash 環境中、主動式壓縮也已針對企業工作負載進行徹底測試、且未對效能產生任何影響。有些客戶甚至報告使用壓縮技術時效能會提高、因為資料會保持在快取中的壓縮、有效增加控制器中可用的快取數量。

ONTAP 以 4KB 單位管理實體區塊。自適應壓縮使用 8KB 的預設壓縮區塊大小、也就是以 8KB 為單位壓縮資料。這與關係式資料庫最常使用的 8KB 區塊大小相符。隨著將更多資料壓縮成單一單元、壓縮演算法就會變得更有效率。32 KB 壓縮區塊大小比 8 KB 壓縮區塊單元更具空間效率。這表示使用預設 8KB 區塊大小的調適式

壓縮確實會導致效率稍微降低、但使用較小的壓縮區塊大小也有很大的好處。資料庫工作負載包括大量的覆寫活動。若要覆寫 32 KB 壓縮資料區塊的 8KB 資料、必須讀回整個 32 KB 的邏輯資料、將其解壓縮、更新所需的 8 KB 區域、重新壓縮、然後將整個 32 KB 寫入磁碟機。這對儲存系統來說是非常昂貴的作業、也是因為某些競爭儲存陣列以較大的壓縮區塊大小為基礎、也會對資料庫工作負載造成重大效能損失的原因。

調適式壓縮所使用的區塊大小最多可增加至 32KB 。這可能會改善儲存效率、而且當大量的這類資料儲存在陣列上時、應該考慮用於靜態檔案、例如交易記錄檔和備份檔案。在某些情況下、使用 16KB 或 32KB 區塊大小的作用中資料庫、也可能因為增加適應式壓縮的區塊大小而受惠。請洽詢 NetApp 或合作夥伴代表、瞭解這是否適合您的工作負載。

在串流備份目的地上、不應將大於 8KB 的壓縮區塊大小與重複資料刪除一起使用。原因是備份資料的細微變更會影響 32KB 壓縮時間。如果視窗移動、則產生的壓縮資料會在整個檔案中有所不同。重複資料刪除是在壓縮之後進行、這表示重複資料刪除引擎會以不同的方式檢視每個壓縮備份。如果需要重複資料刪除串流備份、則只應使用 8KB 區塊調適性壓縮。調適性壓縮較為理想、因為它的區塊大小較小、不會中斷重複資料刪除的效率。由於類似的原因、主機端壓縮也會影響重複資料刪除的效率。

壓縮對齊

資料庫環境中的調適性壓縮需要考量壓縮區塊對齊。這樣做只是對隨機覆寫非常特定區塊的資料的考量。這種方法的概念與整體檔案系統對齊方式類似、檔案系統的開始必須與 4K 裝置邊界對齊、檔案系統的區塊大小必須是 4K 的倍數。

例如、只有在檔案與檔案系統本身的 8KB 邊界對齊時、才會壓縮寫入 8KB 檔案。這表示它必須落在檔案的前 8KB 、檔案的第二 8KB 等。確保正確對齊的最簡單方法是使用正確的 LUN 類型、建立的任何分割區都應該與 8K 的倍數裝置開始偏移、並使用資料庫區塊大小的倍數檔案系統區塊大小。

備份或交易記錄等資料會循序寫入跨越多個區塊的作業、所有這些區塊都會被壓縮。因此、不需要考慮對齊。唯一令人擔憂的 I/O 模式是隨機覆寫檔案。

資料壓縮

資料壓縮技術可改善壓縮效率。如前所述、僅有調適式壓縮功能、最多可節省 2 ： 1 、因為它僅限於在 4KB WAFL 區塊中儲存 8KB I/O 。較大區塊大小的壓縮方法可提供更好的效率。不過、這些資料不適合受到小型區塊覆寫的資料。解壓縮 32KB 的資料單元、更新 8KB 部分、重新壓縮及回寫磁碟機、都會產生額外的負荷。

資料壓縮的運作方式是允許將多個邏輯區塊儲存在實體區塊內。例如、含有高度壓縮資料（例如文字或部分完整區塊）的資料庫、可能會從 8KB 壓縮至 1KB 。如果沒有壓縮、 1KB 的資料仍會佔用整個 4KB 區塊。即時資料壓縮功能可將 1KB 的壓縮資料與其他壓縮資料一起儲存在 1KB 的實體空間中。這不是一項壓縮技術、只是在磁碟機上分配空間的一種更有效率的方法、因此不應產生任何可偵測的效能影響。

節省的程度各不相同。已壓縮或加密的資料通常無法進一步壓縮、因此資料集無法從資料壓縮中獲益。相反地、新初始化的資料檔案僅包含區塊中繼資料和零、最多可壓縮至 80 ： 1 。

對溫度敏感的儲存效率

溫度敏感儲存效率（ TSSE ）是 ONTAP 9.8 及更新版本中提供的產品、它仰賴區塊存取熱圖來識別不常存取的區塊、並以更高的效率加以壓縮。

重複資料刪除

重複資料刪除是從資料集移除重複的區塊大小。例如、如果 10 個不同的檔案中存在相同的 4KB 區塊、重複資

料刪除會將所有 10 個檔案中的 4KB 區塊重新導向至相同的 4KB 實體區塊。結果是該資料的效率提升 10 ： 1 。

VMware 來賓開機 LUN 等資料通常會極好地刪除重複資料、因為這些資料包含相同作業系統檔案的多個複本。效率達到 100 ： 1 以上。

部分資料不包含重複資料。例如、 Oracle 區塊包含資料庫的全域唯一標頭、以及近乎唯一的標尾。因此、 Oracle 資料庫的重複資料刪除功能很少能節省 1% 以上的成本。使用 MS SQL 資料庫進行重複資料刪除的效果稍微好一些、但區塊層級的獨特中繼資料仍是一項限制。

在某些情況下、使用 16KB 和大型區塊大小的資料庫可節省高達 15% 的空間。每個區塊的初始 4KB 包含全域唯一的標頭、最後 4KB 區塊則包含近乎獨特的標尾。內部區塊是重複資料刪除的候選項目、但實際上、這幾乎完全歸功於重複資料刪除零位資料。

許多競爭陣列都宣稱能夠根據資料庫複製多次的假設來刪除重複的資料庫。在這方面，也可以使用 NetApp 重複資料刪除技術，但 ONTAP 提供更好的選擇： NetApp FlexClone 技術。最終結果相同；資料庫的多個複本會建立共用大部分基礎實體區塊。使用 FlexClone 比花時間複製資料庫檔案然後刪除複製檔案更有效率。實際上，它是不重複數據刪除，而不是重複數據刪除，因爲從一開始就不會創建重複數據。

效率與精簡配置

效率功能是精簡配置的形式。例如、佔用 100GB 磁碟區的 100GB LUN 可能會壓縮至 50GB 。由於磁碟區仍為 100GB 、因此尚未實現實際節省。必須先縮小磁碟區的大小、才能將儲存的空間用於系統的其他位置。如果稍後變更為 100GB LUN 、則資料的壓縮性會降低、 LUN 的大小會增加、而且磁碟區可能會填滿。

強烈建議採用精簡配置、因為它可以簡化管理、同時大幅改善可用容量、並節省相關成本。原因很簡單：資料庫環境通常包含大量的空空間、大量的磁碟區和 LUN 、以及可壓縮的資料。如果磁碟區和 LUN 的儲存空間有一天 100% 滿、而且包含 100% 不可壓縮的資料、則大量資源配置會導致保留空間。這種情況不太可能發生。精簡配置可回收空間並在其他地方使用、並可讓容量管理以儲存系統本身為基礎、而非許多較小的磁碟區和 LUN 。

有些客戶偏好針對特定工作負載使用完整資源配置、或是根據既定的營運和採購實務做法。

- 注意： * 如果磁碟區是完整配置的磁碟區、則必須小心將該磁碟區的所有效率功能完全停用、包括使用解壓縮和移除重複資料刪除 `sis undo` 命令。Volume 不應出現在中 `volume efficiency show` 輸出。如果有、則磁碟區仍會部分設定為使用效率功能。因此、覆寫保證會以不同的方式運作、這會增加組態超視導致磁碟區意外用盡空間的機會、進而導致資料庫 I/O 錯誤。

效率最佳實務做法

NetApp 建議：

AFF 預設值

在 All Flash AFF 系統上執行的 ONTAP 上建立的磁碟區會自動精簡佈建、並啟用所有的內嵌效率功能。雖然資料庫通常無法從重複資料刪除中獲益、而且可能包含不可壓縮的資料、但預設設定仍適用於幾乎所有的工作負載。ONTAP 旨在有效處理所有類型的資料和 I/O 模式、無論是否能節省成本。只有在充分瞭解理由且有偏離的好處時、才應變更預設值。

一般建議

- 如果磁碟區和（或） LUN 並未精簡配置、您必須停用所有效率設定、因為使用這些功能並不會節省成本、而將複雜資源配置與啟用空間效率的組合、可能會導致非預期的行為、包括空間不足的錯誤。

- 如果資料不需要覆寫、例如備份或資料庫交易記錄檔、您可以在冷卻週期較短的情況下啟用 TSSE 、以達到更高的效率。
- 某些檔案可能包含大量不可壓縮的資料、例如、當檔案的應用程式層級已啟用壓縮時、就會進行加密。如果上述任何情況屬實、請考慮停用壓縮、以便在包含可壓縮資料的其他磁碟區上執行更有效率的作業。
- 請勿將 32KB 壓縮和重複資料刪除同時用於資料庫備份。請參閱一節 [自適應壓縮] 以取得詳細資料。

使用 **Oracle** 資料庫進行精簡配置

簡化 Oracle 資料庫的資源配置需要仔細規劃、因為結果是在儲存系統上設定的空間比實際可用的空間更多。這項工作非常值得、因為正確完成後、可大幅節省成本、並改善管理能力。

精簡配置有多種形式、是 ONTAP 為企業應用程式環境提供的許多功能不可或缺的一部分。精簡配置也與效率技術密切相關、原因相同：效率功能可儲存比儲存系統技術更多的邏輯資料。

幾乎任何快照的使用都需要精簡配置。例如、 NetApp 儲存設備上的典型 10TB 資料庫、包含約 30 天的快照。這種配置可在作用中的檔案系統中看到大約 10TB 的資料、而在快照中則有 300TB 的資料。總儲存容量為 310TB 、通常位於大約 12TB 到 15TB 的空間上。作用中資料庫消耗 10TB 、而其餘 300TB 的資料僅需要 2TB 到 5TB 的空間、因為只會儲存對原始資料所做的變更。

複製也是精簡配置的範例。一位主要 NetApp 客戶建立 40 個 80 TB 資料庫的複本、供開發人員使用。如果使用這些複本的 40 位開發人員都在每個資料檔案中覆寫每個區塊、則需要超過 3.2PB 的儲存空間。實際上，週轉率較低，而且由於磁碟機上只儲存變更，因此集體空間需求接近 40 TB 。

空間管理

由於資料變更率可能會意外增加、因此精簡配置應用程式環境時必須謹慎處理。例如、如果資料庫資料表重新編製索引、或是將大規模的修補套用至 VMware 來賓系統、快照所造成的空間使用量就會迅速增加。錯誤的備份可能會在很短的時間內寫入大量資料。最後、如果檔案系統在非預期的情況下用盡可用空間、可能很難恢復某些應用程式。

幸運的是、這些風險可以透過仔細設定來解決 `volume-autogrow` 和 `snapshot-autodelete` 原則。如同其名稱所暗示、這些選項可讓使用者建立原則、以自動清除快照佔用的空間、或是增加磁碟區以容納額外資料。有許多選項可供選擇、需求因客戶而異。

請參閱 "邏輯儲存管理文件" 以完整討論這些功能。

部分保留

「部分保留」是指磁碟區中 LUN 在空間效率方面的行為。選項 `fractional-reserve` 設為 100% 、磁碟區中的所有資料在任何資料模式下都能達到 100% 的營業額、而不會耗盡磁碟區上的空間。

例如、假設資料庫位於 1TB 磁碟區中的單一 250GB LUN 上。建立快照將會立即在磁碟區中保留額外的 250GB 空間、以確保磁碟區不會因任何原因而用盡空間。使用分數保留通常是浪費、因為資料庫磁碟區中的每個位元組都不太可能需要覆寫。沒有理由為永遠不會發生的事件預留空間。不過、如果客戶無法監控儲存系統中的空間使用量、而且必須確定空間永遠不會用盡、則使用快照需要 100% 的部分保留。

壓縮與重複資料刪除

壓縮和重複資料刪除都是精簡配置的形式。例如、 50TB 的資料佔用空間可能會壓縮至 30TB 、因此可節省 20TB 。為了讓壓縮產生任何效益、其中某些 20TB 必須用於其他資料、或是儲存系統必須購買的容量低於

50TB 。因此、儲存的資料量比儲存系統技術上的資料量還多。從資料觀點來看、即使磁碟機僅佔用 30TB 、資料仍有 50TB 。

資料集的可壓縮性隨時都會變更、這會導致實際空間的使用量增加。這種使用量的增加意味著、在監控和使用方面、必須像其他形式的精簡配置一樣管理壓縮 `volume-autogrow` 和 `snapshot-autodelete`。

有關壓縮和重複資料刪除的詳細資訊、請參閱連結： efficiency.html 一節

壓縮與部分保留

壓縮是一種精簡配置形式。部分保留會影響壓縮的使用、並附有一個重要附註；在建立快照之前、會保留空間。通常、只有存在快照時、部分保留才會很重要。如果沒有快照、則部分保留並不重要。這不是壓縮的情況。如果在具有壓縮功能的磁碟區上建立 LUN 、 ONTAP 會保留空間以容納快照。在組態期間、這種行為可能會令人困惑、但這是預期的。

舉例來說、請考慮使用 5GB LUN 的 10GB 磁碟區、該磁碟區已壓縮至 2.5GB 、但沒有快照。請考慮以下兩種情況：

- 分數保留 = 100 會導致 7.5 GB 使用率
- 部分保留量 =0 會導致使用率為 2.5GB

第一個案例包括目前資料使用 2.5 GB 的空間、以及 5 GB 的空間、可在預期使用快照時、讓來源的營業額達到 100% 。第二個案例不會保留額外空間。

雖然這種情況可能令人困惑、但實際上並不可能發生。壓縮意味著精簡配置、而 LUN 環境中的精簡配置則需要部分保留。壓縮資料永遠可以被無法壓縮的東西覆寫、這表示必須精簡配置磁碟區以進行壓縮、以節省任何成本。

- NetApp 建議 * 下列保留組態：
- 設定 `fractional-reserve` 與一起進行基本容量監控時為 0 `volume-autogrow` 和 `snapshot-autodelete`。
- 設定 `fractional-reserve` 如果沒有監控能力、或在任何情況下都無法排放空間、則達到 100 。

可用空間和 **LVM** 空間分配

在檔案系統環境中自動精簡配置作用中 LUN 的效率、可能會隨著資料刪除而隨時間而喪失。除非刪除的資料會以零覆寫（另請參閱 "ASMRU" 或是隨著修剪 / 取消對應空間回收而釋放空間、「清除」資料會佔用檔案系統中越來越多的未分配空白空間。此外、在許多資料庫環境中、主動式 LUN 的精簡配置功能有限、因為資料檔案會在建立時初始化為全尺寸。

仔細規劃 LVM 組態可提高效率、並將儲存資源配置和 LUN 調整大小的需求降至最低。當使用 Veritas VxVM 或 Oracle ASM 等 LVM 時、基礎 LUN 會分割成僅在需要時才使用的範圍。例如、如果資料集的大小從 2TB 開始、但隨時間而成長至 10TB 、則此資料集可放置在配置在 LVM 磁碟群組中的 10TB 精簡配置 LUN 上。在建立時、它只會佔用 2TB 的空間、而且只會在分配範圍以容納資料成長時、才會要求額外的空間。只要監控空間、此程序就安全無虞。

Oracle 資料庫和 ONTAP 控制器容錯移轉 / 切換

需要瞭解儲存設備接管和切入功能、才能確保 Oracle 資料庫作業不會因這些作業而中斷。

此外、如果不正確使用、接管和切入作業所使用的引數可能會影響資料完整性。

- 在正常情況下、傳入的寫入資料會同步鏡射至指定的控制器、以供其合作夥伴使用。在 NetApp MetroCluster 環境中、寫入也會鏡射到遠端控制器。除非寫入儲存在所有位置的非揮發性媒體中、否則不會對主機應用程式進行確認。
- 儲存寫入資料的媒體稱為非揮發性記憶體或 NVMEM 。它有時也稱為非揮發性隨機存取記憶體（ NVRAM ）、雖然它是日誌、但仍可視為寫入快取。在正常作業中、不會讀取來自 NVMEM 的資料；只有在軟體或硬體故障時、才會用來保護資料。當資料寫入磁碟機時、資料會從系統的 RAM 傳輸、而非從 NVMEM 傳輸。
- 在接管作業期間、高可用度（ HA ）配對中的一個節點會接管其合作夥伴的作業。切換基本上相同、但適用於遠端節點接管本機節點功能的 MetroCluster 組態。

在例行維護作業期間、儲存設備接管或切換作業應該是透明的、但網路路徑變更時、操作可能會短暫暫停。然而、網路連線可能很複雜、而且容易出錯、因此 NetApp 強烈建議您在將儲存系統投入生產之前、先徹底測試接管和轉換作業。這樣做是確保正確設定所有網路路徑的唯一方法。在 SAN 環境中、請仔細檢查命令的輸出 `sanlun lun show -p` 以確保所有預期的主要和次要路徑都可用。

發出強制接管或關機時、請務必小心。使用這些選項強制變更儲存組態、表示會忽略擁有磁碟機的控制器狀態、而替代節點則強制控制磁碟機。不正確地強制接管可能會導致資料遺失或毀損。這是因為強制接管或變更會捨棄 NVMEM 的內容。在接管或切換完成後、資料遺失表示儲存在磁碟機上的資料可能會從資料庫的角度還原到稍微舊的狀態。

很少需要強制接管正常的 HA 配對。在幾乎所有故障情況下、節點都會關機並通知合作夥伴、以便進行自動容錯移轉。有些邊緣情況、例如發生滾動故障、節點之間的互連中斷、然後一個控制器遺失、需要強制接管。在這種情況下、節點之間的鏡像會在控制器故障之前遺失、這表示當機的控制器將不再擁有正在進行的寫入複本。然後需要強制接管、這表示資料可能會遺失。

同樣的邏輯也適用於 MetroCluster 轉換。在正常情況下、可進行的作業幾乎透明化。然而、災難可能會導致仍在運作的站台和災難站台之間的連線中斷。從仍在運作的站台觀點來看、問題可能只是站台之間的連線中斷、而原始站台可能仍在處理資料。如果節點無法驗證主控制器的狀態、則只能強制進行移轉。

- NetApp 建議 * 採取下列預防措施：
- 請務必小心、避免意外強制接管或切入。一般而言、不應強制、強制變更可能會導致資料遺失。
- 如果需要強制接管或移除、請確定應用程式已關機、所有檔案系統均已卸除、且邏輯 Volume Manager （ LVM ）磁碟區群組已移除。必須卸載 ASM 磁碟群組。
- 在強制 MetroCluster 轉換的情況下、請將故障節點從所有仍在運作的儲存資源中隔離。如需詳細資訊、請參閱 MetroCluster 管理與災難恢復指南、以取得相關版本的 ONTAP 。

MetroCluster 和多個集合體

MetroCluster 是一種同步複寫技術、可在連線中斷時切換至非同步模式。這是客戶最常提出的要求、因為保證同步複寫意味著站台連線中斷會導致資料庫 I/O 完全停止、使資料庫停止運作。

透過 MetroCluster 、集合體在連線恢復後會快速重新同步。與其他儲存技術不同、 MetroCluster 在站台故障後絕不應要求完整的重新鏡射。只能運送差異變更。

在跨集合體的資料集中、在循環災難案例中需要額外的資料恢復步驟、風險很小。具體而言、如果（ a ）站台之間的連線中斷、（ b ）連線恢復、（ c ）集合體會達到某種狀態、其中有些是同步的、有些則不是同步的、然後（ d ）主站台會遺失、結果是無法運作的站台、而集合體彼此之間不會同步。如果發生這種情況、資料集的某些部分會彼此同步、因此無法在沒有恢復的情況下啟動應用程式、資料庫或資料存放區。如果資料集橫跨整

個集合體、 NetApp 強烈建議您利用快照式備份、搭配眾多可用工具之一、在這種不尋常的情況下驗證快速的可恢復性。

資料庫組態

Oracle 資料庫區塊大小

ONTAP 內部使用可變的區塊大小、這表示 Oracle 資料庫可以設定任何所需的區塊大小。不過、檔案系統區塊大小可能會影響效能、在某些情況下、較大的重做區塊大小可能會改善效能。

資料檔案區塊大小

部分作業系統提供多種檔案系統區塊大小選擇。對於支援 Oracle 資料檔案檔案的檔案系統、使用壓縮時區塊大小應為 8KB 。不需要壓縮時、可以使用 8KB 或 4KB 的區塊大小。

如果資料檔案放在具有 512 位元組區塊的檔案系統上、則可能會有未對齊的檔案。根據 NetApp 建議、 LUN 和檔案系統可能已正確對齊、但檔案 I/O 可能未對齊。這種錯誤的調整會導致嚴重的效能問題。

支援重做記錄的檔案系統必須使用重做區塊大小的倍數。這通常需要重做記錄檔系統和重做記錄本身都使用 512 位元組的區塊大小。

重做區塊大小

重做率極高時、 4KB 區塊大小可能會執行得更好、因為重做率較高、可在較少且更有效率的作業中執行 I/O 。如果重做速率大於 50Mbps 、請考慮測試 4KB 區塊大小。

在具有 4KB 區塊大小和許多極小型交易的檔案系統上、使用 512 位元組區塊大小的重做記錄檔來識別資料庫中的一些客戶問題。將多個 512 位元組的變更套用到單一 4KB 檔案系統區塊所涉及的額外負荷、導致效能問題、而這些問題已透過將檔案系統變更為使用 512 位元組的區塊大小來解決。

* NetApp 建議 * 除非相關客戶支援或專業服務組織建議您變更重做區塊大小、否則請勿變更重做區塊大小、否則變更將以正式產品文件為基礎。

Oracle 資料庫參數： DB_FILE_Multifblock_read_count

。 `db_file_multiblock_read_count` 參數控制 Oracle 在連續 I/O 期間讀取為單一作業的 Oracle 資料庫區塊數量上限

不過、此參數不會影響 Oracle 在任何及所有讀取作業期間讀取的區塊數、也不會影響隨機 I/O只有連續 I/O 的區塊大小會受到影響。

Oracle 建議使用者不要設定此參數。如此可讓資料庫軟體自動設定最佳值。這通常表示此參數設為可產生 1MB I/O 大小的值。例如、 1MB 讀取 8KB 區塊需要 128 個區塊才能讀取、因此此參數的預設值為 128 。

NetApp 在客戶站台上觀察到的大多數資料庫效能問題、都涉及此參數的設定不正確。使用 Oracle 版本 8 和 9 變更此值的理由是正確的。因此、參數可能會在不知情的情況下出現在中 `init.ora` 檔案、因為資料庫已就地升級至 Oracle 10 及更新版本。傳統設定為 8 或 16 、而預設值為 128 、會大幅損害連續 I/O 效能。

* NetApp 建議 * 設定 `db_file_multiblock_read_count` 參數不應出現在中 `init.ora` 檔案：NetApp 從未遇到過變更此參數可改善效能的情況、但在許多情況下、它會對連續 I/O 處理量造成明顯損害。

Oracle 資料庫參數： filesystemio_options

Oracle 初始化參數 `filesystemio_options` 控制非同步和直接 I/O 的使用

與一般的看法相反、非同步和直接 I/O 並不相互排斥。NetApp 發現、在客戶環境中、此參數經常設定錯誤、而這種錯誤設定直接導致許多效能問題。

非同步 I/O 表示 Oracle I/O 作業可以平行化。在各種作業系統上均可使用非同步 I/O 之前、使用者已設定數個 dbwriter 程序、並變更伺服器程序組態。透過非同步 I/O 、作業系統本身就能以高效率且平行的方式代表資料庫軟體執行 I/O 。此程序不會讓資料面臨風險、而且關鍵作業（例如 Oracle 重做記錄）仍會同步執行。

直接 I/O 會略過作業系統緩衝區快取。UNIX 系統上的 I/O 通常會流經作業系統緩衝區快取。這對不維護內部快取的應用程式很有用、但 Oracle 在 SGA 中擁有自己的緩衝區快取。在幾乎所有情況下、最好是啟用直接 I/O 並將伺服器 RAM 分配給 SGA 、而非仰賴作業系統緩衝區快取。Oracle SGA 更有效率地使用記憶體。此外、當 I/O 流經作業系統緩衝區時、它會受到額外處理、因此會增加延遲。當低延遲是關鍵需求時、在大量寫入 I/O 時、延遲特別明顯。

的選項 `filesystemio_options` 是：

- * 非同步 * 。 * Oracle 將 I/O 要求提交給作業系統以進行處理。此程序可讓 Oracle 執行其他工作、而非等待 I/O 完成、進而增加 I/O 平行化。
- **directio.** Oracle 直接針對實體檔案執行 I/O 、而非透過主機作業系統快取來路由 I/O 。
- * 無。 * Oracle 使用同步和緩衝 I/O在此組態中、選擇共享與專用伺服器程序與 dbWriters 數量更為重要。
- **setall.** Oracle 同時使用非同步和直接 I/O在幾乎所有情況下、都是使用 `setall` 是最佳的。

。 `filesystemio_options` 參數在 DNFS 和 ASM 環境中無效。使用 DNFS 或 ASM 時、會自動同時使用非同步和直接 I/O

有些客戶過去曾遇到非同步 I/O 問題、尤其是先前的 Red Hat Enterprise Linux 4 （ RHEL4 ）版本。網際網路上有些過時的建議仍建議避免非同步 IO 、因為資訊過時。在所有目前的作業系統上、非同步 I/O 都是穩定的。沒有理由停用它、作業系統沒有已知的錯誤。

如果資料庫已使用緩衝 I/O 、則直接 I/O 的交換器也可能需要變更 SGA 大小。停用緩衝 I/O 可消除主機作業系統快取為資料庫提供的效能優勢。將 RAM 新增回 SGA 可修復此問題。最終結果應該是 I/O 效能的改善。

雖然 Oracle SGA 使用 RAM 幾乎比使用 OS 緩衝區快取更好、但可能無法判斷最佳值。例如、最好在資料庫伺服器上使用具有極小型 SGA 大小的緩衝 I/O 、其中有許多間歇性作用中的 Oracle 執行個體。這種配置可讓所有執行中的資料庫執行個體靈活使用作業系統上的剩餘可用 RAM 。這是非常不尋常的情況、但在某些客戶據點已發現這種情況。

* NetApp 建議 * 設定 `filesystemio_options` 至 `setall`但請注意、在某些情況下、遺失主機緩衝區快取可能需要增加 Oracle SGA 。

Oracle Real Application Clusters （ RAC ）逾時

Oracle RAC 是一款叢集軟體產品、內含多種類型的內部活動訊號處理程序、可監控叢集的健全狀況。

中的資訊 "遺失計數" 本節包含使用網路儲存設備的 Oracle RAC 環境的重要資訊、在許多情況下、需要變更預設 Oracle RAC 設定、以確保 RAC 叢集在網路路徑變更和儲存設備容錯移轉 / 切換作業之後仍能順利運作。

磁碟逾時

主要儲存相關 RAC 參數為 `disktimeout`。此參數控制投票檔案 I/O 必須完成的臨界值。如果是 `disktimeout` 超過參數、 RAC 節點就會從叢集中移出。此參數的預設值為 200 。此值應足以用於標準儲存設備接管和恢復程序。

NetApp 強烈建議您在將 RAC 組態投入生產之前、先徹底測試這些組態、因為許多因素會影響接管或恢復作業。除了完成儲存容錯移轉所需的時間之外、連結集合化控制傳輸協定（ LACP ）變更也需要額外的時間才能傳播。此外、 SAN 多重路徑軟體必須偵測 I/O 逾時、然後在替代路徑上重試。如果資料庫處於極活躍狀態、則必須在處理投票磁碟 I/O 之前、先佇列並重新嘗試大量 I/O 。

如果無法執行實際的儲存接管或恢復、則可以在資料庫伺服器上進行纜線拉出測試來模擬影響。

- NetApp 建議 * 下列事項：
- 離開 `disktimeout` 預設值為 200 的參數。
- 務必徹底測試 RAC 組態。

遺失計數

。 `misscount` 參數通常只會影響 RAC 節點之間的網路心跳。預設值為 30 秒。如果網格二進位檔位於儲存陣列上、或作業系統開機磁碟機不是本機磁碟機、此參數可能會變得很重要。這包括 FC SAN 上具有開機磁碟機的主機、 NFS 開機作業系統、以及位於虛擬化資料存放區（例如 VMDK 檔案）上的開機磁碟機。

如果因儲存接管或恢復而中斷開機磁碟機的存取、網格二進位位置或整個作業系統可能會暫時停止運作。ONTAP 完成儲存作業所需的時間、以及作業系統變更路徑和恢復 I/O 所需的時間、可能會超過 `misscount` 臨界值。因此、節點會在連線到開機 LUN 或網格二進位檔恢復後立即停止。在大多數情況下、會發生遷離和後續重新開機、而不會出現記錄訊息來指出重新開機的原因。並非所有組態都會受到影響、因此請在 RAC 環境中測試任何 SAN 開機、 NFS 開機或資料存放區型主機、以便在與開機磁碟機的通訊中斷時、 RAC 保持穩定。

非本機開機磁碟機或非本機檔案系統代管的情況 `grid` 二進位檔案 `misscount` 需要變更以符合 `disktimeout`。如果變更此參數、請進行進一步測試、以識別對 RAC 行為的任何影響、例如節點容錯移轉時間。

- NetApp 建議 * 下列事項：
- 離開 `misscount` 參數的預設值為 30 、除非符合下列其中一項條件：
  - `grid` 二進位檔位於網路附加磁碟機上、包括 NFS 、 iSCSI 、 FC 和資料存放區型磁碟機。
  - 作業系統是 SAN 開機。
- 在這種情況下、請評估網路中斷對 OS 或的存取造成的影響 `GRID_HOME` 檔案系統。在某些情況下、這類中斷會導致 Oracle RAC 精靈停止運作、進而導致 `misscount`**根據的逾時和遷離。逾時預設為 `27` 秒、即的值** `misscount` 減號 `reboottime`。在這種情況下、請增加 `misscount` 200 比對 `disktimeout`。

主機組態

使用 **IBM AIX** 的 **Oracle** 資料庫

IBM AIX 與 ONTAP 上 Oracle 資料庫的組態主題。

並行 **I/O**

若要在 IBM AIX 上達到最佳效能、必須同時使用 I/O如果沒有並行 I/O 、效能限制很可能是因為 AIX 執行序列化的原子 I/O 、這會產生重大的負荷。

NetApp 最初建議使用 `cio` 掛載選項可強制在檔案系統上使用並行 I/O 、但此程序有缺點、不再需要。自從推出 AIX 5.2 和 Oracle 10gR1 之後、 AIX 上的 Oracle 就可以開啟個別檔案來同時執行 IO 、而不是強制在整個檔案系統上同時執行 I/O 。

啟用並行 I/O 的最佳方法是設定 `init.ora` 參數 `filesystemio_options` 至 `setall`。這樣做可讓 Oracle 開啟特定檔案、以與並行 I/O 搭配使用

使用 `cio` 作爲掛載選項，強制使用並行 I/O ，這可能會產生負面影響。例如、強制並行 I/O 會停用檔案系統上的預先讀取、這可能會損害 Oracle 資料庫軟體以外的 I/O 效能、例如複製檔案和執行磁帶備份。此外、 Oracle GoldenGate 和 SAP BR* Tools 等產品與使用不相容 `cio` 裝載選項搭配特定版本的 Oracle 。

- NetApp 建議 * 下列事項：
- 請勿使用 `cio` 檔案系統層級的掛載選項。而是透過使用來啟用並行 I/O `filesystemio_options=setall`。
- 請僅使用 `cio` 如果無法設定掛載選項、則應選擇掛載選項 `filesystemio_options=setall`。

AIX NFS 裝載選項

下表列出 Oracle 單一執行個體資料庫的 AIX NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144`

檔案類型	掛載選項
控制檔 資料檔案 重作記錄	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144
ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,intr

下表列出 RAC 的 AIX NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144
控制檔 資料檔案 重作記錄	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac
CRS/Voting	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr,noac
專屬 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144
共享 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsize=262144,nointr

單一執行個體與 RAC 掛載選項之間的主要差異在於新增 noac 至掛載選項。此新增功能可停用主機作業系統快取、使 RAC 叢集中的所有執行個體都能一致地檢視資料狀態。

雖然使用 cio 裝載選項和 init.ora 參數 filesystemio_options=setall 停用主機快取的效果相同、仍需使用 noac。 noac 為共享的必要項目 ORACLE_HOME 部署以促進 Oracle 密碼檔案和等檔案的一致性 spfile 參數檔。如果 RAC 叢集中的每個執行個體都有專用的 ORACLE_HOME，則不需要此參數。

AIX jfs/JFS2 掛載選項

下表列出 AIX jfs/JFS2 掛載選項。

檔案類型	掛載選項
ADR 首頁	預設值
控制檔 資料檔案 重作記錄	預設值
Oracle_Home	預設值

使用 AIX 之前 hdisk 任何環境中的裝置（包括資料庫）都請檢查參數 queue_depth。此參數不是 HBA 佇列深度、而是與個別主機的 SCSI 佇列深度相關 hdisk device. Depending on how the LUNs are configured, the value for `queue_depth 效能可能太低。測試顯示最佳值為 64 。

使用 **HP-UX** 的 **Oracle** 資料庫

適用於 HP-UX with ONTAP 上 Oracle 資料庫的組態主題。

HP-UX NFS 掛載選項

下表列出單一執行個體的 HP-UX NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
控制檔 資料檔案 重作記錄	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144,forc edirectio, nointr,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`

下表列出適用於 RAC 的 HP-UX NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, noac,suid`
控制檔 資料檔案 重作記錄	`rw, bg,hard, [vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio,suid`
CRS/ 投票	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac, forcedirectio,suid`
專屬 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
共享 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,suid`

單一執行個體與 RAC 掛載選項之間的主要差異在於新增 `noac` 和 `forcedirectio` 至掛載選項。此新增功能可停用主機作業系統快取、使 RAC 叢集中的所有執行個體都能一致地檢視資料狀態。雖然使用 `init.ora` 參數 `filesystemio_options=setall` 停用主機快取的效果相同、仍需使用 `noac` 和 `forcedirectio`。

原因 `noac` 為共享的必要項目 `ORACLE_HOME` 部署是為了促進檔案的一致性、例如 Oracle 密碼檔案和 spfiles 。如果 RAC 叢集中的每個執行個體都有專用的 `ORACLE_HOME`，不需要此參數。

HP-UX VxFS 掛載選項

對於託管 Oracle 二進位檔的檔案系統、請使用下列掛載選項：

```
delaylog,nodatainlog
```

對於包含資料檔案、重做記錄檔、歸檔記錄檔和控制檔的檔案系統、若 HP-UX 版本不支援並行 I/O 、請使用下列掛載選項：

```
nodatainlog,mincache=direct,convosync=direct
```

支援並行 I/O （ VxFS 5.0.1 及更新版本、或 ServiceGuard Storage Management Suite ）時、請針對包含資料檔案、重作記錄檔、歸檔記錄檔和控制檔的檔案系統、使用這些掛載選項：

```
delaylog,cio
```

> 參數 `db_file_multiblock_read_count` 在 VxFS 環境中尤其重要。Oracle 建議在 Oracle 10g R1 及更新版本中保留此參數、除非另有特別指示。Oracle 8KB 區塊大小的預設值為 128 。如果此參數的值強制為 16 或更少、請移除 `convosync=direct` 裝載選項、因為它可能會損害連續 I/O 效能。此步驟會損害其他效能層面、只有在的價值下才應採取 `db_file_multiblock_read_count` 必須從預設值變更。

使用 Linux 的 Oracle 資料庫

Linux 作業系統專屬的組態主題。

Linux NFSv3 TCP 插槽表

TCP 插槽表是與主機匯流排介面卡（ HBA ）佇列深度相當的 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 。

注意
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。

Linux NFS 裝載選項

下表列出單一執行個體的 Linux NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
控制檔 資料檔案 重作記錄	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr`
Oracle_Home	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr`

下表列出 RAC 的 Linux NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, actimeo=0`
控制檔 資料檔案 重作記錄	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,actimeo=0`
CRS/ 投票	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,actimeo=0`
專屬 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144`
共享 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,actimeo=0`

單一執行個體與 RAC 掛載選項之間的主要差異在於新增 `actimeo=0` 至掛載選項。此新增功能可停用主機作業系統快取、使 RAC 叢集中的所有執行個體都能一致地檢視資料狀態。雖然使用 `init.ora` 參數 `filesystemio_options=setall` 停用主機快取的效果相同、仍需使用 `actimeo=0`。

原因 `actimeo=0` 為共享的必要項目 `ORACLE_HOME` 部署是為了促進檔案的一致性、例如 Oracle 密碼檔案和 spfiles 。如果 RAC 叢集中的每個執行個體都有專用的 `ORACLE_HOME`，則不需要此參數。

一般而言、非資料庫檔案的掛載應該與單一執行個體資料檔案所使用的選項相同、不過特定應用程式可能有不同的需求。避免使用掛載選項 `noac` 和 `actimeo=0` 如果可能、因為這些選項會停用檔案系統層級的預先讀取和緩

衝處理。這可能會對擷取、轉譯和載入等程序造成嚴重的效能問題。

存取與 GetAttr

有些客戶指出、存取和 GetAttr 等極高層級的其他 IOPS 、可能會主導他們的工作負載。在極端情況下、讀取和寫入等作業可低於總作業量的 10% 。這是任何包含使用的資料庫的正常行為 `actimeo=0` 和/或 `noac` 在 Linux 上、因為這些選項會導致 Linux 作業系統持續從儲存系統重新載入檔案中繼資料。存取和 GetAttr 等作業是從資料庫環境中的 ONTAP 快取提供服務的低影響作業。它們不應被視為真正的 IOPS 、例如讀寫、而會對儲存系統產生真正的需求。不過、這些其他 IOPS 確實會產生一些負載、尤其是在 RAC 環境中。若要解決這種情況、請啟用 DNFS 、以略過作業系統緩衝區快取、避免這些不必要的中繼資料作業。

Linux Direct NFS

另一個掛載選項、稱為 `nosharecache`（ a） DNFS 啟用時、需要使用、（ b）來源磁碟區多次裝載於具有巢狀 NFS 裝載的單一伺服器（ c）上。此組態主要出現在支援 SAP 應用程式的環境中。例如、 NetApp 系統上的單一磁碟區可能會有位於的目錄 `/vol/oracle/base` 再來一次 `/vol/oracle/home`。如果 `/vol/oracle/base` 安裝於 `/oracle` 和 `/vol/oracle/home` 安裝於 `/oracle/home`，結果是來自相同來源的巢狀 NFS 掛載。

作業系統可以偵測到這個事實 `/oracle` 和 `/oracle/home` 位於同一個磁碟區、即相同的來源檔案系統。作業系統接著會使用相同的裝置控制代碼來存取資料。這樣做可以改善 OS 快取和某些其他作業的使用、但會干擾 DNFS 。如果 DNFS 必須存取檔案、例如 `spfile`、開啟 `/oracle/home`，它可能會錯誤地嘗試使用錯誤的資料路徑。結果是 I/O 作業失敗。在這些組態中、新增 `nosharecache` 裝載選項至任何與該主機上其他 NFS 檔案系統共用來源 FlexVol 磁碟區的 NFS 檔案系統。這樣做會強制 Linux 作業系統為該檔案系統分配獨立的裝置控制代碼。

Linux Direct NFS 和 Oracle RAC

使用 DNFS 對 Linux 作業系統上的 Oracle RAC 有特殊的效能優勢、因為 Linux 沒有強制直接 I/O 的方法、而 RAC 需要此方法才能在節點之間保持一致。因應措施是 Linux 需要使用 `actimeo=0` 掛載選項、會使檔案資料從作業系統快取立即過期。此選項會強制 Linux NFS 用戶端持續重新讀取屬性資料、進而損害延遲並增加儲存控制器的負載。

啟用 DNFS 會略過主機 NFS 用戶端、避免此損害。多家客戶在啟用 DNFS 時、報告 RAC 叢集的效能大幅提升、 ONTAP 負載大幅降低（特別是其他 IOPS ）。

Linux Direct NFS 和 oranfstab 檔案

在 Linux 上搭配多重路徑選項使用 DNFS 時、必須使用多個子網路。在其他作業系統上、可使用建立多個 DNFS 通道 `LOCAL` 和 `DONTROUTE` 可在單一子網路上設定多個 DNFS 通道的選項。不過、這在 Linux 上無法正常運作、因此可能會產生非預期的效能問題。在 Linux 中、用於 DNFS 流量的每個 NIC 都必須位於不同的子網路上。

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。測試顯示、截止日期通常會提供最佳結果、但有時 NOOP 會稍微好一點。效能差異最小、但如果需要從資料庫組態擷取最大可能效能、請測試這兩個選項。在許多組態中、 CFQ 是預設值、而且已證明資料庫工作負載的效能有重大問題。

如需設定 I/O 排程器的指示、請參閱相關的 Linux 廠商文件。

部分客戶在網路中斷期間遭遇當機、因為多重路徑常駐程式未在其系統上執行。在最新版本的 Linux 上、作業系統的安裝程序和多重路徑常駐程式可能會讓這些作業系統容易受到此問題的影響。套件已正確安裝、但未設定為在重新開機後自動啟動。

例如、 RHEL5.5 上多重路徑常駐程式的預設值可能如下所示：

```
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

您可以使用下列命令來修正此問題：

```
[root@host1 iscsi]# chkconfig multipathd on
[root@host1 iscsi]# chkconfig --list | grep multipath
multipathd      0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ASM 鏡像

ASM鏡射可能需要變更Linux多重路徑設定、以允許ASM辨識問題並切換至其他故障群組。大部分關於「不完整」的ASM組態ONTAP 都使用外部備援、這表示資料保護是由外部陣列提供、而ASM不會鏡射資料。某些站台使用具有一般備援的ASM來提供雙向鏡像、通常是跨不同站台。

中顯示的 Linux 設定 "NetApp 主機公用程式文件" 包含會導致 I/O 無限期佇列的多重路徑參數這表示 LUN 裝置上沒有作用中路徑的 I/O 會在 I/O 完成所需的時間內等待。這通常是很理想的做法、因為 Linux 主機會在 SAN 路徑變更完成、 FC 交換器重新開機或儲存系統完成容錯移轉所需的時間內等待。

這種不受限制的佇列行為會導致 ASM 鏡像發生問題、因為 ASM 必須收到 I/O 故障、才能在替代 LUN 上重試 I/O 。

在 Linux 中設定下列參數 `multipath.conf` 用於 ASM 鏡像的 ASM LUN 檔案：

```
polling_interval 5
no_path_retry 24
```

這些設定會為 ASM 裝置建立 120 秒的逾時。逾時會計算為 `polling_interval` * `no_path_retry` 秒。在某些情況下可能需要調整確切的值、但 120 秒的逾時時間應足以滿足大部分的使用需求。具體而言、 120 秒的時間應該能讓控制器接管或恢復、而不會產生 I/O 錯誤、導致故障群組離線。

較低 `no_path_retry` 此值可縮短 ASM 切換至替代故障群組所需的時間、但這也會增加在維護活動（例如控制器接管）期間不必要的容錯移轉風險。仔細監控 ASM 鏡像狀態、即可降低風險。如果發生不必要的容錯移轉、只要執行重新同步的速度相對較快、鏡像就能快速重新同步。如需更多資訊、請參閱 ASM Fast Mirror Resync 上的 Oracle 說明文件、以瞭解所使用的 Oracle 軟體版本。

Linux xfs 、 **ext3** 和 **ext4** 掛載選項

* NetApp 建議 * 使用預設掛載選項。

使用 **ASMLib/AFD** 的 **Oracle** 資料庫（ **ASM** 篩選器驅動程式）

使用 AFD 和 ASMLib 的 Linux 作業系統專屬組態主題

ASMLib 區塊大小

ASMLib 是選用的 ASM 管理程式庫和相關公用程式。其主要值是將 LUN 或 NFS 型檔案標記為具有人類可讀標籤的 ASM 資源。

ASMLib 的最新版本會偵測稱為每個實體區塊指數（ LBPPBE ）的邏輯區塊的 LUN 參數。ONTAP SCSI 目標直到最近才回報此值。現在會傳回一個值、表示偏好 4KB 區塊大小。這不是區塊大小的定義、但它是使用 LBPPBE 的任何應用程式的提示、可能會更有效率地處理特定大小的 I/O 。不過、 ASMLib 會將 LBPPBE 解譯為區塊大小、並在建立 ASM 裝置時持續標記 ASM 標頭。

此程序可能會以多種方式造成升級和移轉問題、全部是因為無法在同一個 ASM 磁碟群組中混合使用不同區塊大小的 ASMLib 裝置。

例如、較舊的陣列通常回報 LBPPBE 值為 0 、或根本沒有回報此值。ASMLib 會將此解譯為 512 位元組的區塊大小。較新的陣列會被解譯為具有 4KB 區塊大小。無法在同一個 ASM 磁碟群組中混合使用 512 位元組和 4KB 的裝置。這樣做會阻止用戶使用兩個陣列中的 LUN 或使用 ASM 作爲遷移工具來增加 ASM 磁盤組的大小。在其他情況下、 RMAN 可能不允許在具有 512 位元組區塊大小的 ASM 磁碟群組和具有 4KB 區塊大小的 ASM 磁碟群組之間複製檔案。

首選的解決方案是修補 ASMLib 。Oracle 錯誤 ID 為 13999609 、而 Oracle 錯誤 ID 則存在於 oracleas-support-2.1.8-1 及更高版本中。此修補程式可讓使用者設定參數 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 至 `true` 在中 `/etc/sysconfig/oracleasm` 組態檔。這樣做會阻止 ASMLib 使用 LBPPBE 參數、這表示新陣列上的 LUN 現在會被辨識為 512 位元組區塊裝置。

> ⓘ 此選項不會變更先前由 ASMLib 戳記的 LUN 區塊大小。例如、如果具有 512 位元組區塊的 ASM 磁碟群組必須移轉至回報 4KB 區塊的新儲存系統、則選項 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 必須先設定、才能使用 ASMLib 標記新的 LUN 。 如果裝置已被 oracleasm 戳記、則必須先重新格式化、然後再重新設定新的區塊大小。首先、請使用取消設定裝置 `oracleasm deletedisk`、然後使用清除裝置的前 1GB `dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`。最後、如果裝置先前已分割、請使用 `kpartx` 命令移除過時的分割區、或只是重新開機作業系統。

如果無法修補 ASMLib 、可以從組態中移除 ASMLib 。這項變更會造成中斷、需要在 ASM 磁碟上加蓋戳記、並確定 `asm_diskstring` 參數設定正確。不過、這項變更並不需要移轉資料。

ASM Filter Drive （ **AFD** ）區塊大小

AFD 是選用的 ASM 管理程式庫、正在取代 ASMLib 。從儲存角度來看、它與 ASMLib 非常類似、但它還包含其他功能、例如能夠封鎖非 Oracle I/O 、以降低使用者或應用程式錯誤可能毀損資料的機會。

裝置區塊大小

如同 ASMLib 、 AFD 也會讀取 LUN 參數每個實體區塊指數（ LBPPBE ）的邏輯區塊、並依預設使用實體區塊大小、而非邏輯區塊大小。

如果將 AFD 新增至現有組態、而 ASM 裝置已格式化為 512 位元組區塊裝置、則可能會造成問題。AFD 驅動程式會將 LUN 辨識為 4K 裝置、而 ASM 標籤與實體裝置之間的不符將會妨礙存取。同樣地、移轉也會受到影響、因為無法在同一個 ASM 磁碟群組中混合使用 512 位元組和 4KB 的裝置。這樣做會阻止用戶使用兩個陣列中的

LUN 或使用 ASM 作爲遷移工具來增加 ASM 磁盤組的大小。在其他情況下、 RMAN 可能不允許在具有 512 位元組區塊大小的 ASM 磁碟群組和具有 4KB 區塊大小的 ASM 磁碟群組之間複製檔案。

解決方案很簡單 - AFD 包含一個參數、可控制它是否使用邏輯區塊或實體區塊大小。這是影響系統上所有裝置的全域參數。若要強制 AFD 使用邏輯區塊大小、請設定 `options oracleafd oracleafd_use_logical_block_size=1` 在中 `/etc/modprobe.d/oracleafd.conf` 檔案：

多重路徑傳輸大小

最近的 Linux 核心變更會強制執行傳送至多重路徑裝置的 I/O 大小限制、而 AFD 則不遵守這些限制。然後會拒絕 I/O 、導致 LUN 路徑離線。結果是無法安裝 Oracle Grid 、設定 ASM 或建立資料庫。

解決方案是在 ONTAP LUN 的 multipath.conf 檔案中手動指定傳輸長度上限：

```
devices {
            device {
                vendor "NETAPP"
                product "LUN.*"
                max_sectors_kb 4096
        }
    }
```

即使目前沒有問題、如果使用 AFD 來確保未來的 Linux 升級不會意外造成問題、也應設定此參數。

使用 **Microsoft Windows** 的 **Oracle** 資料庫

Microsoft Windows with ONTAP 上 Oracle 資料庫的組態主題。

NFS

Oracle 支援搭配直接 NFS 用戶端使用 Microsoft Windows 。這項功能提供 NFS 管理效益的途徑、包括跨環境檢視檔案、動態調整磁碟區大小、以及使用較便宜的 IP 傳輸協定。如需在 Microsoft Windows 上使用 DNFS 安裝及設定資料庫的詳細資訊、請參閱正式的 Oracle 文件。不存在任何特殊的最佳實務做法。

SAN

為達到最佳壓縮效率、請確保 NTFS 檔案系統使用 8K 或更大的分配單元。使用 4K 分配單元（通常是預設）會對壓縮效率造成負面影響。

Oracle 資料庫與 **Solaris**

特定於 Solaris OS 的組態主題。

Solaris NFS 掛載選項

下表列出單一執行個體的 Solaris NFS 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1], roto=tcp, timeo=600,rsize=262144,wsize=262144`
控制檔 資料檔案 重作記錄	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,llock,suid`
`ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`

的用途 `llock` 已獲證實、可移除在儲存系統上取得和釋放鎖定的相關延遲、大幅提升客戶環境的效能。在將多個伺服器設定為掛載相同檔案系統的環境中、請謹慎使用此選項、並將 Oracle 設定為掛載這些資料庫。雖然這是非常不尋常的組態、但只有少數客戶使用。如果第二次意外啟動某個執行個體、可能會因為 Oracle 無法偵測到外部伺服器上的鎖定檔案而導致資料毀損。NFS 鎖定不會提供保護；如同 NFS 第 3 版一樣、它們只是建議事項。

因為 `llock` 和 `forcedirectio` 參數是互斥的、這一點很重要 `filesystemio_options=setall` 存在於中 `init.ora` 檔案就是這樣 `directio` 已使用。如果沒有此參數、就會使用主機作業系統緩衝區快取、而且效能可能會受到負面影響。

下表列出了 Solaris NFS RAC 掛載選項。

檔案類型	掛載選項
ADR 首頁	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, noac`
控制檔 資料檔案 重作記錄	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio`
CRS/ 投票	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,forcedirectio`
專屬 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, suid`
共享 `ORACLE_HOME`	`rw,bg,hard,[vers=3,vers=4.1],proto=tcp, timeo=600,rsize=262144,wsize=262144, nointr,noac,suid`

單一執行個體與 RAC 掛載選項之間的主要差異在於新增 `noac` 和 `forcedirectio` 至掛載選項。此新增功能可停用主機作業系統快取、使 RAC 叢集中的所有執行個體都能一致地檢視資料狀態。雖然使用 `init.ora` 參數 `filesystemio_options=setall` 停用主機快取的效果相同、仍需使用 `noac` 和 `forcedirectio`。

原因 `actimeo=0` 為共享的必要項目 `ORACLE_HOME` 部署是為了促進檔案的一致性、例如 Oracle 密碼檔案和 spfiles 。如果 RAC 叢集中的每個執行個體都有專用的 `ORACLE_HOME`，不需要此參數。

Solaris UFS 掛載選項

NetApp 強烈建議您使用記錄掛載選項、以便在 Solaris 主機當機或 FC 連線中斷時保留資料完整性。記錄掛載選項也可保留 Snapshot 備份的使用性。

Solaris ZFS

必須仔細安裝和設定 Solaris ZFS 、才能提供最佳效能。

mvector

Solaris 11 變更了 IT 處理大型 I/O 作業的方式、可能會在 SAN 儲存陣列上造成嚴重的效能問題。NetApp 錯誤報告 630173 「 Solaris 11 ZFS 效能回歸」中詳細說明了此問題。" 解決方案是變更名為的 OS 參數 `zfs_mvector_max_size`。

以 root 執行下列命令：

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t131072" |mdb -kw
```

如果這項變更發生任何非預期的問題、您可以以 root 執行下列命令、輕鬆地將其還原：

```
[root@host1 ~]# echo "zfs_mvector_max_size/W 0t1048576" |mdb -kw
```

核心

可靠的 ZFS 效能需要修補 Solaris 核心、以因應 LUN 對齊問題。此修正程式是隨 Solaris 10 中的修補程式 147440-19 和適用於 Solaris 11 的 SRU 10.5 一起推出的。只能將 Solaris 10 及更新版本與 ZFS 搭配使用。

LUN 組態

若要設定 LUN 、請完成下列步驟：

1. 建立類型的 LUN `solaris`。
2. 安裝所指定的適當主機公用程式套件（ Huk ） "NetApp互通性對照表工具IMT （不含）"。
3. 請依照 Huk 中的說明進行操作、完全符合上述說明。以下概述基本步驟、但請參閱 "最新文件" 以瞭解正確的程序。
   a. 執行 `host_config` 更新的公用程式 `sd.conf/sdd.conf` 檔案：這樣做可讓 SCSI 磁碟機正確探索 ONTAP LUN 。
   b. 請遵循所提供的指示 `host_config` 啟用多重路徑輸入 / 輸出（ MPIO ）的公用程式。
   c. 重新開機。此步驟是必要步驟、以便在整個系統中辨識任何變更。
4. 分割 LUN 並確認它們已正確對齊。請參閱「附錄 B ： WAFL 校準驗證」、瞭解如何直接測試及確認校準。

zPools

只有在中的步驟之後才應建立 zpool "LUN 組態" 執行。如果程序未正確執行、可能會因為 I/O 對齊而導致嚴重的效能降低。ONTAP 的最佳效能要求 I/O 必須與磁碟機上的 4K 邊界對齊。在 zpool 上建立的檔案系統使用有效

的區塊大小、並透過稱為的參數加以控制 `ashift`，您可以執行命令來檢視 `zdb -C`。

的價值 `ashift` 預設為 9 、表示 2^9 或 512 位元組。為了獲得最佳效能 `ashift` 值必須為 12 （ 2^12=4K ）。此值是在創建 zpool 時設置的，不能更改，這意味着 zpool 中的數據 `ashift` 除 12 個以外、應將資料複製到新建立的 zPool 、以進行移轉。

建立 zPool 之後、請驗證的值 `ashift` 繼續之前。如果值不是 12 、則表示未正確探索到 LUN 。銷毀 zpool 、確認相關主機公用程式文件中顯示的所有步驟均已正確執行、然後重新建立 zPool 。

zPools 和 Solaris LDoms

Solaris LDoms 還需要確保 I/O 對齊正確無誤。雖然 LUN 可能會被正確發現為 4K 裝置、但 LDOM 上的虛擬 vdsk 裝置不會繼承 I/O 網域的組態。以該 LUN 為基礎的 vdsk 預設為 512 位元組區塊。

需要額外的組態檔案。首先、必須針對 Oracle 錯誤 15824910 修補個別的 LDOM 、才能啟用其他組態選項。此修補程式已移轉至所有目前使用的 Solaris 版本。一旦 LDOM 獲得修補、就可以依照下列方式設定新的正確對齊 LUN ：

1. 識別要在新的 zPool 中使用的 LUN 或 LUN 。在此範例中、它是 c2d1 裝置。

```
[root@LDOM1 ~]# echo | format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2d0 <Unknown-Unknown-0001-100.00GB>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c2d1 <SUN-ZFS Storage 7330-1.0 cyl 1623 alt 2 hd 254 sec 254>
     /virtual-devices@100/channel-devices@200/disk@1
```

2. 擷取要用於 ZFS Pool 的裝置之 VDC 執行個體：

```
[root@LDOM1 ~]#  cat /etc/path_to_inst
#
# Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/virtual-devices@100" 0 "vnex"
"/virtual-devices@100/channel-devices@200" 0 "cnex"
"/virtual-devices@100/channel-devices@200/disk@0" 0 "vdc"
"/virtual-devices@100/channel-devices@200/pciv-communication@0" 0 "vpci"
"/virtual-devices@100/channel-devices@200/network@0" 0 "vnet"
"/virtual-devices@100/channel-devices@200/network@1" 1 "vnet"
"/virtual-devices@100/channel-devices@200/network@2" 2 "vnet"
"/virtual-devices@100/channel-devices@200/network@3" 3 "vnet"
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc" << We want
this one
```

3. 編輯 `/platform/sun4v/kernel/drv/vdc.conf`：

```
block-size-list="1:4096";
```

這表示裝置執行個體 1 的區塊大小為 4096 。

另一個範例是假設需要將 vdsk 執行個體 1 至 6 設定為 4K 區塊大小和 `/etc/path_to_inst` 內容如下：

```
"/virtual-devices@100/channel-devices@200/disk@1" 1 "vdc"
"/virtual-devices@100/channel-devices@200/disk@2" 2 "vdc"
"/virtual-devices@100/channel-devices@200/disk@3" 3 "vdc"
"/virtual-devices@100/channel-devices@200/disk@4" 4 "vdc"
"/virtual-devices@100/channel-devices@200/disk@5" 5 "vdc"
"/virtual-devices@100/channel-devices@200/disk@6" 6 "vdc"
```

4. 最終結果 `vdc.conf` 檔案應包含下列項目：

```
block-size-list="1:8192","2:8192","3:8192","4:8192","5:8192","6:8192";
```

注意
設定 VC.conf 並建立 vdsk 之後、必須重新啟動 LDOM 。無法避免此步驟。區塊大小變更只會在重新開機後生效。繼續使用 zpool 組態、並確保如前所述、移位已正確設定為 12 。

ZFS Intent Log （ ZIL ）

一般而言、沒有理由在不同的裝置上找到 ZFS Intent Log （ ZIL ）。記錄檔可以與主集區共用空間。獨立 ZIL 的主要用途是使用缺乏現代儲存陣列寫入快取功能的實體磁碟機。

logbias

設定 `logbias` 託管 Oracle 資料的 ZFS 檔案系統參數。

```
zfs set logbias=throughput <filesystem>
```

使用此參數可降低整體寫入層級。根據預設值、寫入的資料會先提交至 ZIL 、然後再提交至主儲存池。此方法適用於使用純磁碟機組態的組態、包括 SSD 型 ZIL 裝置和主儲存池的旋轉媒體。這是因為它允許在可用的最低延遲媒體上、在單一 I/O 交易中進行認可。

使用包含其快取功能的現代化儲存陣列時、通常不需要使用此方法。在極少數情況下、可能需要在單一交易中寫入記錄檔、例如由高度集中、對延遲敏感的隨機寫入所組成的工作負載。寫入放大的形式會產生影響、因為記錄的資料最終會寫入主儲存池、導致寫入活動加倍。

直接 I/O

許多應用程式（包括 Oracle 產品）都可以啟用直接 I/O 、藉此略過主機緩衝區快取此策略無法在 ZFS 檔案系統中正常運作。雖然會略過主機緩衝區快取、但 ZFS 本身仍會繼續快取資料。使用 Fio 或 Sio 等工具執行效能測試時、這項動作可能會產生誤導性的結果、因為很難預測 I/O 是否到達儲存系統、或是是否在作業系統中本機快取。此動作也會讓使用此類模擬測試來比較 ZFS 效能與其他檔案系統的情況變得非常困難。實際上、在真實使用者工作負載下、檔案系統效能幾乎沒有任何差異。

多個 zPools

必須在 zpool 層級執行快照型備份、還原、複製及歸檔 ZFS 型資料、而且通常需要多個 zPools 。zpool 類似於 LVM 磁碟群組、應使用相同的規則進行設定。例如、資料庫的配置最好是存放在資料檔案上 `zpool1` 以及駐留在上的歸檔記錄、控制檔和重做記錄 `zpool2`。此方法允許標準熱備份、將資料庫置於熱備份模式、然後是的快照 `zpool1`。接著會從熱備份模式移除資料庫、強制進行記錄歸檔、並建立快照 `zpool2` 已建立。還原作業需要卸載 zfs 檔案系統、並在執行 SnapRestore 還原作業之後、將 zPool 完全離線。然後可以重新上線並恢復資料庫。

filesystemio_options

Oracle 參數 `filesystemio_options` 使用 ZFS 的方式不同。如果 `setall` 或 `directio` 使用時、寫入作業會同步並略過 OS 緩衝區快取、但讀取會由 ZFS 進行緩衝。此動作會導致效能分析方面的困難、因為有時會被 ZFS 快取攔截和服務 I/O 、使儲存延遲和總 I/O 比預期的要少。

網路組態

Oracle 資料庫的邏輯介面設計

Oracle 資料庫需要存取儲存設備。邏輯介面（ Lifs ）是將儲存虛擬機器（ SVM ）連接到網路、然後再連接到資料庫的網路配送。需要適當的 LIF 設計、才能確保每個資料庫工作負載都有足夠的頻寬、而且容錯移轉不會導致儲存服務遺失。

本節概述 LIF 的主要設計原則。如需更完整的文件、請參閱 "ONTAP 網路管理文件"。與資料庫架構的其他層面一樣、儲存虛擬機器（ SVM 、在 CLI 稱為 vserver ）和邏輯介面（ LIF ）設計的最佳選項、在很大程度上取決於擴充需求和業務需求。

建置 LIF 策略時、請考量下列主要主題：

- * 效能。 * 網路頻寬是否足夠？
- * 恢復能力。 * 設計中是否有任何單點故障？
- * 管理能力。 * 網路是否能不中斷地擴充？

這些主題適用於端點對端點解決方案、從主機到交換器、再到儲存系統。

LIF 類型

有多種 LIF 類型。 "LIF 類型的 ONTAP 文件" 提供更完整的本主題資訊、但從功能觀點來看、生命可分為下列群組：

- * 用於管理儲存叢集的叢集與節點管理生命期 * 。
- * SVM 管理階層。 * 允許透過 REST API 或 ONTAPI （也稱為 ZAPI ）存取 SVM 的介面、以執行快照建立或磁碟區調整大小等功能。SnapManager for Oracle （ SMO ）等產品必須能夠存取 SVM 管理 LIF 。
- * 資料生命。 * FC 、 iSCSI 、 NVMe / FC 、 NVMe / TCP 、 NFS 、 或 SMB/CIFS 資料。

> ⓘ 用於 NFS 流量的資料 LIF 也可透過變更防火牆原則來進行管理 `data` 至 `mgmt` 或其他允許 HTTP 、 HTTPS 或 SSH 的原則。這項變更可避免每部主機的組態設定、以同時存取 NFS 資料 LIF 和個別的管理 LIF 、進而簡化網路組態。儘管這兩者都使用 IP 傳輸協定、但無法同時為 iSCSI 和管理流量設定介面。iSCSI 環境需要個別的管理 LIF 。

SAN LIF 設計

SAN 環境中的 LIF 設計相對簡單、原因有一：多重路徑。所有現代化的 SAN 實作均可讓用戶端透過多個、不受限制的網路路徑存取資料、並選擇最佳的存取路徑或路徑。因此、 LIF 設計的效能更容易因應、因為 SAN 用戶端會在最佳可用路徑之間自動平衡 I/O 負載。

如果路徑無法使用、用戶端會自動選取不同的路徑。因此設計簡易性讓 SAN 的工作更容易管理。這並不表示 SAN 環境總是更容易管理、因為 SAN 儲存設備還有許多其他層面比 NFS 複雜得多。這只是表示 SAN LIF 設計更簡單。

效能

在 SAN 環境中、 LIF 效能的最重要考量是頻寬。例如、雙節點 ONTAP AFF 叢集每個節點有兩個 16GB FC 連接埠、可在每個節點之間提供高達 32GB 的頻寬。

恢復能力

AFF 儲存系統上的 SAN Lifs 不會容錯移轉。如果 SAN LIF 因控制器容錯移轉而失敗、則用戶端的多重路徑軟體會偵測路徑遺失、並將 I/O 重新導向至不同的 LIF 。使用 ASA 儲存系統時、在短暫延遲後將會容錯移轉、但這不會中斷 IO 、因為其他控制器上已經有作用中的路徑。發生容錯移轉程序是為了在所有定義的連接埠上還原主機存取。

管理能力

LIF 移轉是 NFS 環境中較常見的工作、因為 LIF 移轉通常與在叢集周圍重新放置磁碟區有關。當磁碟區移轉至 HA 配對內時、無需在 SAN 環境中移轉 LIF 。這是因為在磁碟區移動完成之後、 ONTAP 會傳送路徑變更通知給 SAN 、而 SAN 用戶端會自動重新最佳化。與 SAN 的 LIF 移轉主要與重大實體硬體變更有關。例如、如果需要不中斷營運的控制器升級、則 SAN LIF 會移轉至新硬體。如果發現 FC 連接埠故障、 LIF 就可以移轉至未使用的連接埠。

設計建議

NetApp 提出下列建議：

- 請勿建立超過所需的路徑。過多的路徑會使整體管理更為複雜、並可能導致部分主機上路徑容錯移轉的問題。此外、有些主機對 SAN 開機等組態有非預期的路徑限制。
- 極少數組態需要四條以上的路徑才能連接到 LUN 。如果擁有 LUN 的節點及其 HA 合作夥伴故障、則無法存取主控 LUN 的集合、因此限制將超過兩個節點的路徑通告至 LUN 的價值。在非主要 HA 配對的節點上建立路徑、在這種情況下並無幫助。
- 雖然可視 LUN 路徑的數量可以透過選擇 FC 區域中包含哪些連接埠來進行管理、但通常較容易在 FC 區域中包含所有潛在目標點、並控制 ONTAP 層級的 LUN 可見度。
- 在 ONTAP 8.3 及更新版本中、選擇性 LUN 對應（ SLM ）功能為預設功能。透過 SLM 、任何新的 LUN 都會自動從擁有基礎 Aggregate 的節點和節點的 HA 合作夥伴通告。這種安排可避免建立連接埠集或設定分區、以限制連接埠存取。每個 LUN 都可在最佳效能和恢復能力所需的最小節點數上使用。
- 如果必須在兩個控制器之外移轉 LUN 、則可以使用新增額外的節點 `lun mapping add-reporting-nodes` 命令，以便在新節點上通告 LUN 。這樣做會建立通往 LUN 的額外 SAN 路徑、以進行 LUN 移轉。但是、主機必須執行探索作業、才能使用新路徑。
- 不要過度擔心間接流量。最好在 I/O 密集環境中避免間接流量、因為每微秒的延遲都是關鍵、但對於一般工作負載而言、可見的效能影響卻微不足道。

NFS LIF 設計

與 SAN 通訊協定不同、 NFS 定義多個資料路徑的能力有限。NFSv4 的平行 NFS （ pNFS ）擴充解決了這項限制、但由於乙太網路速度已達 100GB 、而且在新增其他路徑時、很少會有價值。

效能與恢復能力

雖然測量 SAN LIF 效能主要是從所有主要路徑計算總頻寬、但判斷 NFS LIF 效能需要仔細瞭解確切的網路組態。例如、兩個 10Gb 連接埠可設定為原始實體連接埠、或是可設定為連結集合體控制傳輸協定（ LACP ）介面群組。如果將它們設定為介面群組、則可根據流量是交換還是路由、使用不同的多個負載平衡原則。最後、 Oracle Direct NFS （ DNFS ）提供目前不存在於任何 OS NFS 用戶端的負載平衡組態。

與 SAN 通訊協定不同的是、 NFS 檔案系統需要在通訊協定層恢復能力。例如、 LUN 一律設定為啟用多重路徑、表示儲存系統可使用多個備援通道、每個通道都使用 FC 傳輸協定。另一方面、 NFS 檔案系統則取決於單一 TCP/IP 通道的可用度、而該通道只能在實體層受到保護。這種配置是為何存在連接埠容錯移轉和 LACP 連接埠集合等選項。

在 NFS 環境中、網路傳輸協定層會同時提供效能和恢復能力。因此、這兩個主題彼此交織在一起、必須一起討論。

將生命體繫結至連接埠群組

若要將 LIF 繫結至連接埠群組、請將 LIF IP 位址與一組實體連接埠建立關聯。將實體連接埠集合在一起的主要方法是 LACP 。LACP 的容錯功能相當簡單； LACP 群組中的每個連接埠都會受到監控、並在發生故障時從連接埠群組中移除。不過、對於 LACP 在效能方面的運作方式、有許多誤解：

- LACP 不需要在交換器上進行組態以符合端點。例如、 ONTAP 可設定 IP 型負載平衡、而交換器則可使用 MAC 型負載平衡。
- 使用 LACP 連線的每個端點可以個別選擇封包傳輸連接埠、但無法選擇用於接收的連接埠。這表示從 ONTAP 到特定目的地的流量會連結到特定連接埠、而傳回流量可能會到達不同的介面。但這不會造成問題。
- LACP 不會一直平均分配流量。在擁有許多 NFS 用戶端的大型環境中、通常甚至會使用 LACP 集合中的所有連接埠。不過、環境中的任何一個 NFS 檔案系統都只能使用一個連接埠的頻寬、而非整個集合。
- 雖然 ONTAP 上有資源配置資源配置資源 LACP 原則、但這些原則並不會解決從交換器到主機的連線問題。例如、主機上有四埠 LACP 主幹的組態、 ONTAP 上有四埠 LACP 主幹的組態、仍只能使用單一連接埠讀取檔案系統。雖然 ONTAP 可以透過所有四個連接埠傳輸資料、但目前沒有任何交換器技術可以透過所有四個連接埠從交換器傳送到主機。僅使用一個。

在包含許多資料庫主機的大型環境中、最常見的方法是使用 IP 負載平衡、建立一個包含適當數量 10Gb （或更快）介面的 LACP 集合體。只要有足夠的用戶端、這種方法就能讓 ONTAP 提供所有連接埠的均勻使用。當組態中的用戶端較少時、負載平衡會中斷、因為 LACP 主幹不會動態重新分配負載。

建立連線後、特定方向的流量只會放置在一個連接埠上。例如、對透過四埠 LACP 主幹連接的 NFS 檔案系統執行完整表格掃描的資料庫、只會透過一個網路介面卡（ NIC ）讀取資料。如果只有三個資料庫伺服器在這種環境中、則可能所有三個都從同一個連接埠讀取、而其他三個連接埠則處於閒置狀態。

將生命與實體連接埠繫結

將 LIF 繫結至實體連接埠、可更精細地控制網路組態、因為 ONTAP 系統上的指定 IP 位址一次只與一個網路連接埠相關聯。然後、可透過設定容錯移轉群組和容錯移轉原則來實現恢復能力。

容錯移轉原則和容錯移轉群組

網路中斷期間的生命行為是由容錯移轉原則和容錯移轉群組所控制。不同版本的 ONTAP 已變更組態選項。請參閱 "適用於容錯移轉群組和原則的 ONTAP 網路管理文件" 以取得所部署 ONTAP 版本的特定詳細資料。

ONTAP 8.3 及更高版本可根據廣播網域來管理 LIF 容錯移轉。因此、系統管理員可以定義所有可存取指定子網路的連接埠、並允許 ONTAP 選取適當的容錯移轉 LIF 。這種方法可由部分客戶使用、但由於缺乏可預測性、因此在高速儲存網路環境中有限制。例如、環境可同時包含 1Gb 連接埠、以供例行檔案系統存取、而 10Gb 連接埠則可用於資料檔案 I/O如果兩種連接埠都存在於同一個廣播網域中、 LIF 容錯移轉可能會導致資料檔案 I/O 從 10Gb 連接埠移至 1Gb 連接埠。

總而言之、請考慮下列實務做法：

1. 將容錯移轉群組設定為使用者定義。
2. 將儲存容錯移轉（ SFO ）合作夥伴控制器上的連接埠填入容錯移轉群組、以便在儲存容錯移轉期間、生命體跟隨集合體。如此可避免產生間接流量。

3. 使用效能特性與原始 LIF 相符的容錯移轉連接埠。例如、單一實體 10Gb 連接埠上的 LIF 應包含單一 10Gb 連接埠的容錯移轉群組。四埠 LACP LIF 應容錯移轉至另一個四埠 LACP LIF 。這些連接埠將是廣播網域中定義的連接埠子集。
4. 將容錯移轉原則設為僅限 SFO 合作夥伴。這樣做可確保 LIF 在容錯移轉期間跟隨集合體。

自動還原

設定 `auto-revert` 視需要設定參數。大多數客戶偏好將此參數設為 `true` 讓 LIF 還原至其主連接埠。不過、在某些情況下、客戶將此設定為「假」、表示在將 LIF 傳回其主連接埠之前、可以調查非預期的容錯移轉。

LIF 與 **Volume** 比率

常見的誤解是、磁碟區和 NFS 生命體之間必須有一對一的關係。雖然在叢集中的任何位置移動磁碟區都需要此組態、但絕不會產生額外的互連流量、但絕對不需要此組態。必須考慮叢集間流量、但僅存在叢集間流量並不會造成問題。為 ONTAP 所發佈的許多基準測試主要包括間接 I/O

例如、資料庫專案中包含相對少數的效能關鍵資料庫、只需要總共 40 個磁碟區、可能需要將 1 ： 1 磁碟區轉換為 LIF 策略、這種安排需要 40 個 IP 位址。然後、任何磁碟區都可以連同相關的 LIF 一起移至叢集中的任何位置、而且流量永遠是直接的、即使在微秒層級、也能將每個延遲來源減至最低。

舉例來說、大型託管環境的管理可能更容易、因為客戶與生命的關係是一對一。隨著時間的推移、可能需要將磁碟區移轉至不同的節點、這會造成一些間接流量。但是、除非互連交換器上的網路連接埠飽和、否則效能影響應該無法偵測。如果有疑慮、可以在其他節點上建立新的 LIF 、並在下一個維護時段更新主機、以移除組態中的間接流量。

用於 **Oracle** 資料庫的 **TCP/IP** 和乙太網路組態

ONTAP 上的許多 Oracle 客戶都使用乙太網路、 NFS 、 iSCSI 、 NVMe / TCP 的網路傳輸協定、尤其是雲端。

主機作業系統設定

大多數應用程式廠商文件都包含特定的 TCP 和乙太網路設定、以確保應用程式能以最佳方式運作。這些相同的設定通常足以提供最佳的 IP 型儲存效能。

乙太網路流量控制

這項技術可讓用戶端要求傳送者暫時停止資料傳輸。這通常是因為接收者無法快速處理傳入的資料。一次、要求傳送者停止傳輸的中斷程度比接收者丟棄封包的中斷程度低、因為緩衝區已滿。現今作業系統中使用的 TCP 堆疊已不再如此。事實上、流量控制所造成的問題比解決的問題還多。

近年來、乙太網路流量控制所造成的效能問題不斷增加。這是因為乙太網路流量控制是在實體層運作。如果網路組態允許任何主機作業系統將乙太網路流量控制要求傳送至儲存系統、則所有連線的用戶端都會暫停 I/O 。由於單一儲存控制器服務的用戶端數量不斷增加、因此其中一或多個用戶端傳送流量控制要求的可能性會增加。在擁有廣泛作業系統虛擬化的客戶據點、經常會發現這個問題。

NetApp 系統上的 NIC 不應接收流量控制要求。實現此結果的方法因網路交換器製造商而異。在大多數情況下、可將乙太網路交換器上的流量控制設定為 `receive desired` 或 `receive on`，這意味着流控制請求不會轉發到存儲控制器。在其他情況下、儲存控制器上的網路連線可能不允許停用流程控制。在這些情況下、用戶端必須設定為永遠不要傳送流量控制要求、方法是變更至主機伺服器本身的 NIC 組態、或是變更主機伺服器所連接的交換器連接埠。

* NetApp 建議 * 確保 NetApp 儲存控制器不會接收乙太網路流量控制封包。這通常可以透過設定控制器所連接的交換器連接埠來完成、但有些交換器硬體有限制、可能需要改用用戶端端變更。

MTU 大小

使用巨型框架的結果顯示、透過降低 CPU 和網路成本、可在速度較低的網路中提供一些效能改善、但效益通常並不顯著。

* NetApp 建議 * 盡可能實作巨型框架、以實現任何可能的效能效益、並確保解決方案符合未來需求。

在 10Gb 網路中使用巨型框架幾乎是強制性的。這是因為大多數的 10Gb 實作都達到每秒封包數的限制、而不需要巨型框架、就能達到 10Gb 標誌。使用巨型框架可改善 TCP/IP 處理效率、因為它可讓作業系統、伺服器、NIC 和儲存系統處理較少但較大的封包。效能的改善因 NIC 而異、但成效相當顯著。

對於巨型框架實作、通常但不正確的看法是、所有連線的裝置都必須支援巨型框架、而且 MTU 大小必須與端點對端點相符而是在建立連線時、兩個網路端點會協商最高的雙方可接受的框架大小。在一般環境中、網路交換器的 MTU 大小設為 9216 、 NetApp 控制器設為 9000 、用戶端則設為 9000 和 1514 的混合。支援 9000 MTU 的用戶端可以使用巨型框架、而只支援 1514 的用戶端可以協商較低的值。

在完全交換的環境中、這種配置的問題很少發生。不過、在沒有中繼路由器被迫分割巨型框架的路由環境中、請務必小心。

* NetApp 建議 * 設定下列項目：

- 使用 1 GB 乙太網路（ GbE ）時、巨型框架是理想的選擇、但不是必要的。
- 使用 10GbE 及更快的速度、需要巨型框架才能達到最佳效能。

TCP 參數

三項設定通常設定錯誤： TCP 時間戳記、選擇性認可（ SACK ）和 TCP 視窗縮放。網際網路上的許多過時文件建議停用一或多個這些參數、以改善效能。這項建議在多年前就有一些優點、因為 CPU 功能較低、因此有助於盡可能降低 TCP 處理的成本。

然而、在現代化的作業系統中、停用任何這些 TCP 功能通常會導致無法偵測的效益、同時也可能造成效能受損。在虛擬化網路環境中、效能受損的可能性特別大、因為這些功能是有效處理封包遺失和網路品質變更所必需的。

* NetApp 建議 * 在主機上啟用 TCP 時間戳記、 SACK 和 TCP 視窗縮放功能、而且在任何目前的作業系統中、這三個參數都應該預設為開啟。

Oracle 資料庫的 FC 組態

為 Oracle 資料庫設定 FC SAN 主要是為了遵循日常的 SAN 最佳實務做法。

這包括典型的規劃措施、例如確保主機和儲存系統之間的 SAN 上有足夠的頻寬、使用 FC 交換器廠商所需的 FC 連接埠設定、檢查所有必要裝置之間是否存在所有 SAN 路徑、避免 ISL 爭用、 並使用適當的 SAN 架構監控。

分區

FC 區域不得包含多個啟動器。這種安排一開始可能會運作、但啟動器之間的串擾最終會影響效能和穩定性。

雖然在極少數情況下、來自不同廠商的 FC 目標連接埠行為造成問題、但多目標區域通常被視為安全區域。例如、避免將 NetApp 和非 NetApp 儲存陣列的目標連接埠同時納入同一區域。此外、將 NetApp 儲存系統和磁帶裝置置於同一個區域、更有可能造成問題。

Oracle 資料庫與直接連線 ONTAP 連線

儲存管理員有時偏好從組態中移除網路交換器、以簡化其基礎架構。在某些情況下可能會支援這項功能。

iSCSI 和 NVMe / TCP

使用 iSCSI 或 NVMe / TCP 的主機可以直接連線至儲存系統、並正常運作。原因是路徑。直接連線至兩個不同的儲存控制器、會產生兩個不同的資料流路徑。遺失路徑、連接埠或控制器並不會妨礙其他路徑的使用。

NFS

可以使用直接連線的 NFS 儲存設備、但有很大的限制：如果沒有大量的指令碼工作、容錯移轉將無法運作、這是客戶的責任。

直接連線的 NFS 儲存設備會造成不中斷的容錯移轉複雜化、這是因為本機作業系統上會發生路由。例如、假設主機的 IP 位址為 192.168.1.1/24 、並直接連線至 IP 位址為 192.168.1.50/24 的 ONTAP 控制器。在容錯移轉期間、該位址 192.168.1.50 可以容錯移轉至其他控制器、而且主機可以使用該位址、但主機如何偵測其存在？原來的 192.168.1.1 位址仍然存在於不再連線至作業系統的主機 NIC 上。目的地為 192.168.1.5 的流量將繼續傳送至無法運作的網路連接埠。

第二個 OS NIC 可設定為 19 可以與故障的 over 192.168.1.50 位址進行通訊、但本機路由表預設會使用一個 * 且只有一個 * 位址來與 192.168.1.0/24 子網路通訊。系統管理員可以建立指令碼架構、以偵測失敗的網路連線、並變更本機路由表或使介面正常運作。具體程序取決於所使用的作業系統。

實際上、 NetApp 客戶確實有直接連線的 NFS 、但通常僅適用於容錯移轉期間 IO 暫停的工作負載。使用硬掛載時、在這類暫停期間不應有任何 IO 錯誤。IO 應該會暫停運作、直到服務還原為止、無論是透過容錯回復或手動介入、在主機上的 NIC 之間移動 IP 位址。

FC 直接連線

無法使用 FC 傳輸協定將主機直接連接至 ONTAP 儲存系統。原因是使用 NPIV 。用於識別 FC 網路的 ONTAP FC 連接埠的 WWN 使用稱為 NPIV 的虛擬化類型。任何連接至 ONTAP 系統的裝置都必須能夠辨識 NPIV WWN 。目前沒有任何 HBA 廠商提供可安裝在能夠支援 NPIV 目標的主機上的 HBA 。

儲存組態

FC SAN

Oracle 資料庫 I/O 的 LUN 對齊

LUN 對齊是指針對基礎檔案系統配置最佳化 I/O 。

在 ONTAP 系統上、儲存設備是以 4KB 為單位進行組織。資料庫或檔案系統 8KB 區塊應對應至兩個 4KB 區塊。如果 LUN 組態發生錯誤、在任一方向將對齊移至 1KB 、則每個 8KB 區塊會存在於三個不同的 4KB 儲存區塊、而非兩個。這種安排會導致延遲增加、並導致在儲存系統中執行額外的 I/O 。

對齊也會影響 LVM 架構。如果在整個磁碟機裝置上定義邏輯磁碟區群組內的實體磁碟區（不建立分割區）、 LUN 上的前 4KB 區塊會與儲存系統上的前 4KB 區塊對齊。這是正確的對齊方式。磁碟分割發生問題、因為它們會移轉作業系統使用 LUN 的起始位置。只要偏移量以 4KB 的整體單位移動、 LUN 就會對齊。

在 Linux 環境中、在整個磁碟機裝置上建立邏輯磁碟區群組。當需要磁碟分割時、請執行檢查對齊 `fdisk -u` 並驗證每個分割區的開始時間為八個之倍數。這表示分割區從八個 512 位元組磁區的倍數開始、即 4KB 。

另請參閱一節中有關壓縮區塊對齊的討論 "效率"。任何與 8KB 壓縮區塊邊界對齊的配置、也會與 4KB 邊界對齊。

錯誤對齊警告

資料庫重做 / 交易記錄通常會產生未對齊的 I/O 、導致 ONTAP 上未對齊 LUN 的錯誤警告。

記錄會以不同大小的寫入方式、連續寫入記錄檔。不符合 4KB 界限的記錄寫入作業通常不會造成效能問題、因為下一個記錄寫入作業會完成區塊。結果是 ONTAP 幾乎能將所有寫入作業視為完整的 4KB 區塊來處理、即使某些 4KB 區塊中的資料是以兩個不同的作業來寫入。

使用公用程式（例如）來驗證對齊 `sio` 或 `dd` 可在定義的區塊大小下產生 I/O 。您可以使用檢視儲存系統上的 I/O 對齊統計資料 `stats` 命令。請參閱 "WAFL 對齊驗證" 以取得更多資訊。

在 Solaris 環境中進行對齊更為複雜。請參閱 "SAN主機組態ONTAP" 以取得更多資訊。

注意
在 Solaris x86 環境中、由於大多數組態都有多層分割區、因此請格外注意正確的對齊方式。Solaris x86 分割區磁碟片通常位於標準主開機記錄分割區表格的上方。

Oracle 資料庫 LUN 規模調整和 LUN 數量

選擇最佳 LUN 大小和要使用的 LUN 數量、對於 Oracle 資料庫的最佳效能和管理性至關重要。

LUN 是 ONTAP 上的虛擬化物件、存在於託管集合體中的所有磁碟機中。因此、 LUN 的效能不受其大小影響、因為無論選擇何種大小、 LUN 都會充分發揮彙總的效能潛力。

為了方便起見、客戶可能想要使用特定大小的 LUN 。例如、如果資料庫建置在由兩個 LUN 組成的 LVM 或 Oracle ASM 磁碟群組上、每個 LUN 均為 1TB 、則該磁碟群組必須以 1TB 為增量來擴充。最好是從八個 LUN （每個 LUN 爲 500GB ）構建磁盤組，以便可以以更小的增量來增加磁盤組。

我們不鼓勵建立通用標準 LUN 大小的做法、因為這樣做可能會使管理變得複雜。例如、當資料庫或資料存放區的範圍介於 1TB 到 2TB 時、 100GB 的標準 LUN 大小可能運作良好、但大小為 20TB 的資料庫或資料存放區需要 200 個 LUN 。這表示伺服器重新開機時間較長、不同 UI 中需要管理的物件較多、而 SnapCenter 等產品必須在許多物件上執行探索。使用較少、較大的 LUN 可避免此類問題。

- LUN 數量比 LUN 大小更重要。
- LUN 大小大多由 LUN 數需求控制。
- 避免建立超過所需數量的 LUN 。

LUN 計數

與 LUN 大小不同、 LUN 數量確實會影響效能。應用程式效能通常取決於透過 SCSI 層執行平行 I/O 的能力。因此、兩個 LUN 的效能優於單一 LUN 。使用 LVM （例如 Veritas VxVM 、 Linux LVM2 或 Oracle ASM ）是提高平行度的最簡單方法。

NetApp 客戶通常從 LUN 數量增加到 16 個以上獲得最小的效益、不過測試 100% SSD 環境時、隨機 I/O 非常繁重、這已證實可進一步改善至 64 個 LUN 。

- NetApp 建議 * 下列事項：

一般而言、四到十六個 LUN 足以支援任何特定資料庫工作負載的 I/O 需求。由於主機 SCSI 實作的限制、少於四個 LUN 可能會造成效能限制。

Oracle 資料庫 LUN 放置

資料庫 LUN 在 ONTAP 磁碟區內的最佳放置方式、主要取決於如何使用各種 ONTAP 功能。

磁碟區

與剛接觸 ONTAP 的客戶混淆的一個常見點是使用 FlexVols 、通常稱為「 Volume 」。

磁碟區不是 LUN 。這些詞彙與許多其他廠商產品（包括雲端供應商）同義。ONTAP Volume 是簡單的管理容器。它們本身不會提供資料、也不會佔用空間。它們是檔案或 LUN 的容器、可改善及簡化管理、尤其是大規模管理。

磁碟區和 LUN

相關 LUN 通常位於單一磁碟區中。例如、需要 10 個 LUN 的資料庫通常會將所有 10 個 LUN 放在同一個磁碟區上。

- 使用 LUN 對磁碟區的比例 1 ： 1 表示每個磁碟區有一個 LUN 、這是 * 非 * 正式最佳實務做法。
- 而是應將磁碟區視為工作負載或資料集的容器。每個磁碟區可能只有一個 LUN 、或者可能有許多 LUN 。正確的答案取決於管理需求。
- 在不必要數量的磁碟區之間分散 LUN 、可能會導致額外的額外負荷和排程問題、例如快照作業、 UI 中顯示的物件過多、並導致在達到 LUN 限制之前達到平台磁碟區限制。

磁碟區、 LUN 和快照

Snapshot 原則和排程會放置在磁碟區上、而非 LUN 上。如果資料集由 10 個 LUN 組成、則當這些 LUN 位於同一個磁碟區中時、只需要單一快照原則。

此外、在單一磁碟區中共同定位給定資料集的所有相關 LUN 、可提供原子快照作業。例如、如果基礎 LUN 全部放在單一磁碟區上、則位於 10 個 LUN 上的資料庫、或是由 10 個不同作業系統組成的 VMware 應用程式環境、都可以作為單一且一致的物件加以保護。如果將快照放在不同的磁碟區上、則即使同時排程、快照仍可能保持 100% 同步。

在某些情況下、由於恢復需求、相關的 LUN 集可能需要分割成兩個不同的磁碟區。例如、資料庫可能有四個 LUN 用於資料檔案、兩個 LUN 用於記錄。在這種情況下、具有 4 個 LUN 的資料檔案磁碟區和具有 2 個 LUN

的記錄磁碟區可能是最佳選擇。原因在於可進行的可恢復性是不相關的。例如、資料檔案磁碟區可以選擇性地還原為較早的狀態、這表示所有四個 LUN 都會還原為快照狀態、而記錄磁碟區與其重要資料則不會受到影響。

Volume 、 LUN 和 SnapMirror

SnapMirror 原則和作業就像快照作業一樣、是在磁碟區上執行、而不是在 LUN 上執行。

在單一磁碟區中共同定位相關 LUN 、可讓您建立單一 SnapMirror 關係、並透過單一更新來更新所有包含的資料。與快照一樣、更新也將是一項原子作業。SnapMirror 目的地將保證擁有來源 LUN 的單一時間點複本。如果 LUN 分散在多個磁碟區、則複本可能彼此一致、也可能不一致。

磁碟區、 LUN 和 QoS

雖然 QoS 可以選擇性地套用至個別 LUN 、但通常在磁碟區層級設定 QoS 會比較容易。例如、指定 ESX 伺服器中的來賓所使用的所有 LUN 都可以放置在單一磁碟區上、然後就可以套用 ONTAP 調適性 QoS 原則。結果是將每 TB IOPS 的自我擴充限制套用至所有 LUN 。

同樣地、如果資料庫需要 10 萬次 IOPS 、而且佔用 10 個 LUN 、則在單一磁碟區上設定單一的 10 萬次 IOPS 限制、比在每個 LUN 上設定 10 個個別的 10K IOPS 限制更容易。

多重 Volume 配置

在某些情況下、跨多個磁碟區散佈 LUN 可能會有幫助。主要原因是控制器分段。例如、 HA 儲存系統可能會裝載單一資料庫、其中需要每個控制器的完整處理與快取潛力。在這種情況下、典型的設計是將一半的 LUN 放在控制器 1 的單一磁碟區、而另一半的 LUN 則放在控制器 2 的單一磁碟區中。

同樣地、控制器分段也可用於負載平衡。HA 系統託管 100 個資料庫、每個資料庫各有 10 個 LUN 、每個資料庫可在兩個控制器上接收 5 個 LUN 磁碟區。如此一來、每個控制器就能以對稱的方式進行對稱載入、同時還能配置額外的資料庫。

不過、這些範例都不涉及 1 ： 1 的磁碟區對 LUN 比率。目標仍然是透過在磁碟區中共同定位相關 LUN 來最佳化管理性。

其中一個例子是、 1 ： 1 LUN 對磁碟區比率非常合理、其中每個 LUN 可能真正代表單一工作負載、而且每個工作負載都需要個別管理。在這種情況下、 1 ： 1 的比率可能是最佳的。

Oracle 資料庫 LUN 調整大小和以 LVM 為基礎的調整大小

當 SAN 型檔案系統達到容量上限時、有兩個選項可以增加可用空間：

- 增加 LUN 的大小
- 將 LUN 新增至現有的磁碟區群組、並擴充內含的邏輯磁碟區

雖然 LUN 調整大小是增加容量的選項、但通常最好使用 LVM 、包括 Oracle ASM 。LVM 存在的主要原因之一、是為了避免需要調整 LUN 大小。使用 LVM 時、多個 LUN 會結合在一個虛擬儲存池中。從該池中切出的邏輯卷由 LVM 管理，可以輕鬆調整大小。另一項優點是在所有可用 LUN 之間分配給定的邏輯磁碟區、以避免在特定磁碟機上出現熱點。通常可以使用 Volume Manager 將邏輯磁碟區的基礎範圍重新放置到新的 LUN 、以執行透明移轉。

使用 Oracle 資料庫的 LVM 分拆

LVM 分拆是指在多個 LUN 之間分配資料。如此一來、許多資料庫的效能大幅提升。

在快閃磁碟機時代之前、使用區塊延展來協助克服旋轉磁碟機的效能限制。例如、如果作業系統需要執行 1MB 讀取作業、則從單一磁碟機讀取 1MB 的資料時、需要大量的磁碟機磁頭搜尋和讀取、因為 1MB 會緩慢傳輸。如果將 1MB 的資料分散在 8 個 LUN 上、則作業系統可能會同時執行 8 個 128K 讀取作業、並縮短完成 1MB 傳輸所需的時間。

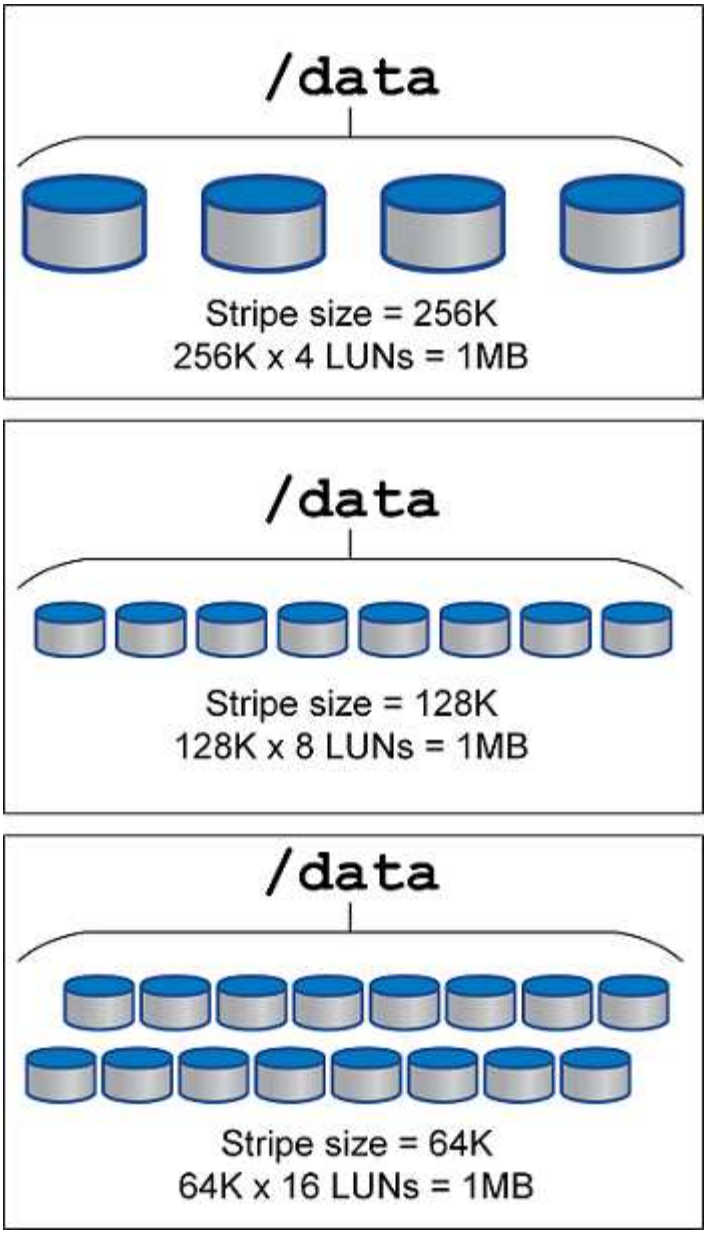
由於必須事先知道 I/O 模式、因此使用旋轉磁碟機進行分拆會更困難。如果串列區塊延展未針對真正的 I/O 模式正確調整、則等量區塊配置可能會損害效能。使用 Oracle 資料庫、特別是搭配 All Flash 組態、分拆作業更容易設定、並經證實可大幅提升效能。

依預設、邏輯磁碟區管理程式（例如 Oracle ASM 等量磁碟區）不屬於原生 OS LVM 。其中有些 LUN 會將多個 LUN 連結在一起、成為串連的裝置、導致資料檔案存在於一台 LUN 裝置上、而只存在於一台 LUN 裝置上。這會造成熱點。其他 LVM 實作預設為分散式擴充。這與分拆類似、但卻是比較粗糙的。磁碟區群組中的 LUN 會切成大型片段、稱為區段、通常以百萬位元組為單位測量、然後邏輯磁碟區會分佈在這些區段中。結果是對檔案進行隨機 I/O 、應該能在 LUN 之間妥善分配、但連續 I/O 作業的效率卻不如以前那麼高。

效能密集的應用程式 I/O 幾乎總是（ a ）以基本區塊大小為單位、或（ b ） 1 MB 。

等量分配組態的主要目標是確保單一檔案 I/O 可作為單一單元執行、而多區塊 I/O 的大小應為 1MB 、可在等量磁碟區中的所有 LUN 之間平均平行處理。這表示等量磁碟區大小不得小於資料庫區塊大小、且等量磁碟區大小乘以 LUN 數量應為 1MB 。

下圖顯示等量磁碟區大小和寬度調校的三個可能選項。選擇 LUN 數量以滿足上述效能需求、但在所有情況下、單一等量磁碟區內的總資料為 1MB 。



NFS

Oracle 資料庫的 NFS 組態

NetApp 已提供企業級 NFS 儲存設備超過 30 年、由於其簡易性、隨著雲端型基礎架構的推向、其使用量也不斷增加。

NFS 傳輸協定包含多個不同需求的版本。如需 ONTAP 的 NFS 組態完整說明、請參閱 "TR-4067 ONTAP 最佳實務做法上的 NFS"。下列各節涵蓋一些較重要的需求和一般使用者錯誤。

NFS 版本

NetApp 必須支援作業系統 NFS 用戶端。

- NFSv3 支援符合 NFSv3 標準的作業系統。
- Oracle DNFS 用戶端支援 NFSv3 。

- 所有遵循 NFSv4 標準的作業系統都支援 NFSv4 。
- NFSv4.1 和 NFSv4.2 需要特定的作業系統支援。請參閱 "NetApp IMT" 適用於支援的作業系統。
- Oracle DNFS 支援 NFSv4.1 需要 Oracle 12.2.0.2 或更高版本。

。 "NetApp 支援對照表" 對於 NFSv3 和 NFSv4 、不包含特定的作業系統。一般支援所有遵守 RFC 的作業系統。搜尋線上 IMT 以取得 NFSv3 或 NFSv4 支援時、請勿選取特定的作業系統、因為不會顯示任何相符項目。一般原則隱含支援所有作業系統。

Linux NFSv3 TCP 插槽表

TCP 插槽表是與主機匯流排介面卡（ HBA ）佇列深度相當的 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 。

注意
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。

ADR 和 NFS

部分客戶回報的效能問題是由於中的資料 I/O 過多所造成 `ADR` 位置。問題通常不會在累積許多效能資料之前發生。I/O 過多的原因不明、但此問題似乎是 Oracle 處理程序重複掃描目標目錄以進行變更所致。

移除 `noac` 和/或 `actimeo=0` 掛載選項可執行主機作業系統快取、並降低儲存 I/O 層級。

* NetApp 建議 * 不要放置 `ADR` 檔案系統上的資料 `noac` 或 `actimeo=0` 因為效能問題很可能會發生。獨立 `ADR` 如有必要、請將資料移至不同的掛載點。

NFS-rootonly 和 mount-rootonly

ONTAP 包含一個稱為的 NFS 選項 `nfs-rootonly` 控制伺服器是否接受來自高連接埠的 NFS 流量連線。為了安全起見、只有 root 使用者可以使用低於 1024 的來源連接埠來開啟 TCP/IP 連線、因為這類連接埠通常是保留給作業系統使用、而非使用者處理程序。此限制有助於確保 NFS 流量來自實際的作業系統 NFS 用戶端、而非模擬 NFS 用戶端的惡意程序。Oracle DNFS 用戶端是 userspace 驅動程式、但程序是以 root 執行、因此通常不需要變更的值 `nfs-rootonly`。連線是從低連接埠建立。

。 `mount-rootonly` 選項僅適用於 NFSv3 。它控制是否從大於 1024 的連接埠接受 RPC 掛載呼叫。使用 DNFS 時、用戶端會再次以 root 執行、因此能夠開啟低於 1024 的連接埠。此參數無效。

透過 NFS 4.0 及更高版本開啟與 DNFS 連線的程序不會以 root 執行、因此需要 1024 以上的連接埠。。 `nfs-rootonly` 參數必須設為停用、 DNFS 才能完成連線。

如果 `nfs-rootonly` 啟用時、結果會在掛載階段開啟 DNFS 連線時暫停。sqlplus 輸出類似於以下內容：

```
SQL>startup
ORACLE instance started.
Total System Global Area 4294963272 bytes
Fixed Size                  8904776 bytes
Variable Size             822083584 bytes
Database Buffers         3456106496 bytes
Redo Buffers                7868416 bytes
```

參數可變更如下：

```
Cluster01::> nfs server modify -nfs-rootonly disabled
```

在極少數情況下、您可能需要將 NFS-rootonly 和 mount-rootonly 都變更為停用。如果伺服器正在管理大量的 TCP 連線、則可能沒有低於 1024 的連接埠可用、而且作業系統必須使用較高的連接埠。需要變更這兩個 ONTAP 參數、才能完成連線。

NFS 匯出原則：超級使用者和 setuid

如果 Oracle 二進位檔位於 NFS 共用區、則匯出原則必須包含超級使用者和 setuid 權限。

用於一般檔案服務（例如使用者主目錄）的共享 NFS 匯出通常會佔用 root 使用者。這表示已掛載檔案系統的主機上 root 使用者的要求、會重新對應為具有較低權限的不同使用者。這有助於防止特定伺服器上的根使用者存取共用伺服器上的資料、進而保護資料安全。在共享環境中， setuid 位元也可能是安全性風險。setuid 位元可讓處理程序以不同於使用者的身分執行、而非以使用者的身分執行指令。例如、 root 擁有的 Shell 指令碼搭配 setuid 位元、會以 root 執行。如果其他使用者可以變更該 Shell 指令碼、任何非 root 使用者都可以透過更新指令碼、以 root 身分發出命令。

Oracle 二進位檔包含 root 擁有的檔案、並使用 setuid 位元。如果在 NFS 共用上安裝 Oracle 二進位檔、匯出原則必須包含適當的超級使用者和 setuid 權限。在以下範例中、這兩項規則都包含在內 `allow-suid` 及許可 `superuser` （ root ）使用系統驗證來存取 NFS 用戶端。

```
Cluster01::> export-policy rule show -vserver vserver1 -policyname orabin
-fields allow-suid,superuser
vserver   policyname ruleindex superuser allow-suid
--------- ---------- --------- --------- ----------
vserver1  orabin     1         sys       true
```

NFSv4/4.1 組態

對於大多數應用程式、 NFSv3 和 NFSv4 之間的差異很小。應用程式 I/O 通常是非常簡單的 I/O 、而且不會從 NFSv4 中提供的某些進階功能中獲得顯著效益。較高版本的 NFS 不應從資料庫儲存的角度視為「升級」、而應視為包含其他功能的 NFS 版本。例如、如果需要 Kerberos 隱私模式（ krb5p ）的端點對端安全性、則需要 NFSv4 。

* 如果需要 NFSv4 功能、 NetApp 建議 * 使用 NFSv4.1 。NFSv4.1 中的 NFSv4 傳輸協定有一些功能性增強功能、可改善某些邊緣情況的恢復能力。

切換至 NFSv4 比單純將掛載選項從 ves=3 變更為 ves=4.1 更複雜。如需更完整的 NFSv4 組態與 ONTAP 說明、包括作業系統設定指南、請參閱 "TR-4067 ONTAP 最佳實務做法上的 NFS"。本 TR 的下列各節說明使用 NFSv4 的一些基本要求。

NFSv4 網域

NFSv4/4.1 組態的完整說明已超出本文件的範圍、但常見的問題之一是網域對應不相符。從系統管理員的角度來看、 NFS 檔案系統的行為似乎正常、但應用程式會報告某些檔案的權限和 / 或 setuid 錯誤。在某些情況下、系統管理員不正確地判斷應用程式二進位檔的權限已受損、並在實際問題是網域名稱時執行 chown 或 chmod 命令。

NFSv4 網域名稱是在 ONTAP SVM 上設定：

```
Cluster01::> nfs server show -fields v4-id-domain
vserver   v4-id-domain
--------- ------------
vserver1  my.lab
```

主機上的 NFSv4 網域名稱是在中設定 `/etc/idmap.cfg`

```
[root@host1 etc]# head /etc/idmapd.conf
[General]
#Verbosity = 0
# The following should be set to the local NFSv4 domain name
# The default is the host's DNS domain name.
Domain = my.lab
```

網域名稱必須相符。如果沒有、則會在中顯示類似下列的對應錯誤 `/var/log/messages`：

```
Apr 12 11:43:08 host1 nfsidmap[16298]: nss_getpwnam: name 'root@my.lab'
does not map into domain 'default.com'
```

應用程式二進位檔（例如 Oracle 資料庫二進位檔）包含 root 擁有的具有 setuid 位元的檔案、這表示 NFSv4 網域名稱不相符會導致 Oracle 啟動失敗、並會發出呼叫檔案擁有權或權限的警告 `oradism`、位於 `$ORACLE_HOME/bin` 目錄。其內容應如下所示：

```
[root@host1 etc]# ls -l /orabin/product/19.3.0.0/dbhome_1/bin/oradism
-rwsr-x--- 1 root oinstall 147848 Apr 17  2019
/orabin/product/19.3.0.0/dbhome_1/bin/oradism
```

如果此檔案的擁有權為 nobody 、則可能是 NFSv4 網域對應問題。

```
[root@host1 bin]# ls -l oradism
-rwsr-x--- 1 nobody oinstall 147848 Apr 17  2019 oradism
```

若要修正此問題、請參閱 `/etc/idmap.cfg` 根據 ONTAP 上的 vv4 識別碼網域設定來建立檔案、並確保檔案一致。如果沒有、請進行必要的變更、然後執行 `nfsidmap -c`，然後等待一段時間讓變更傳播。接著、檔案擁有權應正確辨識為 root 。如果使用者嘗試執行 `chown root` 在 NFS 網域設定修正之前、可能需要在這個檔案上執行 `chown root` 再一次。

Oracle directNFS

Oracle 資料庫可使用 NFS 的方式有兩種。

首先、它可以使用以作業系統一部分的原生 NFS 用戶端所掛載的檔案系統。這有時稱為核心 NFS 或 kNFS 。NFS 檔案系統是由 Oracle 資料庫安裝及使用、與任何其他應用程式使用 NFS 檔案系統的方式完全相同。

第二種方法是 Oracle Direct NFS （ DNFS ）。這是在 Oracle 資料庫軟體中實作 NFS 標準。它不會改變 DBA 設定或管理 Oracle 資料庫的方式。只要儲存系統本身有正確的設定、就應該對 DBA 團隊和終端使用者透明使用 DNFS 。

啟用 DNFS 功能的資料庫仍會掛載一般的 NFS 檔案系統。資料庫開啟後、 Oracle 資料庫會開啟一組 TCP/IP 工作階段、並直接執行 NFS 作業。

Direct NFS

Oracle Direct NFS 的主要值是略過主機 NFS 用戶端、並直接在 NFS 伺服器上執行 NFS 檔案作業。啟用它只需要變更 Oracle 磁碟管理程式（ ODM ）程式庫。Oracle 說明文件中提供此程序的說明。

使用 DNFS 可大幅提升 I/O 效能、並降低主機和儲存系統的負載、因為 I/O 是以最有效率的方式執行。

此外、 Oracle DNFS 還包含 * 選項 * 、可用於網路介面多重路徑和容錯功能。例如、兩個 10Gb 介面可以結合在一起、以提供 20Gb 的頻寬。如果某個介面發生故障、則會在另一個介面上重試 I/O 。整體作業與 FC 多重路徑非常類似。多重路徑在數年前是最常見的標準、那就是 1 個乙太網路。10Gb NIC 足以應付大多數 Oracle 工作負載、但如果需要更多 10Gb NIC 、則可加以連結。

使用 DNFS 時、必須安裝 Oracle Doc 1495104.1 中所述的所有修補程式。如果無法安裝修補程式、則必須評估環境、確保該文件中所述的錯誤不會造成問題。在某些情況下、無法安裝所需的修補程式會導致無法使用 DNFS 。

請勿將 DNFS 用於任何類型的循環名稱解析、包括 DNS 、 DDNS 、 NIS 或任何其他方法。這包括 ONTAP 中可用的 DNS 負載平衡功能。當使用 DNFS 的 Oracle 資料庫將主機名稱解析為 IP 位址時、後續查詢時不得變更。這可能會導致 Oracle 資料庫當機、並可能導致資料毀損。

直接 NFS 和主機檔案系統存取

使用 DNFS 有時會導致依賴掛載在主機上的可見檔案系統的應用程式或使用者活動發生問題、因為 DNFS 用戶端會從主機作業系統不定期存取檔案系統。DNFS 用戶端可以在不瞭解作業系統的情況下建立、刪除及修改檔案。

使用單一執行個體資料庫的掛載選項時、會啟用檔案和目錄屬性的快取、這也表示目錄內容會快取。因此、DNFS 可以建立檔案、而且在作業系統重新讀取目錄內容和讓使用者看到檔案之前、會有短暫的延遲。這通常不是問題、但在極少數情況下、 SAP BR*Tools 等公用程式可能會發生問題。如果發生這種情況、請變更掛載選項、以使用 Oracle RAC 的建議來解決此問題。這項變更會導致停用所有主機快取。

只有在使用（ a ） DNFS 時才變更掛載選項、且（ b ）檔案可見度延遲所造成的問題。如果未使用 DNFS 、在單一執行個體資料庫上使用 Oracle RAC 掛載選項會導致效能降低。

請參閱附註 `nosharecache` 在中 "Linux NFS 裝載選項" 針對可能產生異常結果的 Linux 特定 DNFS 問題。

Oracle 資料庫和 NFS 會租用和鎖定

NFSv3 無狀態。這實際上意味著 NFS 伺服器（ ONTAP ）無法追蹤哪些檔案系統是掛載的、由誰掛載、或哪些鎖定是真的就位。

ONTAP 確實有一些功能會記錄掛載嘗試、因此您可以知道哪些用戶端可能正在存取資料、而且可能會出現諮詢鎖定、但這項資訊並不保證 100% 完整。無法完成、因為追蹤 NFS 用戶端狀態並非 NFSv3 標準的一部分。

NFSv4 狀態

相反地、 NFSv4 是有狀態的。NFSv4 伺服器會追蹤哪些用戶端正在使用哪些檔案系統、哪些檔案存在、哪些檔案和 / 或檔案區域被鎖定等 這表示 NFSv4 伺服器之間需要定期通訊、才能保持狀態資料最新。

NFS 伺服器所管理的最重要狀態是 NFSv4 鎖定和 NFSv4 租賃、它們彼此之間有很大的關聯。您必須瞭解每個項目本身的運作方式、以及它們彼此之間的關係。

NFSv4 鎖定

有了 NFSv3 、鎖定是建議事項。NFS 用戶端仍可修改或刪除「鎖定」檔案。NFSv3 鎖本身不會過期、必須將其移除。這會造成問題。例如、如果叢集式應用程式會建立 NFSv3 鎖定、而其中一個節點發生故障、您該怎麼做？您可以在仍在運作的節點上對應用程式進行編碼、以移除鎖定、但您如何知道這是安全的？可能是「故障」節點可以運作、但無法與叢集的其他部分通訊？

有了 NFSv4 、鎖定的持續時間有限。只要持有鎖定的用戶端繼續與 NFSv4 伺服器簽入、就不允許其他用戶端取得這些鎖定。如果用戶端無法使用 NFSv4 進行存回、伺服器最終會撤銷鎖定、而其他用戶端則能要求並取得鎖定。

NFSv4 租賃

NFSv4 鎖定與 NFSv4 租用相關聯。當 NFSv4 用戶端與 NFSv4 伺服器建立連線時、它會取得租用。如果用戶端取得鎖定（鎖定類型眾多）、則鎖定會與租用相關聯。

此租用具有定義的逾時時間。根據預設、 ONTAP 會將逾時值設為 30 秒：

```
Cluster01::*> nfs server show -vserver vserver1 -fields v4-lease-seconds

vserver   v4-lease-seconds
--------- ----------------
vserver1  30
```

這表示 NFSv4 用戶端需要每 30 秒與 NFSv4 伺服器簽入一次、才能續約。

租賃會自動由任何活動續約、因此如果用戶端正在工作、就不需要執行額外作業。如果某個應用程式變得很安靜、而且沒有真正的工作、則需要改為執行某種保持活動狀態的作業（稱為順序）。基本上只是說：「我還在這裏、請重新整理我的租約。」

Question: What happens if you lose network connectivity for 31 seconds?
NFSv3 無狀態。這並不需要用戶端的通訊。NFSv4
可設定狀態、一旦租用期間結束、租用即會過期、鎖定會被撤銷、而鎖定的檔案會提供給其他用戶端使用。

有了 NFSv3 、您可以四處移動網路纜線、重新啟動網路交換器、進行組態變更、並確保不會發生任何問題。應用程式通常只會耐心等待網路連線再次運作。

有了 NFSv4 、您有 30 秒的時間（除非您已在 ONTAP 中增加該參數的值）來完成工作。如果您超過此上限、您的租約將會逾時。這通常會導致應用程式當機。

舉例來說、如果您有 Oracle 資料庫、而且網路連線中斷（有時稱為「網路分割區」）超過租用逾時、您就會使資料庫當機。

以下是 Oracle 警示記錄中發生這種情況的範例：

```
2022-10-11T15:52:55.206231-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00202: control file: '/redo0/NTAP/ctrl/control01.ctl'
ORA-27072: File I/O error
Linux-x86_64 Error: 5: Input/output error
Additional information: 4
Additional information: 1
Additional information: 4294967295
2022-10-11T15:52:59.842508-04:00
Errors in file /orabin/diag/rdbms/ntap/NTAP/trace/NTAP_ckpt_25444.trc:
ORA-00206: error in writing (block 3, # blocks 1) of control file
ORA-00202: control file: '/redo1/NTAP/ctrl/control02.ctl'
ORA-27061: waiting for async I/Os failed
```

如果您查看系統記錄檔、您應該會看到以下幾個錯誤：

```
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
Oct 11 15:52:55 host1 kernel: NFS: nfs4_reclaim_open_state: Lock reclaim
failed!
```

記錄訊息通常是問題的第一個徵象、而非應用程式凍結。通常、在網路中斷期間、您完全看不到任何內容、因為程序和作業系統本身都遭到封鎖、無法嘗試存取 NFS 檔案系統。

網路重新運作後、就會出現錯誤。在上述範例中、一旦重新建立連線、作業系統就會嘗試重新取得鎖定、但時間太晚了。租約已到期、鎖定已移除。這會導致一個錯誤、該錯誤會傳播到 Oracle 層、並導致警示記錄中出現訊息。根據資料庫的版本和組態、您可能會看到這些模式的變化。

總之、 NFSv3 可容忍網路中斷、但 NFSv4 更敏感、並規定了一段定義的租用期。

如果無法接受 30 秒的逾時、該怎麼辦？如果您管理一個動態變化的網路、在其中重新啟動交換器或重新放置纜線、導致網路偶爾中斷、該怎麼辦？您可以選擇延長租用期、但是否需要說明 NFSv4 寬限期。

NFSv4 寬限期

如果 NFSv3 伺服器重新開機、幾乎可以立即為 IO 服務。它並未維持任何形式的用戶端狀態。結果是、 ONTAP 接管作業通常似乎接近即時。當控制器準備好開始提供資料時、就會傳送 ARP 給網路、以表示拓撲的變化。客戶端通常幾乎立即檢測到這種情況，數據恢復流動。

不過 NFSv4 會短暫暫停。這只是 NFSv4 運作方式的一部分。

NFSv4 伺服器需要追蹤租用、鎖定、以及使用何種資料的人員。如果 NFS 伺服器出現問題並重新開機、或停電一段時間、或在維護活動期間重新啟動、則會導致租約 / 鎖定、而其他用戶端資訊也會遺失。伺服器需要先找出哪個用戶端正在使用哪些資料、才能恢復作業。這就是寬限期的開始。

如果您突然關閉 NFSv4 伺服器的電源、當恢復 IO 時、嘗試恢復 IO 的用戶端會收到回應、基本上說：「我遺失了租用 / 鎖定資訊。您是否要重新登錄鎖定？」 這就是寬限期的開始。ONTAP 預設為 45 秒：

```
Cluster01::> nfs server show -vserver vserver1 -fields v4-grace-seconds

vserver   v4-grace-seconds
--------- ----------------
vserver1  45
```

結果是、在重新啟動之後、控制器會暫停 IO 、而所有用戶端都會回收租用和鎖定。寬限期結束後、伺服器將恢復 IO 作業。

租用逾時與寬限期比較

寬限期與租用期間已連線。如上所述、預設的租用逾時為 30 秒、這表示 NFSv4 用戶端必須至少每 30 秒與伺服器簽入一次、否則就會遺失租約、進而導致鎖定。存在寬限期、可讓 NFS 伺服器重建租用 / 鎖定資料、預設為 45 秒。ONTAP 要求寬限期比租用期長 15 秒。如此可確保設計為至少每 30 秒續約的 NFS 用戶端環境、在重新

啟動後能夠與伺服器簽入。45 秒的寬限期可確保所有預期至少每 30 秒續約一次的客戶都有機會續約。

如果無法接受 30 秒的逾時、您可以選擇延長租用期。如果您想要將租用逾時延長至 60 秒、以便承受 60 秒的網路中斷、您必須將寬限期延長至至少 75 秒。ONTAP 要求比租用期高 15 秒。這表示您將會在控制器容錯移轉期間經歷更長的 IO 暫停時間。

這通常不是問題。一般使用者每年只會更新 ONTAP 控制器一或兩次、而且由於硬體故障而造成的非計畫性容錯移轉極少。此外、如果您的網路發生 60 秒網路中斷的可能性、而您需要將租用逾時時間延長至 60 秒、那麼您可能不會反對罕見的儲存系統容錯移轉、導致暫停時間也達 75 秒。您已確認網路暫停超過 60 秒、而且速度較快。

使用 **Oracle** 資料庫進行 **NFS** 快取

如果存在下列任一掛載選項、則會停用主機快取：

```
cio, actimeo=0, noac, forcedirectio
```

這些設定可能會嚴重影響軟體安裝、修補及備份 / 還原作業的速度。在某些情況下、尤其是叢集式應用程式、這些選項是必要的、因為必須在叢集中的所有節點之間提供快取一致性。在其他情況下、客戶誤用這些參數、結果是不必要的效能損害。

許多客戶在安裝或修補應用程式二進位檔時、會暫時移除這些掛載選項。如果使用者在安裝或修補程序過程中確認沒有其他處理程序正在使用目標目錄、則可安全地執行此移除。

Oracle 資料庫的 **NFS** 傳輸大小

根據預設、 ONTAP 將 NFS I/O 大小限制為 64K 。

大多數應用程式和資料庫的隨機 I/O 使用的區塊大小要小得多、遠低於 64K 的最大值。大型區塊 I/O 通常是平行處理的、因此 64K 的最大值也不是取得最大頻寬的限制。

有些工作負載的上限為 64K 、因此會造成限制。特別是、如果資料庫執行的 I/O 數量較少、但容量較大、則備份或還原作業或資料庫完整表格掃描等單執行緒作業、會更快、更有效率地執行。ONTAP 的最佳 I/O 處理大小為 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
Cluster01::*>
```

注意
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。

Oracle 資料庫與 NVFAIL

NVFAIL 是 ONTAP 中的一項功能、可確保災難性容錯移轉案例期間的完整性。

資料庫在儲存設備容錯移轉事件期間容易受損、因為它們會維持大型內部快取。如果災難性事件需要強制 ONTAP 容錯移轉或強制 MetroCluster 切換、無論整體組態的健全狀況為何、都可能會有效捨棄先前確認的變更。儲存陣列的內容會及時向後跳轉、而且資料庫快取的狀態不再反映磁碟上資料的狀態。這種不一致會導致資料毀損。

快取可能發生在應用程式或伺服器層。例如、 Oracle Real Application Cluster （ RAC ）組態、主站台和遠端站台上的伺服器都處於作用中狀態、可在 Oracle SGA 中快取資料。強制切入作業會導致資料遺失、因此資料庫可能會發生毀損、因為儲存在 SGA 中的區塊可能與磁碟上的區塊不符。

較不明顯的快取用途是在作業系統檔案系統層。來自掛載 NFS 檔案系統的區塊可能會快取到作業系統中。或者、以位於主要站台上的 LUN 為基礎的叢集式檔案系統、可以掛載到遠端站台的伺服器上、然後再次快取資料。在這些情況下、 NVRAM 故障或強制接管或強制性的作業系統、可能會導致檔案系統毀損。

ONTAP 使用 NVFAIL 及其相關設定、保護資料庫和作業系統不受此案例影響。

ASM 回收公用程式和 ONTAP 零區塊偵測

啟用即時壓縮時、 ONTAP 可有效移除寫入檔案或 LUN 的歸零區塊。Oracle ASM 回收公用程式（ ASRU ）等公用程式的運作方式是將零寫入未使用的 ASM 範圍。

這可讓 DBA 在資料刪除後回收儲存陣列上的空間。ONTAP 會攔截零並取消分配 LUN 的空間。回收程序非常快速、因為儲存系統中沒有寫入資料。

從資料庫的角度來看、 ASM 磁碟群組包含零、讀取 LUN 的這些區域會產生零串流、但 ONTAP 不會將零儲存在磁碟機上。而是進行簡單的中繼資料變更、在內部將 LUN 的歸零區域標記為任何資料的空白。

由於類似的原因、涉及零位資料的效能測試無效、因為零區塊實際上並未在儲存陣列內以寫入方式處理。

使用 ASRU 時、請確定已安裝所有 Oracle 建議的修補程式。

Oracle 資料庫虛擬化

使用 VMware 、 Oracle OLVM 或 KVM 來虛擬化資料庫、對於選擇虛擬化技術的 NetApp 客戶而言、這是越來越常見的選擇、即使是他們最關鍵的關鍵任務資料庫也一樣。

支援能力

對於 Oracle 虛擬化支援政策、尤其是 VMware 產品、存在許多誤解。聽說 Oracle 完全不支援虛擬化並不罕

見。這個概念不正確、導致錯失從虛擬化中獲益的機會。Oracle Doc ID 249212.1 討論實際需求、客戶很少會將此視為疑慮。

如果虛擬化伺服器發生問題、而 Oracle Support 先前不知道該問題、可能會要求客戶在實體硬體上重現問題。執行產品尖端版本的 Oracle 客戶可能不想使用虛擬化技術、因為可能會發生支援問題、但這種情況對於虛擬化客戶而言、並不是一個真正的世界、因為他們使用的是 Oracle 產品版本。

儲存簡報

考慮將資料庫虛擬化的客戶應根據其業務需求做出儲存決策。雖然這是所有 IT 決策的一般陳述、但資料庫專案尤其重要、因為需求的大小和範圍差異極大。

儲存簡報有三個基本選項：

- Hypervisor 資料存放區上的虛擬化 LUN
- iSCSI LUN 由虛擬機器上的 iSCSI 啟動器管理、而非 Hypervisor
- 由虛擬機器掛載的 NFS 檔案系統（非從 NFS 型資料存放區）
- 直接裝置對應。客戶不喜歡 VMware RDM 、但實體裝置通常與 KVM 和 OLVM 虛擬化類似。

效能

將儲存設備呈現給虛擬化來賓作業系統的方法通常不會影響效能。主機作業系統、虛擬化網路驅動程式和 Hypervisor 資料存放區實作均經過高度最佳化、只要遵循基本最佳實務做法、通常都能在 Hypervisor 和儲存系統之間使用所有可用的 FC 或 IP 網路頻寬。在某些情況下、使用一種儲存呈現方法比使用另一種方法來獲得最佳效能可能會稍微容易一些、但最終結果應該是可比較的。

管理能力

決定如何將儲存設備呈現給虛擬化來賓作業系統的關鍵因素是可管理性。沒有正確或錯誤的方法。最佳方法取決於 IT 營運需求、技能和偏好。

考量因素包括：

- * 透明度。 * VM 管理其檔案系統時、資料庫管理員或系統管理員更容易識別其資料的檔案系統來源。檔案系統和 LUN 的存取方式與實體伺服器的存取方式完全相同。
- * 一致性。 * VM 擁有其檔案系統時、使用或不使用 Hypervisor 層會影響管理能力。資源配置、監控、資料保護等程序也可在整個資產中使用、包括虛擬化和非虛擬化環境。

  另一方面、在另一個 100% 虛擬化的資料中心中、最好還是在整個佔用空間中使用資料存放區型儲存設備、但前提是上述相同的理由（一致性）、也就是能夠使用相同的程序來進行資源配置、保護、監控和資料保護。

- * 穩定性與疑難排解。 * VM 擁有其檔案系統時、由於整個儲存堆疊都存在於 VM 上、因此提供良好、穩定的效能與疑難排解問題變得更簡單。Hypervisor 唯一的角色是傳輸 FC 或 IP 框架。當資料存放區包含在組態中時、它會引入另一組逾時、參數、記錄檔和潛在錯誤、使組態複雜化。
- * 可攜性。 * VM 擁有其檔案系統時、移動 Oracle 環境的程序會變得更簡單。檔案系統可在虛擬化與非虛擬化的來賓作業系統之間輕鬆移動。
- * 廠商鎖定 * 將資料放入資料存放區後、使用不同的 Hypervisor 或將資料從虛擬化環境中移出、將變得非常困難。

- * 啟用 Snapshot ： * 虛擬化環境中的傳統備份程序可能會因為頻寬相對有限而成為問題。例如、四埠 10GbE 主幹可能足以支援許多虛擬化資料庫的日常效能需求、但這類主幹可能不足以使用 RMAN 或其他需要串流完整資料複本的備份產品來執行備份。因此、日益整合的虛擬化環境需要透過儲存快照來執行備份。如此可避免僅為了支援備份時間內的頻寬和 CPU 需求而需要過度建置 Hypervisor 組態。

  使用來賓擁有的檔案系統有時會讓您更輕鬆地利用快照型備份和還原、因為需要保護的儲存物件可以更輕鬆地鎖定目標。然而、越來越多的虛擬化資料保護產品能夠與資料存放區和快照完美整合。在決定如何將儲存設備呈現給虛擬化主機之前、應充分考慮備份策略。

半虛擬化驅動程式

為了達到最佳效能、使用半虛擬化網路驅動程式至關重要。使用資料存放區時、需要半虛擬化 SCSI 驅動程式。半虛擬化的裝置驅動程式可讓來賓更深入地整合至 Hypervisor 、而非模擬的驅動程式、在該驅動程式中、Hypervisor 會花費更多的 CPU 時間來模擬實體硬體的行為。

過度使用 **RAM**

過度使用 RAM 意味著在不同主機上設定的虛擬化 RAM 多於實體硬體上的虛擬化 RAM 。否則可能會造成非預期的效能問題。虛擬化資料庫時、 Oracle SGA 的基礎區塊不得由 Hypervisor 交換至儲存設備。這樣做會導致效能結果極不穩定。

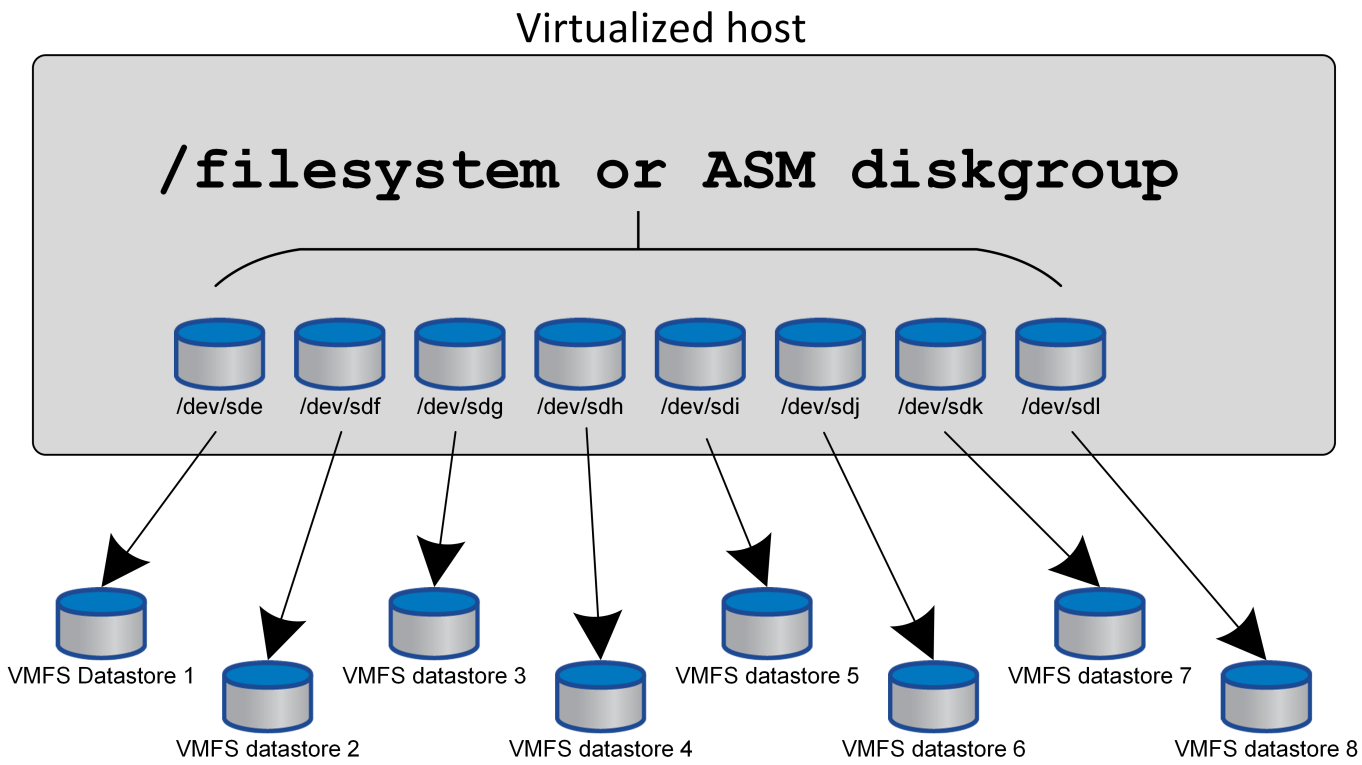
資料存放區等量分割

在使用資料庫搭配資料存放區時、效能方面有一個關鍵因素需要考量、那就是分段。

VMFS 等資料存放區技術可以跨越多個 LUN 、但它們不是等量磁區的裝置。LUN 會串聯。最終結果可能是 LUN 熱點。例如、典型的 Oracle 資料庫可能有 8-LUN ASM 磁碟群組。所有 8 個虛擬化 LUN 均可在 8-LUN VMFS 資料存放區上進行佈建、但無法保證資料所在的 LUN 。產生的組態可能是全部 8 個虛擬化 LUN 、佔用 VMFS 資料存放區內的單一 LUN 。這會成為效能瓶頸。

通常需要分拆。有些 Hypervisor （包括 KVM ）可以使用 LVM 等量分拆來建置資料存放區、如前所述 "請按這裡"。有了 VMware 、架構看起來有點不同。每個虛擬化 LUN 都必須放置在不同的 VMFS 資料存放區上。

例如：



這種方法的主要驅動因素不是 ONTAP 、而是因為單一 VM 或 Hypervisor LUN 可平行處理的作業數量、有固有的限制。單一 ONTAP LUN 通常支援的 IOPS 遠高於主機所能要求的 IOPS 。單一 LUN 效能限制幾乎是主機作業系統的普遍結果。結果是大多數資料庫需要 4 到 8 個 LUN 才能滿足效能需求。

VMware 架構需要仔細規劃其架構、以確保此方法不會遇到資料存放區和 / 或 LUN 路徑最大化問題。此外、每個資料庫都不需要一組唯一的 VMFS 資料存放區。主要需求是確保每個主機都有一組乾淨的 4 到 8 個 IO 路徑、從虛擬化 LUN 到儲存系統本身的後端 LUN 。在極少數情況下、更多的資料存取器可能對真正極致的效能需求有所助益、但 4-8 個 LUN 通常足以滿足 95% 的資料庫需求。單一 ONTAP 磁碟區包含 8 個 LUN 、可透過典型的 OS/ONTAP/ 網路組態、支援多達 250,000 個隨機 Oracle 區塊 IOPS 。

分層

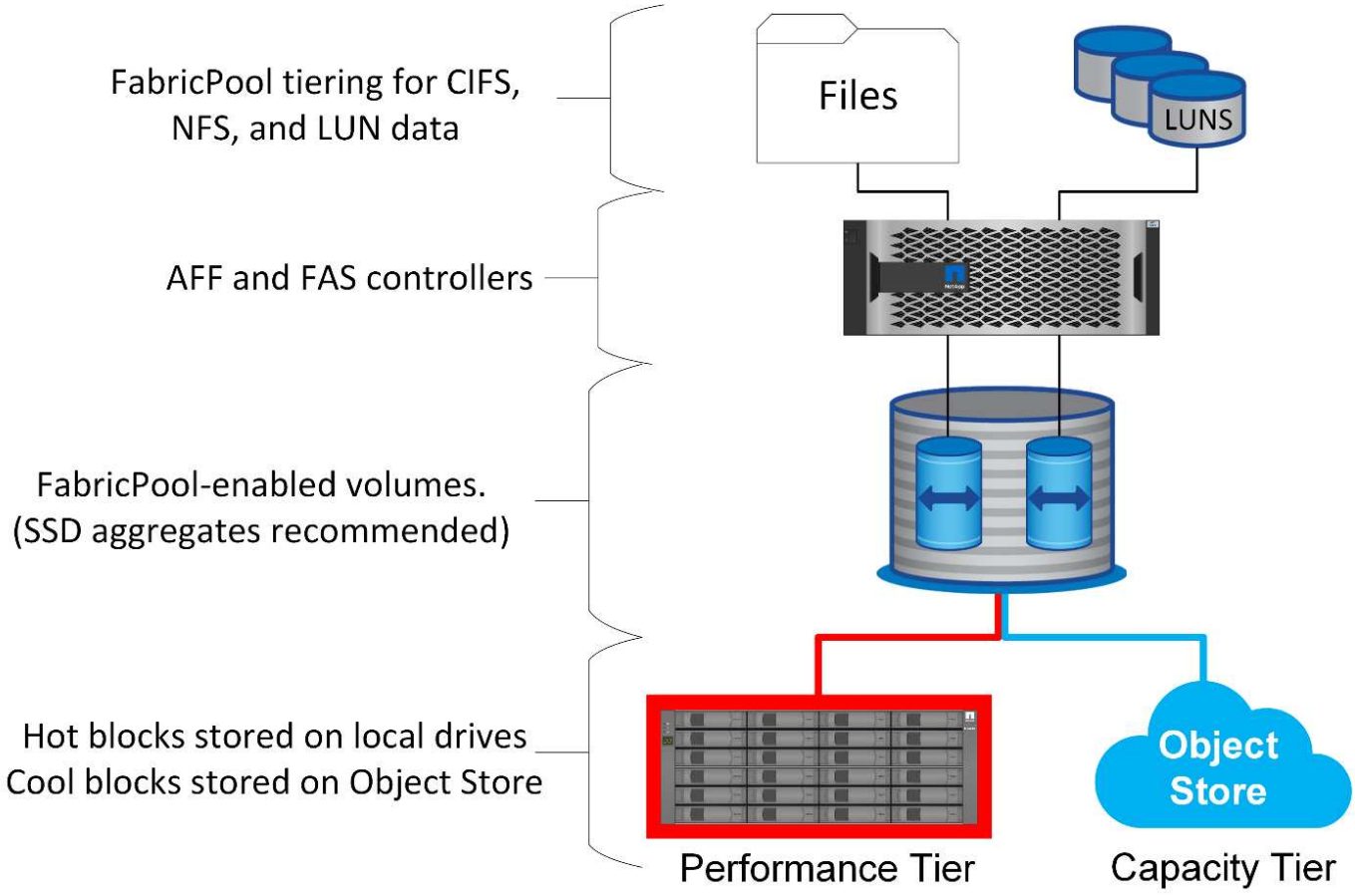
Oracle 資料庫 FabricPool 分層概述

瞭解 FabricPool 分層對 Oracle 和其他資料庫的影響、需要瞭解低階 FabricPool 架構。

架構

FabricPool 是一項分層技術、將區塊歸類為熱區塊或冷卻區塊、並將其置於最適當的儲存層。效能層最常位於 SSD 儲存設備上、並裝載熱資料區塊。容量層位於物件存放區、並裝載 Cool 資料區塊。物件儲存支援包括 NetApp StorageGRID 、 ONTAP S3 、 Microsoft Azure Blob 儲存設備、 Alibaba 雲端物件儲存服務、 IBM 雲端物件儲存設備、 Google 雲端儲存設備和 Amazon AWS S3 。

有多個分層原則可供使用、可控制區塊如何分類為熱或冷區、並可針對每個磁碟區設定原則、並視需要進行變更。只有資料區塊會在效能層和容量層之間移動。定義 LUN 和檔案系統結構的中繼資料一律保留在效能層上。因此、管理作業會集中在 ONTAP 上。檔案和 LUN 與儲存在任何其他 ONTAP 組態上的資料並無不同。NetApp AFF 或 FAS 控制器會套用定義的原則、將資料移至適當的層級。



物件存放區提供者

物件儲存傳輸協定使用簡單的 HTTP 或 HTTPS 要求來儲存大量的資料物件。物件儲存設備的存取必須可靠、因為 ONTAP 的資料存取取決於要求的即時服務。選項包括 Amazon S3 Standard 和 In常用 存取選項、以及 Microsoft Azure Hot 和 Cool Blob Storage 、 IBM Cloud 和 Google Cloud 。不支援 Amazon Glacier 和 Amazon Archive 等歸檔選項、因為擷取資料所需的時間可能超過主機作業系統和應用程式的容許值。

NetApp StorageGRID 也受到支援、是最佳的企業級解決方案。它是高效能、可擴充且高度安全的物件儲存系統、可為 FabricPool 資料及其他物件儲存應用程式提供地理上的備援、這些應用程式越來越可能成為企業應用程式環境的一部分。

StorageGRID 也能避免許多公有雲供應商為了從服務讀取資料而收取的出口費用、進而降低成本。

資料與中繼資料

請注意、此處的「資料」一詞適用於實際的資料區塊、而非中繼資料。只有資料區塊是階層式的、而中繼資料則保留在效能層級中。此外、只有讀取實際的資料區塊、才能影響區塊的「熱」或「冷」狀態。只要讀取檔案的名稱、時間戳記或所有權中繼資料、就不會影響基礎資料區塊的位置。

備份

雖然 FabricPool 可以大幅減少儲存佔用空間、但它本身並不是備份解決方案。NetApp WAFL 中繼資料一律保持在效能層級。如果災難性災難破壞效能層、則無法使用容量層上的資料來建立新環境、因為該環境不包含 WAFL 中繼資料。

不過、 FabricPool 可以成為備份策略的一部分。例如、 FabricPool 可以使用 NetApp SnapMirror 複寫技術進行

設定。每一半的鏡像都可以有自己與物件儲存目標的連線。結果是兩個資料的複本。主要複本包含效能層上的區塊、以及容量層中的相關區塊、而複本則是第二組效能和容量區塊。

分層原則

Oracle 資料庫 FabricPool 分層原則

ONTAP 提供四項原則、可控制效能層上的 Oracle 資料如何成為移轉至容量層的候選對象。

僅限 Snapshot

。 `snapshot-only` `tiering-policy` 僅適用於未與作用中檔案系統共用的區塊。它基本上會導致資料庫備份分層。在建立快照之後、區塊會成為分層的候選項目、然後區塊會被覆寫、導致區塊只存在於快照中。A 之前的延遲 `snapshot-only` 區塊視為冷區、由控制 `tiering-minimum-cooling-days` 音量設定。ONTAP 9.8 的範圍為 2 至 183 天。

許多資料集的變更率都很低、因此這項原則可節省的成本極低。例如、在 ONTAP 上觀察到的典型資料庫每週變更率低於 5% 。資料庫歸檔記錄檔可能佔用大量空間、但通常會繼續存在於作用中的檔案系統中、因此不會成為根據此原則分層的候選項目。

自動

。 `auto` 分層原則可將分層延伸至快照專用區塊、以及作用中檔案系統內的區塊。區塊冷卻前的延遲由控制 `tiering-minimum-cooling-days` 音量設定。ONTAP 9.8 的範圍為 2 至 183 天。

此方法可啟用無法與搭配使用的分層選項 `snapshot-only` 原則。例如、資料保護原則可能需要保留 90 天的特定記錄檔。如果將冷卻期設定為 3 天、則任何超過 3 天的記錄檔都會從效能層中分層移出。此動作可釋放效能層級上的大量空間、同時仍可讓您檢視及管理完整的 90 天資料。

無

。 `none` 分層原則可防止任何額外的區塊從儲存層分層、但容量層中的任何資料仍會保留在容量層中、直到讀取為止。如果接著讀取區塊、則會將其拉回並放置在效能層上。

使用的主要原因 `none` 分層原則旨在防止區塊分層、但隨著時間推移而變更原則可能會很有用。例如、假設某個特定資料集已廣泛分層至容量層、但卻產生對完整效能功能的非預期需求。可變更原則以防止任何額外的分層、並確認隨 IO 增加而讀取的任何區塊仍保留在效能層中。

全部

。 `all` 分層原則取代了 `backup` 原則自 ONTAP 9.6 起。。 `backup` 原則僅套用至資料保護磁碟區、意指 SnapMirror 或 NetApp SnapVault 目的地。。 `all` 原則的功能相同、但不限於資料保護磁碟區。

有了這項原則、就能立即將區塊視為酷炫、並立即分層至容量層。

此原則特別適用於長期備份。它也可以用作階層式儲存管理（ HSM ）的形式。過去、 HSM 通常用於將檔案的資料區塊分層至磁帶、同時讓檔案本身在檔案系統上保持可見。具有的 FabricPool Volume `all` 原則可讓您將檔案儲存在可見且可管理的環境中、但幾乎無需佔用本機儲存層的空間。

Oracle 資料庫和 **FabricPool** 擷取原則

分層原則可控制哪些 Oracle 資料庫區塊從效能層分層分層到容量層。擷取原則可控制已階層的區塊讀取時所發生的情況。

預設

所有 FabricPool 磁碟區的初始設定為 `default`這表示行為是由「雲端擷取原則」所控制。「確切的行為取決於所使用的分層原則。

- `auto`–僅擷取隨機讀取的資料
- `snapshot-only`–擷取所有依序或隨機讀取的資料
- `none`–擷取所有依序或隨機讀取的資料
- `all`–請勿從容量層擷取資料

讀取中

設定 `cloud-retrieval-policy` 對讀取會覆寫預設行為、因此讀取任何階層資料會導致資料傳回效能層。

例如、在的下、一個 Volume 可能已被輕度使用了很長時間 `auto` 分層原則和大多數區塊現在都是分層的。

如果業務發生非預期的變更、需要重複掃描部分資料以準備特定報告、則可能需要變更 `cloud-retrieval-policy` 至 `on-read` 確保讀取的所有資料都會傳回效能層、包括依序和隨機讀取資料。這將改善連續 I/O 相對於磁碟區的效能。

推廣

升級原則的行為取決於分層原則。如果分層原則是 `auto`、然後設定 `cloud-retrieval-policy `to `promote` 在下一次分層掃描時、從容量層移回所有區塊。

如果分層原則是 `snapshot-only`，則只會傳回與作用中檔案系統相關聯的區塊。通常這不會有任何影響、因為唯一的區塊會在底下分層 `snapshot-only` 原則將是與快照完全相關的區塊。作用中檔案系統中不會有階層式區塊。

然而、如果磁碟區上的資料是由 Volume SnapRestore 或快照的檔案複製作業還原、則目前使用中檔案系統可能需要一些因為只與快照相關而分層的區塊。您可能需要暫時變更 `cloud-retrieval-policy` 原則目標 `promote` 快速擷取所有本機需要的區塊。

永不

請勿從容量層擷取區塊。

分層策略

Oracle 資料庫完整檔案 **FabricPool** 分層

雖然 FabricPool 分層是在區塊層級運作、但在某些情況下、它可用於提供檔案層級的分層。

許多應用程式資料集都是按日期組織、而且隨著時間的演進、存取這些資料的可能性通常會越來越小。例如、某家銀行可能會有包含五年客戶對帳單的 PDF 檔案儲存庫、但只有最近幾個月是有效的。FabricPool 可用於將較舊的資料檔案重新放置到容量層。14 天的冷卻期可確保最近 14 天的 PDF 檔案仍保留在效能層級上。此外、至少每 14 天讀取一次的檔案仍會很熱、因此會保留在效能層級上。

原則

若要實作檔案型分層方法、您必須有已寫入且未隨後修改的檔案。。 `tiering-minimum-cooling-days` 原則應該設定得足夠高、以便您可能需要的檔案保留在效能層。例如、需要最近 60 天資料的資料集、且需要設定最佳效能 `tiering-minimum-cooling-days` 期間為 60 。也可以根據檔案存取模式來達成類似的結果。例如、如果需要最近 90 天的資料、而應用程式正在存取該 90 天的資料範圍、則資料會保留在效能層。設定 `tiering-minimum-cooling-days` 期間為 2 、資料變少後、系統會提示您分層。

。 `auto` 由於只有 `auto` 原則會影響作用中檔案系統中的區塊。

任何類型的資料存取都會重設熱圖資料。病毒掃描、索引甚至是讀取來源檔案的備份活動、都會因為必要而防止分層 `tiering-minimum-cooling-days` 從未達到臨界值。

Oracle 部分檔案 **FabricPool** 分層

由於 FabricPool 是在區塊層級運作、因此可能變更的檔案可以部分分層化為物件儲存、而部分保留在效能層級上。

這在資料庫中很常見。已知包含非作用中區塊的資料庫也可用於 FabricPool 分層。例如、供應鏈管理資料庫可能包含歷史資訊、這些資訊必須在必要時提供、但在正常作業期間無法存取。FabricPool 可用於選擇性地重新定位非使用中的區塊。

例如、在 FabricPool 磁碟區上執行的資料檔案 `tiering-minimum-cooling-days` 90 天的期間會保留過去 90 天內在效能層級上存取的任何區塊。然而、 90 天內未存取的任何項目都會重新移至容量層。在其他情況下、正常的應用程式活動會將正確的區塊保留在正確的層級上。例如、如果資料庫通常用於定期處理過去 60 天的資料、則會降低許多 `tiering-minimum-cooling-days` 可以設定期間、因為應用程式的自然活動可確保區塊不會提早重新定位。

。 `auto` 原則應與資料庫一起使用。許多資料庫都有定期活動、例如季末流程或重新編製索引作業。如果這些作業的期間大於 `tiering-minimum-cooling-days` 可能會發生效能問題。例如、如果季度末處理需要 1TB 的資料、而這些資料原本沒有受到影響、則該資料現在可能會出現在容量層。從容量層讀取的速度通常極快、可能不會造成效能問題、但實際結果將取決於物件儲存區組態。

原則

。 `tiering-minimum-cooling-days` 原則的設定應足夠高、以保留效能層上可能需要的檔案。例如、如果資料庫需要最新 60 天的資料、而且效能最佳、則需要設定 `tiering-minimum-cooling-days` 期間為 60 天。也可以根據檔案的存取模式來達成類似的結果。例如、如果需要最近 90 天的資料、而應用程式正在存取該 90 天的資料範圍、則資料會保留在效能層。設定 `tiering-minimum-cooling-days` 在資料變得不活躍之後、將會立即將資料分級至 2 天。

。 `auto` 由於只有 `auto` 原則會影響作用中檔案系統中的區塊。

任何類型的資料存取都會重設熱圖資料。因此、資料庫完整表格掃描、甚至是讀取來源檔案的備份活動、都會因為需要而防止分層 `tiering-minimum-cooling-days` 從未達到臨界值。

Oracle 資料庫歸檔記錄分層

FabricPool 最重要的用途可能是提高已知冷資料的效率、例如資料庫交易記錄。

大部分的關聯式資料庫都是以交易記錄歸檔模式運作、以提供時間點還原。對資料庫所做的變更會記錄交易記錄中的變更、並保留交易記錄而不會被覆寫。結果可能需要保留大量歸檔的交易記錄檔。許多其他應用程式工作流程也有類似的例子、這些工作流程會產生必須保留的資料、但很難存取。

FabricPool 透過提供整合式分層的單一解決方案來解決這些問題。檔案會儲存並保留在其一般位置、但實際上不會佔用主要陣列上的任何空間。

原則

使用 `tiering-minimum-cooling-days` 幾天的原則會導致在效能層上保留最近建立的檔案（近期最可能需要的檔案）中的區塊。之後、舊檔案的資料區塊會移至容量層。

。 `auto` 無論主要檔案系統中的記錄是否已刪除或繼續存在、都會在達到冷卻臨界值時強制執行提示分層。將所有可能需要的記錄儲存在作用中檔案系統的單一位置、也能簡化管理。沒有理由搜尋快照以找出需要還原的檔案。

某些應用程式（例如 Microsoft SQL Server ）會在備份作業期間截斷交易記錄檔、使記錄不再位於作用中的檔案系統中。使用可節省容量 `snapshot-only` 分層原則、但 `auto` 原則對記錄資料並不實用、因為作用中檔案系統中應該很少會冷卻記錄資料。

Oracle 與 FabricPool 快照分層

FabricPool 的初始版本以備份使用案例為目標。唯一可以分層的區塊類型是不再與作用中檔案系統中的資料相關聯的區塊。因此、只能將快照資料區塊移至容量層。當您需要確保效能不受影響時、這仍然是最安全的分層選項之一。

原則 - 本機快照

有兩個選項可將非作用中的快照區塊分層到容量層。首先 `snapshot-only` 原則僅針對快照區塊。儘管如此 `auto` 原則包括 `snapshot-only` 區塊、也會將區塊分層、從作用中的檔案系統中移出。這可能不理想。

。 `tiering-minimum-cooling-days` 值應設為時間週期、以便在效能層上提供還原期間所需的資料。例如、重要正式作業資料庫的大多數還原案例、都會在過去幾天的某個時間加入還原點。設定 A `tiering-minimum-cooling-days` 值 3 可確保檔案的任何還原都會產生可立即提供最大效能的檔案。作用中檔案中的所有區塊仍會顯示在快速儲存設備上、而無需從容量層恢復。

原則 - 複寫的快照

使用 SnapMirror 或 SnapVault 複寫的僅用於恢復的快照通常應使用 FabricPool `all` 原則。使用此原則、會複寫中繼資料、但所有資料區塊都會立即傳送至容量層、以獲得最大效能。大部分的恢復程序都涉及循序 I/O 、這是固有的效率。應該評估物件存放區目的地的恢復時間、但在設計完善的架構中、此恢復程序不需要比從本機資料恢復慢很多。

如果複製的資料也要用於複製、則會使用 `auto` 原則比較適當、請使用 `tiering-minimum-cooling-days` 包含預期在複製環境中經常使用的資料的值。例如、資料庫的作用中工作集可能包含前三天讀取或寫入的資料、但也可能包含另外 6 個月的歷史資料。如果是、則是 `auto` SnapMirror 目的地的原則可讓工作集在效能層上使用。

傳統應用程式備份包括 Oracle Recovery Manager 等產品、可在原始資料庫之外建立檔案型備份。

```
 `tiering-minimum-cooling-days` policy of a few days preserves the most
recent backups, and therefore the backups most likely to be required for
an urgent recovery situation, on the performance tier. The data blocks of
the older files are then moved to the capacity tier.
```

- `auto`
原則是最適合備份資料的原則。如此可確保在達到冷卻臨界值時、無論主要檔案系統中的檔案是否已刪除或繼續存在、都能立即分層。將所有可能需要的檔案儲存在作用中檔案系統的單一位置、也能簡化管理。沒有理由搜尋快照以找出需要還原的檔案。

- `snapshot-only` 原則可以生效、但該原則僅適用於不在作用中檔案系統中的區塊。因此、必須先刪除 NFS 或 SMB 共用上的檔案、才能分層化資料。

使用 LUN 組態時、此原則的效率會更低、因為從 LUN 刪除檔案只會從檔案系統中繼資料中移除檔案參照。LUN 上的實際區塊會一直保留到位、直到被覆寫為止。這種情況可能會造成從刪除檔案到覆寫區塊並成為分層候選項目之間的長時間延遲。移動有一些好處 `snapshot-only` 區塊到容量層、但整體而言、備份資料的 FabricPool 管理最適合搭配使用 `auto` 原則。

> 此方法可協助使用者更有效率地管理備份所需的空間、但 FabricPool 本身並不是備份技術。將備份檔案分層至物件存放區可簡化管理、因為檔案仍可在原始儲存系統上看到、但物件存放區目的地中的資料區塊則取決於原始儲存系統。如果來源磁碟區遺失、物件儲存區資料將無法再使用。

Oracle 資料庫和物件儲存區存取中斷

使用 FabricPool 分層資料集會導致主要儲存陣列與物件存放區層之間的相依性。有許多物件儲存選項可提供不同層級的可用度。請務必瞭解主要儲存陣列與物件儲存層之間可能中斷連線的影響。

如果發給 ONTAP 的 I/O 需要容量層的資料、而 ONTAP 無法到達容量層來擷取區塊、則 I/O 最終會逾時。此逾時的影響取決於所使用的傳輸協定。在 NFS 環境中、 ONTAP 會根據傳輸協定回應 EJUKEBOX 或 EDELAY 回應。有些較舊的作業系統可能會將此視為錯誤、但 Oracle Direct NFS 用戶端目前的作業系統和修補程式層級會將此視為可重試的錯誤、並繼續等待 I/O 完成。

較短的逾時時間適用於 SAN 環境。如果需要物件存放區環境中的區塊、而且無法連線兩分鐘、則會將讀取錯誤傳回主機。ONTAP 磁碟區和 LUN 保持連線、但主機作業系統可能會將檔案系統標示為處於錯誤狀態。

物件儲存連線問題 `snapshot-only` 由於只有備份資料是階層式的、因此原則並不令人擔心。通訊問題會拖慢資料恢復速度、但不會影響使用中的資料。。 `auto` 和 `all` 原則允許從作用中 LUN 分層處理冷資料、這表示物件儲存區資料擷取期間發生的錯誤可能會影響資料庫可用度。採用這些原則的 SAN 部署只能搭配專為高可用度所設計的企業級物件儲存設備和網路連線使用。NetApp StorageGRID 是絕佳的選擇。

Oracle 資料保護

使用 **ONTAP** 保護 **Oracle** 資料

NetApp 知道資料庫中有最重要的關鍵任務資料。

企業無法在沒有資料存取權的情況下運作、有時資料會定義業務。此資料必須受到保護；然而、資料保護不只是確保可用的備份、還要快速可靠地執行備份、而且還要安全地儲存。

資料保護的另一面是資料恢復。當資料無法存取時、企業就會受到影響、而且可能無法運作、直到資料還原為止。此程序必須快速且可靠。最後、大多數資料庫都必須防範災難、這表示必須維護資料庫的複本。複本必須是最新的。複本也必須快速且簡單、才能使其成為完全運作的資料庫。

本文件取代先前發佈的技術報告 _TR-4591 ： Oracle 資料保護：備份、還原及複寫。 _

規劃

正確的企業資料保護架構、取決於資料保留、可恢復性、以及各種事件中斷的容忍度等業務需求。

例如、請考慮範圍內的應用程式、資料庫和重要資料集數量。為單一資料集建立備份策略、以確保符合典型 SLA 的要求相當簡單、因為沒有太多物件需要管理。隨著資料集數量增加、監控作業變得更複雜、系統管理員可能不得不花費更多時間來解決備份故障。當環境達到雲端與服務供應商的規模時、需要完全不同的方法。

資料集大小也會影響策略。例如、由於資料集太小、因此有許多選項可用於 100GB 資料庫的備份與還原。只要使用傳統工具從備份媒體複製資料、通常就能提供足夠的恢復 RTO 。100TB 資料庫通常需要完全不同的策略、除非 RTO 允許多天中斷、否則傳統的複製備份與還原程序可能是可以接受的。

最後、除了備份與還原程序本身之外、還有其他因素。例如、是否有支援關鍵生產活動的資料庫、使恢復成為僅由熟練的 DBA 執行的罕見事件？或者、資料庫是否是大型開發環境的一部分、而在大型開發環境中、恢復是經常發生的事、並由一群通才的 IT 團隊負責管理？

Oracle 資料庫 **RTO** 、 **RPO** 和 **SLA** 規劃

ONTAP 可讓您輕鬆根據業務需求量身打造 Oracle 資料庫資料保護策略。

這些需求包括恢復速度、最大允許資料遺失量、以及備份保留需求等因素。資料保護計畫也必須考量資料保留與還原的各種法規要求。最後、必須考量不同的資料恢復情境、從使用者或應用程式錯誤所造成的典型和可預見的恢復、到包括站點完全遺失的災難恢復情境。

資料保護與還原原則的細微變更、可能會對儲存、備份與還原的整體架構造成重大影響。在開始設計工作之前、務必先定義並記錄標準、以免資料保護架構複雜化。不必要的功能或保護層級會導致不必要的成本和管理成本、而最初忽略的需求可能導致專案方向錯誤、或是需要最後一分鐘的設計變更。

恢復時間目標

恢復時間目標（ RTO ）定義了恢復服務所允許的最長時間。例如、人力資源資料庫的 RTO 可能為 24 小時、因為雖然在工作天內無法存取這些資料、但企業仍可繼續營運。相反地、支援銀行總分類帳的資料庫、其 RTO 會以分鐘甚至秒為單位來衡量。RTO 為零是不可能的、因為必須有辦法區分實際服務中斷和例行事件、例如遺失的網路封包。然而、典型的要求是接近零的 RTO 。

恢復點目標

恢復點目標（ RPO ）定義了最大可容忍的資料遺失量。在許多情況下、 RPO 完全取決於快照或 SnapMirror 更

新的頻率。

在某些情況下、 RPO 可能會變得更具侵略性、因此可選擇性地更頻繁地保護某些資料。在資料庫內容中、 RPO 通常是在特定情況下可能遺失多少記錄資料的問題。在資料庫因產品錯誤或使用者錯誤而受損的典型還原案例中、 RPO 應為零、表示不應有資料遺失。恢復程序包括還原較早的資料庫檔案複本、然後重新播放記錄檔、將資料庫狀態提升至所需的時間點。此作業所需的記錄檔應已在原始位置中。

在不尋常的情況下、記錄資料可能會遺失。例如、意外或惡意 `rm -rf *` 資料庫檔案可能會導致刪除所有資料。唯一的選擇是從備份還原、包括記錄檔、有些資料必然會遺失。在傳統備份環境中改善 RPO 的唯一選項是執行記錄資料的重複備份。然而、由於資料持續移動、而且難以將備份系統維持為持續運作的服務、因此這項功能也有其侷限性。進階儲存系統的優點之一、就是能夠保護資料、避免意外或惡意損壞檔案、進而在不移動資料的情況下提供更好的 RPO 。

災難恢復

災難恢復包括在發生實體災難時恢復服務所需的 IT 架構、原則和程序。這可能包括洪水、火災或惡意或疏忽意圖的人。

災難恢復不只是一套恢復程序。這是識別各種風險、定義資料恢復和服務持續性需求、以及提供適當架構及相關程序的完整程序。

在建立資料保護需求時、必須區分典型的 RPO 和 RTO 需求、以及災難恢復所需的 RPO 和 RTO 需求。有些應用程式環境需要零 RPO 和近乎零的 RTO 、才能因應資料遺失的情況、從相對正常的使用者錯誤到破壞資料中心的火災。然而、這些高層級的保護措施會產生成本和管理上的後果。

一般而言、非災難性資料恢復需求應嚴格、原因有兩個。首先、應用程式錯誤和使用者錯誤會導致資料受損、幾乎是不可避免的。其次、只要儲存系統未遭銷毀、就不難設計出可提供零 RPO 和低 RTO 的備份策略。沒有理由不解決容易補救的重大風險、這就是為何用於本機恢復的 RPO 和 RTO 目標應該積極主動的原因。

災難恢復 RTO 和 RPO 需求因災難可能性及相關資料遺失或業務中斷所造成的後果而異。RPO 和 RTO 需求應以實際業務需求為基礎、而非一般原則。他們必須考慮多種邏輯和實體災難案例。

邏輯災難

邏輯災難包括使用者、應用程式或作業系統錯誤所造成的資料毀損、以及軟體故障。邏輯災難也可能包括由外部人士利用病毒或蠕蟲或利用應用程式弱點發動的惡意攻擊。在這些情況下、實體基礎架構並未受損、但基礎資料已不再有效。

越來越常見的邏輯災難類型稱為勒索軟體、其中攻擊模式是用來加密資料。加密不會損壞資料、但會在付款給第三方之前無法使用。越來越多的企業被勒索軟體攻擊的目標鎖定。針對此威脅、 NetApp 提供防竄改快照、即使儲存管理員也無法在設定的到期日之前變更受保護的資料。

實體災難

實體災難包括基礎架構元件故障、其超過備援功能、導致資料遺失或服務延長中斷。例如、 RAID 保護可提供磁碟機備援、而使用 HBA 則可提供 FC 連接埠和 FC 纜線備援。這類元件的硬體故障是可預見的、不會影響可用度。

在企業環境中、通常可以使用備援元件來保護整個站台的基礎架構、直到唯一可預見的實體災難案例完全遺失站台為止。災難恢復規劃則取決於站台對站台的複寫。

在理想的環境中、所有資料都會在地理上分散的站台之間同步複寫。這類複寫不一定可行、甚至可能、原因有幾個：

- 同步複寫不可避免地會增加寫入延遲、因為所有變更都必須複寫到兩個位置、應用程式 / 資料庫才能繼續處理。因此產生的效能影響有時是不可接受的、無法使用同步鏡射。
- 100% SSD 儲存設備的採用率增加、意味著更有可能注意到額外的寫入延遲、因為效能期望包括數十萬 IOPS 和低於毫秒的延遲。若要充分發揮 100% SSD 的效益、可能需要重新規劃災難恢復策略。
- 資料集會繼續以位元組為單位增加、因此必須確保足夠的頻寬來維持同步複寫、因此會產生挑戰。
- 資料集的複雜度也隨之增加、管理大規模同步複寫也會帶來挑戰。
- 雲端型策略通常需要較長的複寫距離和延遲、進一步排除同步鏡射的使用。

NetApp 提供的解決方案包括同步複寫、可滿足最嚴苛的資料恢復需求、並可提供更優異的效能與靈活度的非同步解決方案。此外、 NetApp 技術也能與許多第三方複寫解決方案無縫整合、例如 Oracle DataGuard

保留時間

資料保護策略的最後一個層面是資料保留時間、這可能會大幅改變。

- 一般要求是在主要站台上進行 14 天夜間備份、以及儲存在次要站台上的 90 天備份。
- 許多客戶會建立獨立的季度歸檔、儲存在不同的媒體上。
- 持續更新的資料庫可能不需要歷史資料、而備份只需保留幾天。
- 法規要求可能需要在 365 天內恢復任何任意交易的點。

ONTAP 提供 Oracle 資料庫可用度

ONTAP 旨在提供最大的 Oracle 資料庫可用度。ONTAP 高可用度功能的完整說明不在本文件的範圍之內。然而、與資料保護一樣、在設計資料庫基礎架構時、對此功能的基本瞭解非常重要。

HA 配對

高可用度的基本單位是 HA 配對。每對都包含備援連結、可支援將資料複寫到 NVRAM 。NVRAM 不是寫入快取。控制器內的 RAM 會做為寫入快取。NVRAM 的用途是暫時記錄資料、以防止發生非預期的系統故障。在這方面、它與資料庫重做記錄類似。

NVRAM 和資料庫重做記錄都可用來快速儲存資料、讓資料的變更能夠儘快提交。磁碟機（或資料檔案）上的持續資料更新直到稍後在 ONTAP 和大多數資料庫平台上稱為檢查點的程序期間才會進行。正常作業期間不會讀取 NVRAM 資料或資料庫重做記錄。

如果控制器突然故障、可能會有擱置中的變更、這些變更儲存在 NVRAM 中、但尚未寫入磁碟機。合作夥伴控制器會偵測故障、控制磁碟機、並套用儲存在 NVRAM 中的必要變更。

接管與恢復

接管與恢復是指在 HA 配對中的節點之間轉移儲存資源責任的程序。接管和恢復有兩個層面：

- 管理可存取磁碟機的網路連線能力
- 管理磁碟機本身

支援 CIFS 和 NFS 流量的網路介面、都是設定在主位置和容錯移轉位置。接管包括將網路介面移至與原始位置位於同一子網路的實體介面上的暫存主目錄。贈品包括將網路介面移回其原始位置。您可以視需要調整確切行為。

支援 SAN 區塊傳輸協定（例如 iSCSI 和 FC ）的網路介面不會在接管和恢復期間重新定位。而是應使用包含完整 HA 配對的路徑來佈建 LUN 、以產生主要路徑和次要路徑。

您也可以設定其他控制器的路徑、以支援在較大叢集中的節點之間重新放置資料、但這並不屬於 HA 程序的一部分。

接管與恢復的第二個層面是磁碟擁有權的轉移。確切的程序取決於多種因素、包括接管 / 恢復的原因、以及發出的命令列選項。目標是盡可能有效率地執行作業。雖然整體程序可能需要幾分鐘的時間、但磁碟機的實際擁有權從節點移轉至節點的時間通常只需幾秒鐘。

接管時間

主機 I/O 在接管和恢復作業期間會短暫暫停 I/O 、但在正確設定的環境中不應發生應用程式中斷。I/O 延遲的實際轉換程序通常是以秒為單位來測量、但主機可能需要額外的時間來識別資料路徑的變更並重新提交 I/O 作業。

中斷的性質取決於傳輸協定：

- 支援 NFS 和 CIFS 流量的網路介面會在移轉至新實體位置後、向網路發出位址解析傳輸協定（ ARP ）要求。這會導致網路交換器更新其媒體存取控制（ MAC ）位址表、並繼續處理 I/O在計畫性接管和恢復的情況下、中斷通常以秒為單位來衡量、在許多情況下都無法偵測到。有些網路可能會較慢、無法完全辨識網路路徑的變更、有些作業系統可能會在很短的時間內排入大量 I/O 、因此必須重新嘗試。這可能會延長恢復 I/O 所需的時間
- 支援 SAN 通訊協定的網路介面不會轉換到新位置。主機作業系統必須變更使用中的路徑。主機觀察到 I/O 暫停的情形取決於多種因素。從儲存系統的角度來看、無法提供 I/O 的時間只有幾秒鐘。不過、不同的主機作業系統可能需要額外的時間、才能讓 I/O 逾時、再重試。較新的作業系統更能更快辨識路徑變更、但較舊的作業系統通常需要 30 秒才能辨識變更。

下表顯示儲存系統無法將資料提供給應用程式環境的預期接管時間。在任何應用程式環境中都不應發生任何錯誤、而是在 IO 處理過程中、接管應該會顯示為短暫的暫停。

	NFS	AFF	ASA
計畫性接管	15 秒	6-10 秒	2-3 秒
非計畫性接管	30 秒	6-10 秒	2-3 秒

Checksum 與 Oracle 資料庫完整性

ONTAP 及其支援的通訊協定包含多項保護 Oracle 資料庫完整性的功能、包括靜態資料和透過網路傳輸的資料。

ONTAP 內的邏輯資料保護包含三項關鍵需求：

- 資料必須受到保護、以免資料毀損。

- 資料必須受到保護、避免磁碟機故障。
- 資料變更必須受到保護、以免遺失。

以下各節將討論這三項需求。

網路毀損： **Checksum**

最基本的資料保護層級是 Checksum 、這是儲存在資料旁的特殊錯誤偵測程式碼。在網路傳輸期間、會使用 Checksum 和（在某些情況下）多個 Checksum 來偵測資料毀損。

例如、 FC 框架包含稱為循環備援檢查（ CRC ）的校驗和形式、以確保有效負載不會在傳輸過程中毀損。傳輸器會同時傳送資料和資料的 CRC 。FC 訊框的接收器會重新計算接收資料的 CRC 、以確保其符合傳輸的 CRC 。如果新計算的 CRC 與附加至框架的 CRC 不相符、則資料會毀損、 FC 框架會遭到捨棄或拒絕。iSCSI I/O 作業包括 TCP/IP 層和乙太網路層的校驗和、此外、為了提供額外的保護、也可在 SCSI 層提供選用的 CRC 保護。TCP 層或 IP 層偵測到線路上的任何位元毀損、導致封包重新傳輸。與 FC 一樣、 SCSI CRC 中的錯誤也會導致作業遭到捨棄或拒絕。

磁碟機毀損： **Checksum**

Checksum 也可用來驗證儲存在磁碟機上的資料完整性。寫入磁碟機的資料區塊會以 Checksum 功能儲存、產生與原始資料相關的不可預測數字。從磁碟機讀取資料時、會重新計算總和檢查碼、並與儲存的總和檢查碼進行比較。如果不相符、則資料已毀損、必須由 RAID 層還原。

資料毀損：寫入遺失

最難偵測的毀損類型之一是遺失或錯誤寫入。確認寫入後、必須將其寫入正確位置的媒體。就地資料毀損使用儲存在資料中的簡單檢查碼、相當容易偵測。但是、如果寫入資料只是遺失、則先前版本的資料可能仍然存在、而且總和檢查碼是正確的。如果寫入放置在錯誤的實體位置、則相關的 Checksum 將再次對儲存的資料有效、即使寫入已銷毀其他資料。

這項挑戰的解決方案如下：

- 寫入作業必須包含中繼資料、以指出寫入的預期位置。
- 寫入作業必須包含某種版本識別碼。

ONTAP 寫入區塊時、會包含區塊所屬的資料。如果後續讀取識別出某個區塊、但中繼資料指出該區塊在位置 456 找到時屬於位置 123 、則表示該寫入已放錯位置。

更難偵測完全遺失的寫入。這項說明非常複雜、但基本上 ONTAP 是以寫入作業導致磁碟機上兩個不同位置的更新方式來儲存中繼資料。如果寫入遺失、後續的資料讀取和相關中繼資料會顯示兩個不同的版本識別。這表示磁碟機未完成寫入。

遺失或放錯位置的寫入毀損極少發生、但隨著磁碟機持續成長、資料集也逐漸擴充至 EB 規模、風險也會增加。任何支援資料庫工作負載的儲存系統都應包含遺失寫入偵測。

磁碟機故障： **RAID** 、 **RAID DP** 和 **RAID-TEC**

如果發現磁碟機上的資料區塊毀損、或整個磁碟機故障且完全無法使用、則必須重新建立資料。這是在 ONTAP 中使用同位元磁碟機來完成的。資料會在多個資料磁碟機之間進行等量分割、然後產生同位元檢查資料。這會與原始資料分開儲存。

ONTAP 最初使用 RAID 4 、每組資料磁碟機使用單一同位元檢查磁碟機。結果是群組中的任何一個磁碟機都可

能發生故障、而不會導致資料遺失。如果同位元磁碟機故障、則沒有資料受損、也可以建構新的同位元磁碟機。如果單一資料磁碟機故障、其餘磁碟機可與同位元磁碟機搭配使用、以重新產生遺失的資料。

當磁碟機很小時、同時發生兩個磁碟機故障的統計機率可忽略不計。隨著磁碟機容量的增加、磁碟機故障後重建資料所需的時間也隨之增加。這增加了第二個磁碟機故障會導致資料遺失的時間。此外、重建程序也會在正常運作的磁碟機上建立許多額外的 I/O 。隨著磁碟機老化、導致第二個磁碟機故障的額外負載風險也會增加。最後、即使持續使用 RAID 4 、資料遺失的風險並未增加、資料遺失的後果也會更加嚴重。在 RAID 群組故障時遺失的資料越多、還原資料所需的時間就越長、業務中斷也就越長。

這些問題導致 NetApp 開發 NetApp RAID DP 技術、這是 RAID 6 的變體。此解決方案包含兩個同位元檢查磁碟機、表示 RAID 群組中的任何兩個磁碟機都可能發生故障、而不會造成資料遺失。磁碟機的大小持續成長、最終導致 NetApp 開發 NetApp RAID-TEC 技術、引進第三個同位元磁碟機。

某些歷史資料庫最佳實務做法建議使用 RAID-10 、也稱為等量鏡射。因為有多個雙磁碟故障案例、因此資料保護功能比 RAID DP 更少、而在 RAID DP 中則沒有。

由於效能考量、有些歷史資料庫最佳實務做法表示 RAID-10 較 RAID-4/5/6 選項更為偏好。這些建議有時是指 RAID 罰款。雖然這些建議一般都是正確的、但不適用於在 ONTAP 中實作 RAID 。效能考量與同位元重生有關。在傳統的 RAID 實作中、處理資料庫執行的例行隨機寫入作業需要多個磁碟讀取才能重新產生同位元資料並完成寫入。其懲罰定義為執行寫入作業所需的額外讀取 IOPS 。

ONTAP 不會發生 RAID 損失、因為寫入會分段在記憶體中產生同位元檢查、然後以單一 RAID 等量磁碟寫入磁碟。完成寫入作業不需要讀取。

總而言之、相較於 RAID 10 、 RAID DP 和 RAID-TEC 可提供更多可用容量、更好的磁碟機故障防護、而且不會犧牲效能。

硬體故障保護： **NVRAM**

任何服務資料庫工作負載的儲存陣列、都必須儘快服務寫入作業。此外、寫入作業也必須受到保護、避免因電源故障等非預期事件而遺失。這表示任何寫入作業都必須安全地儲存在至少兩個位置。

AFF 和 FAS 系統仰賴 NVRAM 來滿足這些需求。寫入程序的運作方式如下：

1. 傳入寫入資料儲存在 RAM 中。
2. 必須對磁碟上的資料所做的變更、會同時記入本機節點和合作夥伴節點上的 NVRAM 。NVRAM 不是寫入快取、而是類似資料庫重做記錄的日誌。在正常情況下、系統不會讀取。它僅用於恢復、例如在 I/O 處理期間發生電源故障後。
3. 然後寫入會被確認給主機。

此階段的寫入程序從應用程式的角度來看已完成、而且資料會受到保護、不會遺失、因為資料會儲存在兩個不同的位置。最後、變更會寫入磁碟、但此程序會從應用程式的觀點超出頻外、因為它會在寫入確認之後發生、因此不會影響延遲。此程序再次類似於資料庫記錄。對資料庫的變更會盡快記錄在重做記錄檔中、然後將變更確認為已認可。資料檔案的更新會在稍後進行、不會直接影響處理速度。

如果控制器發生故障、合作夥伴控制器會取得所需磁碟的所有權、並重新執行 NVRAM 中記錄的資料、以恢復發生故障時正在執行的任何 I/O 作業。

硬體故障保護： **NVFAIL**

如前所述、寫入必須先登入本機 NVRAM 及至少一個其他控制器上的 NVRAM 、才會被確認。此方法可確保硬體故障或停電不會導致機內 I/O 遺失如果本機 NVRAM 故障或連線至 HA 合作夥伴失敗、則無法再鏡射此傳輸中

的資料。

如果本機 NVRAM 回報錯誤、節點會關機。此關機會導致容錯移轉至 HA 合作夥伴控制器。由於發生故障的控制器尚未確認寫入作業、因此不會遺失任何資料。

除非強制容錯移轉、否則 ONTAP 不允許在資料不同步時進行容錯移轉。以這種方式強制變更條件、即表示資料可能會留在原始控制器中、而且資料遺失是可以接受的。

如果強制容錯移轉、則資料庫特別容易受損、因為資料庫會在磁碟上保留大量的內部資料快取。如果發生強制容錯移轉、先前確認的變更將會有效捨棄。儲存陣列的內容會有效地及時向後跳轉、而且資料庫快取的狀態不再反映磁碟上資料的狀態。

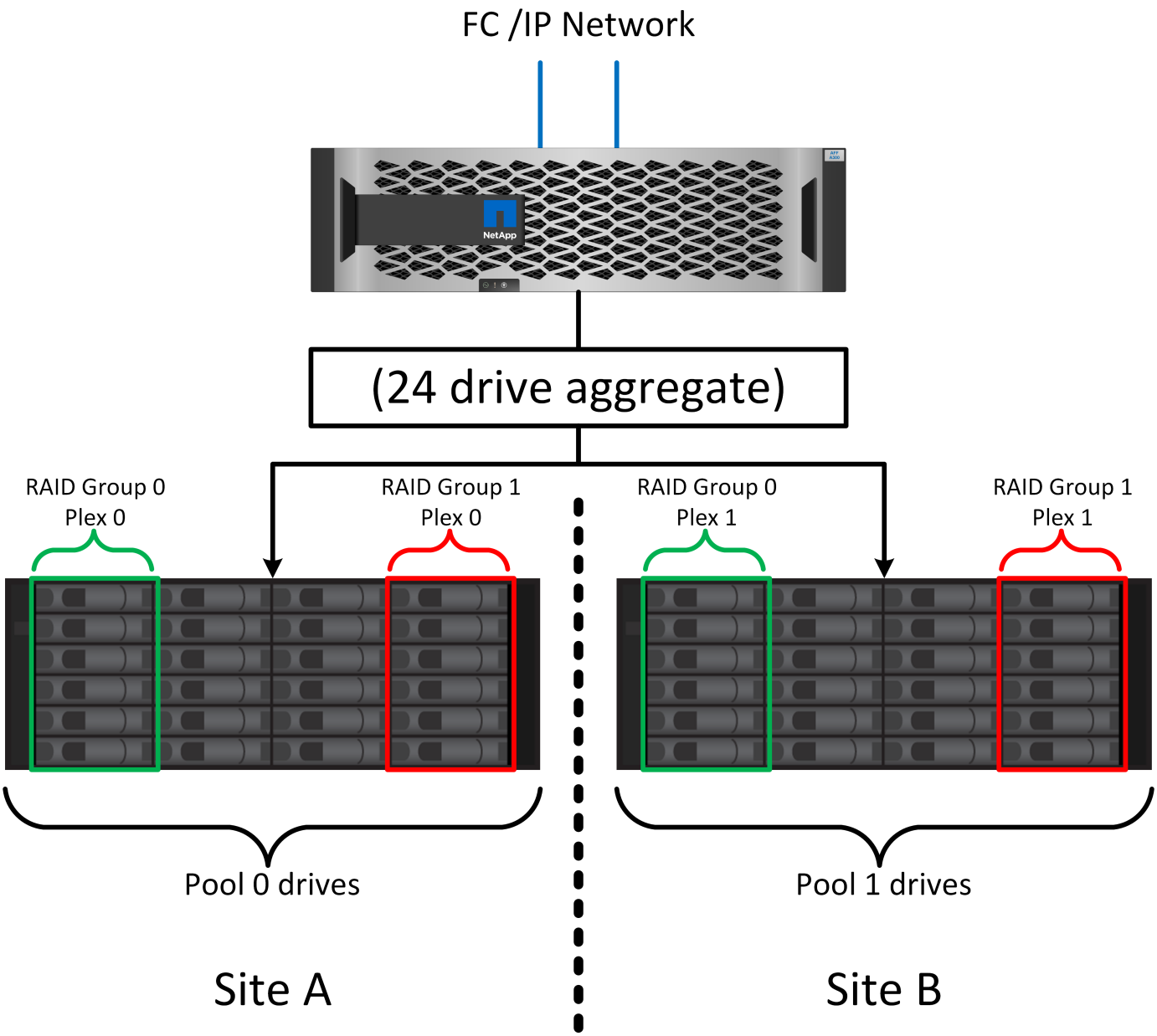
為了保護資料不受這種情況影響、 ONTAP 允許設定磁碟區、以針對 NVRAM 故障提供特殊保護。觸發時、此保護機制會導致磁碟區進入稱為 NVFAIL 的狀態。此狀態會導致 I/O 錯誤、導致應用程式關機、使其不使用過時的資料。資料不應遺失、因為儲存陣列上應該存在任何已確認的寫入資料。

通常的後續步驟是讓系統管理員在手動將 LUN 和磁碟區重新上線之前、先完全關閉主機。雖然這些步驟可能涉及一些工作、但這種方法是確保資料完整性的最安全方法。並非所有資料都需要這項保護、因此 NVFAIL 行為可依每個磁碟區設定。

站台和機櫃故障保護： **SyncMirror** 和叢

SyncMirror 是一項鏡射技術、可增強但不取代 RAID DP 或 RAID-TEC 。它會鏡射兩個不同 RAID 群組的內容。邏輯組態如下：

- 磁碟機會根據位置設定成兩個集區。一個集區由站台 A 上的所有磁碟機組成、第二個集區由站台 B 上的所有磁碟機組成
- 接著會根據鏡射的 RAID 群組集建立通用儲存池（稱為 Aggregate ）。從每個站台擷取的磁碟機數量相等。例如、 20 個磁碟機的 SyncMirror Aggregate 將由站台 A 的 10 個磁碟機和站台 B 的 10 個磁碟機組成
- 指定站台上的每組磁碟機都會自動設定為一個或多個完全備援的 RAID-DP 或 RAID-TEC 群組、而不受鏡像的使用影響。即使在站台遺失之後、也能提供持續的資料保護。



上圖說明 SyncMirror 組態範例。在控制器上建立了 24 個磁碟機的集合體、其中 12 個磁碟機來自於站台 A 上配置的機櫃、 12 個磁碟機來自於站台 B 上配置的機櫃磁碟機分為兩個鏡射 RAID 群組。RAID 群組 0 包括站台 A 上的 6 磁碟機叢、鏡射到站台 B 上的 6 磁碟機叢同樣地、 RAID Group 1 也包含站台 A 的 6 磁碟叢、鏡射到站台 B 的 6 磁碟叢叢

SyncMirror 通常用於提供 MetroCluster 系統的遠端鏡射、每個站台都有一份資料複本。有時候、它是用來在單一系統中提供額外的備援層級。特別是提供機架層級的備援。磁碟機櫃已包含雙電源供應器和控制器、整體上比金屬板稍多、但在某些情況下、可能需要額外的保護。例如、有一位 NetApp 客戶部署 SyncMirror 、用於汽車測試期間使用的行動即時分析平台。系統由獨立的 UPS 系統提供獨立的電源供應器、分為兩個實體機架。

=checksum

對於習慣使用 Oracle RMAN 串流備份的 DBA 來說、檢查和主題特別重要、因為它們會移轉至快照型備份。RMAN 的一項功能是在備份作業期間執行完整性檢查。雖然這項功能有一些價值、但其主要優點是資料庫並未用於現代化的儲存陣列。當實體磁碟機用於 Oracle 資料庫時、幾乎可以確定磁碟機老化時最終會發生毀損、這是由真正儲存陣列中的陣列型校驗和所解決的問題。

使用真正的儲存陣列、資料完整性可在多個層級上使用校驗和加以保護。如果 IP 型網路中的資料毀損、傳輸控制傳輸協定（ TCP ）層會拒絕封包資料並要求重新傳輸。FC 傳輸協定包括校驗和、封裝的 SCSI 資料也一樣。在陣列上之後、 ONTAP 就有 RAID 和 Checksum 保護。可能會發生毀損、但如同大多數企業陣列一樣、系統會偵測並修正毀損。一般而言、整個磁碟機都會故障、導致 RAID 重建、資料庫完整性也不會受到影響。ONTAP 偵測到 Checksum 錯誤的頻率較低、表示磁碟機上的資料已損壞。然後磁碟機故障、 RAID 重建就會開始。資料完整性再次不受影響。

Oracle 資料檔案和重做記錄架構也設計成即使在極端情況下、也能提供最高程度的資料完整性。在最基本的層級、 Oracle 區塊幾乎包含每個 I/O 的 Checksum 和基本邏輯檢查如果 Oracle 尚未當機或將資料表空間離線、則資料會保持不變。資料完整性檢查的程度可調整、 Oracle 也可設定為確認寫入。因此、幾乎所有的當機和故障情況都可以恢復、而且在極罕見的不可恢復情況下、系統會立即偵測到毀損。

大多數使用 Oracle 資料庫的 NetApp 客戶在移轉至快照型備份後、都會停止使用 RMAN 和其他備份產品。仍有一些選項可讓 RMAN 使用 SnapCenter 執行區塊層級的還原。然而、日常使用的 RMAN 、 NetBackup 及其他產品只會偶爾用於建立每月或每季的歸檔複本。

有些客戶選擇執行 `dbv` 定期對現有資料庫執行完整性檢查。NetApp 不鼓勵這種做法、因為它會產生不必要的 I/O 負載。如上所述、如果資料庫先前沒有遇到問題、就有可能發生 `dbv` 偵測問題接近零、此公用程式會在網路和儲存系統上產生非常高的連續 I/O 負載。除非有理由相信存在毀損、例如暴露於已知的 Oracle 錯誤、否則沒有理由執行 `dbv`。

備份與還原基礎知識

Oracle 資料庫和快照型備份

NetApp Snapshot 技術是 ONTAP 上 Oracle 資料庫資料保護的基礎。

關鍵值如下：

- * 簡易性。 * 快照是特定時間點資料容器內容的唯讀複本。
- * 效率。 * 快照在建立時不需要任何空間。只有在資料變更時才會使用空間。
- * 管理能力。 * 由於快照是儲存作業系統的原生部分、因此以快照為基礎的備份策略很容易設定和管理。如果儲存系統已開機、就可以開始建立備份。
- * 擴充性。 * 最多可保留 1024 個檔案和 LUN 的單一容器備份。對於複雜的資料集、可透過一組一致的快照來保護多個資料容器。
- 無論磁碟區包含 1024 個快照或無、效能都不會受到影響。

雖然許多儲存廠商都提供快照技術、但 ONTAP 中的 Snapshot 技術是獨一無二的、可為企業應用程式和資料庫環境帶來顯著效益：

- Snapshot 複本是基礎 Write -Anywhere File Layout （ WAFL ）的一部分。它們不是附加技術或外部技術。這簡化了管理、因為儲存系統是備份系統。
- 快照複本不會影響效能、但某些邊緣情況除外、例如當基礎儲存系統填滿的快照中儲存了大量資料時。
- 「一致性群組」一詞通常是指一組儲存物件、這些物件是以一致的資料集合來管理。特定 ONTAP 磁碟區的快照構成一致性群組備份。

ONTAP 快照的擴充能力也優於競爭技術。客戶可以儲存 5 、 50 或 500 個快照、而不會影響效能。磁碟區目前允許的最大快照數為 1024 。如果需要額外的快照保留、則有選項可將快照串聯至其他磁碟區。

因此、保護託管在 ONTAP 上的資料集非常簡單且具有高度擴充性。備份不需要移動資料、因此備份策略可以根據業務需求量身打造、而非限制網路傳輸率、大量磁帶機或磁碟接移區域。

快照是否為備份？

將快照當作資料保護策略使用的常見問題之一、就是「真實」資料和快照資料位於同一個磁碟機上。遺失這些磁碟機將會導致主要資料和備份遺失。

這是一項有效的考量。本機快照是用於日常備份與還原需求、在這方面、快照是備份。在 NetApp 環境中、將近 99% 的還原案例都仰賴快照來滿足最嚴苛的 RTO 需求。

然而、本機快照不應是唯一的備份策略、因此 NetApp 提供 SnapMirror 和 SnapVault 複寫等技術、可快速有效地將快照複寫至一組不同的磁碟機。在架構正確的解決方案中、快照加上快照複寫功能可將磁帶的使用降至最低、甚至是每季歸檔、或完全消除。

快照型備份

使用 ONTAP Snapshot 複本保護資料有許多選項、快照是許多其他 ONTAP 功能的基礎、包括複寫、災難恢復和複製。快照技術的完整說明不在本文件的範圍內、但以下各節提供一般概觀。

建立資料集快照的主要方法有兩種：

- 損毀一致的備份
- 應用程式一致的備份

資料集的損毀一致備份是指在單一時間點擷取整個資料集結構。如果資料集儲存在單一 NetApp FlexVol Volume 中、則程序很簡單；您可以隨時建立 Snapshot 。如果資料集橫跨多個磁碟區、則必須建立一致性群組（ CG ）快照。建立 CG 快照有多種選項、包括 NetApp SnapCenter 軟體、原生 ONTAP 一致性群組功能、以及使用者維護的指令碼。

當備份點還原足夠時、主要會使用損毀一致的備份。當需要更精細的恢復時、通常需要應用程式一致的備份。

「應用程式一致性」一詞通常是錯誤的。例如、將 Oracle 資料庫置於備份模式稱為應用程式一致的備份、但資料並未以任何方式保持一致或停止。資料會在整個備份過程中持續變更。相反地、大部分的 MySQL 和 Microsoft SQL Server 備份確實會在執行備份之前先將資料關閉。VMware 可能會或可能不會使某些檔案一致。

一致性群組

術語「一致性群組」是指儲存陣列將多個儲存資源視為單一映像來管理的能力。例如、資料庫可能包含 10 個 LUN 。陣列必須能夠以一致的方式備份、還原及複寫這 10 個 LUN 。如果 LUN 的映像在備份時不一致、則無法還原。複寫這 10 個 LUN 需要所有複本彼此完全同步。

討論 ONTAP 時、「一致性群組」一詞並不常使用、因為一致性一向是 ONTAP 內 Volume 和 Aggregate 架構的基本功能。許多其他儲存陣列會將 LUN 或檔案系統視為個別單元進行管理。接著可選擇性地將它們設定為「一致性群組」、以保護資料、但這是組態中的額外步驟。

ONTAP 一向能夠擷取一致的本機和複寫資料映像。雖然 ONTAP 系統上的各種磁碟區通常並未正式描述為一致性群組、但這就是它們的名稱。該磁碟區的快照是一致性群組映像、該快照的還原是一致性群組還原、SnapMirror 和 SnapVault 都提供一致性群組複寫。

一致性群組快照

一致性群組快照（ CG 快照）是基本 ONTAP Snapshot 技術的延伸。標準快照作業可在單一磁碟區內建立所有

資料的一致映像、但有時必須在多個磁碟區甚至跨多個儲存系統建立一致的快照集。結果是一組快照、其使用方式與只有一個個別磁碟區的快照相同。它們可用於本機資料還原、複寫以進行災難恢復、或複製為單一一致的單元。

CG 快照的最大已知用途是資料庫環境、其大小約為 1PB 、跨越 12 個控制器。在此系統上建立的 CG 快照已用於備份、恢復和複製。

大多數情況下、當資料集跨越磁碟區且必須保留寫入順序時、所選管理軟體會自動使用 CG 快照。在此情況下、無需瞭解 CG 快照的技術詳細資料。然而、在某些情況下、複雜的資料保護需求需要對資料保護和複寫程序進行詳細控制。自動化工作流程或使用自訂指令碼來呼叫 CG 快照 API 是其中的一些選項。若要瞭解最佳選項和 CG-snapshot 的角色、需要更詳細的技術說明。

建立一組 CG 快照是兩個步驟：

1. 在所有目標磁碟區上建立寫入屏障。
2. 在圍籬狀態下建立這些磁碟區的快照。

寫入隔離是連續建立的。這表示當隔離程序在多個磁碟區之間設定時、寫入 I/O 會依序凍結在第一個磁碟區上、因為它會繼續提交到稍後出現的磁碟區。這可能一開始就違反了保留寫入順序的要求、但這僅適用於非同步在主機上發出的 I/O 、不需仰賴任何其他寫入。

例如、資料庫可能會發出許多非同步資料檔案更新、並允許作業系統重新排序 I/O 、並根據其本身的排程器組態完成這些更新。這類 I/O 的順序無法保證、因為應用程式和作業系統已釋出保留寫入順序的要求。

以計數器為例、大部分的資料庫記錄活動都是同步的。在確認 I/O 之前、資料庫不會繼續進行記錄寫入、而且必須保留這些寫入的順序。如果記錄 I/O 到達圍籬式磁碟區、則不會予以確認、應用程式會在進一步寫入時加以封鎖。同樣地、檔案系統中繼資料 I/O 通常是同步的。例如、檔案刪除作業不得遺失。如果具有 xfs 檔案系統的作業系統刪除了檔案、而更新 xfs 檔案系統中繼資料的 I/O 則會移除位於圍籬磁碟區上的該檔案參照、則檔案系統活動就會暫停。這可確保 CG 快照作業期間檔案系統的完整性。

在目標磁碟區之間設定寫入屏障之後、就可以開始建立快照。由於磁碟區的狀態會從相關寫入點凍結、因此不需要同時精確建立快照。為了防範建立 CG 快照的應用程式中的瑕疵、初始寫入屏障包含可設定的逾時時間、ONTAP 會在定義的秒數後自動釋放隔離功能並繼續寫入處理。如果所有快照都是在逾時期間發生之前建立的、則產生的一組快照是有效的一致性群組。

相關寫入順序

從技術觀點來看、一致性群組的關鍵在於保留寫入順序、特別是根據寫入順序。例如、寫入 10 個 LUN 的資料庫會同時寫入所有 LUN 。許多寫入都是以非同步方式發出、這表示完成的順序不重要、實際完成的順序會因作業系統和網路行為而異。

某些寫入作業必須存在於磁碟上、資料庫才能繼續進行其他寫入作業。這些關鍵寫入作業稱為「相關寫入」。後續寫入 I/O 則取決於磁碟上是否有這些寫入資料。這 10 個 LUN 的任何快照、恢復或複寫都必須確保相關寫入順序受到保證。檔案系統更新是寫入順序相關寫入的另一個範例。必須保留檔案系統變更的順序、否則整個檔案系統可能會毀損。

策略

以快照為基礎的備份主要有兩種方法：

- 損毀一致的備份
- 快照保護的熱備份

資料庫的損毀一致備份是指在單一時間點擷取整個資料庫結構、包括資料檔案、重做記錄和控制檔。如果資料庫儲存在單一 NetApp FlexVol Volume 中、則程序很簡單；您可以隨時建立 Snapshot 。如果資料庫橫跨磁碟區、則必須建立一致性群組（ CG ）快照。建立 CG 快照有多種選項、包括 NetApp SnapCenter 軟體、原生 ONTAP 一致性群組功能、以及使用者維護的指令碼。

當備份點還原足夠時、主要會使用損毀一致的 Snapshot 備份。在某些情況下可以套用歸檔記錄檔、但如果需要更精細的時間點還原、則最好使用線上備份。

快照型線上備份的基本程序如下：

1. 將資料庫放入 `backup` 模式。
2. 建立所有託管資料檔案的磁碟區快照。
3. 結束 `backup` 模式。
4. 執行命令 `alter system archive log current` 強制記錄歸檔。
5. 為所有託管歸檔記錄的磁碟區建立快照。

此程序會產生一組快照、其中包含備份模式中的資料檔案、以及在備份模式中產生的重要歸檔記錄。這是恢復資料庫的兩項需求。控制檔等檔案也應受到保護、以方便使用、但唯一的絕對需求是保護資料檔案和歸檔記錄。

雖然不同的客戶可能有非常不同的策略、但幾乎所有這些策略最終都是以下列相同原則為基礎。

快照型還原

在設計 Oracle 資料庫的 Volume 配置時、第一個決定是是否使用 Volume NetApp SnapRestore （ VBSR ）技術。

Volume 型 SnapRestore 可讓磁碟區立即還原至較早的時間點。由於磁碟區上的所有資料都已還原、因此 VBSR 可能不適用於所有使用案例。例如、如果整個資料庫（包括資料檔案、重做記錄和歸檔記錄）儲存在單一磁碟區上、且此磁碟區使用 VBSR 還原、則資料會遺失、因為較新的歸檔記錄和重做資料會被捨棄。

還原不需要 VSR 。許多資料庫都可以使用檔案型單一檔案 SnapRestore （ SFSR ）來還原、或只是將檔案從快照複製回作用中的檔案系統。

當資料庫非常大或必須盡快恢復時、最好使用 VBSR 、而使用 VSR 需要隔離資料檔案。在 NFS 環境中、指定資料庫的資料檔案必須儲存在專用的磁碟區中、而這些磁碟區不會受到任何其他類型的檔案污染。在 SAN 環境中、資料檔案必須儲存在專用 FlexVol 磁碟區上的專用 LUN 中。如果使用 Volume Manager （包括 Oracle 自動儲存管理 [AS] ）、則磁碟群組也必須專用於資料檔案。

以這種方式隔離資料檔案、可讓檔案還原至較早的狀態、而不會損壞其他檔案系統。

Snapshot保留

對於 SAN 環境中具有 Oracle 資料的每個 Volume `percent-snapshot-space` 應設為零、因為在 LUN 環境中保留快照空間並不實用。如果百分比保留設為 100 、則具有 LUN 的磁碟區快照需要在磁碟區中有足夠的可用空間、但不包括快照保留空間、以吸收所有資料 100% 的營業額。如果將百分比保留設為較低的值、則需要相對較小的可用空間、但它一律會排除快照保留。這表示 LUN 環境中的快照保留空間會被浪費。

在 NFS 環境中、有兩個選項：

- 設定 `percent-snapshot-space` 根據預期的快照空間使用量。

- 設定 `percent-snapshot-space` 以歸零並統整管理作用中和快照空間使用量。

使用第一個選項、 `percent-snapshot-space` 設為非零值、通常約 20% 。然後、使用者就會隱藏此空間。不過、此值並不會限制使用率。如果具有 20% 保留的資料庫擁有 30% 的營業額、則快照空間可能會超出 20% 保留空間的範圍、並佔用無保留空間。

將保留設定為 20% 等值的主要優點是驗證某些空間永遠可供快照使用。例如、保留 20% 的 1TB 磁碟區只允許資料庫管理員（ DBA ）儲存 800GB 的資料。此組態保證至少有 200GB 的空間可供快照使用。

何時 `percent-snapshot-space` 設為零、則使用者可以使用磁碟區中的所有空間、以提供更好的可見度。DBA 必須瞭解、如果他 / 她看到 1TB 的磁碟區運用快照、則這 1TB 的空間會在使用中資料和 Snapshot 週轉之間共享。

終端使用者之間的選項 1 和選項 2 之間沒有明確的偏好設定。

ONTAP 和第三方快照

Oracle Doc ID 604683.1 說明第三方快照支援的需求、以及備份與還原作業的多種選項。

第三方廠商必須保證公司的快照符合下列要求：

- 快照必須與 Oracle 建議的還原與還原作業整合。
- 快照必須在快照點保持一致的資料庫損毀。
- 快照中的每個檔案都會保留寫入順序。

ONTAP 和 NetApp Oracle 管理產品符合這些要求。

使用 **SnapRestore** 快速恢復 **Oracle** 資料庫

NetApp SnapRestore 技術可從快照快速還原 ONTAP 中的資料。

當關鍵資料集無法使用時、關鍵業務營運就會中斷。磁帶可能會中斷、甚至是從磁碟型備份還原、在網路上傳輸速度可能會很慢。SnapRestore 可提供近乎即時的資料集還原功能、避免這些問題。即使是 PB 規模的資料庫、只要花幾分鐘的時間就能完全還原。

SnapRestore 有兩種形式：檔案 /LUN 型和磁碟區型。

- 無論是 2TB LUN 或 4KB 檔案、個別檔案或 LUN 都能在數秒內還原。
- 無論是 10GB 或 100TB 的資料、檔案或 LUN 的容器都能在數秒內還原。

「檔案或 LUN 的容器」通常指的是 FlexVol Volume 。例如、您可以在單一磁碟區中擁有 10 個組成 LVM 磁碟群組的 LUN 、或是一個磁碟區可以儲存 1000 位使用者的 NFS 主目錄。您可以將整個磁碟區還原為單一作業、而不需為每個個別檔案或 LUN 執行還原作業。此程序也適用於包含多個磁碟區的橫向擴充容器、例如 FlexGroup 或 ONTAP 一致性群組。

SnapRestore 之所以能如此快速有效地運作、是因為快照的本質、基本上是在特定時間點對磁碟區內容進行平行唯讀檢視。作用中區塊是可以變更的實際區塊、而快照則是建立快照時構成檔案和 LUN 之區塊狀態的唯讀檢視。

ONTAP 僅允許唯讀存取快照資料、但可透過 SnapRestore 重新啟動資料。快照會重新啟用為資料的讀寫檢視、並將資料恢復至先前的狀態。SnapRestore 可以在磁碟區或檔案層級運作。這項技術基本上相同、但行為上有

一些小差異。

Volume SnapRestore

Volume 型 SnapRestore 會將整個資料量傳回至較早的狀態。這項作業不需要資料移動、也就是說還原程序基本上是即時的、雖然 API 或 CLI 作業可能需要幾秒鐘的時間才能處理。還原 1GB 的資料並不比還原 1PB 的資料複雜或耗時。這項功能是許多企業客戶移轉至 ONTAP 儲存系統的主要原因。即使是最大的資料集、也能以秒為單位提供 RTO 。

Volume 型 SnapRestore 的缺點之一、是因為磁碟區內的變更會隨時間累積。因此、每個快照和作用中檔案資料都取決於到該點之前的變更。將磁碟區還原為較早的狀態、表示捨棄所有後續對資料所做的變更。然而、不太明顯的是、這包括後續建立的快照。這並不總是理想的。

例如、資料保留 SLA 可能會指定每晚備份 30 天。將資料集還原至五天前以 Volume SnapRestore 建立的快照、將會捨棄前五天建立的所有快照、違反 SLA 。

有許多選項可解決此限制：

1. 資料可從先前的快照複製、而非執行整個 Volume 的 SnapRestore 。此方法最適合較小的資料集。
2. 您可以複製快照、而非還原快照。此方法的限制在於來源快照是複本的相依性。因此、除非也刪除複本、或將其分割成一個不同的 Volume 、否則無法將其刪除。
3. 使用檔案型 SnapRestore 。

File SnapRestore

檔案型 SnapRestore 是更精細的快照型還原程序。個別檔案或 LUN 的狀態會還原、而非還原整個磁碟區的狀態。不需要刪除快照、此作業也不需要對先前的快照建立任何相依性。檔案或 LUN 會立即在作用中磁碟區中可用。

SnapRestore 還原檔案或 LUN 時不需要移動資料。不過、需要進行一些內部中繼資料更新、以反映檔案或 LUN 中的基礎區塊現在同時存在於快照和作用中磁碟區中。不應影響效能、但此程序會封鎖快照的建立、直到快照完成為止。根據還原的檔案總大小、處理速度約為 5Gbps （ 18TB/ 小時）。

Oracle 資料庫線上備份

在備份模式中保護和恢復 Oracle 資料庫需要兩組資料。請注意、這不是唯一的 Oracle 備份選項、而是最常見的選項。

- 備份模式中資料檔案的快照
- 資料檔案處於備份模式時所建立的歸檔記錄

如果需要完整恢復（包括所有已提交的交易）、則需要第三個項目：

- 一組目前的重做記錄

有許多方法可以推動線上備份的還原。許多客戶使用 ONTAP CLI 還原快照、然後使用 Oracle RMAN 或 sqlplus 來完成還原。這在大型正式作業環境中尤其常見、因為資料庫還原的可能性和頻率極低、而且任何還原程序都是由熟練的 DBA 來處理。為了實現完整的自動化、 NetApp SnapCenter 等解決方案包含 Oracle 外掛程式、其中包含命令列和圖形介面。

有些大型客戶已在主機上設定基本指令碼、以便在特定時間將資料庫置於備份模式、以準備排程的快照、藉此採

取更簡單的方法。例如、排程命令 `alter database begin backup` 23 時 58 分、 `alter database end backup` 於 00 ： 02 、然後於午夜直接在儲存系統上排程快照。如此一來、就能實現簡單易用、擴充性極高的備份策略、無需外部軟體或授權。

資料配置

最簡單的配置是將資料檔案隔離到一個或多個專用磁碟區。它們必須不受任何其他檔案類型的污染。這是為了確保資料檔案磁碟區可以透過 SnapRestore 作業快速還原、而不會破壞重要的重做記錄檔、控制檔或歸檔記錄。

SAN 對專用磁碟區內的資料檔案隔離有類似的需求。在 Microsoft Windows 等作業系統中、單一磁碟區可能包含多個資料檔案 LUN 、每個 LUN 都有 NTFS 檔案系統。在其他作業系統中、通常會有邏輯 Volume Manager 。例如、使用 Oracle ASM 時、最簡單的選項是將 ASM 磁碟群組的 LUN 限制在單一磁碟區、以便作為一個單元進行備份和還原。如果基於效能或容量管理的理由而需要額外的磁碟區、則在新磁碟區上建立額外的磁碟群組、將可簡化管理。

如果遵循這些準則、則可直接在儲存系統上排程快照、而無需執行一致性群組快照。原因是 Oracle 備份不需要同時備份資料檔案。線上備份程序旨在讓資料檔案在數小時內緩慢串流至磁帶、因此能夠繼續更新。

在使用分佈於不同磁碟區的 ASM 磁碟群組等情況下、會產生複雜性。在這種情況下、必須執行 CG 快照、以確保 ASM 中繼資料在所有組成磁碟區之間一致。

- 注意： * 驗證 ASM `spfile` 和 `passwd` 檔案不在主控資料檔案的磁碟群組中。這會影響選擇性還原資料檔案和僅還原資料檔案的能力。

本機恢復程序— **NFS**

此程序可以手動或透過 SnapCenter 等應用程式來驅動。基本程序如下：

1. 關閉資料庫。
2. 在所需還原點之前、立即將資料檔案磁碟區復原至快照。
3. 將歸檔記錄重播至所需的點。
4. 如果需要完整還原、請重新播放目前的重做記錄。

此程序假設所需的歸檔記錄檔仍存在於作用中的檔案系統中。如果沒有、則必須還原歸檔記錄、或將 RMAN/sqlplus 導向快照目錄中的資料。

此外、對於較小的資料庫、終端使用者可以直接從中復原資料檔案 `.snapshot` 目錄、無需自動化工具或儲存管理員協助執行 `snaprestore` 命令。

本機恢復程序— **SAN**

此程序可以手動或透過 SnapCenter 等應用程式來驅動。基本程序如下：

1. 關閉資料庫。
2. 將託管資料檔案的磁碟群組置於系統中。此程序會因所選的邏輯磁碟區管理程式而異。使用 ASM 時、此程序需要卸除磁碟群組。在 Linux 中、必須卸除檔案系統、且必須停用邏輯磁碟區和磁碟區群組。目標是停止要還原之目標 Volume 群組上的所有更新。
3. 在所需還原點之前、立即將資料檔案磁碟群組還原至快照。
4. 重新啟動新還原的磁碟群組。

5. 將歸檔記錄重播至所需的點。
6. 如果需要完整還原、請重新播放所有重做記錄。

此程序假設所需的歸檔記錄檔仍存在於作用中的檔案系統中。如果沒有、則必須將歸檔記錄 LUN 離線並執行還原、以還原歸檔記錄。這也是將歸檔記錄分割成專用磁碟區的範例。如果歸檔記錄與重做記錄共用一個磁碟區群組、則必須先將重做記錄複製到其他位置、才能還原整個 LUN 集。此步驟可防止這些最終記錄的交易遺失。

Oracle 資料庫儲存快照最佳化備份

當 Oracle 12c 發行時、快照型備份與還原變得更簡單、因為不需要將資料庫置於熱備份模式。結果是能夠直接在儲存系統上排程快照式備份、同時仍保留執行完整或時間點還原的能力。

雖然 DBA 較熟悉熱備份還原程序、但很長一段時間以來、它仍能使用資料庫處於熱備份模式時未建立的快照。恢復期間、 Oracle 10g 和 11g 需要額外的手動步驟、才能使資料庫保持一致。使用 Oracle 12c 、 `sqlplus` 和 `rman` 包含額外的邏輯、可在非熱備份模式的資料檔案備份上重播歸檔記錄。

如前所述、復原快照型熱備份需要兩組資料：

- 在備份模式下建立的資料檔案快照
- 資料檔案處於熱備份模式時所產生的歸檔記錄

在還原期間、資料庫會從資料檔案讀取中繼資料、以選取所需的歸檔記錄進行還原。

儲存快照最佳化的還原需要稍微不同的資料集、才能達到相同的結果：

- 資料檔案的快照、加上一種識別快照建立時間的方法
- 從最新資料檔案檢查點的時間到快照的確切時間、都會歸檔記錄檔

在還原期間、資料庫會從資料檔案讀取中繼資料、以識別所需的最早歸檔記錄。可以執行完整或時間點恢復。執行時間點還原時、必須知道資料檔案快照的時間。指定的恢復點必須在快照建立時間之後。NetApp 建議您在快照時間中加入至少幾分鐘、以因應時鐘變化。

如需完整的詳細資料、請參閱 Oracle 各版本的 Oracle 12c 說明文件中有關「使用儲存 Snapshot 最佳化進行恢復」主題的 Oracle 文件。此外、請參閱 Oracle 文件 ID 文件 ID 604683.1 、瞭解 Oracle 協力廠商快照支援。

資料配置

最簡單的配置是將資料檔案隔離為一個或多個專用磁碟區。它們必須不受任何其他檔案類型的污染。這是為了確保資料檔案磁碟區可以透過 SnapRestore 作業快速還原、而不會破壞重要的重做記錄檔、控制檔或歸檔記錄檔。

SAN 對專用磁碟區內的資料檔案隔離有類似的需求。在 Microsoft Windows 等作業系統中、單一磁碟區可能包含多個資料檔案 LUN 、每個 LUN 都有 NTFS 檔案系統。在其他作業系統中、通常也會有邏輯 Volume Manager 。例如、使用 Oracle ASM 時、最簡單的選項是將磁碟群組限制在單一磁碟區、以便作為一個單元進行備份和還原。如果基於效能或容量管理的理由而需要額外的磁碟區、則在新磁碟區上建立額外的磁碟群組、將可更輕鬆地進行管理。

如果遵循這些準則、則可直接在 ONTAP 上排程快照、而無需執行一致性群組快照。原因是快照最佳化備份不需要同時備份資料檔案。

在 ASM 磁碟群組等情況下、會發生複雜的情況、而 ASM 磁碟群組會分散在不同的磁碟區中。在這種情況下、必須執行 CG 快照、以確保 ASM 中繼資料在所有組成磁碟區之間一致。

[注意] 確認 ASM spfile 和 passwd 檔案不在主控資料檔案的磁碟群組中。這會影響選擇性還原資料檔案和僅還原資料檔案的能力。

本機恢復程序— NFS

此程序可以手動或透過 SnapCenter 等應用程式來驅動。基本程序如下：

1. 關閉資料庫。
2. 在所需還原點之前、立即將資料檔案磁碟區復原至快照。
3. 將歸檔記錄重播至所需的點。

此程序假設所需的歸檔記錄檔仍存在於作用中的檔案系統中。如果沒有、則必須還原歸檔記錄、或 `rman` 或 `sqlplus` 可導向至中的資料 `.snapshot` 目錄。

此外、對於較小的資料庫、終端使用者可以直接從中復原資料檔案 `.snapshot` 無需自動化工具或儲存管理員協助執行 SnapRestore 命令的目錄。

本機恢復程序— SAN

此程序可以手動或透過 SnapCenter 等應用程式來驅動。基本程序如下：

1. 關閉資料庫。
2. 將託管資料檔案的磁碟群組置於系統中。此程序會因所選的邏輯磁碟區管理程式而異。使用 ASM 時、此程序需要卸除磁碟群組。在 Linux 中、必須卸除檔案系統、並停用邏輯磁碟區和磁碟區群組。目標是停止要還原之目標 Volume 群組上的所有更新。
3. 在所需還原點之前、立即將資料檔案磁碟群組還原至快照。
4. 重新啟動新還原的磁碟群組。
5. 將歸檔記錄重播至所需的點。

此程序假設所需的歸檔記錄檔仍存在於作用中的檔案系統中。如果沒有、則必須將歸檔記錄 LUN 離線並執行還原、以還原歸檔記錄。這也是將歸檔記錄分割成專用磁碟區的範例。如果歸檔記錄與重做記錄共用磁碟區群組、則必須在還原整體 LUN 組之前、將重做記錄複製到其他位置、以免遺失最終記錄的交易。

完整恢復範例

假設資料檔案已毀損或毀損、且需要完整還原。執行程序如下：

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover automatic;
Media recovery complete.
SQL> alter database open;
Database altered.
SQL>
```

時間點恢復範例

整個恢復過程只需一個命令： `recover automatic`。

如果需要時間點恢復、則必須知道快照的時間戳記、並可識別如下：

```
Cluster01::> snapshot show -vserver vserver1 -volume NTAP_oradata -fields
create-time
vserver   volume        snapshot   create-time
--------  ------------  ---------  -------------------------
vserver1  NTAP_oradata  my-backup  Thu Mar 09 10:10:06 2017
```

快照建立時間列於 3 月 9 日和 10 ： 10 ： 06 。為了安全起見、快照時間會增加一分鐘：

```
[oracle@host1 ~]$ sqlplus / as sysdba
Connected to an idle instance.
SQL> startup mount;
ORACLE instance started.
Total System Global Area 1610612736 bytes
Fixed Size                  2924928 bytes
Variable Size            1040191104 bytes
Database Buffers          553648128 bytes
Redo Buffers               13848576 bytes
Database mounted.
SQL> recover database until time '09-MAR-2017 10:44:15' snapshot time '09-
MAR-2017 10:11:00';
```

恢復作業現在已啟動。它指定的快照時間為 10 ： 11 ： 00 、記錄時間後一分鐘、以計算可能的時鐘差異、目標恢復時間為 10 ： 44 。接下來、 sqlplus 會要求所需的歸檔記錄檔、以達到所需的 10 ： 44 恢復時間。

```
ORA-00279: change 551760 generated at 03/09/2017 05:06:07 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_31_930813377.dbf
ORA-00280: change 551760 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 552566 generated at 03/09/2017 05:08:09 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_32_930813377.dbf
ORA-00280: change 552566 for thread 1 is in sequence #32
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 553045 generated at 03/09/2017 05:10:12 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_33_930813377.dbf
ORA-00280: change 553045 for thread 1 is in sequence #33
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 753229 generated at 03/09/2017 05:15:58 needed for
thread 1
ORA-00289: suggestion : /oralogs_nfs/arch/1_34_930813377.dbf
ORA-00280: change 753229 for thread 1 is in sequence #34
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
Log applied.
Media recovery complete.
SQL> alter database open resetlogs;
Database altered.
SQL>
```

使用來完成資料庫的快照還原 `recover automatic` 命令不需要特定授權、而是使用時間點還原 `snapshot time` 需要 Oracle 進階壓縮授權。

Oracle 資料庫管理與自動化工具

ONTAP 在 Oracle 資料庫環境中的主要價值來自核心 ONTAP 技術、例如即時 Snapshot 複本、簡單的 SnapMirror 複寫、以及快速建立 FlexClone Volume 。

在某些情況下、直接在 ONTAP 上簡單設定這些核心功能即可滿足需求、但更複雜的需求則需要協調層。

SnapCenter

SnapCenter 是 NetApp 資料保護的旗艦產品。在極低的層級上、它與 SnapManager 產品在執行資料庫備份的方式上類似、但它是從頭開始打造、提供單一窗口來管理 NetApp 儲存系統上的資料保護。

SnapCenter 包括快照式備份與還原、 SnapMirror 與 SnapVault 複寫等基本功能、以及大型企業大規模營運所需的其他功能。這些進階功能包括擴充的角色型存取控制（ RBAC ）功能、可與協力廠商協調化產品整合的 RESTful API 、資料庫主機上 SnapCenter 外掛程式的不中斷中央管理、以及專為雲端規模環境設計的使用者介

面。

休息

ONTAP 也包含豐富的 RESTful API 集。這可讓協力廠商建立資料保護及其他管理應用程式、並與 ONTAP 進行深度整合。此外、想要建立自己的自動化工作流程和公用程式的客戶也能輕鬆使用 RESTful API 。

Oracle 災難恢復

使用 **ONTAP** 進行 **Oracle** 資料庫災難恢復

災難恢復是指在發生災難性事件（例如破壞儲存系統甚至整個站台的火災）之後還原資料服務。

本文件取代先前發佈的技術報告 _TR-4591 ： Oracle Data Protection _ 和 _TR-4592 ： Oracle on MetroCluster 。 _

當然、災難恢復可以透過使用 SnapMirror 簡單複寫資料來完成、許多客戶會在每小時更新鏡射複本。

對於大多數客戶而言、 DR 不只需要擁有遠端資料複本、還需要能夠快速使用該資料。NetApp 提供兩種技術來滿足這種需求： MetroCluster 和 SnapMirror 主動同步

MetroCluster 指的是硬體組態中的 ONTAP 、其中包括低階同步鏡射儲存設備和許多其他功能。MetroCluster 等整合式解決方案可簡化現今複雜的橫向擴充資料庫、應用程式及虛擬化基礎架構。它以一個簡單的中央儲存陣列取代多種外部資料保護產品和策略。它也能在單一叢集式儲存系統中提供整合式備份、還原、災難恢復和高可用度（ HA ）。

SnapMirror 主動同步是以 SnapMirror Synchronous 為基礎。使用 MetroCluster 、每個 ONTAP 控制器都負責將其磁碟機資料複寫到遠端位置。有了 SnapMirror 主動式同步、您基本上擁有兩個不同的 ONTAP 系統、可維護 LUN 資料的獨立複本、但可以合作呈現該 LUN 的單一執行個體。從主機的角度來看、這是單一 LUN 實體。

雖然 SnapMirror 主動式同步和 MetroCluster 在內部的運作方式截然不同、但對於主機而言、結果卻非常相似。主要差異在於精細度。如果您只需要選取要同步複寫的工作負載、 SnapMirror 主動同步是更好的選擇。如果您需要複寫整個環境、甚至是資料中心、 MetroCluster 是更好的選擇。此外、 SnapMirror 主動式同步目前僅適用於 SAN 、而 MetroCluster 則是多重傳輸協定、包括 SAN 、 NFS 和 SMB 。

MetroCluster

MetroCluster 實體架構和 **Oracle** 資料庫

瞭解 Oracle 資料庫在 MetroCluster 環境中的運作方式、需要對 MetroCluster 系統的實體設計進行一些說明。

本文件取代先前發佈的技術報告 _TR-4592 ： Oracle on MetroCluster 。 _

MetroCluster 可在 **3** 種不同組態中使用

- HA 可與 IP 連線配對
- HA 可與 FC 連線配對

- 單一控制器、具備 FC 連線能力

[注意] 「連線」一詞是指用於跨站台複寫的叢集連線。它並不指主機協定。無論叢集間通訊所使用的連線類型為何、 MetroCluster 組態中的所有主機端通訊協定都會如常支援。

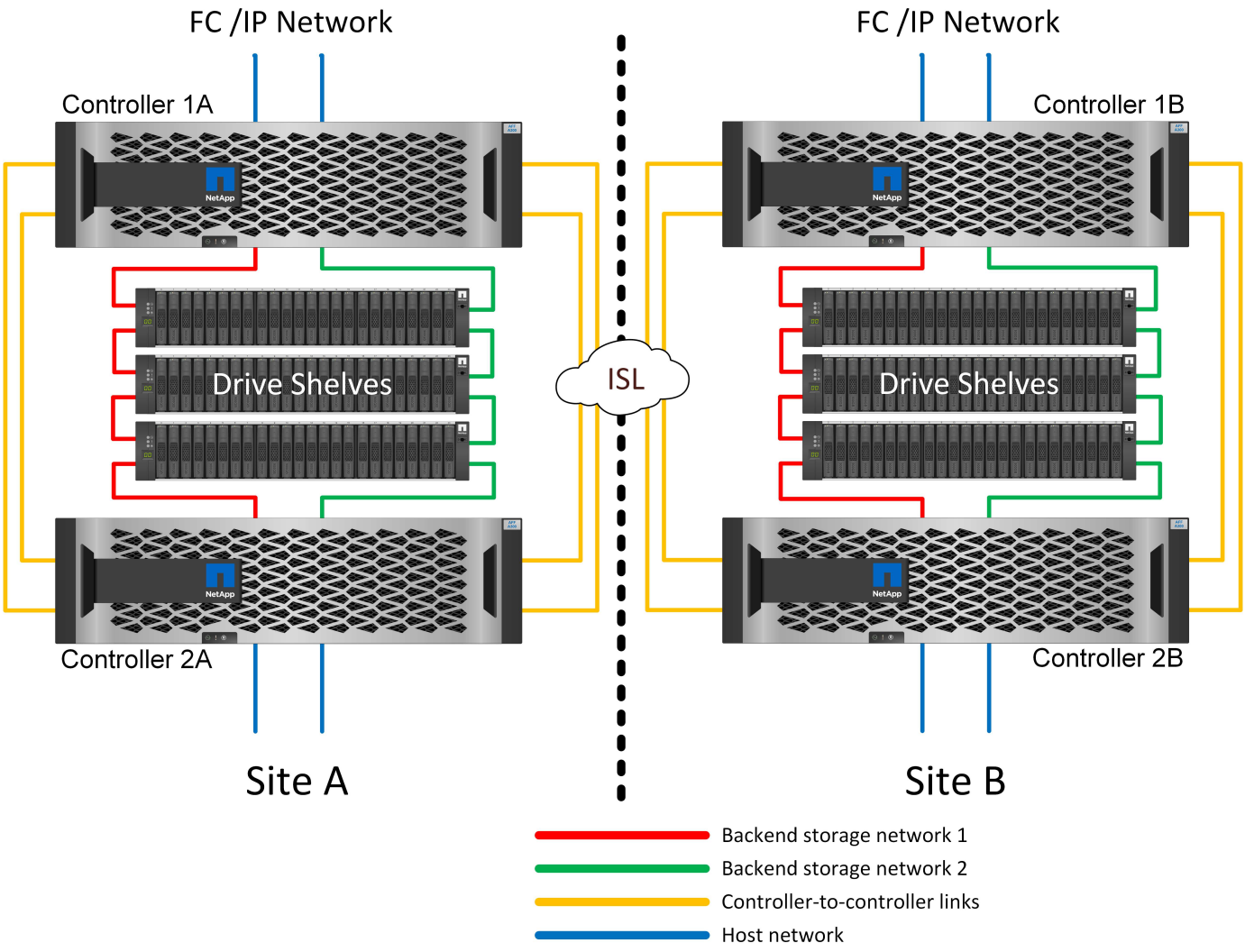
知識產權**MetroCluster**

HA 配對 MetroCluster IP 組態每個站台使用兩或四個節點。此組態選項可增加與雙節點選項相關的複雜度和成本、但它提供重要的優點：站台內備援。簡單的控制器故障不需要透過 WAN 存取資料。透過替代本機控制器、資料存取仍保持在本機狀態。

大多數客戶都選擇 IP 連線、因為基礎架構需求較為簡單。過去、高速跨站台連線通常較容易使用深色光纖和 FC 交換器進行配置、但如今、高速、低延遲的 IP 電路更容易使用。

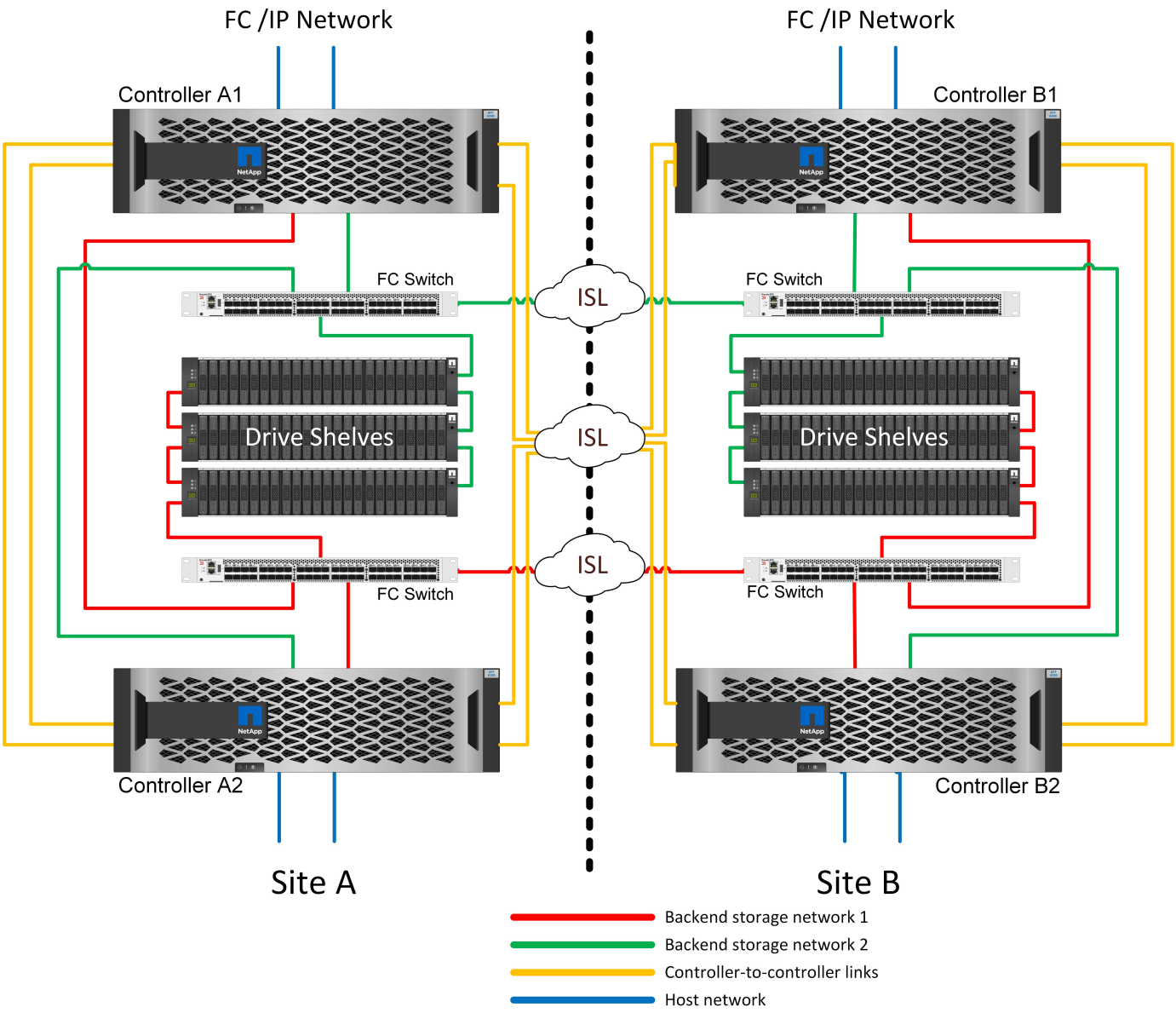
由於唯一的跨站台連線適用於控制器、因此架構也更簡單。在 FC SAN 附加 MetroCluster 中、控制器會直接寫入另一個站台上的磁碟機、因此需要額外的 SAN 連線、交換器和橋接器。相反地、 IP 組態中的控制器會透過控制器寫入相對的磁碟機。

如需其他資訊、請參閱 ONTAP 正式文件和 "SIP解決方案架構與設計MetroCluster"。



HA 配對 FC SAN 附加 MetroCluster

HA 配對 MetroCluster FC 組態每個站台使用兩個或四個節點。此組態選項可增加與雙節點選項相關的複雜度和成本、但它提供重要的優點：站台內備援。簡單的控制器故障不需要透過 WAN 存取資料。透過替代本機控制器、資料存取仍保持在本機狀態。



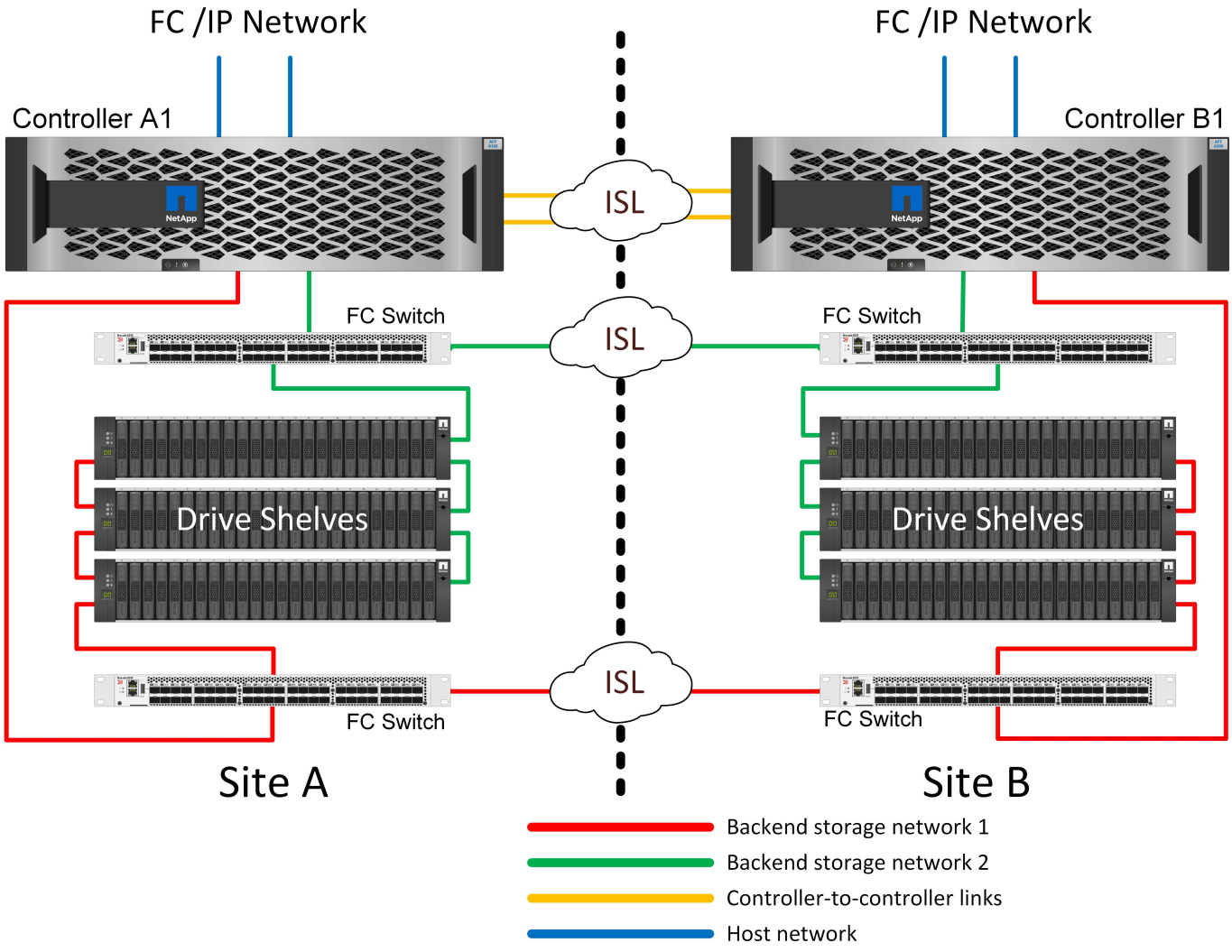
有些多站台基礎架構並非設計用於主動式作業、而是更多用於主要站台和災難恢復站台。在這種情況下、 HA 配對 MetroCluster 選項通常較為理想、原因如下：

- 雖然雙節點 MetroCluster 叢集是 HA 系統、但控制器意外故障或規劃的維護作業需要資料服務必須在相反的站台上線。如果站台之間的網路連線能力不支援所需的頻寬、效能就會受到影響。唯一的選項是將各種主機作業系統和相關服務容錯移轉至替代站台。HA 配對 MetroCluster 叢集可消除此問題、因為遺失控制器會導致同一個站台內的簡單容錯移轉。
- 有些網路拓撲並非設計用於跨站台存取、而是使用不同的子網路或隔離的 FC SAN 。在這種情況下、雙節點 MetroCluster 叢集不再作為 HA 系統運作、因為替代控制器無法將資料提供給位於相反站台的伺服器。HA 配對 MetroCluster 選項是提供完整備援的必要條件。
- 如果將雙站台基礎架構視為單一的高可用度基礎架構、則雙節點 MetroCluster 組態很適合。不過、如果系統

在站台故障後必須長時間運作、則最好使用 HA 配對、因為它會繼續在單一站台內提供 HA 。

雙節點 **FC SAN** 附加 **MetroCluster**

雙節點 MetroCluster 組態每個站台僅使用一個節點。此設計比 HA 配對選項簡單、因為要設定和維護的元件較少。此外、它也降低了佈線和 FC 交換方面的基礎架構需求。最後、它能降低成本。



這項設計的明顯影響是、控制器在單一站台上故障、表示資料可從另一個站台取得。這種限制不一定是個問題。許多企業都有多站台資料中心作業、並有延伸、高速、低延遲的網路、基本上是一個基礎架構。在這些情況下、MetroCluster 的雙節點版本是慣用的組態。多家服務供應商目前以 PB 規模使用雙節點系統。

MetroCluster 恢復功能

MetroCluster 解決方案沒有單點故障：

- 每個控制器都有兩條通往本機站台磁碟櫃的路徑。
- 每個控制器都有兩條通往遠端站台磁碟機櫃的路徑。
- 每個控制器都有兩條通往另一個站台上控制器的路徑。
- 在 HA 配對組態中、每個控制器都有兩條路徑通往本機合作夥伴。

總而言之、您可以移除組態中的任何一個元件、而不會影響 MetroCluster 提供資料的能力。這兩個選項之間恢復能力的唯一差異是 HA 配對版本在站台故障後仍是整個 HA 儲存系統。

MetroCluster 邏輯架構和 **Oracle** 資料庫

瞭解 Oracle 資料庫在 MetroCluster 環境中的運作方式需要對 MetroCluster 系統的邏輯功能進行一些說明。

站台故障保護： **NVRAM** 和 **MetroCluster**

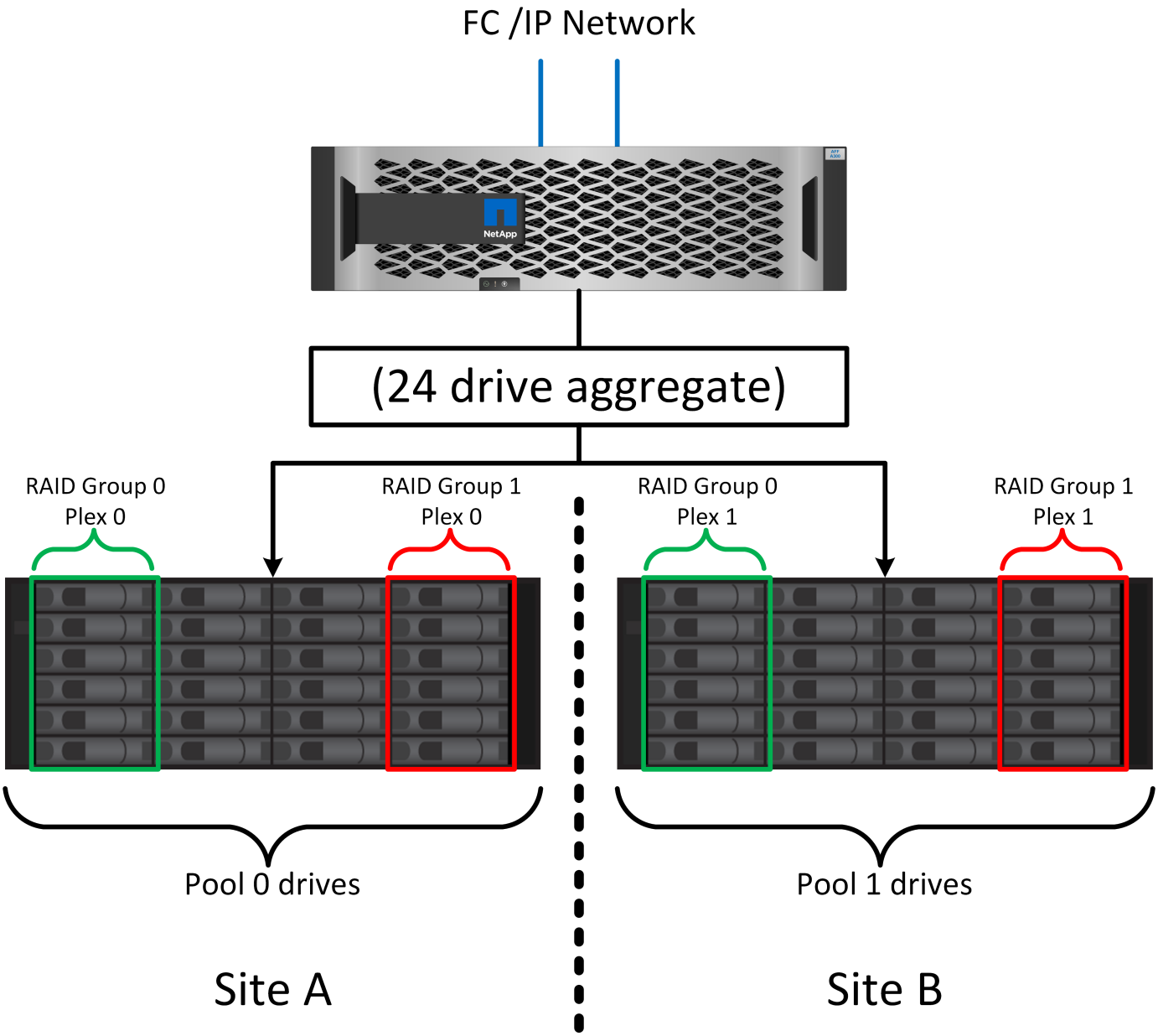
MetroCluster 以下列方式擴充 NVRAM 資料保護：

- 在雙節點組態中、 NVRAM 資料會使用交換器間連結（ ISL ）複寫到遠端合作夥伴。
- 在 HA 配對組態中、 NVRAM 資料會同時複寫到本機合作夥伴和遠端合作夥伴。
- 寫入內容必須複寫到所有合作夥伴、才能予以確認。此架構可將 NVRAM 資料複寫至遠端合作夥伴、保護機上 I/O 不受站台故障影響。此程序不涉及磁碟機層級的資料複寫。擁有該集合體的控制器負責將資料複寫至集合體中的兩個叢集、但在站台遺失時仍必須保護資料、避免在執行中遺失 I/O 。只有當合作夥伴控制器必須接管故障控制器時、才會使用複寫的 NVRAM 資料。

站台和機櫃故障保護： **SyncMirror** 和叢

SyncMirror 是一項鏡射技術、可增強但不取代 RAID DP 或 RAID-TEC 。它會鏡射兩個不同 RAID 群組的內容。邏輯組態如下：

1. 磁碟機會根據位置設定成兩個集區。一個集區由站台 A 上的所有磁碟機組成、第二個集區由站台 B 上的所有磁碟機組成
2. 接著會根據鏡射的 RAID 群組集建立通用儲存池（稱為 Aggregate ）。從每個站台擷取的磁碟機數量相等。例如、 20 個磁碟機的 SyncMirror Aggregate 將由站台 A 的 10 個磁碟機和站台 B 的 10 個磁碟機組成
3. 指定站台上的每組磁碟機都會自動設定為一個或多個完全備援的 RAID DP 或 RAID-TEC 群組、而不受鏡像的使用影響。在鏡射下使用 RAID 、即使在站台遺失之後、也能提供資料保護。



上圖說明 SyncMirror 組態範例。在控制器上建立了 24 個磁碟機的集合體、其中 12 個磁碟機來自於站台 A 上配置的機櫃、 12 個磁碟機來自站台 B 上配置的機櫃磁碟機分為兩個鏡射 RAID 群組。RAID 群組 0 包含站台 A 的 6 磁碟機叢、鏡射到站台 B 的 6 磁碟機叢同樣地、 RAID 群組 1 也包含站台 A 的 6 磁碟叢叢、鏡射到站台 B 的 6 磁碟叢叢

SyncMirror 通常用於提供 MetroCluster 系統的遠端鏡射、每個站台都有一份資料複本。有時候、它是用來在單一系統中提供額外的備援層級。特別是提供機架層級的備援。磁碟機櫃已包含雙電源供應器和控制器、整體上比金屬板稍多、但在某些情況下、可能需要額外的保護。例如、有一位 NetApp 客戶部署 SyncMirror 、用於汽車測試期間使用的行動即時分析平台。系統分為兩個實體機架、分別隨附獨立的電源饋送和獨立的 UPS 系統。

備援故障： **NVFAIL**

如前所述、寫入必須先登入本機 NVRAM 及至少一個其他控制器上的 NVRAM 、才會被確認。此方法可確保硬體故障或停電不會導致機內 I/O 遺失如果本機 NVRAM 故障或連線至其他節點失敗、則資料將不再鏡射。

如果本機 NVRAM 回報錯誤、節點會關機。當使用 HA 配對時、此關機會導致容錯移轉至合作夥伴控制器。使

用 MetroCluster 時、行為取決於所選的整體組態、但可能會導致自動容錯移轉至遠端記事。無論如何、由於發生故障的控制器尚未確認寫入作業、因此不會遺失任何資料。

站台對站台連線故障會封鎖 NVRAM 複寫至遠端節點、這種情況更為複雜。寫入不再複寫到遠端節點、因此如果控制器發生災難性錯誤、可能會導致資料遺失。更重要的是、在這些情況下、嘗試容錯移轉至其他節點會導致資料遺失。

控制因素是 NVRAM 是否同步。如果 NVRAM 已同步、則節點對節點容錯移轉可安全地繼續進行、不會有資料遺失的風險。在 MetroCluster 組態中、如果 NVRAM 和基礎 Aggregate plex 同步、則可以安全地繼續進行轉換、而不會有資料遺失的風險。

除非強制進行容錯移轉或切換、否則 ONTAP 不允許在資料不同步時進行容錯移轉或切換。以這種方式強制變更條件、即表示資料可能會留在原始控制器中、而且資料遺失是可以接受的。

如果強制進行容錯移轉或切換、資料庫和其他應用程式尤其容易毀損、因為它們會在磁碟上保留較大的內部資料快取。如果發生強制容錯移轉或切換、先前確認的變更將會有效捨棄。儲存陣列的內容會有效地及時向後跳轉、而且快取狀態不再反映磁碟上資料的狀態。

為了避免這種情況發生、 ONTAP 允許設定磁碟區、以針對 NVRAM 故障提供特殊保護。觸發時、此保護機制會導致磁碟區進入稱為 NVFAIL 的狀態。此狀態會導致 I/O 錯誤、導致應用程式當機。這項當機會導致應用程式關機、使其不使用過時的資料。資料不應遺失、因為記錄中應存在任何已認可的交易資料。通常的後續步驟是讓系統管理員在手動將 LUN 和磁碟區重新上線之前、先完全關閉主機。雖然這些步驟可能涉及一些工作、但這種方法是確保資料完整性的最安全方法。並非所有資料都需要這項保護、因此 NVFAIL 行為可依每個磁碟區設定。

HA 配對與 MetroCluster

MetroCluster 提供兩種組態：雙節點和 HA 配對。雙節點組態在 NVRAM 上的運作方式與 HA 配對相同。如果發生突然故障、合作夥伴節點可以重新執行 NVRAM 資料、以確保磁碟機一致、並確保沒有遺失任何已確認的寫入資料。

HA 配對組態也會將 NVRAM 複寫到本機合作夥伴節點。簡單的控制器故障會在合作夥伴節點上重新執行 NVRAM 、而獨立 HA 配對則不使用 MetroCluster 。萬一突然完全遺失站台、遠端站台也需要 NVRAM 、才能讓磁碟機保持一致、開始提供資料。

MetroCluster 的一個重要層面是、在正常作業條件下、遠端節點無法存取合作夥伴資料。每個站台基本上都是一個可假設對方站台特性的個別系統。此程序稱為「轉換」、包含計畫性的轉換、可在不中斷營運的情況下、將站台作業移轉至另一個站台。它也包括站台遺失的非計畫性情況、以及災難恢復需要手動或自動切換。

切換與切換

術語切換和切換是指在 MetroCluster 組態中、在遠端控制器之間轉換磁碟區的程序。此程序僅適用於遠端節點。在四個磁碟區組態中使用 MetroCluster 時、本機節點容錯移轉是先前所述的相同接管和恢復程序。

計畫性切換與切換

規劃的切換或切換類似於節點之間的接管或恢復。此程序有多個步驟、可能需要幾分鐘的時間、但實際發生的是儲存設備和網路資源的多階段順暢轉換。控制傳輸的速度比執行完整命令所需的時間快得多。

接管 / 恢復與切換 / 切換回復之間的主要差異在於對 FC SAN 連線能力的影響。使用本機接管 / 恢復功能、主機會遺失通往本機節點的所有 FC 路徑、並仰賴其原生 MPIO 來切換至可用的替代路徑。連接埠不會重新定位。透過切換和切換、控制器上的虛擬 FC 目標連接埠會轉換到另一個站台。它們在 SAN 上實際上已經停用一段時間、然後重新出現在替代控制器上。

SyncMirror 逾時

SyncMirror 是一項 ONTAP 鏡射技術、可針對機櫃故障提供保護。當機櫃之間相隔一段距離時、就能獲得遠端資料保護。

SyncMirror 無法提供通用同步鏡像。因此、可用度更高。有些儲存系統使用固定的全或全自動鏡射、有時稱為 Domino 模式。這種形式的鏡像在應用程式中受到限制、因為如果與遠端站台的連線中斷、所有寫入活動都必須停止。否則、寫字會存在於某個站台、但不會存在於另一個站台。一般而言、如果站台對站台連線中斷超過一段短時間（例如 30 秒）、這類環境就會設定為使 LUN 離線。

這種行為是小型環境子集的理想選擇。不過、大多數應用程式都需要一套解決方案、能夠在正常作業條件下提供保證同步複寫、但能夠暫停複寫。站台對站台連線能力完全中斷通常被視為近乎災難的情況。一般而言、這類環境會保持在線上狀態並提供資料、直到連線能力修復或正式決定關閉環境以保護資料為止。純粹因為遠端複寫失敗而需要自動關閉應用程式、這是不尋常的。

SyncMirror 支援同步鏡射需求、並可靈活調整逾時時間。如果與遠端控制器和 / 或叢的連線中斷、 30 秒定時器就會開始倒數。當計數器達到 0 時、會使用本機資料繼續寫入 I/O 處理。資料的遠端複本可以使用、但會在連線恢復之前、及時凍結。重新同步利用 Aggregate 層級快照、將系統儘快恢復至同步模式。

值得注意的是、在許多情況下、這種通用的「全或全無」 Domino 模式複寫功能更適合在應用程式層上實作。例如、 Oracle DataGuard 包括最大保護模式、可在任何情況下保證執行個體的長時間複寫。如果複寫連結失敗超過可設定的逾時時間、資料庫就會關閉。

使用 **Fabric** 附加 **MetroCluster** 自動進行無人值守切換

自動無人值守切換（ AUSO ）是一項 Fabric 附加 MetroCluster 功能、可提供一種跨站台 HA 的形式。如前所述、 MetroCluster 有兩種類型：每個站台上只有一個控制器、或每個站台上有一個 HA 配對。HA 選項的主要優點是、計畫性或非計畫性控制器關機仍可讓所有 I/O 成為本機。單一節點選項的優勢在於降低成本、複雜度和基礎架構。

AUSO 的主要價值在於改善 Fabric 附加 MetroCluster 系統的 HA 功能。每個站台都會監控相對站台的健全狀況、如果沒有節點仍可提供資料、 AUSO 就會導致快速的轉換。這種方法在每個站台只有一個節點的 MetroCluster 組態中特別有用、因為在可用度方面、它使組態更接近 HA 配對。

AUSO 無法在 HA 配對層級提供全方位監控。HA 配對可提供極高的可用度、因為它包含兩條備援實體纜線、可用於直接節點對節點通訊。此外、 HA 配對中的兩個節點都能存取備援迴圈上的同一組磁碟、為一個節點提供另一條路由來監控另一個節點的健全狀況。

MetroCluster 叢集存在於站台之間、節點對節點通訊和磁碟存取都仰賴站台對站台網路連線。監控叢集其餘部分的活動訊號的能力有限。AUSO 必須區分其他站台實際停機、而非因為網路問題而無法使用的情況。

因此、如果 HA 配對中的控制器偵測到因特定原因（例如系統異常）而發生的控制器故障、就會提示接管。如果連線完全中斷、也可能會提示接管、有時也稱為「失去心跳」。

只有在原始站台偵測到特定故障時、 MetroCluster 系統才能安全地執行自動切換。此外、擁有儲存系統所有權的控制器必須能夠保證磁碟和 NVRAM 資料同步。控制器無法保證進行變更的安全性、因為它與來源站台失去接觸、而該站台仍可運作。如需將交換作業自動化的其他選項、請參閱下一節中的 MetroCluster tiebreaker （ MCTB ）解決方案資訊。

MetroCluster tiebreaker 搭配網路附加 **MetroCluster**

。 "NetApp MetroCluster tiebreaker" 軟體可在第三個站台上執行、以監控 MetroCluster 環境的健全狀況、傳送通知、並在災難情況下強制切換。如需有關斷路器的完整說明、請參閱 "NetApp 支援網站"但 MetroCluster 斷路

器的主要用途是偵測站台遺失。它還必須區分站台遺失和連線中斷。例如、不應因為斷路器無法到達主要站台而進行切入、這就是為什麼斷路器也會監控遠端站台與主要站台聯絡的能力。

與 AUSO 的自動切換功能也相容於 MCTB 。AUSO 反應非常迅速、因為它的設計是偵測特定故障事件、然後只有在 NVRAM 和 SyncMirror 叢同步時才叫用切入。

相反地、斷路器位於遠端位置、因此必須等到定時器結束後才會宣告站台停機。tiebreaker 最終會偵測 AUSO 涵蓋的控制器故障類型、但一般而言、 AUSO 已經開始進行開關作業、而且可能會在 tiebreaker 運作之前完成開關作業。產生的第二個來自 tiebreaker 的切換命令將會遭到拒絕。

- 注意： * 強制切入時、 MCTB 軟體無法驗證 NVRAM 是否與 / 或叢同步。如果已設定自動切換、則應在維護活動期間停用、導致 NVRAM 或 SyncMirror 叢同步中斷。

此外、 MCTB 可能無法因應導致下列事件順序的滾動災難：

1. 站台之間的連線中斷超過 30 秒。
2. SyncMirror 複寫逾時、且作業會繼續在主要站台上執行、使遠端複本過時。
3. 主站台會遺失。結果是主站台上存在未複寫的變更。因此、由於下列幾個原因、可能不希望進行任何一次的重新操作：
    - 關鍵資料可能會出現在主要站台上、而且該資料最終可能會恢復。允許應用程式繼續作業的轉換作業、將會有效捨棄該關鍵資料。
    - 當站台遺失時、使用主要站台上儲存資源的仍在運作中站台上的應用程式可能已快取資料。切入會導致資料的過時版本與快取不相符。
    - 當發生站台遺失時、使用主要站台上儲存資源的仍在運作中站台上的作業系統、可能已快取資料。切入會導致資料的過時版本與快取不相符。最安全的選項是將斷路器設定為在偵測到站台故障時傳送警示、然後讓人員決定是否強制進行轉換。應用程式和（或）作業系統可能需要先關機、才能清除任何快取資料。此外、 NVFAIL 設定也可用於新增進一步的保護、並協助簡化容錯移轉程序。

ONTAP Mediator 搭配 MetroCluster IP

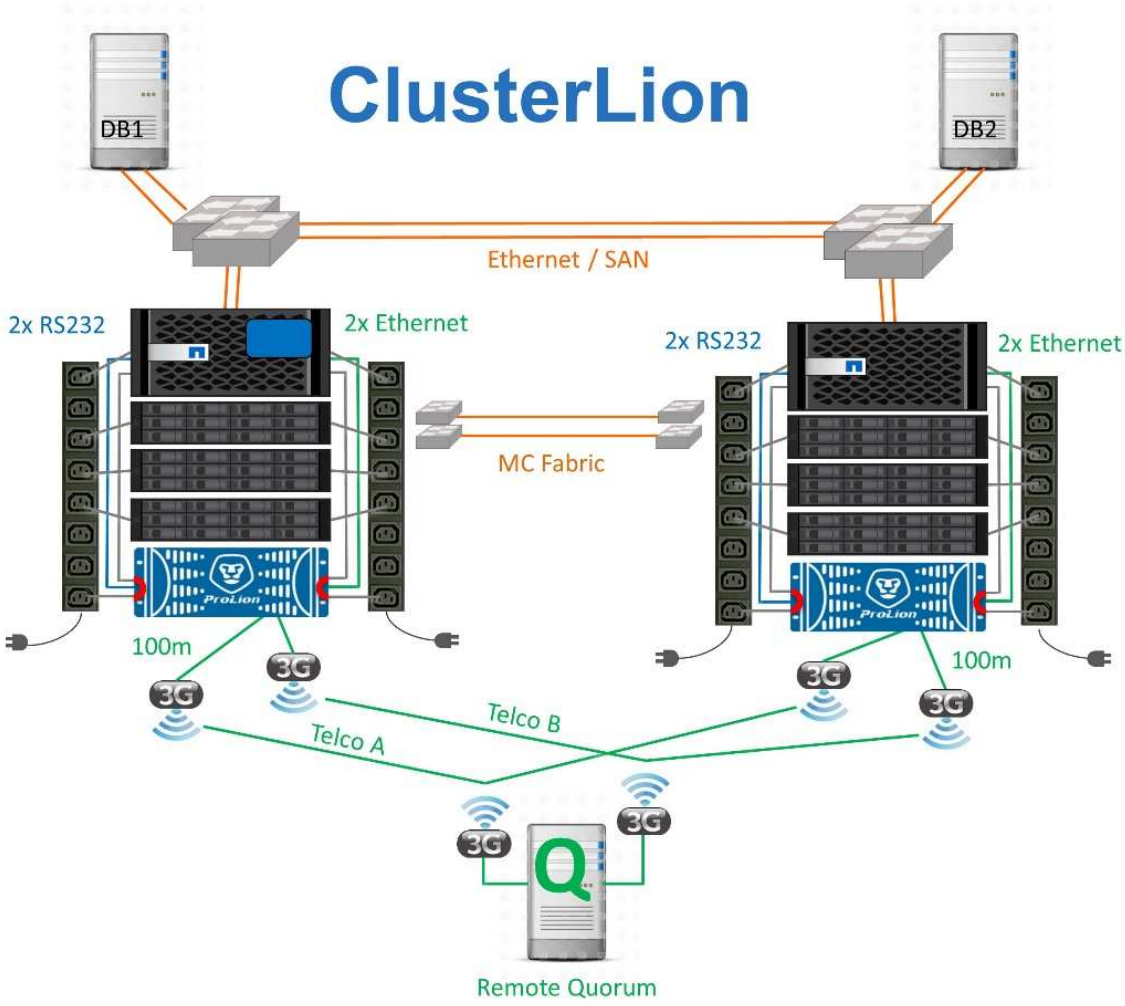
ONTAP Mediator 可搭配 MetroCluster IP 和某些其他 ONTAP 解決方案使用。它是一項傳統的斷路器服務、就像上述的 MetroCluster tiebreaker 軟體一樣、但也包含一項重要功能、即執行自動無人值守的移除。

光纖連接的 MetroCluster 可直接存取位於相對站台的儲存裝置。這可讓一個 MetroCluster 控制器從磁碟機讀取心跳資料、以監控其他控制器的健全狀況。這可讓一個控制器辨識另一個控制器的故障、並執行切換。

相反地、 MetroCluster IP 架構只會透過控制器控制器連線路由所有 I/O 、而無法直接存取遠端站台上的儲存裝置。這會限制控制器偵測故障和執行轉換的能力。因此、 ONTAP Mediator 必須作為斷路器裝置、才能偵測站台遺失並自動執行轉換。

使用 ClusterLion 的虛擬第三站點

ClusterLion 是一款先進的 MetroCluster 監控設備、可作為虛擬第三站點使用。此方法可讓 MetroCluster 安全部署在雙站台組態中、並具備全自動的轉換功能。此外、 ClusterLion 還能執行額外的網路層級監控、並執行後置作業。完整文件可從 ProLion 取得。



- ClusterLion 設備會使用直接連接的乙太網路和序列纜線來監控控制器的健全狀況。
- 這兩台設備透過備援的 3G 無線連線彼此連線。
- ONTAP 控制器的電源會透過內部中繼路由傳送。發生站台故障時、包含內部 UPS 系統的 ClusterLion 會先切斷電源連線、然後再啟動切入。此程序可確保不會發生任何大腦分割狀況。
- ClusterLion 會在 30 秒 SyncMirror 逾時內執行切換、或完全不執行。
- 除非 NVRAM 和 SyncMirror 叢集的狀態同步、否則 ClusterLion 不會執行切入。
- 由於 ClusterLion 只會在 MetroCluster 完全同步時執行切入、因此不需要 NVFAIL 。此組態可讓擴充 Oracle RAC 等站台跨距環境保持連線、即使在非計畫性的轉換期間亦然。
- 支援包括光纖連接的 MetroCluster 和 MetroCluster IP

SyncMirror 的 **Oracle** 資料庫

SyncMirror 是 MetroCluster 系統的 Oracle 資料保護基礎、是最大效能的橫向擴充同步鏡射技術。

使用 **SyncMirror** 保護資料

在最簡單的層級上、同步複寫表示必須先對鏡射儲存設備的兩側進行任何變更、然後才會被確認。例如、如果資料庫正在寫入記錄檔、或是正在修補 VMware 來賓作業系統、則寫入作業絕不能遺失。作爲一種協議級別，在

兩個站點上的非易失性介質被認可之前，存儲系統不得確認寫入內容。只有這樣、在不遺失資料的風險下繼續作業是安全的。

使用同步複寫技術是設計和管理同步複寫解決方案的第一步。最重要的考量是瞭解在各種計畫性和非計畫性失敗案例中可能發生的情況。並非所有同步複寫解決方案都提供相同的功能。如果您需要提供零恢復點目標（ RPO ）的解決方案、亦即零資料遺失、則必須考慮所有故障情況。特別是、當站台之間的連線中斷而無法進行複寫時、預期會產生什麼結果？

SyncMirror 資料可用度

MetroCluster 複寫是以 NetApp SyncMirror 技術為基礎、其設計旨在有效率地切換至同步模式及從同步模式切換到同步模式。這項功能符合要求同步複寫、但也需要高可用度資料服務的客戶需求。例如、如果中斷與遠端站台的連線、通常最好讓儲存系統繼續以非複寫狀態運作。

許多同步複寫解決方案只能以同步模式運作。這種類型的全或全無複寫有時稱為 Domino 模式。這類儲存系統會停止提供資料、而不允許資料的本機和遠端複本進行非同步處理。如果複寫被強制中斷、重新同步可能會非常耗時、而且可能會讓客戶在重新建立鏡像期間暴露在完全資料遺失的風險中。

SyncMirror 不僅可以在無法連線到遠端站台時、無縫切換至同步模式、也可以在連線恢復時、快速重新同步至 RPO = 0 狀態。遠端站台的資料過時複本也可在重新同步期間保留為可用狀態、以確保資料的本機和遠端複本隨時都存在。

在需要 Domino 模式的情況下、 NetApp 提供 SnapMirror 同步（ SM-S ）。應用程式層級選項也存在、例如 Oracle DataGuard 或主機端磁碟鏡射的延長逾時。如需其他資訊和選項、請洽詢您的 NetApp 或合作夥伴客戶團隊。

使用 **MetroCluster** 進行 **Oracle** 資料庫容錯移轉

```
 Metrocluster is an ONTAP feature that can protect your Oracle databases
with RPO=0 synchronous mirroring across sites, and it scales up to support
hundreds of databases on a single MetroCluster system. It's also simple to
use. The use of MetroCluster does not necessarily add to or change any
best practices for operating a enterprise applications and databases.
```

通常的最佳實務做法仍適用、如果您的需求只需要 `RPO = 0` 資料保護、則 `MetroCluster` 會滿足您的需求。然而、大多數客戶不僅使用 `MetroCluster` 來保護 `RPO =0` 資料、還能在災難期間改善 `RTO` 、並在站台維護活動中提供透明的容錯移轉。

使用預先設定的作業系統進行容錯移轉

SyncMirror 在災難恢復站點上提供資料的同步複本、但要讓資料可用、則需要作業系統和相關應用程式。基本自動化可大幅改善整體環境的容錯移轉時間。Oracle RAC 、 Veritas 叢集伺服器（ VCS ）或 VMware HA 等叢集式產品通常用於在站台之間建立叢集、在許多情況下、容錯移轉程序可以使用簡單的指令碼來驅動。

如果主節點遺失、叢集軟體（或指令碼）會設定為在替代站台上線應用程式。其中一個選項是建立預先針對構成應用程式的 NFS 或 SAN 資源所預先設定的待命伺服器。如果主站台發生故障、叢集軟體或指令碼替代方案會執行類似下列的一系列動作：

1. 強制 MetroCluster 進行重新操作
2. 執行 FC LUN 探索（僅限 SAN ）

3. 掛載檔案系統
4. 啟動應用程式

此方法的主要需求是在遠端站台上執行作業系統。它必須預先設定應用程式二進位檔、也就是說、修補等工作必須在主要站台和待命站台上執行。或者、應用程式二進位檔可鏡射至遠端站台、並在宣告災難時掛載。

實際的啟動程序很簡單。LUN 探索等命令每個 FC 連接埠只需要幾個命令。檔案系統掛載只不過是 `mount` 只需一個命令、即可在 CLI 上啟動和停止資料庫和 ASM 。如果在切換之前、磁碟區和檔案系統並未在災難恢復站台上使用、則無需設定 `dr-force- nvfail` 在磁碟區上。

使用虛擬化作業系統進行容錯移轉

資料庫環境的容錯移轉可延伸至包含作業系統本身。理論上、此容錯移轉可以使用開機 LUN 來完成、但通常是使用虛擬化的作業系統來完成。此程序類似於下列步驟：

1. 強制 MetroCluster 進行重新操作
2. 裝載託管資料庫伺服器虛擬機器的資料存放區
3. 啟動虛擬機器
4. 手動啟動資料庫、或將虛擬機器設定為自動啟動資料庫

例如、 ESX 叢集可以跨越站台。在發生災難時、虛擬機器可在移至災難恢復站台後上線。只要主控虛擬化資料庫伺服器的資料存放區在災難發生時並未使用、就不需要設定 `dr-force- nvfail` 在相關的磁碟區上。

Oracle 資料庫、 **MetroCluster** 和 **NVFAIL**

NVFAIL 是 ONTAP 中的一般資料完整性功能、其設計可讓資料庫發揮最大的資料完整性保護。

本節將進一步說明基本的 ONTAP NVFAIL 、以涵蓋 MetroCluster 專屬主題。

使用 MetroCluster 時、寫入必須登入至少一個其他控制器的本機 NVRAM 和 NVRAM 、才能被確認。此方法可確保硬體故障或停電不會導致機內 I/O 遺失如果本機 NVRAM 故障或連線至其他節點失敗、則資料將不再鏡射。

如果本機 NVRAM 回報錯誤、節點會關機。當使用 HA 配對時、此關機會導致容錯移轉至合作夥伴控制器。使用 MetroCluster 時、行為取決於所選的整體組態、但可能會導致自動容錯移轉至遠端記事。無論如何、由於發生故障的控制器尚未確認寫入作業、因此不會遺失任何資料。

站台對站台連線故障會封鎖 NVRAM 複寫至遠端節點、這種情況更為複雜。寫入不再複寫到遠端節點、因此如果控制器發生災難性錯誤、可能會導致資料遺失。更重要的是、在這些情況下、嘗試容錯移轉至其他節點會導致資料遺失。

控制因素是 NVRAM 是否同步。如果 NVRAM 已同步、則節點對節點容錯移轉可安全地繼續進行、而不會有資料遺失的風險。在 MetroCluster 組態中、如果 NVRAM 和基礎 Aggregate plex 同步、則在不遺失資料的情況下繼續進行轉換是安全的。

除非強制進行容錯移轉或切換、否則 ONTAP 不允許在資料不同步時進行容錯移轉或切換。以這種方式強制變更條件、即表示資料可能會留在原始控制器中、而且資料遺失是可以接受的。

如果強制進行容錯移轉或切換、則資料庫特別容易遭到毀損、因為資料庫會在磁碟上保留較大的內部資料快取。如果發生強制容錯移轉或切換、先前確認的變更將會有效捨棄。儲存陣列的內容會有效地及時向後跳轉、而且資

料庫快取的狀態不再反映磁碟上資料的狀態。

為了保護應用程式不受這種情況影響、 ONTAP 允許設定磁碟區、以針對 NVRAM 故障提供特殊保護。觸發時、此保護機制會導致磁碟區進入稱為 NVFAIL 的狀態。此狀態會導致 I/O 錯誤、導致應用程式關機、使其不使用過時的資料。資料不應遺失、因為儲存系統上仍有任何已確認的寫入資料、而資料庫則應在記錄中顯示任何已認可的交易資料。

通常的後續步驟是讓系統管理員在手動將 LUN 和磁碟區重新上線之前、先完全關閉主機。雖然這些步驟可能涉及一些工作、但這種方法是確保資料完整性的最安全方法。並非所有資料都需要這項保護、因此 NVFAIL 行為可依每個磁碟區設定。

手動強制 **NVFAIL**

最安全的選項是透過指定來強制轉換跨站台散佈的應用程式叢集（包括 VMware 、 Oracle RAC 及其他）`-force-nvfail-all` 在命令列。此選項可作為緊急措施使用、以確保所有快取資料均已清除。如果主機使用的儲存資源原本位於災難性站台上、則會收到 I/O 錯誤或過時的檔案處理 (`ESTALE`）錯誤。Oracle 資料庫當機、檔案系統可能完全離線、或切換至唯讀模式。

在完成重新操作之後、 `in-nvfailed-state` 需要清除旗標、且 LUN 必須置於線上。完成此活動後、即可重新啟動資料庫。這些工作可以自動化、以降低 RTO 。

dr-force-nvfail

作為一般安全措施、請設定 `dr-force-nvfail` 在所有可能在正常作業期間從遠端站台存取的磁碟區上加上旗標、表示這些磁碟區是在容錯移轉之前使用的活動。此設定的結果是、選取的遠端磁碟區在進入時無法使用 `in-nvfailed-state` 在進行重新操作時。在完成重新操作之後、 `in-nvfailed-state` 旗標必須清除、且 LUN 必須置於線上。這些活動完成後、即可重新啟動應用程式。這些工作可以自動化、以降低 RTO 。

結果就像使用 `-force-nvfail-all` 手動切換的旗標。然而、受影響的磁碟區數量可能僅限於必須受到保護的磁碟區、不受具有過時快取的應用程式或作業系統的影響。

對於不使用的環境、有兩項關鍵需求 `dr-force-nvfail` 在應用程式磁碟區上：

- 在主站台遺失後、強制進行的重新操作不得超過 30 秒。
- 在維護工作期間、或是在 SyncMirror 叢或 NVRAM 複寫不同步的任何其他情況下、切勿進行切入。第一項需求可以透過使用已設定為在站台故障 30 秒內執行轉換的斷路器軟體來達成。這並不表示切入作業必須在偵測站台故障的 30 秒內執行。這表示、如果站台確認運作已過 30 秒、就不再安全地強制進行轉換。

第二項需求可在已知 MetroCluster 組態不同步時停用所有自動切換功能、以部分滿足。更好的選擇是擁有可監控 NVRAM 複寫和 SyncMirror 叢的健全狀況的斷路器解決方案。如果叢集未完全同步、則斷路器不應觸發切入。

NetApp MCTB 軟體無法監控同步處理狀態、因此當 MetroCluster 因任何原因而未同步時、應該停用同步處理狀態。ClusterLion 確實包含 NVRAM 監控和叢監視功能、除非 MetroCluster 系統確認完全同步、否則可將其設定為不觸發切入。

MetroCluster 上的 **Oracle** 單一執行個體

如前所述、 MetroCluster 系統的存在並不一定會新增或變更任何操作資料庫的最佳實務做法。目前在客戶 MetroCluster 系統上執行的大多數資料庫都是單一執行個體、並遵循 Oracle on ONTAP 文件中的建議。

使用預先設定的作業系統進行容錯移轉

SyncMirror 在災難恢復站點上提供資料的同步複本、但要讓資料可用、則需要作業系統和相關應用程式。基本自動化可大幅改善整體環境的容錯移轉時間。例如 Veritas Cluster Server （ VCS ）等叢集件產品通常用於在站台之間建立叢集、而且在許多情況下、容錯移轉程序可以使用簡單的指令碼來驅動。

如果主節點遺失、叢集軟體（或指令碼）會設定為在替代站台上線資料庫。其中一個選項是建立預先設定為 NFS 或 SAN 資源的備用伺服器、以供組成資料庫。如果主站台發生故障、叢集軟體或指令碼替代方案會執行類似下列的一系列動作：

1. 強制 MetroCluster 進行重新操作
2. 執行 FC LUN 探索（僅限 SAN ）
3. 掛載檔案系統和 / 或掛載 ASM 磁碟群組
4. 啟動資料庫

此方法的主要需求是在遠端站台上執行作業系統。它必須預先設定 Oracle 二進位檔、這也表示 Oracle 修補等工作必須在主要站台和待命站台上執行。或者、 Oracle 二進位檔可鏡射至遠端站台、並在宣告災難時掛載。

實際的啟動程序很簡單。LUN 探索等命令每個 FC 連接埠只需要幾個命令。檔案系統掛載只不過是 `mount` 只需一個命令、即可在 CLI 上啟動和停止資料庫和 ASM 。如果在切換之前、磁碟區和檔案系統並未在災難恢復站台上使用、則無需設定 `dr-force- nvfail` 在磁碟區上。

使用虛擬化作業系統進行容錯移轉

資料庫環境的容錯移轉可延伸至包含作業系統本身。理論上、此容錯移轉可以使用開機 LUN 來完成、但通常是使用虛擬化的作業系統來完成。此程序類似於下列步驟：

1. 強制 MetroCluster 進行重新操作
2. 裝載託管資料庫伺服器虛擬機器的資料存放區
3. 啟動虛擬機器
4. 手動啟動資料庫、或將虛擬機器設定為自動啟動資料庫、例如 ESX 叢集可能跨越站台。在發生災難時、虛擬機器可在移至災難恢復站台後上線。只要主控虛擬化資料庫伺服器的資料存放區在災難發生時並未使用、就不需要設定 `dr-force- nvfail` 在相關的磁碟區上。

MetroCluster 上的延伸 **Oracle RAC**

許多客戶透過在各個站台之間延伸 Oracle RAC 叢集來最佳化 RTO 、進而實現完全主動式的組態。整體設計變得更複雜、因為它必須包含 Oracle RAC 的仲裁管理。此外、從兩個站台存取資料、這表示強制轉換可能會導致使用過時的資料複本。

雖然兩個站台上都有資料複本、但只有目前擁有 Aggregate 的控制器才能提供資料。因此、使用擴充的 RAC 叢集時、遠端節點必須透過站台對站台連線來執行 I/O 。結果會增加 I/O 延遲、但這種延遲通常不是問題。RAC 互連網路也必須延伸至站台、這表示無論如何都需要高速、低延遲的網路。如果增加的延遲確實造成問題、則叢集可以主動被動方式運作。接著、需要將 I/O 密集作業導向至擁有該集合體的控制器本機的 RAC 節點。然後、遠端節點會執行較輕的 I/O 作業、或純粹作為暖待機伺服器使用。

如果需要雙主動式擴充 RAC 、則應考慮使用 ASM 鏡像來取代 MetroCluster 。ASM 鏡像可讓您偏好資料的特定複本。因此、可以內建擴充 RAC 叢集、讓所有讀取作業都在本機進行。讀取 I/O 永遠不會跨越網站、因此可提供最低的延遲。所有寫入活動仍必須傳輸站台間連線、但任何同步鏡射解決方案都無法避免此類流量。

如果開機 LUN （包括虛擬化開機磁碟）與 Oracle RAC 搭配使用、請使用 `misscount` 可能需要變更參數。如需 RAC 逾時參數的詳細資訊、請參閱 "Oracle RAC 搭配 ONTAP"。

雙站台組態

雙站台擴充 RAC 組態可提供雙主動式資料庫服務、可在不中斷營運的情況下、在許多（但並非全部）災難案例中順利運作。

RAC 投票檔案

在 MetroCluster 上部署擴充 RAC 時、首先應考慮仲裁管理。Oracle RAC 有兩種機制可管理仲裁：磁碟心跳和網路心跳。磁碟心跳會使用投票檔案來監控儲存設備存取。只要基礎儲存系統提供 HA 功能、單一投票資源就足以搭配單一站台 RAC 組態。

在早期版本的 Oracle 中、投票檔案會放置在實體儲存裝置上、但在目前版本的 Oracle 中、投票檔案會儲存在 ASM 磁碟群組中。

NFS 支援 Oracle RAC 。在網格安裝程序期間、會建立一組 ASM 程序、將用於網格檔案的 NFS 位置顯示為 ASM 磁碟群組。此程序對終端使用者來說幾乎透明、安裝完成後不需要持續進行 ASM 管理。

雙站台組態的第一項需求是確保每個站台都能以保證不中斷災難恢復程序的方式存取超過半數的投票檔案。這項工作在投票檔案儲存在 ASM 磁碟群組之前很簡單、但現在管理員必須瞭解 ASM 備援的基本原則。

ASM 磁碟群組有三種備援選項 `external`、 `normal`和 `high`。換句話說、非鏡射、鏡射和 3 向鏡射。名為的較新選項 `Flex` 也可以使用、但很少使用。備援裝置的備援層級和放置位置可控制故障情況發生的情況。例如：

- 將投票檔案放在上 `diskgroup` 與 `external` 如果站台間連線中斷、備援資源保證可收回一個站台。
- 將投票檔案放在上 `diskgroup` 與 `normal` 如果站台間連線中斷、每個站台只有一個 ASM 磁碟的備援功能可確保兩個站台的節點遷離、因為兩個站台都不會有大部分的仲裁。
- 將投票檔案放在上 `diskgroup` 與 `high` 當兩個站台都可以運作且彼此可連線時、一個站台上有兩個磁碟和另一個站台上的單一磁碟的備援功能可讓雙主動式作業運作。但是、如果單一磁碟站台與網路隔離、則該站台會被逐出。

RAC 網路心跳

Oracle RAC 網路活動訊號可監控叢集互連中的節點可連性。若要保留在叢集中、節點必須能夠連絡其他節點的一半以上。在雙站台架構中、此需求會為 RAC 節點數建立下列選項：

- 如果每個站台放置相同數量的節點、則會在網路連線中斷時、在某個站台上造成遷離。
- 在另一個站台上放置 N 個節點、在另一個站台上放置 N+1 個節點、可確保站台之間的連線中斷、導致站台的網路仲裁中剩餘節點數量較多、而節點移出數量較少的站台。

在 Oracle 12cR2 之前、無法控制哪一方在站台遺失時會發生遷離。當每個站台的節點數量相等時、會由主要節點控制遷離、這通常是第一個要開機的 RAC 節點。

Oracle 12cR2 引進節點加權功能。這項功能可讓管理員更有效地控制 Oracle 如何解決大腦分裂狀況。例如、下列命令可設定 RAC 中特定節點的偏好設定：

```
[root@host-a ~]# /grid/bin/crsctl set server css_critical yes
CRS-4416: Server attribute 'CSS_CRITICAL' successfully changed. Restart
Oracle High Availability Services for new value to take effect.
```

重新啟動 Oracle 高可用度服務後、組態如下所示：

```
[root@host-a lib]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
```

節點 `host-a` 現已指定為關鍵伺服器。如果兩個 RAC 節點是隔離的、 `host-a` 生存、和 `host-b` 被逐出。

如需完整詳細資料、請參閱 Oracle 白皮書《 Oracle Clusterware 12c Release 2 Technical Overview 》。」

對於 12cR2 之前的 Oracle RAC 版本、可透過檢查 CRS 記錄來識別主節點、如下所示：

```
[root@host-a ~]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
 [root@host-a ~]# grep -i 'master node' /grid/diag/crs/host-
a/crs/trace/crsd.trc
2017-05-04 04:46:12.261525 :   CRSSE:2130671360: {1:16377:2} Master Change
Event; New Master Node ID:1 This Node's ID:1
2017-05-04 05:01:24.979716 :   CRSSE:2031576832: {1:13237:2} Master Change
Event; New Master Node ID:2 This Node's ID:1
2017-05-04 05:11:22.995707 :   CRSSE:2031576832: {1:13237:221} Master
Change Event; New Master Node ID:1 This Node's ID:1
2017-05-04 05:28:25.797860 :   CRSSE:3336529664: {1:8557:2} Master Change
Event; New Master Node ID:2 This Node's ID:1
```

此記錄表示主節點為 2 和節點 `host-a` ID 為 1。這意味著 `host-a` 不是主節點。您可以使用命令確認主節點的身分識別 `olsnodes -n`。

```
[root@host-a ~]# /grid/bin/olsnodes -n
host-a  1
host-b  2
```

識別碼為的節點 2 是 host-b，這是主節點。在每個站台上節點數量相等的組態中、站台為 host-b 如果這兩組因為任何原因而失去網路連線、則該站台仍可生存。

識別主節點的記錄項目可能會超出系統的使用期限。在這種情況下、可以使用 Oracle 叢集登錄（ OCR ）備份的時間戳記。

```
[root@host-a ~]#  /grid/bin/ocrconfig -showbackup
host-b     2017/05/05 05:39:53     /grid/cdata/host-cluster/backup00.ocr
0
host-b     2017/05/05 01:39:53     /grid/cdata/host-cluster/backup01.ocr
0
host-b     2017/05/04 21:39:52     /grid/cdata/host-cluster/backup02.ocr
0
host-a     2017/05/04 02:05:36     /grid/cdata/host-cluster/day.ocr     0
host-a     2017/04/22 02:05:17     /grid/cdata/host-cluster/week.ocr     0
```

此範例顯示主節點是 host-b。它也表示主節點的變更來源 host-a 至 host-b 5 月 4 日下午 2 ： 05 至 21 ： 39 之間。這種識別主節點的方法只有在也檢查了 CRS 記錄檔時才安全使用、因為主節點可能自上一次的 OCR 備份後變更。如果發生此變更、則應可在 OCR 記錄中看到。

大多數客戶選擇單一投票磁碟群組來服務整個環境、以及每個站台上相同數量的 RAC 節點。磁碟群組應放置在包含資料庫的網站上。結果是連線中斷會導致遠端站台被逐出。遠端站台將不再擁有仲裁、也無法存取資料庫檔案、但本機站台會繼續如常運作。連線恢復後、遠端執行個體即可重新上線。

發生災難時、需要進行轉換、才能讓資料庫檔案和投票磁碟群組在正常運作的網站上線。如果災難允許 AUSO 觸發切換、則不會觸發 NVFAIL 、因為已知叢集處於同步狀態、且儲存資源正常上線。AUSO 是一項非常快速的作業、應在完成之前完成 disktimeout 期間過期。

由於只有兩個站台、因此無法使用任何類型的自動外部中斷軟體、這表示強制切換必須是手動操作。

三站台組態

擴充的 RAC 叢集可更輕鬆地建構三個站台。裝載 MetroCluster 系統每一半的兩個站台也支援資料庫工作負載、而第三個站台則是資料庫和 MetroCluster 系統的斷路器。Oracle tiebreaker 組態可能只需在第三站台上放置用於投票的 ASM 磁碟群組成員、也可能在第三站台上加入作業執行個體、以確保 RAC 叢集中有奇數個節點。

有關在擴展 RAC 配置中使用 NFS 的重要信息，請參閱 Oracle 文檔中的“ quorum failure group （仲裁故障組）”。總而言之、 NFS 掛載選項可能需要修改以包含軟選項、以確保主仲裁資源所在的第三站台連線中斷、不會使主 Oracle 伺服器或 Oracle RAC 程序掛起。

SnapMirror 主動同步

SnapMirror 主動同步可針對個別 Oracle 資料庫和應用程式環境、啟用選擇性的 RPO =0 同步鏡射。

SnapMirror 主動同步基本上是 SAN 的強化 SnapMirror 功能、可讓主機從主控 LUN 的系統以及主控其複本的系統存取 LUN 。

SnapMirror 主動式同步和 SnapMirror 同步可共用複寫引擎、不過 SnapMirror 主動式同步則包含其他功能、例如企業應用程式的透明應用程式容錯移轉和容錯回復。

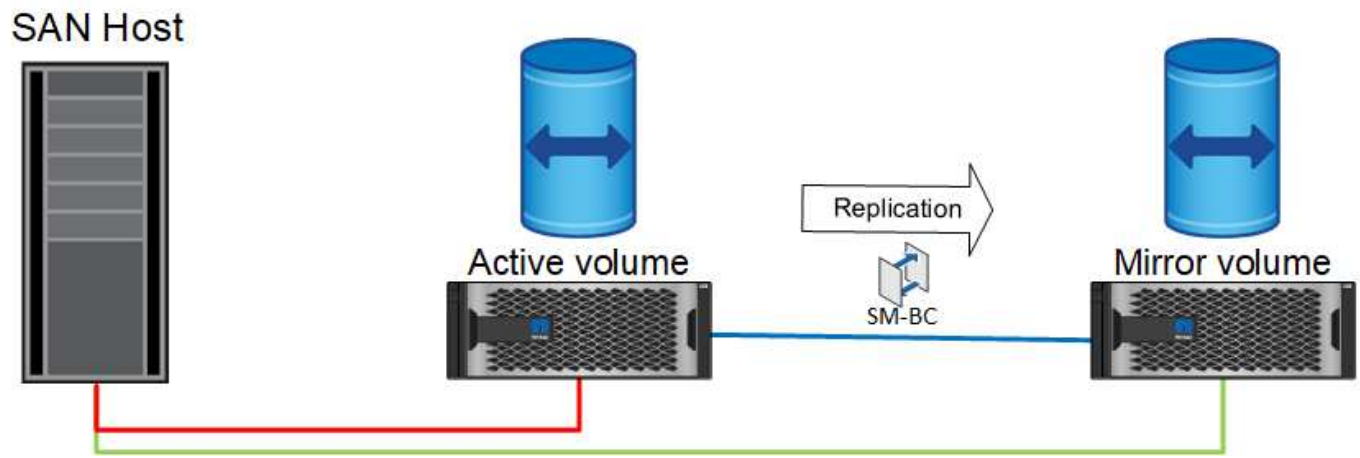
實際上、它的運作方式類似於精細的 MetroCluster 版本、可針對個別工作負載啟用選擇性且精細的 RPO = 0 同步複寫。低階路徑行為與 MetroCluster 非常不同、但主機觀點的最終結果卻相似。

路徑存取

透過 SnapMirror 主動式同步、儲存裝置可從主要和遠端儲存陣列的主機作業系統中看到。路徑是透過非對稱式邏輯單元存取（ ALUA ）進行管理、這是一種業界標準傳輸協定、用於識別儲存系統與主機之間的最佳化路徑。

存取 I/O 最短的裝置路徑被視為主動 / 最佳化路徑、其餘路徑則被視為主動 / 非最佳化路徑。

SnapMirror 主動同步關係是位於不同叢集上的一對 SVM 之間的關係。兩個 SVM 都能提供資料、但 ALUA 會優先使用 SVM 、而 SVM 目前擁有 LUN 所在磁碟機的所有權。透過 SnapMirror 主動同步互連、將 IO 透過代理至遠端 SVM 。



同步複寫

在正常作業中、遠端複本始終為 RPO = 0 同步複本、但有一個例外。如果無法複寫資料、 SnapMirror 主動式同步將會釋放複寫資料和恢復服務 IO 的需求。如果客戶認為複寫連結遺失的情況近乎災難、或是不想在無法複寫資料時停止業務作業、則偏好使用此選項。

儲存硬體

與其他儲存災難恢復解決方案不同、 SnapMirror 主動式同步提供非對稱式平台靈活度。每個站台的硬體不一定相同。此功能可讓您調整支援 SnapMirror 主動同步所用硬體的大小。如果遠端儲存系統需要支援完整的正式作業工作負載、則它可以與主要站台相同、但如果災難導致 I/O 減少、遠端站台上較小的系統可能會更具成本效益。

中間器**ONTAP**

ONTAP Mediator 是從 NetApp 支援下載的軟體應用程式。Mediator 可自動執行主與遠端站台儲存叢集的容錯移轉作業。它可以部署在內部部署或雲端的小型虛擬機器（ VM ）上。設定之後、它會成為第三個站台、用來監控兩個站台的容錯移轉案例。

Oracle 資料庫容錯移轉搭配 **SnapMirror** 主動式同步

在 SnapMirror 主動式同步上託管 Oracle 資料庫的主要原因、是在計畫性和非計畫性儲存事件期間提供透明的容錯移轉。

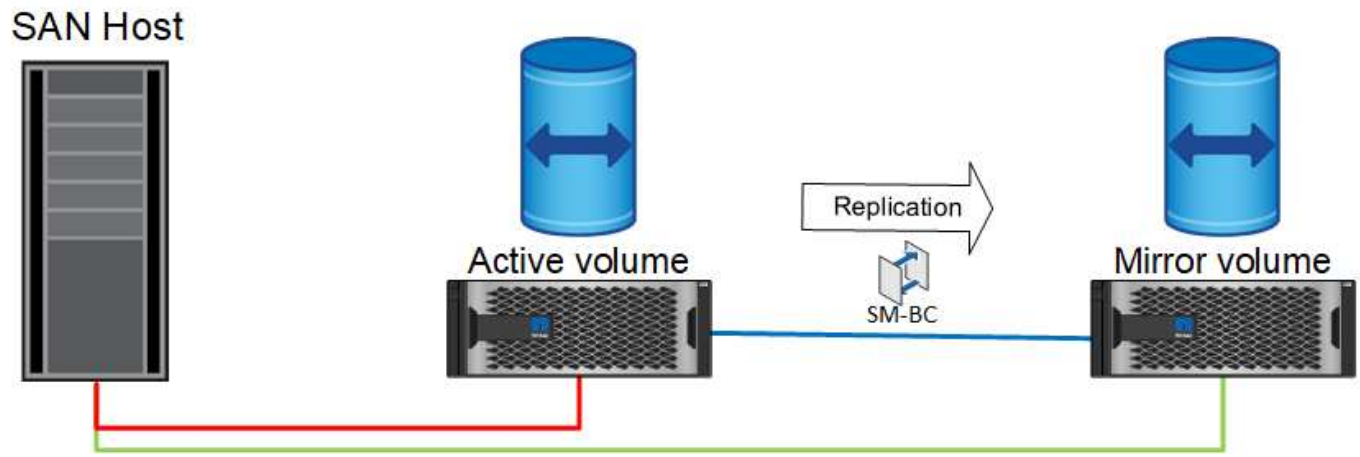
SnapMirror 主動式同步支援兩種儲存容錯移轉作業：計畫性和非計畫性、運作方式略有不同。系統管理員會手動啟動計畫性容錯移轉、以快速切換至遠端站台、而非計畫性容錯移轉則由第三站台的協調員自動啟動。計畫性容錯移轉的主要目的是執行漸進式修補與升級、執行災難恢復測試、或是採用正式的原則、在一年內在站台之間切換作業、以證明完整的主動式同步功能。

下圖顯示正常、容錯移轉及容錯回復作業期間的情況。為了便於說明、它們描述了複寫的 LUN 。在實際的 SnapMirror 主動式同步組態中、複寫是以磁碟區為基礎、其中每個磁碟區都包含一個或多個 LUN 、但為了讓圖片更簡單、磁碟區層已經移除。

正常運作

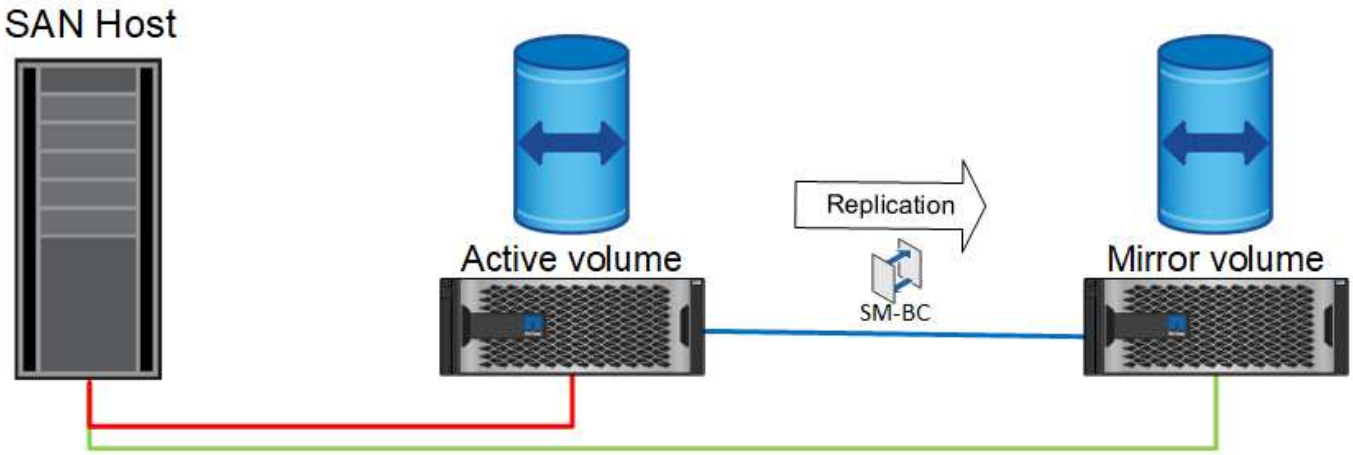
在正常作業中、可以從本機或遠端複本存取 LUN 。紅色線表示 ALUA 所通告的最佳化路徑、結果應該是 IO 優先傳送至此路徑。

綠線是一條活動路徑，但由於該路徑上的 IO 需要通過 SnapMirror 活動同步路徑傳遞，因此會產生更多延遲。額外的延遲時間取決於用於 SnapMirror 主動同步的站台之間互連的速度。



故障

如果主動鏡像複本因為計畫性或非計畫性容錯移轉而無法使用、則顯然無法再使用。然而、遠端系統已擁有通往遠端站台的同步複本和 SAN 路徑。遠端系統能夠為該 LUN 提供 IO 服務。

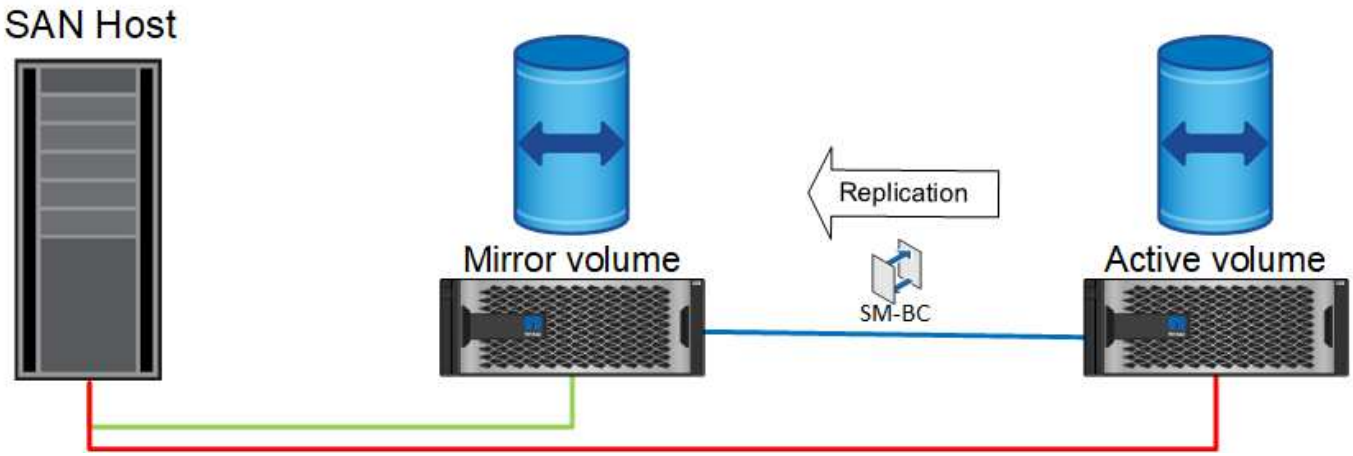


容錯移轉

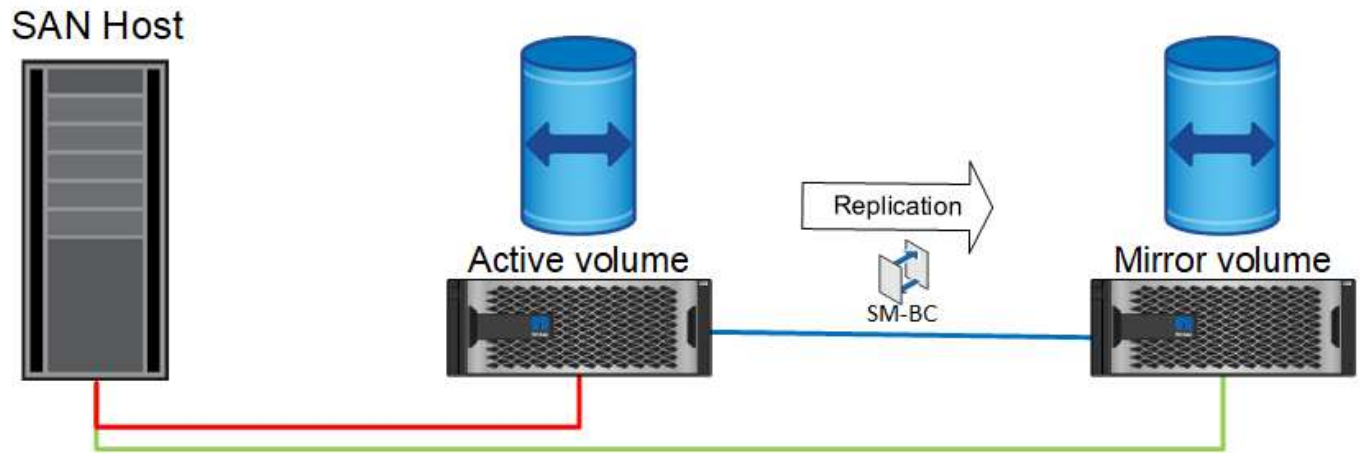
容錯移轉會導致遠端複本變成使用中複本。路徑會從「 Active 」變更為「 Active/Optimized 」、 IO 也會持續提供服務、不會遺失資料。

修復

一旦來源系統恢復服務、 SnapMirror 主動式同步就能重新同步複寫、但會執行其他方向。現在的組態基本上與起點相同、只是主動鏡射站台已經翻轉。



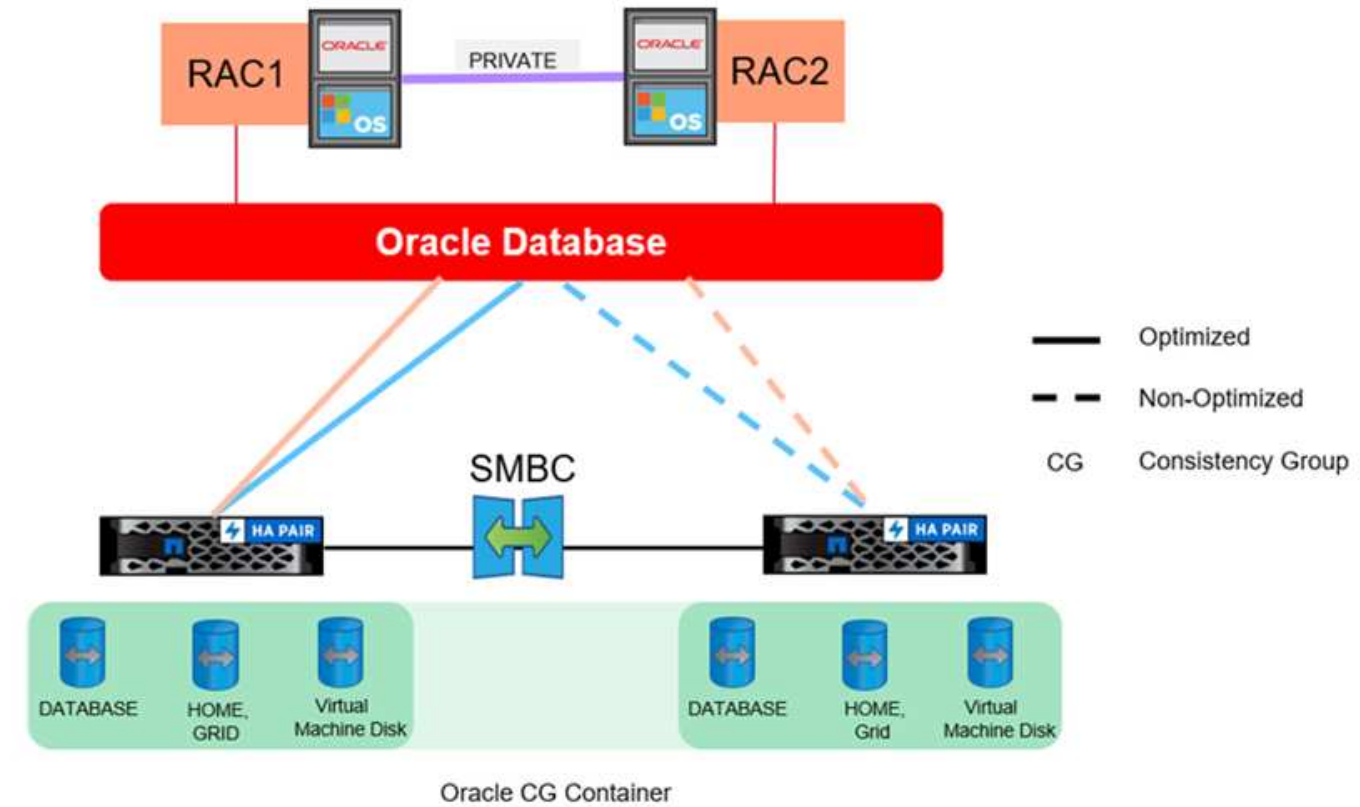
如果需要、管理員可以執行容錯回復、並將 LUN 的作用中複本移回原始控制器。



單一執行個體 **Oracle** 資料庫搭配 **SnapMirror** 主動式同步

下圖顯示一個簡單的部署模式、您可以將儲存裝置分區或從主要和遠端儲存叢集連線至 Oracle 資料庫。

Oracle 僅在主要系統上設定。此模式可因應儲存端災難時的無縫儲存容錯移轉、在沒有任何應用程式停機的情況下、不會造成資料遺失。然而、此模型無法在站台故障期間提供資料庫環境的高可用度。這類架構對於想要尋找零資料遺失解決方案、且儲存服務可用度高的客戶而言非常實用、但請接受、資料庫叢集的整體遺失需要手動作業。



這種方法也能節省 Oracle 授權成本。在遠端站台上預先設定 Oracle 資料庫節點時、所有核心都必須根據大多數 Oracle 授權合約獲得授權。如果安裝 Oracle 資料庫伺服器和掛載仍在運作的資料複本所需的時間所造成的延遲是可以接受的、則此設計可能非常具成本效益。

Oracle RAC 搭配 SnapMirror 主動式同步

SnapMirror 主動式同步可針對資料集複寫提供精細的控制、例如負載平衡或個別應用程式容錯移轉。整體架構看起來像延伸 RAC 叢集、但有些資料庫是專供特定站台使用、整體負載則分散。

例如、您可以建置一個 Oracle RAC 叢集、主控六個個別資料庫。其中三個資料庫的儲存設備主要託管於站台 A 、其他三個資料庫的儲存設備則裝載於站台 B此組態可將跨網站流量降至最低、以確保最佳效能。此外、應用程式也會設定為使用儲存系統本機的資料庫執行個體、並使用作用中路徑。如此可將 RAC 互連流量降至最低。最後、這項整體設計可確保所有運算資源均能平均使用。隨著工作負載改變、資料庫可以在站台之間選擇性地來回容錯、以確保負載均勻。

除了精細度之外、使用 SnapMirror 主動式 SynCare 的 Oracle RAC 基本原則和選項與相同 "MetroCluster 上的 Oracle RAC"

Oracle 資料庫和 SnapMirror 主動式同步失敗案例

有多種 SnapMirror 主動式同步（ SM-AS ）故障案例、每種案例的結果各不相同。

案例	結果
複寫連結失敗	中介程序可辨識這種分割腦案例、並在保留主複本的節點上恢復 I/O 。當站台之間的連線恢復上線時、替代站台會執行自動重新同步。
主要站台儲存設備故障	自動非計畫性容錯移轉是由 Mediator 啟動。 無 I/O 中斷。
遠端站台儲存設備故障	沒有 I/O 中斷。由於網路造成同步複寫中斷、主機確認其擁有者是合法的 I/O 服務者（共識）、因此暫時暫停。因此、 I/O 暫停數秒、然後 I/O 就會恢復。 網站上線時會自動重新同步。
Mediator 或 Mediator 與儲存陣列之間的連結遺失	I/O 會繼續並與遠端叢集保持同步、但如果沒有 Mediator 、則無法自動進行非計畫性 / 計畫性容錯移轉和容錯回復。
HA 叢集中的其中一個儲存控制器遺失	HA 叢集中的合作夥伴節點會嘗試接管（ n ）。如果接管失敗、 Mediator 會注意到儲存設備中的兩個節點都已關閉、並自動執行非計畫性容錯移轉至遠端叢集。
磁碟遺失	IO 會持續連續三次發生磁碟故障。這是 RAID-TEC 的一部分。

案例	結果
在一般部署中遺失整個站台	故障站台上的伺服器顯然無法再使用。支援叢集的應用程式可設定為在兩個站台上執行、並在其他站台上繼續作業、不過大多數的應用程式都需要類似於 SM 要求中介者的第三站台斷路器。 如果沒有應用程式層級的叢集、應用程式就必須在正常運作的站台上啟動。這會影響可用性、但會保留 RPO =0 。不會遺失任何資料。

Oracle 資料庫移轉

將 Oracle 資料庫移轉至 ONTAP 儲存系統

利用新儲存平台的功能、有一項不可避免的需求；資料必須放在新的儲存系統上。ONTAP 讓移轉程序變得簡單、包括 ONTAP 到 ONTAP 的移轉與升級、外部 LUN 匯入、以及直接使用主機作業系統或 Oracle 資料庫軟體的程序。

本文件取代先前發佈的技術報告 _TR-4534 ：將 Oracle 資料庫移轉至 NetApp 儲存系統

若是新的資料庫專案、這並不是問題、因為資料庫和應用程式環境都已建置就緒。然而、移轉對於業務中斷、完成移轉所需的時間、所需的技能組合、以及將風險降至最低等方面、都會帶來特殊挑戰。

指令碼

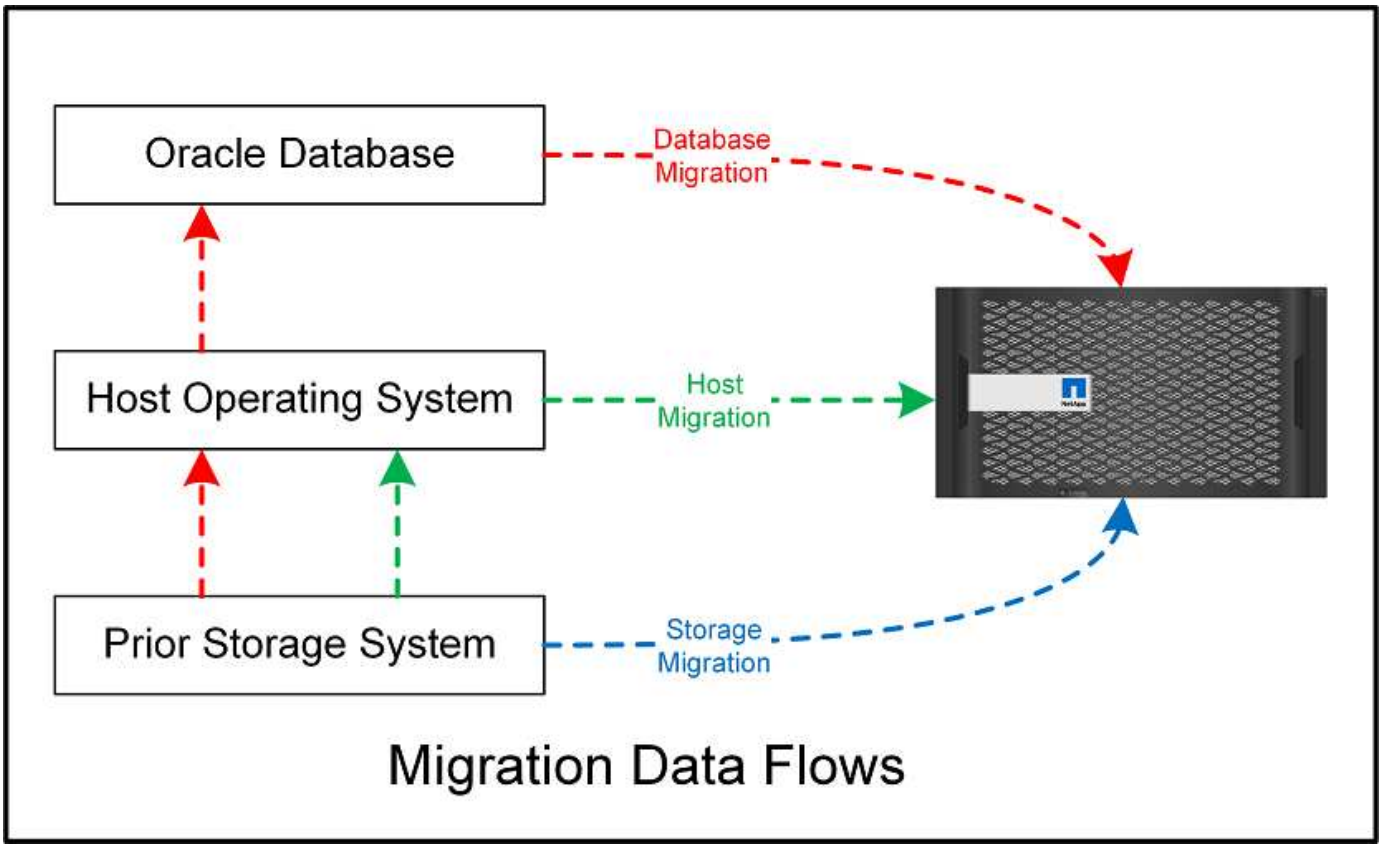
本文件提供範例指令碼。這些指令碼提供自動化移轉各個層面的範例方法、以降低使用者錯誤的機率。這些指令碼可降低 IT 人員對於移轉作業的整體需求、並加速整體程序。這些指令碼都是從 NetApp 專業服務和 NetApp 合作夥伴執行的實際移轉專案中擷取而來。本文件中會顯示使用範例。

Oracle 資料庫移轉規劃

Oracle 資料移轉可在下列三個層級中進行：資料庫、主機或儲存陣列。

不同之處在於整體解決方案的哪個元件負責移動資料：資料庫、主機作業系統或儲存系統。

下圖顯示移轉層級和資料流的範例。在資料庫層級移轉的情況下、資料會從原始儲存系統透過主機和資料庫層移至新環境。主機層級移轉類似、但資料不會通過應用程式層、而是使用主機程序寫入新位置。最後、隨著儲存層級移轉、 NetApp FAS 系統等陣列也會負責資料移動。



資料庫層級移轉通常指透過待命資料庫傳送 Oracle 記錄以完成 Oracle 層級的移轉。主機層級的移轉是使用主機作業系統組態的原生功能來執行。此組態包括檔案複製作業、使用 CP 、 tar 和 Oracle Recovery Manager （ RMAN ）等命令、或使用邏輯 Volume Manager （ LVM ）來重新定位檔案系統的基礎位元組。Oracle 自動儲存管理（ ASM ）被歸類為主機層級功能、因為它的執行層級低於資料庫應用程式層級。ASM 取代主機上一般的邏輯磁碟區管理程式。最後、資料可以在儲存陣列層級上移轉、也就是在作業系統層級之下。

規劃考量

最佳移轉選項取決於多種因素、包括要移轉的環境規模、避免停機的需求、以及執行移轉所需的整體工作。大型資料庫顯然需要更多時間和精力來進行移轉、但這類移轉的複雜度極低。小型資料庫可以快速移轉、但如果要移轉數千個資料庫、則工作規模可能會造成複雜問題。最後、資料庫越大、業務關鍵的可能性就越大、因此需要將停機時間降至最低、同時保留一條後端路徑。

此處將討論規劃移轉策略的一些考量事項。

資料大小

要移轉的資料庫大小顯然會影響移轉規劃、但大小不一定會影響轉換時間。當必須移轉大量資料時、主要考量的是頻寬。複製作業通常是以高效率的連續 I/O 來執行保守估計、假設複製作業的可用網路頻寬使用率為 50% 。例如、 8GB FC 連接埠理論上可傳輸約 800MBps 。假設使用率為 50% 、則資料庫的複製速度約為 400Mbps 。因此、 10TB 資料庫可在此速率下在大約七小時內複製。

遠距離移轉通常需要更具創意的方法、例如中所說明的記錄傳送程序 "線上資料檔案移動"。遠距 IP 網路在任何接近 LAN 或 SAN 速度的地方、都很少會有頻寬。在某種情況下、 NetApp 協助 220 TB 資料庫進行遠距移轉、而且產生的歸檔記錄率非常高。所選的資料傳輸方法是每天運送磁帶、因為這種方法提供最大可能的頻寬。

資料庫計數

在許多情況下、移動大量資料的問題並不是資料大小、而是支援資料庫的組態複雜度。只要知道必須移轉 50TB 的資料庫、就無法獲得足夠的資訊。它可以是單一 50TB 關鍵任務資料庫、 4 、 000 個舊資料庫的集合、或是正式作業和非正式作業資料的混合。在某些情況下、大部分資料是由來源資料庫的複本所組成。這些複本完全不需要移轉、因為它們可以輕鬆地重新建立、特別是當新架構設計為使用 NetApp FlexClone Volume 時。

在移轉規劃方面、您必須瞭解範圍內有多少資料庫、以及它們的優先順序。隨著資料庫數量的增加、偏好的移轉選項在堆疊中會較低和較低。例如、在 RMAN 和短暫停機的情況下、複製單一資料庫可能很容易。這是主機層級的複寫。

如果有 50 個資料庫、可能會更容易避免設定新的檔案系統結構來接收 RMAN 複本、而改為將資料移到適當位置。此程序可透過利用主機型 LVM 移轉來將資料從舊 LUN 重新放置到新的 LUN 來完成。這樣做會將責任從資料庫管理員（ DBA ）團隊移轉至作業系統團隊、因此資料會以透明方式移轉至資料庫。檔案系統組態不變。

最後、如果必須移轉 200 部伺服器上的 500 個資料庫、則可使用 ONTAP 外部 LUN 匯入（ FLI ）功能等儲存型選項來執行 LUN 的直接移轉。

重新架構需求

一般而言、必須變更資料庫檔案配置才能運用新儲存陣列的功能、但情況並非總是如此。例如、 EF 系列 All Flash 陣列的功能主要是針對 SAN 效能和 SAN 可靠性。在大多數情況下、資料庫可以移轉至 EF 系列陣列、而無需特別考量資料配置。唯一的要求是高 IOPS 、低延遲和強大的可靠性。雖然 RAID 組態或動態磁碟集區等因素都有最佳實務做法、但 EF 系列專案很少需要對整體儲存架構進行任何重大變更、才能運用這些功能。

相反地、移轉至 ONTAP 通常需要更多考量資料庫配置、以確保最終組態能提供最大價值。ONTAP 本身可為資料庫環境提供許多功能、即使沒有任何特定的架構工作也沒問題。最重要的是、當目前的硬體達到使用壽命時、它能夠不中斷地移轉至新的硬體。一般而言、移轉至 ONTAP 是您最後需要執行的移轉作業。隨後的硬體即會就地升級、資料也不會中斷營運地移轉至新媒體。

有了一些規劃、就能獲得更多效益。使用快照時最重要的考量事項。快照是執行近乎即時的備份、還原及複製作業的基礎。作為快照功能的範例、已知最大的用途是在 6 個控制器上的約 250 個 LUN 上執行單一資料庫 996TB 。此資料庫可在 2 分鐘內備份、 2 分鐘內還原、 15 分鐘內複製完成。其他優點包括：能夠在叢集內移動資料、以因應工作負載的變化、以及應用服務品質（ QoS ）控制來在多資料庫環境中提供良好且一致的效能。

QoS 控制、資料重新配置、快照和複製等技術幾乎可在任何組態中運作。不過、一般需要考慮一些方法才能最大化效益。在某些情況下、資料庫儲存配置可能需要變更設計、以最大化對新儲存陣列的投資。這類設計變更可能會影響移轉策略、因為主機型或儲存型移轉會複寫原始資料配置。完成移轉並提供針對 ONTAP 最佳化的資料配置、可能需要其他步驟。中所示的程序 "Oracle 移轉程序概述" 稍後、我們將示範一些方法、讓您不只是移轉資料庫、還能以最少的心力將資料庫移轉至最佳的最終配置。

轉換時間

應決定轉換期間允許的服務中斷上限。假設整個移轉程序會造成中斷、這是常見的錯誤。許多工作都可以在服務中斷開始之前完成、許多選項都能在不中斷或中斷的情況下完成移轉。即使無法避免中斷、您仍必須定義允許的服務中斷上限、因為轉換時間的持續時間因程序而異。

例如、複製 10TB 資料庫通常需要大約七小時才能完成。如果企業需要中斷七小時、檔案複製是輕鬆安全的移轉選項。如果五小時不可接受、則只需簡單的記錄傳送程序（請參閱 "Oracle 記錄傳送"）只需最少的努力就能設定、將轉換時間縮短至約 15 分鐘。在此期間、資料庫管理員可以完成此程序。如果 15 分鐘是不可接受的、則可透過指令碼將最後的轉換程序自動化、將轉換時間縮短至幾分鐘。您可以隨時加快移轉速度、但這樣做的代價是時間和精力。轉換時間目標應以企業可接受的內容為基礎。

回溯路徑

沒有移轉作業完全沒有風險。即使技術運作正常、使用者也永遠可能發生錯誤。與所選移轉路徑相關的風險、必須與移轉失敗的後果一併考量。例如、 Oracle ASM 的透明線上儲存移轉功能是其重要功能之一、這是最可靠的方法之一。然而、資料正以這種方法進行無法扭轉的複製。在 ASM 發生問題的極不可能發生的事件中、沒有簡單的回傳路徑。唯一的選項是還原原始環境、或使用 ASM 將移轉回復至原始 LUN 。如果系統能夠執行此類作業、則可在原始儲存系統上執行快照類型備份、將風險降至最低、但不會消除。

排練

某些移轉程序必須在執行前經過完整驗證。移轉和排練轉換程序是關鍵任務資料庫的常見要求、移轉必須成功、停機時間必須降至最低。此外、使用者驗收測試通常是移轉後工作的一部分、只有在完成這些測試之後、才能將整個系統恢復正常運作。

如果需要排練、有幾項 ONTAP 功能可以讓流程更輕鬆。特別是、快照可以重設測試環境、並快速建立資料庫環境的多個具空間效益的複本。

程序

Oracle 移轉程序概述

Oracle 移轉資料庫有許多可用的程序。正確的選擇取決於您的業務需求。

在許多情況下、系統管理員和 DBA 都有自己偏好的方法來重新定位實體磁碟區資料、鏡像和磁碟鏡射、或是利用 Oracle RMAN 來複製資料。

這些程序主要是為不熟悉某些可用選項的 IT 人員提供指引。此外、這些程序還說明每種移轉方法的工作、時間需求和專長類別需求。如此一來、 NetApp 和合作夥伴專業服務或 IT 管理等其他方就能更充分瞭解每個程序的要求。

建立移轉策略沒有單一最佳實務做法。建立計畫需要先瞭解可用度選項、然後選擇最符合業務需求的方法。下圖說明客戶所做的基本考量和典型結論、但並不適用於所有情況。

例如、一個步驟會引發資料庫總大小的問題。下一步取決於資料庫是否大於或小於 1TB 。建議的步驟就是根據一般客戶實務做法提出建議。大多數客戶不會使用 DataGuard 複製小型資料庫、但有些客戶可能會使用。大多數客戶不會因為所需時間而嘗試複製 50TB 資料庫、但有些客戶可能會有足夠大的維護時間來允許這類作業。

您可以找到最適合移轉路徑的考量類型流程圖 "請按這裡"。

線上資料檔案移動

Oracle 12cR1 及更高版本可在資料庫保持連線時移動資料檔案。此外、它還能在不同的檔案系統類型之間運作。例如、資料檔案可從 xfs 檔案系統重新定位至 ASM 。由於需要個別資料檔案移動作業的數量、因此通常不會大規模使用此方法、但這是一個值得考慮的選項、因為較小的資料庫資料檔案數量較少。

此外、單純移動資料檔案是移轉部分現有資料庫的好選項。例如、較不活躍的資料檔案可重新放置到更具成本效益的儲存設備、例如可在物件儲存區中儲存閒置區塊的 FabricPool Volume 。

資料庫層級移轉

資料庫層級的移轉意味著允許資料庫重新配置資料。具體而言、這表示記錄傳送。RMAN 和 ASM 等技術是 Oracle 產品、但為了進行移轉、它們會在主機層級運作、在主機層級複製檔案和管理磁碟區。

記錄傳送

資料庫層級移轉的基礎是 Oracle 歸檔記錄檔、其中包含資料庫變更的記錄檔。歸檔記錄通常是備份與還原策略的一部分。恢復程序從還原資料庫開始、然後重新播放一或多個歸檔記錄檔、將資料庫恢復到所需的狀態。這項相同的基本技術可用於執行移轉、幾乎不會中斷營運。更重要的是、這項技術可在不影響原始資料庫的情況下進行移轉、並保留一條向外移轉的路徑。

移轉程序從將資料庫備份還原至次要伺服器開始。您可以透過多種方式進行、但大多數客戶都會使用正常的備份應用程式來還原資料檔案。資料檔案還原後、使用者便可建立記錄傳送方法。目標是建立由主要資料庫所產生的持續歸檔記錄摘要、並在還原的資料庫上重新播放、使兩者都接近相同的狀態。轉換時間到達時、來源資料庫會完全關閉、最後的歸檔記錄會複製並重新播放、有時則會複製重做記錄。重做記錄也必須列入考量、因為它們可能包含已提交的部分最終交易。

在傳輸和重播這些記錄之後、兩個資料庫彼此之間的一致性。此時、大多數客戶都會執行一些基本測試。如果在移轉過程中發生任何錯誤、則記錄重新執行應會報告錯誤並失敗。還是建議您根據已知的查詢或應用程式導向的活動來執行一些快速測試、以驗證組態是否最佳化。在關閉原始資料庫之前建立一個最終測試表格、以確認該表格是否存在於移轉的資料庫中、這也是常見的做法。此步驟可確保在最終記錄同步期間不會發生任何錯誤。

簡單的記錄傳送移轉作業可針對原始資料庫進行頻外設定、這對關鍵任務資料庫特別有用。來源資料庫不需要變更組態、移轉環境的還原和初始組態也不會影響正式作業。設定記錄傳送之後、它會對正式作業伺服器提出一些 I/O 需求。不過、記錄傳送是由簡單的歸檔記錄循序讀取所組成、這對正式作業資料庫效能不太可能有任何影響。

已證實、記錄傳送對於長期、高變更率的移轉專案特別有用。在某個案例中、單一 220TB 資料庫已移轉至距離約 500 英哩的新位置。變更率極高、安全性限制無法使用網路連線。記錄傳送是使用磁帶和快遞業者來執行。原始資料庫的複本最初是使用下列程序來還原。然後、快遞業者每週寄送記錄、直到最後一組磁帶送達時、記錄才會套用至複本資料庫。

Oracle DataGuard

在某些情況下、保證提供完整的 DataGuard 環境。使用術語 DataGuard 來指稱任何記錄傳送或待命資料庫組態是不正確的。Oracle DataGuard 是管理資料庫複寫的全方位架構、但它不是複寫技術。在移轉工作中、完整的 DataGuard 環境的主要優點是能從一個資料庫透明切換到另一個資料庫。如果發現問題（例如新環境的效能或網路連線問題）、 Dataguard 也能將切換回原始資料庫。完整設定的 DataGuard 環境不僅需要設定資料庫層、也需要設定應用程式、以便應用程式能夠偵測主要資料庫位置的變更。一般而言、不需要使用 DataGuard 來完成移轉、但有些客戶內部擁有豐富的 DataGuard 專業知識、而且已經仰賴它來進行移轉工作。

重新架構

如前所述、運用儲存陣列的進階功能有時需要變更資料庫配置。此外、從 ASM 移轉至 NFS 檔案系統等儲存傳輸協定變更、也必然會改變檔案系統配置。

記錄傳送方法（包括 DataGuard ）的主要優點之一是複寫目的地不需要與來源相符。使用記錄傳送方法從 ASM 移轉至一般檔案系統時沒有任何問題、反之亦然。您可以在目的地變更資料檔案的精確配置、以最佳化易插拔資料庫（ PDB ）技術的使用、或選擇性地設定特定檔案的 QoS 控制。換句話說、以記錄傳送為基礎的移轉程序可讓您輕鬆安全地最佳化資料庫儲存配置。

伺服器資源

資料庫層級移轉的一項限制是需要第二部伺服器。使用第二部伺服器的方法有兩種：

1. 您可以使用第二部伺服器作為資料庫的永久新主目錄。
2. 您可以使用第二部伺服器做為暫存伺服器。在資料移轉至新儲存陣列完成並測試之後、 LUN 或 NFS 檔案系

統會從暫存伺服器中斷連線、然後重新連線至原始伺服器。

第一個選項是最簡單的、但在需要非常強大伺服器的大型環境中、使用它可能不可行。第二個選項需要額外的工作、才能將檔案系統重新放置回原始位置。這可以是一項簡單的作業、使用 NFS 做為儲存傳輸協定、因為檔案系統可以從暫存伺服器卸載、然後重新掛載到原始伺服器上。

區塊型檔案系統需要額外的工作、才能更新 FC 分區或 iSCSI 啟動器。對於大多數邏輯磁碟區管理員（包括 ASM ）、 LUN 會在原始伺服器上可用後自動偵測並上線。不過、某些檔案系統和 LVM 實作可能需要更多工作才能匯出和匯入資料。確切的程序可能會有所不同、但通常很容易建立簡單且可重複的程序、以完成移轉並將資料重新存放在原始伺服器上。

雖然可以在單一伺服器環境中設定記錄傳送和複寫資料庫、但新執行個體必須具有不同的處理程序 SID 、才能重新播放記錄。您可以使用不同的 SID 、在不同的處理序識別碼集下暫時開啟資料庫、稍後再變更。然而、這樣做可能會導致許多複雜的管理活動、並使資料庫環境面臨使用者錯誤的風險。

主機層級移轉

在主機層級移轉資料是指使用主機作業系統和相關公用程式來完成移轉。此程序包括複製資料的任何公用程式、包括 Oracle RMAN 和 Oracle ASM 。

資料複製

不應低估簡單複製作業的價值。現代化的網路基礎架構可以以每秒 GB 的速度來移動資料、而檔案複製作業則是以高效率的連續讀寫 I/O 為基礎相較於記錄傳送、主機複本作業無法避免造成更多中斷、但移轉不只是資料移動而已。通常包括網路變更、資料庫重新啟動時間和移轉後測試。

複製資料所需的實際時間可能並不重要。此外、複製作業會保留保證的回傳路徑、因為原始資料不會受到影響。如果在移轉過程中遇到任何問題、可以重新啟動原始資料的原始檔案系統。

重新建立平台

重組是指 CPU 類型的變更。當資料庫從傳統的 Solaris 、 AIX 或 HP-UX 平台移轉至 x86 Linux 時、由於 CPU 架構的變更、資料必須重新格式化。SPARC 、 IA64 和 Power CPU 稱為 Big endian 處理器、而 x86 和 x86_64 架構則稱為小 endian 。因此、 Oracle 資料檔案中的某些資料會根據使用中的處理器而有不同的訂購方式。

傳統上、客戶都使用 DataPump 跨平台複寫資料。datapump 是一種公用程式、可建立特殊類型的邏輯資料匯出、以便更快地匯入目的地資料庫。因為它會建立資料的邏輯複本、所以 DataPump 會將處理器位準的相依性留在背後。有些客戶仍使用資料平台來重新建立平台、但 Oracle 11g 提供更快速的選項：跨平台可攜式表格空間。這項進階功能可將資料表空間轉換成不同的 endian 格式。這是一種實體轉型、效能優於 DataPump 匯出、它必須將實體位元組轉換為邏輯資料、然後再轉換回實體位元組。

關於 DataPump 和可攜式資料表空間的完整討論不在 NetApp 文件的範圍之內、但 NetApp 根據我們協助客戶移轉至具有新 CPU 架構的新儲存陣列記錄的經驗、提供一些建議：

- 如果使用 DataPump 、則應在測試環境中測量完成移轉所需的時間。客戶有時會對完成移轉所需的時間感到驚訝。這種非預期的額外停機可能會造成中斷。
- 許多客戶誤以為跨平台可攜式資料表空間不需要資料轉換。當使用具有不同序位元組的 CPU 時、會使用 RMAN `convert` 必須事先對資料檔案執行作業。這不是即時操作。在某些情況下、轉換程序可以透過在不同資料檔案上執行多個執行緒來加速、但無法避免轉換程序。

邏輯 **Volume Manager** 導向的移轉

LVMS 的運作方式是將一組或多個 LUN 拆分為一般稱為擴充的小型單元。然後將擴充集區用作建立邏輯磁碟區的來源、這些邏輯磁碟區基本上是虛擬化的。此虛擬化層以各種方式提供價值：

- 邏輯磁碟區可以使用從多個 LUN 擷取的範圍。在邏輯磁碟區上建立檔案系統時、它可以使用所有 LUN 的完整效能功能。此外、它也能提升磁碟區群組中所有 LUN 的平均載入速度、提供更可預測的效能。
- 您可以新增邏輯磁碟區、並在某些情況下移除範圍、以調整其大小。在邏輯磁碟區上調整檔案系統大小通常不會中斷營運。
- 透過移動基礎範圍、邏輯磁碟區可以不中斷地移轉。

使用 LVM 移轉的運作方式有兩種：移動範圍或鏡射 / 去除範圍。LVM 移轉使用高效率的大型區塊連續 I/O 、而且很少會造成任何效能問題。如果這確實是問題、通常有節流 I/O 速率的選項。如此可增加完成移轉所需的時間、同時減輕主機和儲存系統的 I/O 負擔。

鏡射與鏡射

某些 Volume 管理程式（例如 AIX LVM ）可讓使用者指定每個範圍的複本數量、並控制裝載每個複本的裝置。移轉作業是透過取得現有的邏輯磁碟區、將基礎範圍鏡射到新磁碟區、等待複本同步、然後丟棄舊複本來完成。如果需要返回路徑、可以在放置鏡射複本之前建立原始資料的快照。或者、您也可以在強制刪除內含的鏡像複本之前、暫時關閉伺服器以遮罩原始 LUN 。這樣做會在資料的原始位置保留可恢復的資料複本。

擴展移轉

幾乎所有的 Volume 管理程式都允許移轉擴充、有時也有多個選項。例如、某些 Volume 管理程式可讓管理員將特定邏輯磁碟區的個別擴充區從舊儲存區重新定位到新儲存區。Volume 管理程式（例如 Linux LVM2 ）提供 `pvmove` 命令、可將指定 LUN 裝置上的所有延伸重新定位至新 LUN 。移除舊 LUN 之後、即可將其移除。

> 作業的主要風險是從組態中移除舊的、未使用的 LUN 。變更 FC 分區和移除過時的 LUN 裝置時、必須格外小心。

Oracle 自動儲存管理

Oracle ASM 是結合邏輯 Volume Manager 與檔案系統的產品。在較高層級、 Oracle ASM 會將 LUN 集合起來、分成小的分配單元、並將其呈現為稱為 ASM 磁碟群組的單一磁碟區。ASM 也能透過設定備援層級來鏡射磁碟群組。磁碟區可以是無鏡射（外部備援）、鏡射（正常備援）或三向鏡射（高備援）。設定備援層級時、請務必謹慎、因為建立後無法變更。

ASM 也提供檔案系統功能。雖然檔案系統無法直接從主機看到、但 Oracle 資料庫仍可在 ASM 磁碟群組上建立、移動及刪除檔案與目錄。此外、您也可以使用 asmcmd 公用程式來瀏覽結構。

與其他 LVM 實作一樣、 Oracle ASM 也會在所有可用 LUN 之間、對每個檔案的 I/O 進行分拆和負載平衡、以最佳化 I/O 效能。其次、基礎擴充可重新定位、以便同時調整 ASM 磁碟群組的大小和移轉。Oracle ASM 會透過重新平衡作業來自動化程序。新的 LUN 會新增至 ASM 磁碟群組、而舊的 LUN 會被丟棄、這會觸發磁碟群組中的磁碟區重新配置及後續刪除已清空的 LUN 。此程序是最獲證實的移轉方法之一、而 ASM 提供透明移轉的可靠性、可能是最重要的功能。

> 由於 Oracle ASM 的鏡射層級是固定的、因此無法搭配鏡射和鏡射移轉方法使用。

儲存層級移轉是指在應用程式和作業系統層級以下執行移轉。過去、這有時是指使用專門的裝置來複製網路層級的 LUN 、但現在這些功能在 ONTAP 中是原生的。

SnapMirror

使用 NetApp SnapMirror 資料複寫軟體、幾乎可以通用地從 NetApp 系統之間移轉資料庫。此程序包括為要移轉的磁碟區設定鏡射關係、允許它們進行同步處理、然後等待轉換時間。當來源資料庫到達時、即會關閉、執行最後一個鏡像更新、而且鏡像也會中斷。然後、複本磁碟區就可以開始使用、方法是掛載包含的 NFS 檔案系統目錄、或是探索包含的 LUN 並啟動資料庫。

在單一 ONTAP 叢集中重新放置磁碟區並不視為移轉作業、而是例行作業 `volume move` 營運。SnapMirror 用作叢集中的資料複寫引擎。此程序完全自動化。當磁碟區的屬性（例如 LUN 對應或 NFS 匯出權限）與磁碟區本身一起移動時、無需執行其他移轉步驟。重新配置不會中斷主機作業。在某些情況下、必須更新網路存取、以確保以最有效率的方式存取新重新部署的資料、但這些工作也不會中斷營運。

外部 **LUN** 匯入（ **FLI** ）

FLI 是一項功能、可讓執行 8.3 或更高版本的 Data ONTAP 系統從另一個儲存陣列移轉現有 LUN 。此程序很簡單： ONTAP 系統會分區到現有的儲存陣列、就像是任何其他 SAN 主機一樣。然後 Data ONTAP 控制所需的舊版 LUN 、並移轉基礎資料。此外、匯入程序會在資料移轉時使用新 Volume 的效率設定、也就是說、資料可以在移轉過程中內嵌進行壓縮及刪除重複資料。

Data ONTAP 8.3 中首次實作的 FLI 僅允許離線移轉。這是非常快速的傳輸、但仍表示在移轉完成之前、 LUN 資料無法使用。線上移轉是在 Data ONTAP 8.3.1 中推出。這類移轉可讓 ONTAP 在傳輸過程中提供 LUN 資料、將中斷情形減至最低。當主機重新分區以透過 ONTAP 使用 LUN 時、會發生短暫的中斷。不過、一旦進行這些變更、資料就會再次存取、並在整個移轉程序中保持可存取的狀態。

讀取 I/O 會透過 ONTAP 代理、直到複製作業完成為止、而寫入 I/O 會同步寫入外部和 ONTAP LUN 。這兩個 LUN 複本會以這種方式保持同步、直到系統管理員執行完整的轉換程式來釋放外部 LUN 、而不再複寫寫入內容。

FLI 的設計可與 FC 搭配使用、但如果您想要變更為 iSCSI 、則可在移轉完成後、輕鬆將移轉的 LUN 重新對應為 iSCSI LUN 。

FLI 的功能包括自動對齊偵測與調整。在這種情況下、「對齊」一詞是指 LUN 裝置上的分割區。最佳效能需要將 I/O 與 4K 區塊對齊。如果分割區的偏移量不是 4K 的倍數、效能就會受到影響。

第二個對齊層面無法透過調整分割區偏移（檔案系統區塊大小）來修正。例如、 ZFS 檔案系統通常預設為 512 位元組的內部區塊大小。其他使用 AIX 的客戶偶爾會建立具有 512 或 1 、 024 位元組區塊大小的 JFS2 檔案系統。雖然檔案系統可能會與 4K 邊界對齊、但在該檔案系統中建立的檔案不會受到影響、效能也會受到影響。

在此情況下不應使用 FLI 。雖然資料在移轉後仍可存取、但結果是檔案系統的效能嚴重限制。一般而言、任何支援 ONTAP 上隨機覆寫工作負載的檔案系統、都應該使用 4K 區塊大小。這主要適用於資料庫資料檔案和 VDI 部署等工作負載。區塊大小可使用相關的主機作業系統命令來識別。

例如、在 AIX 上、可以使用檢視區塊大小 `lsfs -q`。使用 Linux 、 `xfs_info` 和 `tune2fs` 可用於 `xfs` 和 `ext3/ext4`。與 `zfs`、命令是 `zdb -C`。

控制區塊大小的參數為 `ashift` 而且通常預設值為 9 ，即 2^9 或 512 位元組。為了獲得最佳效能 `ashift` 值必須為 12 （ 2^12=4K ）。此值是在創建 zpool 時設置的，不能更改，這意味着使用的數據 zpool `ashift` 除 12 個以外、應將資料複製到新建立的 zPool 、以進行移轉。

Oracle ASM 沒有基本區塊大小。唯一的要求是必須正確對齊 ASM 磁碟所在的磁碟分割區。

7-Mode Transition Tool

7-Mode Transition Tool （ 7MTT ）是一種自動化公用程式、用於將大型 7-Mode 組態移轉至 ONTAP 。大多數資料庫客戶發現其他方法都比較容易、部分原因是他們通常會依資料庫來移轉環境資料庫、而非重新配置整個儲存設備佔用空間。此外、資料庫通常只是較大型儲存環境的一部分。因此、資料庫通常會個別移轉、其餘的環境則可以使用 7MTT 進行移轉。

有少數客戶擁有專為複雜資料庫環境設計的儲存系統。這些環境可能包含許多磁碟區、快照和許多組態詳細資料、例如匯出權限、 LUN 啟動器群組、使用者權限和輕量型目錄存取傳輸協定組態。在這種情況下、 7MTT 的自動化功能可簡化移轉作業。

7MTT 可在下列兩種模式中的其中一種運作：

- * 複製型轉換（ CBT ） * 7MTT 搭配 CBT 、可從新環境中現有的 7 模式系統設定 SnapMirror 磁碟區。資料同步後、 7MTT 會協調轉換程序。
- * 複製 - 自由轉換（ CFT ）。 * 採用 CFT 的 7MTT 是根據現有 7-Mode 磁碟櫃的原位轉換而定。不會複製任何資料、也可以重複使用現有的磁碟櫃。保留現有的資料保護與儲存效率組態。

這兩種選項的主要差異在於：無複製轉換是一種非常有效的方法、其中所有連接至原始 7-Mode HA 配對的磁碟櫃都必須重新放置到新環境中。沒有選項可以移動一部分機櫃。複製型方法可讓選取的磁碟區移動。此外、由於可重新儲存磁碟櫃和轉換中繼資料所需的連結、因此也可能會有較長的轉換時間、且無需複製。根據現場經驗、 NetApp 建議允許 1 小時重新配置及重新配置磁碟櫃、 15 分鐘至 2 小時的中繼資料轉換時間。

Oracle 資料檔案移轉

單一命令即可移動個別的 Oracle 資料檔案。

例如、下列命令會將資料檔案 IOPST.dbf 從檔案系統中移出 `/oradata2` 至檔案系統 `/oradata3`。

```
SQL> alter database move datafile  '/oradata2/NTAP/IOPS002.dbf' to
'/oradata3/NTAP/IOPS002.dbf';
Database altered.
```

使用此方法移動資料檔案可能會很慢、但通常不應產生足夠的 I/O 、而會干擾日常資料庫工作負載。相反地、透過 ASM 重新平衡移轉可以更快執行、但卻會在資料移動時降低整體資料庫的速度。

您可以建立測試資料檔、然後移動資料檔案、輕鬆測量移動資料檔案所需的時間。操作所耗用的時間會記錄在 v$ 工作階段資料中：

```
SQL> set linesize 300;
SQL> select elapsed_seconds||':'||message from v$session_longops;
ELAPSED_SECONDS||':'||MESSAGE
------------------------------------------------------------------------
---------------
351:Online data file move: data file 8: 22548578304 out of 22548578304
bytes done
SQL> select bytes / 1024 / 1024 /1024 as GB from dba_data_files where
FILE_ID = 8;
        GB
----------
        21
```

在此範例中、移動的檔案是 datafile 8 、大小為 21 GB 、需要約 6 分鐘才能移轉。所需時間顯然取決於儲存系統、儲存網路的功能、以及移轉時發生的整體資料庫活動。

透過記錄傳送進行 **Oracle** 資料庫移轉

使用記錄傳送進行移轉的目標是在新位置建立原始資料檔案的複本、然後建立將變更傳送到新環境的方法。

一旦建立、記錄傳送和重播就能自動進行、使複本資料庫與來源保持大致同步。例如、 cron 工作可排程至（ a ）將最近的記錄複製到新位置、並（ b ）每 15 分鐘重播一次。這樣做可在轉換時將中斷次數降至最低、因為必須重播不超過 15 分鐘的歸檔記錄。

以下程序基本上也是資料庫複製作業。所示邏輯類似於 NetApp SnapManager for Oracle （ SMO ）和 NetApp SnapCenter Oracle Plug-in 內的引擎。有些客戶已使用指令碼或 WFA 工作流程中所示的程序來進行自訂的複製作業。雖然此程序比使用 SnapCenter 或 SMO 更為手冊化、但仍可輕鬆撰寫指令碼、而 ONTAP 中的資料管理 API 則可進一步簡化程序。

記錄傳送 **-** 檔案系統至檔案系統

本範例示範如何將名為華夫餅的資料庫從一般檔案系統移轉至位於不同伺服器上的其他一般檔案系統。它也說明 SnapMirror 可用來快速複製資料檔案、但這並不是整體程序不可或缺的一部分。

建立資料庫備份

第一步是建立資料庫備份。具體而言、此程序需要一組資料檔案、可用於歸檔記錄重新執行。

環境

在此範例中、來源資料庫位於 ONTAP 系統上。建立資料庫備份最簡單的方法是使用快照。將資料庫置於熱備份模式幾秒鐘 `snapshot create` 在託管資料檔案的磁碟區上執行作業。

```
SQL> alter database begin backup;
Database altered.
```

```
Cluster01::*> snapshot create -vserver vserver1 -volume jfsc1_oradata
hotbackup
Cluster01::*>
```

```
SQL> alter database end backup;
Database altered.
```

結果是磁碟上的快照稱為 `hotbackup` 包含處於熱備份模式時的資料檔案映像。如果結合適當的歸檔記錄以使資料檔案一致、則此快照中的資料可作為還原或複製的基礎。在這種情況下、它會複寫到新的伺服器。

還原至新環境

現在必須在新環境中還原備份。這可以透過多種方式完成、包括 Oracle RMAN 、從備份應用程式（如 NetBackup ）還原、或是簡單複製置於熱備份模式的資料檔案。

在此範例中、 SnapMirror 用於將快照熱備份複寫到新位置。

1. 建立新的磁碟區以接收快照資料。從初始化鏡像 `jfsc1_oradata` 至 `vol_oradata`。

```
Cluster01::*> volume create -vserver vserver1 -volume vol_oradata
-aggregate data_01 -size 20g -state online -type DP -snapshot-policy
none -policy jfsc3
[Job 833] Job succeeded: Successful
```

```
Cluster01::*> snapmirror initialize -source-path vserver1:jfsc1_oradata
-destination-path vserver1:vol_oradata
Operation is queued: snapmirror initialize of destination
"vserver1:vol_oradata".
Cluster01::*> volume mount -vserver vserver1 -volume vol_oradata
-junction-path /vol_oradata
Cluster01::*>
```

2. 在 SnapMirror 設定狀態後、表示同步已完成、請特別根據所需的快照來更新鏡像。

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields state
source-path             destination-path        state
----------------------- ----------------------- ------------
vserver1:jfsc1_oradata vserver1:vol_oradata SnapMirrored
```

```
Cluster01::*> snapmirror update -destination-path vserver1:vol_oradata
-source-snapshot hotbackup
Operation is queued: snapmirror update of destination
"vserver1:vol_oradata".
```

3. 您可以透過檢視來驗證同步成功與否 `newest-snapshot` 鏡射磁碟區上的欄位。

```
Cluster01::*> snapmirror show -destination-path vserver1:vol_oradata
-fields newest-snapshot
source-path             destination-path        newest-snapshot
----------------------- ----------------------- ---------------
vserver1:jfsc1_oradata vserver1:vol_oradata hotbackup
```

4. 然後鏡射可能會中斷。

```
Cluster01::> snapmirror break -destination-path vserver1:vol_oradata
Operation succeeded: snapmirror break for destination
"vserver1:vol_oradata".
Cluster01::>
```

5. 掛載新的檔案系統。使用區塊型檔案系統時、精確的程序會因使用中的 LVM 而異。必須設定 FC 分區或 iSCSI 連線。建立與 LUN 的連線後、命令如 Linux `pvscan` 可能需要探索哪些磁碟區群組或 LUN 需要正確設定、才能讓 ASM 發現。

   在此範例中、使用簡單的 NFS 檔案系統。此檔案系統可直接掛載。

```
fas8060-nfs1:/vol_oradata        19922944   1639360   18283584   9%
/oradata
fas8060-nfs1:/vol_logs            9961472       128    9961344   1%
/logs
```

建立控制檔建立範本

接下來必須建立控制檔範本。。 `backup controlfile to trace` 命令會建立文字命令以重新建立控制檔。在某些情況下、此功能可用於從備份還原資料庫、而且通常用於執行資料庫複製等工作的指令碼。

1. 以下命令的輸出用於爲遷移的數據庫重新創建控制文件。

```
SQL> alter database backup controlfile to trace as '/tmp/waffle.ctrl';
Database altered.
```

2. 建立控制檔之後、請將檔案複製到新伺服器。

```
[oracle@jfsc3 tmp]$ scp oracle@jfsc1:/tmp/waffle.ctrl /tmp/
oracle@jfsc1's password:
waffle.ctrl                                               100% 5199
5.1KB/s   00:00
```

備份參數檔案

在新環境中也需要一個參數檔。最簡單的方法是從目前的 spfile 或 pfile 建立 pfile 。在此範例中、來源資料庫使用的是 spfile 。

```
SQL> create pfile='/tmp/waffle.tmp.pfile' from spfile;
File created.
```

建立 **oratab** 項目

需要建立 oratab 項目、才能正常運作如 oraenv 等公用程式。若要建立 oratab 項目、請完成下列步驟。

```
WAFFLE:/orabin/product/12.1.0/dbhome_1:N
```

準備目錄結構

如果所需目錄尚未存在、您必須建立它們、否則資料庫啟動程序會失敗。若要準備目錄結構、請完成下列最低需求。

```
[oracle@jfsc3 ~]$ . oraenv
ORACLE_SID = [oracle] ? WAFFLE
The Oracle base has been set to /orabin
[oracle@jfsc3 ~]$ cd $ORACLE_BASE
[oracle@jfsc3 orabin]$ cd admin
[oracle@jfsc3 admin]$ mkdir WAFFLE
[oracle@jfsc3 admin]$ cd WAFFLE
[oracle@jfsc3 WAFFLE]$ mkdir adump dpdump pfile scripts xdb_wallet
```

參數檔案更新

1. 若要將參數檔複製到新伺服器、請執行下列命令。預設位置為 `$ORACLE_HOME/dbs` 目錄。在這種情況下、pfile 可以放在任何地方。它只是移轉程序中的中間步驟。

```
[oracle@jfsc3 admin]$ scp oracle@jfsc1:/tmp/waffle.tmp.pfile
$ORACLE_HOME/dbs/waffle.tmp.pfile
oracle@jfsc1's password:
waffle.pfile                                          100%  916
0.9KB/s   00:00
```

1. 視需要編輯檔案。例如、如果歸檔記錄位置已變更、則必須變更 pfile 以反映新位置。在此範例中、只有控制檔正在重新定位、部分是為了在記錄檔和資料檔案系統之間散佈。

```
[root@jfsc1 tmp]# cat waffle.pfile
WAFFLE.__data_transfer_cache_size=0
WAFFLE.__db_cache_size=507510784
WAFFLE.__java_pool_size=4194304
WAFFLE.__large_pool_size=20971520
WAFFLE.__oracle_base='/orabin'#ORACLE_BASE set from environment
WAFFLE.__pga_aggregate_target=268435456
WAFFLE.__sga_target=805306368
WAFFLE.__shared_io_pool_size=29360128
WAFFLE.__shared_pool_size=234881024
WAFFLE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/WAFFLE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/oradata//WAFFLE/control01.ctl','/oradata//WAFFLE/contr
ol02.ctl'
*.control_files='/oradata/WAFFLE/control01.ctl','/logs/WAFFLE/control02.
ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='WAFFLE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=WAFFLEXDB)'
*.log_archive_dest_1='LOCATION=/logs/WAFFLE/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

2. 編輯完成後、請根據此 pfile 建立 spfile 。

```
SQL> create spfile from pfile='waffle.tmp.pfile';
File created.
```

重新建立控制檔

在前一個步驟中、的輸出 `backup controlfile to trace` 已複製到新伺服器。所需輸出的特定部分是 `controlfile recreation` 命令。此資訊可在檔案中標記的區段下找到 `Set #1. NORESETLOGS`。從這條線開始 `create controlfile reuse database` 並應包含這個字 `noresetlogs`。結尾是分號（；）字元。

1. 在此範例程序中、檔案會讀取如下內容。

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/logs/WAFFLE/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/logs/WAFFLE/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/logs/WAFFLE/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

2. 視需要編輯此指令碼、以反映各種檔案的新位置。例如、已知可支援高 I/O 的某些資料檔案、可能會重新導向至高效能儲存層上的檔案系統。在其他情況下、這些變更可能純粹是因為系統管理員的理由、例如在專用磁碟區中隔離指定的 PDB 資料檔案。
3. 在此範例中 `DATAFILE` stanza 保持不變、但重做記錄會移至中的新位置 `/redo` 而非與歸檔登入共用空間 `/logs`。

```
CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
    MAXLOGFILES 16
    MAXLOGMEMBERS 3
    MAXDATAFILES 100
    MAXINSTANCES 8
    MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
  GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
-- STANDBY LOGFILE
DATAFILE
  '/oradata/WAFFLE/system01.dbf',
  '/oradata/WAFFLE/sysaux01.dbf',
  '/oradata/WAFFLE/undotbs01.dbf',
  '/oradata/WAFFLE/users01.dbf'
CHARACTER SET WE8MSWIN1252
;
```

```
SQL> startup nomount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             331353200 bytes
Database Buffers          465567744 bytes
Redo Buffers                5455872 bytes
SQL> CREATE CONTROLFILE REUSE DATABASE "WAFFLE" NORESETLOGS  ARCHIVELOG
  2      MAXLOGFILES 16
  3      MAXLOGMEMBERS 3
  4      MAXDATAFILES 100
  5      MAXINSTANCES 8
  6      MAXLOGHISTORY 292
  7  LOGFILE
  8    GROUP 1 '/redo/redo01.log'  SIZE 50M BLOCKSIZE 512,
  9    GROUP 2 '/redo/redo02.log'  SIZE 50M BLOCKSIZE 512,
 10    GROUP 3 '/redo/redo03.log'  SIZE 50M BLOCKSIZE 512
 11  -- STANDBY LOGFILE
 12  DATAFILE
 13    '/oradata/WAFFLE/system01.dbf',
 14    '/oradata/WAFFLE/sysaux01.dbf',
 15    '/oradata/WAFFLE/undotbs01.dbf',
 16    '/oradata/WAFFLE/users01.dbf'
 17  CHARACTER SET WE8MSWIN1252
 18  ;
Control file created.
SQL>
```

如果有任何檔案放錯位置或參數設定錯誤、就會產生錯誤、指出必須修正的項目。資料庫已掛載、但尚未開啟且無法開啟、因為使用中的資料檔案仍標示為處於熱備份模式。必須先套用歸檔記錄檔、才能使資料庫一致。

初始記錄複寫

為了使資料檔案一致、至少需要執行一項記錄回覆作業。有許多選項可供重播記錄。在某些情況下、原始伺服器上的原始歸檔記錄檔位置可以透過 NFS 共用、而且記錄回覆可以直接完成。在其他情況下、必須複製歸檔記錄。

例如、簡單 `scp` 作業可將所有目前記錄從來源伺服器複製到移轉伺服器：

```
[oracle@jfsc3 arch]$ scp jfsc1:/logs/WAFFLE/arch/* ./
oracle@jfsc1's password:
1_22_912662036.dbf                                          100%   47MB
47.0MB/s   00:01
1_23_912662036.dbf                                          100%   40MB
40.4MB/s   00:00
1_24_912662036.dbf                                          100%   45MB
45.4MB/s   00:00
1_25_912662036.dbf                                          100%   41MB
40.9MB/s   00:01
1_26_912662036.dbf                                          100%   39MB
39.4MB/s   00:00
1_27_912662036.dbf                                          100%   39MB
38.7MB/s   00:00
1_28_912662036.dbf                                          100%   40MB
40.1MB/s   00:01
1_29_912662036.dbf                                          100%   17MB
16.9MB/s   00:00
1_30_912662036.dbf                                          100%  636KB
636.0KB/s   00:00
```

初始記錄重新播放

檔案在歸檔記錄位置後、可以發出命令來重新播放 `recover database until cancel` 接著是回應 `AUTO` 自動重播所有可用的記錄。

```
SQL> recover database until cancel;
ORA-00279: change 382713 generated at 05/24/2016 09:00:54 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_23_912662036.dbf
ORA-00280: change 382713 for thread 1 is in sequence #23
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 405712 generated at 05/24/2016 15:01:05 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_24_912662036.dbf
ORA-00280: change 405712 for thread 1 is in sequence #24
ORA-00278: log file '/logs/WAFFLE/arch/1_23_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
ORA-00278: log file '/logs/WAFFLE/arch/1_30_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_31_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
```

最終的歸檔記錄回覆會回報錯誤、但這是正常現象。記錄會指出這一點 `sqlplus` 正在尋找特定的記錄檔、但找不到該檔案。原因很可能是記錄檔尚未存在。

如果在複製歸檔記錄之前可以關閉來源資料庫、則此步驟只能執行一次。歸檔記錄會複製並重新播放、然後程序會直接繼續進行轉換程序、以複寫重要的重作記錄。

遞增記錄複寫及重新播放

在大多數情況下、移轉作業不會立即執行。移轉程序可能在幾天甚至幾週前完成、這表示記錄必須持續運送至複本資料庫並重新執行。因此、當轉換程式到達時、必須傳輸和重播最少的資料。

這樣做有許多方式可以撰寫指令碼、但其中最受歡迎的方法之一是使用 rsync 、這是通用的檔案複寫公用程式。使用此公用程式最安全的方法是將其設定為常駐程式。例如、 `rsyncd.conf` 下列檔案顯示如何建立名為的資源 `waffle.arch` 使用 Oracle 使用者認證存取、並對應至 `/logs/WAFFLE/arch`。最重要的是、資源設為唯讀、可讀取正式作業資料、但不變更。

```
[root@jfsc1 arch]# cat /etc/rsyncd.conf
[waffle.arch]
   uid=oracle
   gid=dba
   path=/logs/WAFFLE/arch
   read only = true
[root@jfsc1 arch]# rsync --daemon
```

下列命令會將新伺服器的保存檔記錄目的地與 rsync 資源同步 `waffle.arch` 在原始伺服器上。。 `t` 引數 `rsync - potg` 根據時間戳記比較檔案清單、只複製新檔案。此程序提供新伺服器的遞增更新。此命令也可在 cron 中排程為定期執行。

```
[oracle@jfsc3 arch]$ rsync -potg --stats --progress jfsc1::waffle.arch/*
/logs/WAFFLE/arch/
1_31_912662036.dbf
      650240 100%  124.02MB/s    0:00:00 (xfer#1, to-check=8/18)
1_32_912662036.dbf
     4873728 100%  110.67MB/s    0:00:00 (xfer#2, to-check=7/18)
1_33_912662036.dbf
     4088832 100%   50.64MB/s    0:00:00 (xfer#3, to-check=6/18)
1_34_912662036.dbf
     8196096 100%   54.66MB/s    0:00:00 (xfer#4, to-check=5/18)
1_35_912662036.dbf
    19376128 100%   57.75MB/s    0:00:00 (xfer#5, to-check=4/18)
1_36_912662036.dbf
       71680 100%  201.15kB/s    0:00:00 (xfer#6, to-check=3/18)
1_37_912662036.dbf
     1144320 100%    3.06MB/s    0:00:00 (xfer#7, to-check=2/18)
1_38_912662036.dbf
    35757568 100%   63.74MB/s    0:00:00 (xfer#8, to-check=1/18)
1_39_912662036.dbf
      984576 100%    1.63MB/s    0:00:00 (xfer#9, to-check=0/18)
Number of files: 18
Number of files transferred: 9
Total file size: 399653376 bytes
Total transferred file size: 75143168 bytes
Literal data: 75143168 bytes
Matched data: 0 bytes
File list size: 474
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 204
Total bytes received: 75153219
sent 204 bytes  received 75153219 bytes  150306846.00 bytes/sec
total size is 399653376  speedup is 5.32
```

在收到記錄之後、必須重新播放記錄。前面的範例顯示使用 sqlplus 來手動執行 `recover database until cancel`，這是一種可以輕鬆自動化的程序。此處顯示的範例使用中所述的指令碼 "重播資料庫上的記錄"。指令碼會接受指定需要重新執行作業之資料庫的引數。如此可在多資料庫移轉作業中使用相同的指令碼。

```
[oracle@jfsc3 logs]$ ./replay.logs.pl WAFFLE
ORACLE_SID = [WAFFLE] ? The Oracle base remains unchanged with value
/orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu May 26 10:47:16 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 713874 generated at 05/26/2016 04:26:43 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_31_912662036.dbf
ORA-00280: change 713874 for thread 1 is in sequence #31
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 814256 generated at 05/26/2016 04:52:30 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_32_912662036.dbf
ORA-00280: change 814256 for thread 1 is in sequence #32
ORA-00278: log file '/logs/WAFFLE/arch/1_31_912662036.dbf' no longer
needed for
this recovery
ORA-00279: change 814780 generated at 05/26/2016 04:53:04 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_33_912662036.dbf
ORA-00280: change 814780 for thread 1 is in sequence #33
ORA-00278: log file '/logs/WAFFLE/arch/1_32_912662036.dbf' no longer
needed for
this recovery
...
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
ORA-00278: log file '/logs/WAFFLE/arch/1_39_912662036.dbf' no longer
needed for
this recovery
ORA-00308: cannot open archived log '/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

轉換

當您準備好切換至新環境時、必須執行最後一次同步、其中包括歸檔記錄和重做記錄。如果尚未知道原始的重做記錄位置、可以如下所示識別：

```
SQL> select member from v$logfile;
MEMBER
--------------------------------------------------------------------------------
------
/logs/WAFFLE/redo/redo01.log
/logs/WAFFLE/redo/redo02.log
/logs/WAFFLE/redo/redo03.log
```

1. 關閉來源資料庫。
2. 使用所需的方法、在新伺服器上執行歸檔記錄的最後一次同步。
3. 來源重做記錄檔必須複製到新伺服器。在此範例中、重做記錄會重新定位到新的目錄 `/redo`。

   ```
   [oracle@jfsc3 logs]$ scp jfsc1:/logs/WAFFLE/redo/* /redo/
   oracle@jfsc1's password:
   redo01.log
   100%   50MB  50.0MB/s   00:01
   redo02.log
   100%   50MB  50.0MB/s   00:00
   redo03.log
   100%   50MB  50.0MB/s   00:00
   ```

4. 在此階段、新的資料庫環境包含所有必要的檔案、使其與來源完全相同。歸檔記錄必須最後重播一次。

```
SQL> recover database until cancel;
ORA-00279: change 1120099 generated at 05/26/2016 09:59:21 needed for
thread 1
ORA-00289: suggestion : /logs/WAFFLE/arch/1_40_912662036.dbf
ORA-00280: change 1120099 for thread 1 is in sequence #40
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
ORA-00308: cannot open archived log
'/logs/WAFFLE/arch/1_40_912662036.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
```

5. 完成後、必須重新執行重作記錄。如果出現此訊息 `Media recovery complete` 會傳回、程序成功、資料庫會同步、並可開啟。

```
SQL> recover database;
Media recovery complete.
SQL> alter database open;
Database altered.
```

記錄傳送 **- ASM** 至檔案系統

本範例說明如何使用 Oracle RMAN 移轉資料庫。這與先前的檔案系統傳送檔案系統記錄檔範例非常類似、但主機看不到 ASM 上的檔案。唯一用於移轉位於 ASM 裝置上的資料的選項是重新放置 ASM LUN 、或使用 Oracle RMAN 來執行複製作業。

雖然 RMAN 是從 Oracle ASM 複製檔案的必要條件、但 RMAN 的使用不限於 ASM 。RMAN 可用於從任何類型的儲存設備移轉至任何其他類型。

此範例顯示將名為 pake 的資料庫從 ASM 儲存設備重新放置到位於路徑上不同伺服器上的一般檔案系統 `/oradata` 和 `/logs`。

建立資料庫備份

第一步是建立要移轉到替代伺服器的資料庫備份。由於來源使用 Oracle ASM 、因此必須使用 RMAN 。簡單的 RMAN 備份可執行如下。此方法會建立標記備份、可在稍後的程序中由 RMAN 輕鬆識別。

第一個命令定義備份的目的地類型和要使用的位置。第二個只會啟動資料檔案的備份。

```
RMAN> configure channel device type disk format '/rman/pancake/%U';
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT   '/rman/pancake/%U';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT   '/rman/pancake/%U';
new RMAN configuration parameters are successfully stored
RMAN> backup database tag 'ONTAP_MIGRATION';
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=251 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/PANCAKE/system01.dbf
input datafile file number=00002 name=+ASM0/PANCAKE/sysaux01.dbf
input datafile file number=00003 name=+ASM0/PANCAKE/undotbs101.dbf
input datafile file number=00004 name=+ASM0/PANCAKE/users01.dbf
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/1gr6c161_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/1hr6c164_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

備份控制檔

稍後的程序中需要備份控制檔 `duplicate database` 營運。

```
RMAN> backup current controlfile format '/rman/pancake/ctrl.bkp';
Starting backup at 24-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
channel ORA_DISK_1: starting piece 1 at 24-MAY-16
channel ORA_DISK_1: finished piece 1 at 24-MAY-16
piece handle=/rman/pancake/ctrl.bkp tag=TAG20160524T032651 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

備份參數檔案

在新環境中也需要一個參數檔。最簡單的方法是從目前的 spfile 或 pfile 建立 pfile 。在此範例中、來源資料庫使用 spfile 。

```
RMAN> create pfile='/rman/pancake/pfile' from spfile;
Statement processed
```

ASM 檔案重新命名指令碼

移動資料庫時、控制檔中目前定義的數個檔案位置會變更。下列指令碼會建立 RMAN 指令碼、以簡化程序。此範例顯示的資料庫資料檔案數量極少、但資料庫通常包含數百個甚至數千個資料檔案。

此指令碼位於 "ASM 至檔案系統名稱轉換" 它有兩件事。

首先、它會建立一個參數、重新定義稱為的重做記錄位置 `log_file_name_convert`。基本上是交替欄位清單。第一個欄位是目前重做記錄檔的位置、第二個欄位是新伺服器上的位置。然後重複該模式。

第二個功能是提供資料檔案重新命名的範本。指令碼會循環瀏覽資料檔案、擷取名稱和檔案編號資訊、並將其格式化為 RMAN 指令碼。然後、它會對暫存檔案執行相同的操作。結果是一個簡單的 RMAN 指令碼、可視需要加以編輯、以確保檔案還原至所需的位置。

```
SQL> @/rman/mk.rename.scripts.sql
Parameters for log file conversion:
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/NEW_PATH/redo01.log','+ASM0/PANCAKE/redo02.log',
'/NEW_PATH/redo02.log','+ASM0/PANCAKE/redo03.log', '/NEW_PATH/redo03.log'
rman duplication script:
run
{
set newname for datafile 1 to '+ASM0/PANCAKE/system01.dbf';
set newname for datafile 2 to '+ASM0/PANCAKE/sysaux01.dbf';
set newname for datafile 3 to '+ASM0/PANCAKE/undotbs101.dbf';
set newname for datafile 4 to '+ASM0/PANCAKE/users01.dbf';
set newname for tempfile 1 to '+ASM0/PANCAKE/temp01.dbf';
duplicate target database for standby backup location INSERT_PATH_HERE;
}
PL/SQL procedure successfully completed.
```

擷取此畫面的輸出。。 `log_file_name_convert` 參數會如下所述放置在 pfile 中。RMAN 資料檔案重新命名和重複指令碼必須據此編輯、才能將資料檔案放置在所需的位置。在此範例中、所有的項目都放在中 `/oradata/pancake`。

```
run
{
set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
set newname for datafile 4 to '/oradata/pancake/users.dbf';
set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
duplicate target database for standby backup location '/rman/pancake';
}
```

準備目錄結構

指令碼幾乎可以執行、但首先必須有目錄結構。如果所需目錄尚未存在、則必須建立這些目錄、否則資料庫啟動程序會失敗。以下範例反映最低需求。

```
[oracle@jfsc2 ~]$ mkdir /oradata/pancake
[oracle@jfsc2 ~]$ mkdir /logs/pancake
[oracle@jfsc2 ~]$ cd /orabin/admin
[oracle@jfsc2 admin]$ mkdir PANCAKE
[oracle@jfsc2 admin]$ cd PANCAKE
[oracle@jfsc2 PANCAKE]$ mkdir adump dpdump pfile scripts xdb_wallet
```

建立 **oratab** 項目

下列命令是 oraenv 等公用程式正常運作所需的命令。

```
PANCAKE:/orabin/product/12.1.0/dbhome_1:N
```

參數更新

必須更新儲存的 pfile 、以反映新伺服器上的任何路徑變更。資料檔案路徑變更是由 RMAN 複製指令碼所變更、幾乎所有資料庫都需要變更 `control_files` 和 `log_archive_dest` 參數。也可能有必須變更的稽核檔案位置和參數、例如 `db_create_file_dest` 在 ASM 之外可能無關緊要。經驗豐富的 DBA 應仔細審查建議的變更、然後再繼續。

在此範例中、主要變更為控制檔位置、記錄歸檔目的地、以及新增 `log_file_name_convert` 參數。

```
PANCAKE.__data_transfer_cache_size=0
PANCAKE.__db_cache_size=545259520
PANCAKE.__java_pool_size=4194304
PANCAKE.__large_pool_size=25165824
PANCAKE.__oracle_base='/orabin'#ORACLE_BASE set from environment
PANCAKE.__pga_aggregate_target=268435456
PANCAKE.__sga_target=805306368
PANCAKE.__shared_io_pool_size=29360128
PANCAKE.__shared_pool_size=192937984
PANCAKE.__streams_pool_size=0
*.audit_file_dest='/orabin/admin/PANCAKE/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='+ASM0/PANCAKE/control01.ctl','+ASM0/PANCAKE/control02.ctl
'
*.control_files='/oradata/pancake/control01.ctl','/logs/pancake/control02.
ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='PANCAKE'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=PANCAKEXDB)'
*.log_archive_dest_1='LOCATION=+ASM1'
*.log_archive_dest_1='LOCATION=/logs/pancake'
*.log_archive_format='%t_%s_%r.dbf'
'/logs/path/redo02.log'
*.log_file_name_convert = '+ASM0/PANCAKE/redo01.log',
'/logs/pancake/redo01.log', '+ASM0/PANCAKE/redo02.log',
'/logs/pancake/redo02.log', '+ASM0/PANCAKE/redo03.log',
'/logs/pancake/redo03.log'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

確認新參數之後、必須使參數生效。存在多個選項、但大多數客戶會根據文字 pfile 建立 spfile 。

```
bash-4.1$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Fri Jan 8 11:17:40 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> create spfile from pfile='/rman/pancake/pfile';
File created.
```

啟動 **nomount**

複寫資料庫之前的最後一個步驟是啟動資料庫程序、但不要掛載檔案。在此步驟中、 spfile 可能會出現問題。如果是 `startup nomount` 命令因參數錯誤而失敗、關機很簡單、請修正 pfile 範本、將其重新載入為 spfile 、然後再試一次。

```
SQL> startup nomount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             373296240 bytes
Database Buffers          423624704 bytes
Redo Buffers                5455872 bytes
```

複製資料庫

將先前的 RMAN 備份還原至新位置、比此程序中的其他步驟花費更多時間。必須複製資料庫、而不需變更資料庫 ID （ DBID ）或重新設定記錄。這可防止套用記錄、這是完全同步複本的必要步驟。

使用 RMAN AS aux 連線至資料庫、並使用在前一個步驟中建立的指令碼發出重複資料庫命令。

```
[oracle@jfsc2 pancake]$ rman auxiliary /
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 03:04:56
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights
reserved.
connected to auxiliary database: PANCAKE (not mounted)
RMAN> run
2> {
3> set newname for datafile 1 to '/oradata/pancake/pancake.dbf';
4> set newname for datafile 2 to '/oradata/pancake/sysaux.dbf';
5> set newname for datafile 3 to '/oradata/pancake/undotbs1.dbf';
6> set newname for datafile 4 to '/oradata/pancake/users.dbf';
7> set newname for tempfile 1 to '/oradata/pancake/temp.dbf';
8> duplicate target database for standby backup location '/rman/pancake';
9> }
executing command: SET NEWNAME
```

```
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting Duplicate Db at 24-MAY-16
contents of Memory Script:
{
   restore clone standby controlfile from  '/rman/pancake/ctrl.bkp';
}
executing Memory Script
Starting restore at 24-MAY-16
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file name=/oradata/pancake/control01.ctl
output file name=/logs/pancake/control02.ctl
Finished restore at 24-MAY-16
contents of Memory Script:
{
   sql clone 'alter database mount standby database';
}
executing Memory Script
sql statement: alter database mount standby database
released channel: ORA_AUX_DISK_1
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=243 device type=DISK
contents of Memory Script:
{
   set newname for tempfile  1 to
 "/oradata/pancake/temp.dbf";
   switch clone tempfile all;
   set newname for datafile  1 to
 "/oradata/pancake/pancake.dbf";
   set newname for datafile  2 to
 "/oradata/pancake/sysaux.dbf";
   set newname for datafile  3 to
 "/oradata/pancake/undotbs1.dbf";
   set newname for datafile  4 to
 "/oradata/pancake/users.dbf";
   restore
   clone database
   ;
}
executing Memory Script
executing command: SET NEWNAME
```

```
renamed tempfile 1 to /oradata/pancake/temp.dbf in control file
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
executing command: SET NEWNAME
Starting restore at 24-MAY-16
using channel ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: specifying datafile(s) to restore from backup set
channel ORA_AUX_DISK_1: restoring datafile 00001 to
/oradata/pancake/pancake.dbf
channel ORA_AUX_DISK_1: restoring datafile 00002 to
/oradata/pancake/sysaux.dbf
channel ORA_AUX_DISK_1: restoring datafile 00003 to
/oradata/pancake/undotbs1.dbf
channel ORA_AUX_DISK_1: restoring datafile 00004 to
/oradata/pancake/users.dbf
channel ORA_AUX_DISK_1: reading from backup piece
/rman/pancake/1gr6c161_1_1
channel ORA_AUX_DISK_1: piece handle=/rman/pancake/1gr6c161_1_1
tag=ONTAP_MIGRATION
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:07
Finished restore at 24-MAY-16
contents of Memory Script:
{
   switch clone datafile all;
}
executing Memory Script
datafile 1 switched to datafile copy
input datafile copy RECID=5 STAMP=912655725 file
name=/oradata/pancake/pancake.dbf
datafile 2 switched to datafile copy
input datafile copy RECID=6 STAMP=912655725 file
name=/oradata/pancake/sysaux.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=7 STAMP=912655725 file
name=/oradata/pancake/undotbs1.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=8 STAMP=912655725 file
name=/oradata/pancake/users.dbf
Finished Duplicate Db at 24-MAY-16
```

初始記錄複寫

您現在必須將變更從來源資料庫傳送至新位置。這樣做可能需要多個步驟的組合。最簡單的方法是讓來源資料庫

上的 RMAN 將歸檔記錄寫入共用網路連線。如果無法使用共用位置、則另一種方法是使用 RMAN 寫入本機檔案系統、然後使用 rcp 或 rsync 複製檔案。

在此範例中 `/rman` 目錄是一種 NFS 共用、可同時用於原始和移轉的資料庫。

此處的一個重要問題是 `disk format` 條款。備份的磁碟格式為 `%h_%e_%a.dbf`，這表示您必須使用資料庫的執行緒編號、序號和啟動 ID 格式。雖然字母不同、但這與相符 `log_archive_format='%t_%s_%r.dbf` pfile 中的參數。此參數也會以執行緒編號、序號和啟動 ID 的格式來指定封存記錄。最終結果是來源上的記錄檔備份使用資料庫預期的命名慣例。如此一來、就能執行像這樣的作業 `recover database` 更簡單、因為 sqlplus 能正確預測要重新播放的歸檔記錄名稱。

```
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/arch/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
released channel: ORA_DISK_1
RMAN> backup as copy archivelog from time 'sysdate-2';
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=373 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=70 STAMP=912658508
output file name=/rman/pancake/logship/1_54_912576125.dbf RECID=123
STAMP=912659482
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=29 STAMP=912654101
output file name=/rman/pancake/logship/1_41_912576125.dbf RECID=124
STAMP=912659483
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=33 STAMP=912654688
output file name=/rman/pancake/logship/1_45_912576125.dbf RECID=152
STAMP=912659514
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=36 STAMP=912654809
output file name=/rman/pancake/logship/1_47_912576125.dbf RECID=153
STAMP=912659515
channel ORA_DISK_1: archived log copy complete, elapsed time: 00:00:01
Finished backup at 24-MAY-16
```

初始記錄重新播放

檔案在歸檔記錄位置後、可以發出命令來重新播放 `recover database until cancel` 接著是回應 `AUTO` 自動重播所有可用的記錄。參數檔目前正在將歸檔記錄導向 `/logs/archive`但這與 RMAN 用於保存日誌的位置不匹配。在恢復資料庫之前、可依下列方式暫時重新導向位置。

```
SQL> alter system set log_archive_dest_1='LOCATION=/rman/pancake/logship'
scope=memory;
System altered.
SQL> recover standby database until cancel;
ORA-00279: change 560224 generated at 05/24/2016 03:25:53 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_49_912576125.dbf
ORA-00280: change 560224 for thread 1 is in sequence #49
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
AUTO
ORA-00279: change 560353 generated at 05/24/2016 03:29:17 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_50_912576125.dbf
ORA-00280: change 560353 for thread 1 is in sequence #50
ORA-00278: log file '/rman/pancake/logship/1_49_912576125.dbf' no longer
needed
for this recovery
...
ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
ORA-00278: log file '/rman/pancake/logship/1_53_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_54_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
```

最終的歸檔記錄回覆會回報錯誤、但這是正常現象。此錯誤表示 sqlplus 正在尋找特定的記錄檔、但找不到該檔案。原因很可能是記錄檔尚未存在。

如果在複製歸檔記錄之前可以關閉來源資料庫、則此步驟只能執行一次。歸檔記錄會複製並重新播放、然後程序會直接繼續進行轉換程序、以複寫重要的重作記錄。

遞增記錄複寫及重新播放

在大多數情況下、移轉作業不會立即執行。移轉程序可能在幾天甚至幾週前完成、這表示記錄必須持續運送至複本資料庫並重新執行。這樣做可確保轉換程序到達時、必須傳輸和重播最少的資料。

此程序很容易撰寫指令碼。例如、您可以在原始資料庫上排程下列命令、以確保用於記錄傳送的位置持續更新。

```
[oracle@jfsc1 pancake]$ cat copylogs.rman
configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
backup as copy archivelog from time 'sysdate-2';
```

```
[oracle@jfsc1 pancake]$ rman target / cmdfile=copylogs.rman
Recovery Manager: Release 12.1.0.2.0 - Production on Tue May 24 04:36:19
2016
Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights
reserved.
connected to target database: PANCAKE (DBID=3574534589)
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=369 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:22
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=41 RECID=124 STAMP=912659483
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:23
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_41_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
```

```
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:55
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03009: failure of backup command on ORA_DISK_1 channel at 05/24/2016
04:36:57
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf
Recovery Manager complete.
```

在收到記錄之後、必須重新播放記錄。先前的範例顯示使用 sqlplus 來手動執行 `recover database until cancel` 可輕鬆自動化。此處顯示的範例使用中所述的指令碼 "重播待命資料庫上的記錄"。指令碼會接受一個引數、指定需要重新執行作業的資料庫。此程序允許在多資料庫移轉工作中使用相同的指令碼。

```
[root@jfsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 04:47:10 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 560591 generated at 05/24/2016 03:33:56 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_54_912576125.dbf
ORA-00280: change 560591 for thread 1 is in sequence #54
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 562219 generated at 05/24/2016 04:15:08 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_55_912576125.dbf
ORA-00280: change 562219 for thread 1 is in sequence #55
ORA-00278: log file '/rman/pancake/logship/1_54_912576125.dbf' no longer
needed for this recovery
ORA-00279: change 562370 generated at 05/24/2016 04:19:18 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_56_912576125.dbf
ORA-00280: change 562370 for thread 1 is in sequence #56
ORA-00278: log file '/rman/pancake/logship/1_55_912576125.dbf' no longer
needed for this recovery
...
ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
ORA-00278: log file '/rman/pancake/logship/1_64_912576125.dbf' no longer
needed for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_65_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

轉換

準備好切換至新環境時、您必須執行最後一次同步。使用一般檔案系統時、由於原始的重作記錄會複製並重新播放、因此很容易確保移轉的資料庫與原始資料庫 100% 同步。使用 ASM 執行此作業的方法並不理想。只有歸檔日誌可以輕鬆地重新記錄。為了確保不會遺失任何資料、必須謹慎執行原始資料庫的最終關機。

1. 首先、必須將資料庫暫時禁用、確保不會進行任何變更。這種停止可能包括停用排程作業、關閉接聽程式及 / 或關閉應用程式。
2. 執行此步驟後、大多數 DBA 會建立一個虛擬表格、做為關機的標記。
3. 強制記錄歸檔、以確保在歸檔記錄檔中記錄建立虛擬表格。若要這麼做、請執行下列命令：

```
SQL> create table cutovercheck as select * from dba_users;
Table created.
SQL> alter system archive log current;
System altered.
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

4. 若要複製最後一個歸檔記錄檔、請執行下列命令。資料庫必須可用、但不可開啟。

```
SQL> startup mount;
ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             331353200 bytes
Database Buffers          465567744 bytes
Redo Buffers                5455872 bytes
Database mounted.
```

5. 若要複製歸檔記錄檔、請執行下列命令：

```
RMAN> configure channel device type disk format
'/rman/pancake/logship/%h_%e_%a.dbf';
2> backup as copy archivelog from time 'sysdate-2';
3>
4>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT
'/rman/pancake/logship/%h_%e_%a.dbf';
new RMAN configuration parameters are successfully stored
Starting backup at 24-MAY-16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=8 device type=DISK
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=54 RECID=123 STAMP=912659482
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:24
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_54_912576125.dbf
continuing other job steps, job failed will not be re-run
...
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=45 RECID=152 STAMP=912659514
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:58:58
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_45_912576125.dbf
continuing other job steps, job failed will not be re-run
channel ORA_DISK_1: starting archived log copy
input archived log thread=1 sequence=47 RECID=153 STAMP=912659515
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
05/24/2016 04:59:00
ORA-19635: input and output file names are identical:
/rman/pancake/logship/1_47_912576125.dbf
```

6. 最後、在新伺服器上重播剩餘的歸檔記錄。

```
[root@jfsc2 pancake]# ./replaylogs.pl PANCAKE
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 24 05:00:53 2016
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> ORA-00279: change 563137 generated at 05/24/2016 04:36:20 needed
for thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_65_912576125.dbf
ORA-00280: change 563137 for thread 1 is in sequence #65
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
ORA-00279: change 563629 generated at 05/24/2016 04:55:20 needed for
thread 1
ORA-00289: suggestion : /rman/pancake/logship/1_66_912576125.dbf
ORA-00280: change 563629 for thread 1 is in sequence #66
ORA-00278: log file '/rman/pancake/logship/1_65_912576125.dbf' no longer
needed
for this recovery
ORA-00308: cannot open archived log
'/rman/pancake/logship/1_66_912576125.dbf'
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
```

7. 在此階段、複寫所有資料。資料庫已準備好從待命資料庫轉換為作用中的作業資料庫、然後開啟。

```
SQL> alter database activate standby database;
Database altered.
SQL> alter database open;
Database altered.
```

8. 確認虛擬表格是否存在、然後將其丟棄。

```
SQL> desc cutovercheck
 Name                                      Null?    Type
 ----------------------------------------- --------
----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 USER_ID                                   NOT NULL NUMBER
 PASSWORD                                           VARCHAR2(4000)
 ACCOUNT_STATUS                            NOT NULL VARCHAR2(32)
 LOCK_DATE                                          DATE
 EXPIRY_DATE                                        DATE
 DEFAULT_TABLESPACE                        NOT NULL VARCHAR2(30)
 TEMPORARY_TABLESPACE                      NOT NULL VARCHAR2(30)
 CREATED                                   NOT NULL DATE
 PROFILE                                   NOT NULL VARCHAR2(128)
 INITIAL_RSRC_CONSUMER_GROUP                        VARCHAR2(128)
 EXTERNAL_NAME                                      VARCHAR2(4000)
 PASSWORD_VERSIONS                                  VARCHAR2(12)
 EDITIONS_ENABLED                                   VARCHAR2(1)
 AUTHENTICATION_TYPE                                VARCHAR2(8)
 PROXY_ONLY_CONNECT                                 VARCHAR2(1)
 COMMON                                             VARCHAR2(3)
 LAST_LOGIN                                         TIMESTAMP(9) WITH
TIME ZONE
 ORACLE_MAINTAINED                                  VARCHAR2(1)
SQL> drop table cutovercheck;
Table dropped.
```

不中斷的重作記錄移轉

有時資料庫會在整體上正確組織、但重做記錄除外。這可能是因為許多原因、其中最常見的原因與快照有關。SnapManager for Oracle 、 SnapCenter 和 NetApp Snap Creator 儲存管理架構等產品可讓您近乎即時地恢復資料庫、但前提是您必須還原資料檔案磁碟區的狀態。如果重做記錄檔與資料檔案共用空間、則無法安全執行還原、因為還原會導致重做記錄檔毀損、這可能表示資料遺失。因此、重做記錄必須重新定位。

此程序很簡單、可在不中斷營運的情況下執行。

目前的重做記錄組態

1. 識別重做記錄群組的數目及其各自的群組編號。

```
SQL> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
--------------------------------------------------------------------------------
--------
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
rows selected.
```

2. 輸入重做記錄檔的大小。

```
SQL> select group#||' '||bytes from v$log;
GROUP#||''||BYTES
--------------------------------------------------------------------------------
--------
1 524288000
2 524288000
3 524288000
```

建立新記錄

1. 針對每個重做記錄、建立一個大小和成員數目相符的新群組。

```
SQL> alter database add logfile ('/newredo0/redo01a.log',
'/newredo1/redo01b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo02a.log',
'/newredo1/redo02b.log') size 500M;
Database altered.
SQL> alter database add logfile ('/newredo0/redo03a.log',
'/newredo1/redo03b.log') size 500M;
Database altered.
SQL>
```

2. 驗證新組態。

```
SQL> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
--------------------------------------------------------------------------------
--------
1 /redo0/NTAP/redo01a.log
1 /redo1/NTAP/redo01b.log
2 /redo0/NTAP/redo02a.log
2 /redo1/NTAP/redo02b.log
3 /redo0/NTAP/redo03a.log
3 /redo1/NTAP/redo03b.log
4 /newredo0/redo01a.log
4 /newredo1/redo01b.log
5 /newredo0/redo02a.log
5 /newredo1/redo02b.log
6 /newredo0/redo03a.log
6 /newredo1/redo03b.log
12 rows selected.
```

刪除舊記錄

1. 刪除舊記錄（群組 1 、 2 和 3 ）。

```
SQL> alter database drop logfile group 1;
Database altered.
SQL> alter database drop logfile group 2;
Database altered.
SQL> alter database drop logfile group 3;
Database altered.
```

2. 如果您遇到錯誤、導致無法刪除作用中記錄、請強制切換至下一個記錄檔、以釋放鎖定並強制建立全域檢查點。請參閱下列此程序範例。由於此記錄檔中仍有作用中的資料、因此拒絕嘗試丟棄位於舊位置的記錄檔群組 2 。

```
SQL> alter database drop logfile group 2;
alter database drop logfile group 2
*
ERROR at line 1:
ORA-01623: log 2 is current log for instance NTAP (thread 1) - cannot
drop
ORA-00312: online log 2 thread 1: '/redo0/NTAP/redo02a.log'
ORA-00312: online log 2 thread 1: '/redo1/NTAP/redo02b.log'
```

3. 記錄歸檔之後再加上檢查點、可讓您捨棄記錄檔。

```
SQL> alter system archive log current;
System altered.
SQL> alter system checkpoint;
System altered.
SQL> alter database drop logfile group 2;
Database altered.
```

4. 然後從檔案系統刪除記錄。您應該非常小心地執行此程序。

Oracle 資料庫主機資料複本

如同資料庫層級的移轉、主機層的移轉也提供儲存設備廠商的不受侷連的方法。

換句話說、有時候「只複製檔案」是最佳選擇。

雖然這種低技術方法似乎過於基本、但它確實提供了顯著的效益、因為不需要特殊軟體、而且在程序期間、原始資料仍保持安全不變。主要的限制是檔案複製資料移轉是一項破壞性程序、因為必須在複製作業開始之前關閉資料庫。沒有適當的方法可以同步處理檔案中的變更、因此檔案必須在開始複製之前完全處於禁用狀態。

如果複製作業所需的關機不理想、則下一個最佳的主機型選項是使用邏輯 Volume Manager （ LVM ）。包括 Oracle ASM 在內的許多 LVM 選項都具有類似的功能、但也有一些必須考量的限制。在大多數情況下、可在不中斷或停機的情況下完成移轉。

檔案系統複製到檔案系統

不應低估簡單複製作業的效用。這項作業需要在複製程序期間停機、但這是一個非常可靠的程序、不需要操作系統、資料庫或儲存系統的專門知識。此外、它也非常安全、因為它不會影響原始資料。通常、系統管理員會將來源檔案系統變更為唯讀安裝、然後重新啟動伺服器、以保證沒有任何東西會損壞目前的資料。複製程序可以撰寫指令碼、確保能以最快的速度執行、而不會發生使用者錯誤的風險。由於 I/O 類型是簡單的資料循序傳輸、因此具有極高的頻寬效率。

下列範例示範安全快速移轉的一個選項。

環境

要移轉的環境如下：

- 目前的檔案系統

```
ontap-nfs1:/host1_oradata       52428800  16196928  36231872  31%
/oradata
ontap-nfs1:/host1_logs          49807360    548032  49259328   2% /logs
```

- 新檔案系統

```
ontap-nfs1:/host1_logs_new      49807360       128   49807232   1%
/new/logs
ontap-nfs1:/host1_oradata_new   49807360       128   49807232   1%
/new/oradata
```

總覽

資料庫可由 DBA 移轉、只需關閉資料庫並複製檔案即可、但如果必須移轉許多資料庫、或是將停機時間降至最低、則此程序很容易撰寫指令碼。使用指令碼也能降低使用者錯誤的機率。

所示範例指令碼可自動化下列作業：

- 關閉資料庫
- 將現有檔案系統轉換為唯讀狀態
- 將所有資料從來源複製到目標檔案系統、以保留所有檔案權限
- 卸載舊的和新的檔案系統
- 將新檔案系統重新掛載到與先前檔案系統相同的路徑

程序

1. 關閉資料庫。

```
[root@host1 current]# ./dbshut.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 15:58:48 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down
```

2. 將檔案系統轉換為唯讀。如所示、使用指令碼可以更快完成這項工作 "將檔案系統轉換為唯讀"。

```
[root@host1 current]# ./mk.fs.readonly.pl /oradata
/oradata unmounted
/oradata mounted read-only
[root@host1 current]# ./mk.fs.readonly.pl /logs
/logs unmounted
/logs mounted read-only
```

3. 確認檔案系統現在為唯讀。

```
ontap-nfs1:/host1_oradata on /oradata type nfs
(ro,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
ontap-nfs1:/host1_logs on /logs type nfs
(ro,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
```

4. 將檔案系統內容與同步 `rsync` 命令。

```
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /oradata/ /new/oradata/
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
 10737426432 100%  153.50MB/s    0:01:06 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
    22823573 100%   12.09MB/s    0:00:01 (xfer#2, to-check=9/13)
...
NTAP/undotbs02.dbf
  1073750016 100%  131.60MB/s    0:00:07 (xfer#10, to-check=1/13)
NTAP/users01.dbf
     5251072 100%    3.95MB/s    0:00:01 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes  received 228 bytes  162204017.96 bytes/sec
total size is 18570092218  speedup is 1.00
```

```
[root@host1 current]# rsync -rlpogt --stats --progress
--exclude=.snapshot /logs/ /new/logs/
sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
    45523968 100%   95.98MB/s    0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
    40601088 100%   49.45MB/s    0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
    52429312 100%   44.68MB/s    0:00:01 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
    52429312 100%   68.03MB/s    0:00:00 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
sent 527098156 bytes  received 278 bytes  95836078.91 bytes/sec
total size is 527032832  speedup is 1.00
```

5. 卸載舊檔案系統、並重新放置複製的資料。如所示、使用指令碼可以更快完成這項工作 "取代檔案系統"。

```
[root@host1 current]# ./swap.fs.pl /logs,/new/logs
/new/logs unmounted
/logs unmounted
Updated /logs mounted
[root@host1 current]# ./swap.fs.pl /oradata,/new/oradata
/new/oradata unmounted
/oradata unmounted
Updated /oradata mounted
```

6. 確認新檔案系統已就位。

```
ontap-nfs1:/host1_logs_new on /logs type nfs
(rw,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
ontap-nfs1:/host1_oradata_new on /oradata type nfs
(rw,bg,vers=3,rsize=65536,wsize=65536,addr=172.20.101.10)
```

7. 啟動資料庫。

```
[root@host1 current]# ./dbstart.pl NTAP
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 16:10:07 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             390073456 bytes
Database Buffers          406847488 bytes
Redo Buffers                5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
```

全自動轉換

此範例指令碼接受資料庫 SID 的引數、後面接著通用分隔的檔案系統配對。如前所示、命令發出方式如下：

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
```

執行時、範例指令碼會嘗試執行下列順序。如果在任何步驟中遇到錯誤、它都會終止：

1. 關閉資料庫。
2. 將目前的檔案系統轉換為唯讀狀態。
3. 使用每個以逗號分隔的檔案系統引數配對、並將第一個檔案系統同步到第二個檔案系統。
4. 卸除先前的檔案系統。
5. 更新 `/etc/fstab` 檔案如下：
   a. 請在下列位置建立備份 `/etc/fstab.bak`。

b. 註解先前和新檔案系統的先前項目。

c. 為使用舊掛載點的新檔案系統建立新項目。

6. 掛載檔案系統。

7. 啟動資料庫。

下列文字提供此指令碼的執行範例：

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:05:50 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP shut down
sending incremental file list
./
NTAP/
NTAP/1_22_897068759.dbf
    45523968 100%  185.40MB/s    0:00:00 (xfer#1, to-check=15/18)
NTAP/1_23_897068759.dbf
    40601088 100%   81.34MB/s    0:00:00 (xfer#2, to-check=14/18)
...
NTAP/redo/redo02.log
    52429312 100%   70.42MB/s    0:00:00 (xfer#12, to-check=1/18)
NTAP/redo/redo03.log
    52429312 100%   47.08MB/s    0:00:01 (xfer#13, to-check=0/18)
Number of files: 18
Number of files transferred: 13
Total file size: 527032832 bytes
Total transferred file size: 527032832 bytes
Literal data: 527032832 bytes
Matched data: 0 bytes
File list size: 413
File list generation time: 0.001 seconds
```

```
File list transfer time: 0.000 seconds
Total bytes sent: 527098156
Total bytes received: 278
sent 527098156 bytes  received 278 bytes  150599552.57 bytes/sec
total size is 527032832  speedup is 1.00
Succesfully replicated filesystem /logs to /new/logs
sending incremental file list
./
NTAP/
NTAP/IOPS.dbf
 10737426432 100%  176.55MB/s    0:00:58 (xfer#1, to-check=10/13)
NTAP/iops.dbf.zip
    22823573 100%    9.48MB/s    0:00:02 (xfer#2, to-check=9/13)
... NTAP/undotbs01.dbf
   309338112 100%   70.76MB/s    0:00:04 (xfer#9, to-check=2/13)
NTAP/undotbs02.dbf
  1073750016 100%  187.65MB/s    0:00:05 (xfer#10, to-check=1/13)
NTAP/users01.dbf
     5251072 100%    5.09MB/s    0:00:00 (xfer#11, to-check=0/13)
Number of files: 13
Number of files transferred: 11
Total file size: 18570092218 bytes
Total transferred file size: 18570092218 bytes
Literal data: 18570092218 bytes
Matched data: 0 bytes
File list size: 277
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 18572359828
Total bytes received: 228
sent 18572359828 bytes  received 228 bytes  177725933.55 bytes/sec
total size is 18570092218  speedup is 1.00
Succesfully replicated filesystem /oradata to /new/oradata
swap 0 /logs /new/logs
/new/logs unmounted
/logs unmounted
Mounted updated /logs
Swapped filesystem /logs for /new/logs
swap 1 /oradata /new/oradata
/new/oradata unmounted
/oradata unmounted
Mounted updated /oradata
Swapped filesystem /oradata for /new/oradata
ORACLE_SID = [oracle] ? The Oracle base has been set to /orabin
SQL*Plus: Release 12.1.0.2.0 Production on Thu Dec 3 17:08:59 2015
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
```

```
Connected to an idle instance.
SQL> ORACLE instance started.
Total System Global Area  805306368 bytes
Fixed Size                  2929552 bytes
Variable Size             390073456 bytes
Database Buffers          406847488 bytes
Redo Buffers                5455872 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
NTAP started
[root@host1 current]#
```

Oracle ASM spfile 和 passwd 移轉

在完成涉及 ASM 的移轉時、有一個困難是 ASM 專屬的 spfile 和密碼檔案。根據預設、這些關鍵中繼資料檔案會建立在定義的第一個 ASM 磁碟群組上。如果必須撤出和移除特定的 ASM 磁碟群組、則必須重新放置管理該 ASM 執行個體的 spfile 和密碼檔案。

另一個需要重新放置這些檔案的使用案例是在部署資料庫管理軟體時、例如 SnapManager for Oracle 或 SnapCenter Oracle 外掛程式。這些產品的其中一項功能是透過還原代管資料檔案的 ASM LUN 狀態、快速還原資料庫。這樣做需要在執行還原之前將 ASM 磁碟群組離線。只要指定資料庫的資料檔案隔離在專用的 ASM 磁碟群組中、這不是問題。

當該磁碟群組也包含 ASM spfile/passwd 檔案時、唯一可以將磁碟群組離線的方法是關閉整個 ASM 執行個體。這是一項破壞性程序、也就是說、 spfile/passwd 檔案必須重新放置。

環境

1. 資料庫 SID = Toast
2. 目前的資料檔案位於 `+DATA`
3. 上目前的記錄檔和控制檔 `+LOGS`
4. 建立為的新 ASM 磁碟群組 `+NEWDATA` 和 `+NEWLOGS`

ASM spfile/passwd 檔案位置

您可以不中斷地重新放置這些檔案。不過、為了安全起見、 NetApp 建議您關閉資料庫環境、以便確定檔案已重新放置、且組態已正確更新。如果伺服器上有多個 ASM 執行個體、則必須重複此程序。

識別 **ASM** 執行個體

根據中記錄的資料來識別 ASM 執行個體 `oratab` 檔案：ASM 執行個體以 + 符號表示。

```
-bash-4.1$ cat /etc/oratab | grep '^+'
+ASM:/orabin/grid:N             # line added by Agent
```

此伺服器上有一個稱為 +ASM 的 ASM 執行個體。

確定所有資料庫都已關閉

唯一可見的 SMON 程序應該是使用中 ASM 執行個體的 SMON 。另一個 SMON 程序的存在表示資料庫仍在執行中。

```
-bash-4.1$ ps -ef | grep smon
oracle     857     1  0 18:26 ?        00:00:00 asm_smon_+ASM
```

唯一的 SMON 程序是 ASM 執行個體本身。這表示沒有其他資料庫正在執行中、而且在不中斷資料庫作業的風險下繼續作業是安全的。

尋找檔案

使用識別 ASM spfile 和密碼檔案的目前位置 `spget` 和 `pwget` 命令。

```
bash-4.1$ asmcmd
ASMCMD> spget
+DATA/spfile.ora
```

```
ASMCMD> pwget --asm
+DATA/orapwasm
```

這些檔案都位於的基礎上 `+DATA` 磁碟群組。

複製檔案

使用將檔案複製到新的 ASM 磁碟群組 `spcopy` 和 `pwcopy` 命令。如果新磁碟群組是最近建立的、而且目前是空的、則可能需要先掛載。

```
ASMCMD> mount NEWDATA
```

```
ASMCMD> spcopy +DATA/spfile.ora +NEWDATA/spfile.ora
copying +DATA/spfile.ora -> +NEWDATA/spfilea.ora
```

```
ASMCMD> pwcopy +DATA/orapwasm +NEWDATA/orapwasm
copying +DATA/orapwasm -> +NEWDATA/orapwasm
```

檔案現已從複製 `+DATA` 至 `+NEWDATA`。

更新 **ASM** 執行個體

現在必須更新 ASM 執行個體、以反映位置變更。。 `spset` 和 `pwset` 命令會更新啟動 ASM 磁碟群組所需的 ASM 中繼資料。

```
ASMCMD> spset +NEWDATA/spfile.ora
ASMCMD> pwset --asm +NEWDATA/orapwasm
```

使用更新的檔案啟動 **ASM**

此時、 ASM 執行個體仍會使用這些檔案的先前位置。必須重新啟動執行個體、以強制重新讀取新位置的檔案、並釋放先前檔案上的鎖定。

```
-bash-4.1$ sqlplus / as sysasm
SQL> shutdown immediate;
ASM diskgroups volume disabled
ASM diskgroups dismounted
ASM instance shutdown
```

```
SQL> startup
ASM instance started
Total System Global Area 1140850688 bytes
Fixed Size                  2933400 bytes
Variable Size            1112751464 bytes
ASM Cache                  25165824 bytes
ORA-15032: not all alterations performed
ORA-15017: diskgroup "NEWDATA" cannot be mounted
ORA-15013: diskgroup "NEWDATA" is already mounted
```

移除舊的 **spfile** 和密碼檔案

如果程序已成功執行、先前的檔案將不再鎖定、現在可以移除。

```
-bash-4.1$ asmcmd
ASMCMD> rm +DATA/spfile.ora
ASMCMD> rm +DATA/orapwasm
```

Oracle ASM 至 ASM 複本

Oracle ASM 本質上是輕量的組合 Volume Manager 和檔案系統。由於檔案系統並不容易看到、因此 RMAN 必須用於執行複製作業。雖然複製型移轉程序既安全又簡單、但會造成部分中斷。可以將中斷降至最低、但不能完全消除。

如果您想要不中斷地移轉 ASM 型資料庫、最好的方法是利用 ASM 的功能、在移轉舊 LUN 的同時、重新平衡 ASM 擴充至新 LUN 的平衡。這樣做通常是安全且不中斷營運的、但它不提供回溯路徑。如果遇到功能或效能問題、唯一的選項是將資料移回來源。

您可以將資料庫複製到新位置而非移動資料、以避免此風險、避免原始資料受到影響。資料庫可以在新位置進行完整測試後再上線運作、如果發現問題、原始資料庫則可作為回復選項使用。

此程序是 RMAN 的眾多選項之一。其設計允許建立初始備份的兩個步驟程序、然後透過記錄重播進行同步處理。這項程序最適合將停機時間降至最低、因為它可讓資料庫在初始基準複本期間維持運作並提供資料。

複製資料庫

Oracle RMAN 會建立目前位於 ASM 磁碟群組的來源資料庫層級 0 （完整）複本 `+DATA` 移至新位置 `+NEWDATA`。

```
-bash-4.1$ rman target /
Recovery Manager: Release 12.1.0.2.0 - Production on Sun Dec 6 17:40:03
2015
Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights
reserved.
connected to target database: TOAST (DBID=2084313411)
RMAN> backup as copy incremental level 0 database format '+NEWDATA' tag
'ONTAP_MIGRATION';
Starting backup at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=302 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
...
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
output file name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
tag=ONTAP_MIGRATION RECID=5 STAMP=897759622
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWDATA/TOAST/BACKUPSET/2015_12_06/nnsnn0_ontap_migration_0.262.89
7759623 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

強制歸檔記錄切換

您必須強制使用歸檔記錄切換、以確保歸檔記錄包含所有必要資料、使複本完全一致。如果沒有此命令、重做記錄檔中可能仍會有關鍵資料。

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

關閉來源資料庫

由於資料庫已關機、並處於有限存取、唯讀模式、因此在此步驟中就會開始中斷。若要關閉來源資料庫、請執行下列命令：

```
RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                390073456 bytes
Database Buffers             406847488 bytes
Redo Buffers                   5455872 bytes
```

控制檔備份

您必須備份控制檔、以防您必須中止移轉並還原至原始儲存位置。備份控制檔的複本並非 100% 必要、但它確實讓將資料庫檔案位置重設回原始位置的程序變得更簡單。

```
RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=358 device type=DISK
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151206T174753 RECID=6
STAMP=897760073
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

參數更新

目前的 spfile 包含對舊 ASM 磁碟群組內控制檔目前位置的參照。您必須編輯此檔案、只要編輯中繼 pfile 版本即可輕鬆完成。

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```

更新 **pfile**

更新任何參照舊 ASM 磁碟群組的參數、以反映新的 ASM 磁碟群組名稱。然後儲存更新的 pfile 。請確定 `db_create` 有參數存在。

在以下範例中、請參考 +DATA 變更為 +NEWDATA 以黃色反白顯示。兩個主要參數是 db_create 在正確位置建立任何新檔案的參數。

```
*.compatible='12.1.0.2.0'
*.control_files='+NEWLOGS/TOAST/CONTROLFILE/current.258.897683139'
*.db_block_size=8192
*. db_create_file_dest='+NEWDATA'
*. db_create_online_log_dest_1='+NEWLOGS'
*.db_domain=''
*.db_name='TOAST'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB)'
*.log_archive_dest_1='LOCATION=+NEWLOGS'
*.log_archive_format='%t_%s_%r.dbf'
```

更新 **init.ora** 檔案

大多數以 ASM 為基礎的資料庫都使用 init.ora 檔案位於 $ORACLE_HOME/dbs 目錄、指向 ASM 磁碟群組上的 spfile 。此檔案必須重新導向至新 ASM 磁碟群組上的位置。

```
-bash-4.1$ cd $ORACLE_HOME/dbs
-bash-4.1$ cat initTOAST.ora
SPFILE='+DATA/TOAST/spfileTOAST.ora'
```

變更此檔案的方式如下：

```
SPFILE=+NEWLOGS/TOAST/spfileTOAST.ora
```

參數檔案重新建立

spfile 現在已準備好由編輯的 pfile 中的資料填入。

```
RMAN> create spfile from pfile='/tmp/pfile';
Statement processed
```

啟動資料庫以開始使用新的 **spfile**

啟動資料庫以確保它現在使用新建立的 spfile 、並正確記錄對系統參數的任何進一步變更。

```
RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                373296240 bytes
Database Buffers             423624704 bytes
Redo Buffers                   5455872 bytes
```

還原控制檔

RMAN 所建立的備份控制檔也可直接還原至新 spfile 中指定的位置。

```
RMAN> restore controlfile from
'+DATA/TOAST/CONTROLFILE/current.258.897683139';
Starting restore at 06-DEC-15
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=417 device type=DISK
channel ORA_DISK_1: copied control file copy
output file name=+NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
Finished restore at 06-DEC-15
```

裝入資料庫並驗證新控制檔的使用。

```
RMAN> alter database mount;
using target database control file instead of recovery catalog
Statement processed
```

```
SQL> show parameter control_files;
NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
control_files                        string
+NEWLOGS/TOAST/CONTROLFILE/cur
                                                 rent.273.897761061
```

記錄重新播放

資料庫目前使用舊位置的資料檔案。在使用複本之前、必須先進行同步處理。初始複製程序已經過時間、變更主要記錄在歸檔記錄中。這些變更會複寫如下：

1. 執行包含歸檔記錄的 RMAN 遞增備份。

```
RMAN> backup incremental level 1 format '+NEWLOGS' for recover of copy
with tag 'ONTAP_MIGRATION' database;
Starting backup at 06-DEC-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=62 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/TOAST/DATAFILE/system.262.897683141
input datafile file number=00002
name=+DATA/TOAST/DATAFILE/sysaux.260.897683143
input datafile file number=00003
name=+DATA/TOAST/DATAFILE/undotbs1.257.897683145
input datafile file number=00004
name=+DATA/TOAST/DATAFILE/users.264.897683151
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 06-DEC-15
channel ORA_DISK_1: finished piece 1 at 06-DEC-15
piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/ncsnn1_ontap_migration_0.267.
897762697 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15
```

2. 重新播放記錄。

```
RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 06-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001
name=+NEWDATA/TOAST/DATAFILE/system.259.897759609
recovering datafile copy file number=00002
name=+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
recovering datafile copy file number=00003
name=+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
recovering datafile copy file number=00004
name=+NEWDATA/TOAST/DATAFILE/users.258.897759623
channel ORA_DISK_1: reading from backup piece
+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.8977626
93
channel ORA_DISK_1: piece
handle=+NEWLOGS/TOAST/BACKUPSET/2015_12_06/nnndn1_ontap_migration_0.268.
897762693 tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15
```

啟動

還原的控制檔仍會參照原始位置的資料檔案、也會包含複製資料檔案的路徑資訊。

1. 若要變更使用中的資料檔案、請執行 `switch database to copy` 命令。

```
RMAN> switch database to copy;
datafile 1 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/system.259.897759609"
datafile 2 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615"
datafile 3 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619"
datafile 4 switched to datafile copy
"+NEWDATA/TOAST/DATAFILE/users.258.897759623"
```

   使用中的資料檔案現在是複製的資料檔案、但最終的重做記錄檔中可能仍有變更。

2. 若要重播所有剩餘記錄、請執行 `recover database` 命令。如果出現此訊息 `media recovery complete` 出現時、程序成功。

```
RMAN> recover database;
Starting recover at 06-DEC-15
using channel ORA_DISK_1
starting media recovery
media recovery complete, elapsed time: 00:00:01
Finished recover at 06-DEC-15
```

此程序只會變更一般資料檔案的位置。必須重新命名暫存資料檔案、但不需要複製、因為它們只是暫時性的。資料庫目前關閉、因此暫存資料檔案中沒有作用中的資料。

3. 若要重新放置暫存資料檔案、請先識別其位置。

```
RMAN> select file#||' '||name from v$tempfile;
FILE#||''||NAME
-------------------------------------------------------------------------------
--------
1 +DATA/TOAST/TEMPFILE/temp.263.897683145
```

4. 使用 RMAN 命令重新定位暫存資料檔案、為每個資料檔案設定新名稱。使用 Oracle 託管檔案（ OMF ）時、不需要完整名稱； ASM 磁碟群組已足夠。開啟資料庫時、 OMF 會連結至 ASM 磁碟群組上的適當位置。若要重新定位檔案、請執行下列命令：

```
run {
set newname for tempfile 1 to '+NEWDATA';
switch tempfile all;
}
```

```
RMAN> run {
2> set newname for tempfile 1 to '+NEWDATA';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to +NEWDATA in control file
```

重做記錄移轉

移轉程序即將完成、但重做記錄仍位於原始 ASM 磁碟群組中。重作記錄無法直接重新定位。而是會建立新的重做記錄集、並將其新增至組態、然後刪除舊的記錄。

1. 識別重做記錄群組的數目及其各自的群組編號。

```
RMAN> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
------------------------------------------------------------------------------
--------
1 +DATA/TOAST/ONLINELOG/group_1.261.897683139
2 +DATA/TOAST/ONLINELOG/group_2.259.897683139
3 +DATA/TOAST/ONLINELOG/group_3.256.897683139
```

2. 輸入重做記錄檔的大小。

```
RMAN> select group#||' '||bytes from v$log;
GROUP#||''||BYTES
------------------------------------------------------------------------------
--------
1 52428800
2 52428800
3 52428800
```

3. 針對每個重做記錄、建立具有相符組態的新群組。如果您未使用 OMF 、則必須指定完整路徑。這也是使用的範例 `db_create_online_log` 參數。如先前所示、此參數設為 +NEWLOGS 。此組態可讓您使用下列命令來建立新的線上記錄檔、而無需指定檔案位置、甚至是特定的 ASM 磁碟群組。

```
RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed
RMAN> alter database add logfile size 52428800;
Statement processed
```

4. 開啟資料庫。

```
SQL> alter database open;
Database altered.
```

5. 刪除舊記錄。

```
RMAN> alter database drop logfile group 1;
Statement processed
```

6. 如果您遇到錯誤、導致無法刪除作用中記錄、請強制切換至下一個記錄檔、以釋放鎖定並強制建立全域檢查點。範例如下所示。嘗試丟棄位於舊位置的記錄檔群組 3 、因為此記錄檔中仍有作用中資料、因此遭到拒

絕。檢查點之後的記錄封存可讓您刪除記錄檔。

```
RMAN> alter database drop logfile group 3;
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 3 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 3 thread 1:
'+LOGS/TOAST/ONLINELOG/group_3.259.897563549'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 3;
Statement processed
```

7. 檢閱環境、確定所有位置型參數都已更新。

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

8. 下列指令碼示範如何簡化此程序：

```
[root@host1 current]# ./checkdbdata.pl TOAST
TOAST datafiles:
+NEWDATA/TOAST/DATAFILE/system.259.897759609
+NEWDATA/TOAST/DATAFILE/sysaux.263.897759615
+NEWDATA/TOAST/DATAFILE/undotbs1.264.897759619
+NEWDATA/TOAST/DATAFILE/users.258.897759623
TOAST redo logs:
+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123
+NEWLOGS/TOAST/ONLINELOG/group_5.265.897763125
+NEWLOGS/TOAST/ONLINELOG/group_6.264.897763125
TOAST temp datafiles:
+NEWDATA/TOAST/TEMPFILE/temp.260.897763165
TOAST spfile
spfile                                string
+NEWDATA/spfiletoast.ora
TOAST key parameters
control_files +NEWLOGS/TOAST/CONTROLFILE/current.273.897761061
log_archive_dest_1 LOCATION=+NEWLOGS
db_create_file_dest +NEWDATA
db_create_online_log_dest_1 +NEWLOGS
```

9. 如果 ASM 磁碟群組已完全撤出、現在可以使用卸載 `asmcmd`。不過、在許多情況下、屬於其他資料庫或 ASM spfile/passwd 檔案的檔案可能仍存在。

```
-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>
```

Oracle ASM 至檔案系統複本

Oracle ASM 至檔案系統複製程序與 ASM 至 ASM 複製程序非常類似、具有類似的優點和限制。主要差異在於使用可見檔案系統時、不同命令和組態參數的語法、而非使用 ASM 磁碟群組。

複製資料庫

Oracle RMAN 用於建立目前位於 ASM 磁碟群組的來源資料庫層級 0 （完整）複本 `+DATA` 移至新位置 `/oradata`。

```
RMAN> backup as copy incremental level 0 database format
'/oradata/TOAST/%U' tag 'ONTAP_MIGRATION';
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=377 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-
1_01r5fhjg tag=ONTAP_MIGRATION RECID=1 STAMP=911722099
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-
2_02r5fhjo tag=ONTAP_MIGRATION RECID=2 STAMP=911722106
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt tag=ONTAP_MIGRATION RECID=3 STAMP=911722113
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:07
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/oradata/TOAST/cf_D-TOAST_id-2098173325_04r5fhk5
tag=ONTAP_MIGRATION RECID=4 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
output file name=/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-
4_05r5fhk6 tag=ONTAP_MIGRATION RECID=5 STAMP=911722118
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental level 0 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/oradata/TOAST/06r5fhk7_1_1 tag=ONTAP_MIGRATION comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16
```

強制歸檔記錄切換

必須強制使用歸檔記錄交換器、才能確保歸檔記錄包含所有必要資料、使複本完全一致。如果沒有此命令、重做記錄檔中可能仍會有關鍵資料。若要強制使用歸檔記錄交換器、請執行下列命令：

```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

關閉來源資料庫

由於資料庫已關機、並處於有限存取的唯讀模式、因此此步驟開始造成中斷。若要關閉來源資料庫、請執行下列命令：

```
RMAN> shutdown immediate;
using target database control file instead of recovery catalog
database closed
database dismounted
Oracle instance shut down
RMAN> startup mount;
connected to target database (not started)
Oracle instance started
database mounted
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                331353200 bytes
Database Buffers             465567744 bytes
Redo Buffers                   5455872 bytes
```

控制檔備份

備份控制檔、以防您必須中止移轉並還原至原始儲存位置。備份控制檔的複本並非 100% 必要、但它確實讓將資料庫檔案位置重設回原始位置的程序變得更簡單。

```
RMAN> backup as copy current controlfile format '/tmp/TOAST.ctrl';
Starting backup at 08-DEC-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
copying current control file
output file name=/tmp/TOAST.ctrl tag=TAG20151208T194540 RECID=30
STAMP=897939940
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 08-DEC-15
```

參數更新

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```

更新 **pfile**

任何參照舊 ASM 磁碟群組的參數都應該更新、在某些情況下、當不再相關時、就會刪除。更新它們以反映新的檔案系統路徑、並儲存更新的 pfile 。請確定已列出完整的目標路徑。若要更新這些參數、請執行下列命令：

```
*.audit_file_dest='/orabin/admin/TOAST/adump'
*.audit_trail='db'
*.compatible='12.1.0.2.0'
*.control_files='/logs/TOAST/arch/control01.ctl','/logs/TOAST/redo/control
02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='TOAST'
*.diagnostic_dest='/orabin'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=TOASTXDB)'
*.log_archive_dest_1='LOCATION=/logs/TOAST/arch'
*.log_archive_format='%t_%s_%r.dbf'
*.open_cursors=300
*.pga_aggregate_target=256m
*.processes=300
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=768m
*.undo_tablespace='UNDOTBS1'
```

停用原始的 **init.ora** 檔案

此檔案位於 `$ORACLE_HOME/dbs` 目錄和通常位於 pfile 中、作為指向 ASM 磁碟群組上 spfile 的指標。若要確定不再使用原始 spfile 、請重新命名。不過、請勿刪除它、因為如果必須中止移轉、就需要此檔案。

```
[oracle@jfsc1 ~]$ cd $ORACLE_HOME/dbs
[oracle@jfsc1 dbs]$ cat initTOAST.ora
SPFILE='+ASM0/TOAST/spfileTOAST.ora'
[oracle@jfsc1 dbs]$ mv initTOAST.ora initTOAST.ora.prev
[oracle@jfsc1 dbs]$
```

參數檔案重新建立

這是重新定位 spfile 的最後一步。原始 spfile 不再使用、而且資料庫目前是使用中繼檔案啟動（但未掛載）。此檔案的內容可以寫入新的 spfile 位置、如下所示：

```
RMAN> create spfile from pfile='/tmp/pfile';
Statement processed
```

啟動資料庫以開始使用新的 **spfile**

您必須啟動資料庫以釋放中繼檔案上的鎖定、並只使用新的 spfile 檔案來啟動資料庫。啟動資料庫也能證明新的 spfile 位置正確、而且其資料有效。

```
RMAN> shutdown immediate;
Oracle instance shut down
RMAN> startup nomount;
connected to target database (not started)
Oracle instance started
Total System Global Area     805306368 bytes
Fixed Size                     2929552 bytes
Variable Size                331353200 bytes
Database Buffers             465567744 bytes
Redo Buffers                   5455872 bytes
```

還原控制檔

已在路徑上建立備份控制檔 `/tmp/TOAST.ctrl` 請稍早在程序中進行。新的 spfile 將控制檔位置定義為 `/logfs/TOAST/ctrl/ctrlfile1.ctrl` 和 `/logfs/TOAST/redo/ctrlfile2.ctrl`。不過、這些檔案尚不存在。

1. 此命令會將控制檔資料還原至 spfile 中定義的路徑。

   ```
   RMAN> restore controlfile from '/tmp/TOAST.ctrl';
   Starting restore at 13-MAY-16
   using channel ORA_DISK_1
   channel ORA_DISK_1: copied control file copy
   output file name=/logs/TOAST/arch/control01.ctl
   output file name=/logs/TOAST/redo/control02.ctl
   Finished restore at 13-MAY-16
   ```

2. 發出 mount 命令、以便正確探索控制檔並包含有效資料。

   ```
   RMAN> alter database mount;
   Statement processed
   released channel: ORA_DISK_1
   ```

   驗證 `control_files` 參數、請執行下列命令：

```
SQL> show parameter control_files;
NAME                                 TYPE        VALUE
------------------------------------ -----------
------------------------------
control_files                        string
/logs/TOAST/arch/control01.ctl
                                                 ,
/logs/TOAST/redo/control02.c
                                                 tl
```

記錄重新播放

資料庫目前正在使用舊位置的資料檔案。在使用複本之前、必須先同步資料檔案。在初始複製程序期間已經過時間、變更主要記錄在歸檔記錄中。以下兩個步驟會複寫這些變更。

1. 執行包含歸檔記錄的 RMAN 遞增備份。

```
RMAN>  backup incremental level 1 format '/logs/TOAST/arch/%U' for
recover of copy with tag 'ONTAP_MIGRATION' database;
Starting backup at 13-MAY-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=124 device type=DISK
channel ORA_DISK_1: starting incremental level 1 datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+ASM0/TOAST/system01.dbf
input datafile file number=00002 name=+ASM0/TOAST/sysaux01.dbf
input datafile file number=00003 name=+ASM0/TOAST/undotbs101.dbf
input datafile file number=00004 name=+ASM0/TOAST/users01.dbf
channel ORA_DISK_1: starting piece 1 at 13-MAY-16
channel ORA_DISK_1: finished piece 1 at 13-MAY-16
piece handle=/logs/TOAST/arch/09r5fj8i_1_1 tag=ONTAP_MIGRATION
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped
```

2. 重播記錄。

```
RMAN> recover copy of database with tag 'ONTAP_MIGRATION';
Starting recover at 13-MAY-16
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
recovering datafile copy file number=00002 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
recovering datafile copy file number=00003 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
recovering datafile copy file number=00004 name=/oradata/TOAST/data_D-
TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
channel ORA_DISK_1: reading from backup piece
/logs/TOAST/arch/09r5fj8i_1_1
channel ORA_DISK_1: piece handle=/logs/TOAST/arch/09r5fj8i_1_1
tag=ONTAP_MIGRATION
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 13-MAY-16
RMAN-06497: WARNING: control file is not current, control file
AUTOBACKUP skipped
```

啟動

還原的控制檔仍會參照原始位置的資料檔案、也會包含複製資料檔案的路徑資訊。

1. 若要變更使用中的資料檔案、請執行 `switch database to copy` 命令：

```
RMAN> switch database to copy;
datafile 1 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSTEM_FNO-1_01r5fhjg"
datafile 2 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-SYSAUX_FNO-2_02r5fhjo"
datafile 3 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt"
datafile 4 switched to datafile copy "/oradata/TOAST/data_D-TOAST_I-
2098173325_TS-USERS_FNO-4_05r5fhk6"
```

2. 雖然資料檔案應完全一致、但仍需最後一步才能重播線上重作記錄中記錄的其餘變更。使用 `recover database` 命令重播這些變更、並使複本 100% 與原始版本相同。不過、複本尚未開啟。

```
RMAN> recover database;
Starting recover at 13-MAY-16
using channel ORA_DISK_1
starting media recovery
archived log for thread 1 with sequence 28 is already on disk as file
+ASM0/TOAST/redo01.log
archived log file name=+ASM0/TOAST/redo01.log thread=1 sequence=28
media recovery complete, elapsed time: 00:00:00
Finished recover at 13-MAY-16
```

重新部署暫存資料檔案

1. 識別仍在原始磁碟群組中使用的暫存資料檔案位置。

```
RMAN> select file#||' '||name from v$tempfile;
FILE#||''||NAME
------------------------------------------------------------------------
--------
1 +ASM0/TOAST/temp01.dbf
```

2. 若要重新放置資料檔案、請執行下列命令。如果有許多 tempfiles 、請使用文字編輯器建立 RMAN 命令、然後剪下並貼上。

```
RMAN> run {
2> set newname for tempfile 1 to '/oradata/TOAST/temp01.dbf';
3> switch tempfile all;
4> }
executing command: SET NEWNAME
renamed tempfile 1 to /oradata/TOAST/temp01.dbf in control file
```

重做記錄移轉

移轉程序即將完成、但重做記錄仍位於原始 ASM 磁碟群組中。重作記錄無法直接重新定位。而是建立新的重做記錄集、並在刪除舊記錄之後新增至組態。

1. 識別重做記錄群組的數目及其各自的群組編號。

```
RMAN> select group#||' '||member from v$logfile;
GROUP#||''||MEMBER
--------------------------------------------------------------------------------
--------
1 +ASM0/TOAST/redo01.log
2 +ASM0/TOAST/redo02.log
3 +ASM0/TOAST/redo03.log
```

2. 輸入重做記錄檔的大小。

```
RMAN> select group#||' '||bytes from v$log;
GROUP#||''||BYTES
--------------------------------------------------------------------------------
--------
1 52428800
2 52428800
3 52428800
```

3. 對於每個重做記錄、請使用與目前重做記錄群組相同的大小、使用新的檔案系統位置來建立新群組。

```
RMAN> alter database add logfile '/logs/TOAST/redo/log00.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log01.rdo' size
52428800;
Statement processed
RMAN> alter database add logfile '/logs/TOAST/redo/log02.rdo' size
52428800;
Statement processed
```

4. 移除仍位於先前儲存設備上的舊記錄檔群組。

```
RMAN> alter database drop logfile group 4;
Statement processed
RMAN> alter database drop logfile group 5;
Statement processed
RMAN> alter database drop logfile group 6;
Statement processed
```

5. 如果遇到阻止刪除作用中記錄的錯誤、請強制切換至下一個記錄檔、以釋放鎖定並強制建立全域檢查點。範例如下所示。嘗試丟棄位於舊位置的記錄檔群組 3 、因為此記錄檔中仍有作用中資料、因此遭到拒絕。記錄歸檔之後再加上檢查點、即可刪除記錄檔。

```
RMAN> alter database drop logfile group 4;
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-03002: failure of sql statement command at 12/08/2015 20:23:51
ORA-01623: log 4 is current log for instance TOAST (thread 4) - cannot
drop
ORA-00312: online log 4 thread 1:
'+NEWLOGS/TOAST/ONLINELOG/group_4.266.897763123'
RMAN> alter system switch logfile;
Statement processed
RMAN> alter system checkpoint;
Statement processed
RMAN> alter database drop logfile group 4;
Statement processed
```

6. 檢閱環境、確定所有位置型參數都已更新。

```
SQL> select name from v$datafile;
SQL> select member from v$logfile;
SQL> select name from v$tempfile;
SQL> show parameter spfile;
SQL> select name, value from v$parameter where value is not null;
```

7. 下列指令碼示範如何簡化此程序。

```
[root@jfsc1 current]# ./checkdbdata.pl TOAST
TOAST datafiles:
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-3_03r5fhjt
/oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
TOAST redo logs:
/logs/TOAST/redo/log00.rdo
/logs/TOAST/redo/log01.rdo
/logs/TOAST/redo/log02.rdo
TOAST temp datafiles:
/oradata/TOAST/temp01.dbf
TOAST spfile
spfile                                  string
/orabin/product/12.1.0/dbhome_
                                                      1/dbs/spfileTOAST.ora
TOAST key parameters
control_files /logs/TOAST/arch/control01.ctl,
/logs/TOAST/redo/control02.ctl
log_archive_dest_1 LOCATION=/logs/TOAST/arch
```

8. 如果 ASM 磁碟群組已完全撤出、現在可以使用卸載 `asmcmd`。在許多情況下、屬於其他資料庫或 ASM spfile/passwd 檔案的檔案仍會存在。

```
-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>
```

資料檔案清理程序

根據 Oracle RMAN 的使用方式而定、移轉程序可能會導致資料檔案的語法較長或較隱密。在此所示範例中、備份是以的檔案格式執行 `/oradata/TOAST/%U`。 `%U` 表示 RMAN 應為每個資料檔案建立預設的唯一名稱。結果與下列文字所示類似。資料檔案的傳統名稱會內嵌在名稱中。您可以使用中所示的指令碼方法來清除此問題 "ASM 移轉清理"。

```
[root@jfsc1 current]# ./fixuniquenames.pl TOAST
#sqlplus Commands
shutdown immediate;
startup mount;
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSTEM_FNO-1_01r5fhjg
/oradata/TOAST/system.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-SYSAUX_FNO-2_02r5fhjo
/oradata/TOAST/sysaux.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-UNDOTBS1_FNO-
3_03r5fhjt /oradata/TOAST/undotbs1.dbf
host mv /oradata/TOAST/data_D-TOAST_I-2098173325_TS-USERS_FNO-4_05r5fhk6
/oradata/TOAST/users.dbf
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSTEM_FNO-1_01r5fhjg' to '/oradata/TOAST/system.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
SYSAUX_FNO-2_02r5fhjo' to '/oradata/TOAST/sysaux.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
UNDOTBS1_FNO-3_03r5fhjt' to '/oradata/TOAST/undotbs1.dbf';
alter database rename file '/oradata/TOAST/data_D-TOAST_I-2098173325_TS-
USERS_FNO-4_05r5fhk6' to '/oradata/TOAST/users.dbf';
alter database open;
```

Oracle ASM 重新平衡

如前所述、 Oracle ASM 磁碟群組可透過重新平衡程序、以透明方式移轉至新的儲存系統。總而言之、重新平衡程序需要在現有的 LUN 群組中新增大小相同的 LUN 、然後再中斷先前 LUN 的作業。Oracle ASM 會以最佳配置自動將基礎資料重新定位至新儲存設備、然後在完成時釋出舊的 LUN 。

移轉程序使用高效率的循序 I/O 、通常不會造成任何效能中斷、但可視需要調整移轉率。

識別要移轉的資料

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
SQL> select group_number||' '||name from v$asm_diskgroup;
1 NEWDATA
```

建立新的 **LUN**

建立大小相同的新 LUN 、並視需要設定使用者和群組成員資格。LUN 應顯示為 `CANDIDATE` 磁碟。

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
 0 0 /dev/mapper/3600a098038303537762b47594c31586b CANDIDATE
 0 0 /dev/mapper/3600a098038303537762b47594c315869 CANDIDATE
 0 0 /dev/mapper/3600a098038303537762b47594c315858 CANDIDATE
 0 0 /dev/mapper/3600a098038303537762b47594c31586a CANDIDATE
NEWDATA_0003 1 10240 /dev/mapper/3600a098038303537762b47594c315864 MEMBER
NEWDATA_0002 1 10240 /dev/mapper/3600a098038303537762b47594c315863 MEMBER
NEWDATA_0000 1 10240 /dev/mapper/3600a098038303537762b47594c315861 MEMBER
NEWDATA_0001 1 10240 /dev/mapper/3600a098038303537762b47594c315862 MEMBER
```

新增 **LUN**

雖然可以同時執行新增和刪除作業、但通常只需兩個步驟即可輕鬆新增 LUN 。首先、將新 LUN 新增至磁碟群組。此步驟會將一半的擴充從目前的 ASM LUN 移轉至新的 LUN 。

重新平衡的力量代表資料傳輸的速度。資料傳輸的平行度越高、資料傳輸的數量就越多。執行移轉時、必須執行有效率的連續 I/O 作業、而這些作業不太可能造成效能問題。不過、若有需要、可利用調整進行中移轉的重新平衡能力 `alter diskgroup [name] rebalance power [level]` 命令。典型移轉使用 5 個值。

```
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586b' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315869' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c315858' rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA add disk
'/dev/mapper/3600a098038303537762b47594c31586a' rebalance power 5;
Diskgroup altered.
```

監控作業

可透過多種方式監控和管理重新平衡作業。在此範例中、我們使用下列命令。

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
------------ ----- ----
           1 REBAL RUN
           1 REBAL WAIT
```

移轉完成時、不會回報任何重新平衡作業。

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

丟棄舊的 **LUN**

移轉作業現在已完成一半。您可能需要執行一些基本效能測試、以確保環境健全。確認之後、可藉由丟棄舊的 LUN 來重新放置其餘的資料。請注意、這不會導致 LUN 立即發行。此中斷作業會先發出 Oracle ASM 重新定位延伸、然後再釋放 LUN 。

```
sqlplus / as sysasm
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0000 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup NEWDATA drop disk NEWDATA_0001 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0002 rebalance power 5;
Diskgroup altered.
SQL> alter diskgroup newdata drop disk NEWDATA_0003 rebalance power 5;
Diskgroup altered.
```

監控作業

可透過多種方式監控和管理重新平衡作業。在此範例中、我們使用下列命令：

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
------------ ----- ----
           1 REBAL RUN
           1 REBAL WAIT
```

移轉完成時、不會回報任何重新平衡作業。

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

移除舊的 **LUN**

從磁碟群組移除舊 LUN 之前、您應該先對標頭狀態執行一次最後檢查。從 ASM 發佈 LUN 後、它不再列出名稱、而且標頭狀態會列為 `FORMER`。這表示這些 LUN 可以安全地從系統中移除。

```
SQL> select name||' '||group_number||' '||total_mb||' '||path||'
'||header_status from v$asm_disk;
NAME||''||GROUP_NUMBER||''||TOTAL_MB||''||PATH||''||HEADER_STATUS
--------------------------------------------------------------------------
------
 0 0 /dev/mapper/3600a098038303537762b47594c315863 FORMER
 0 0 /dev/mapper/3600a098038303537762b47594c315864 FORMER
 0 0 /dev/mapper/3600a098038303537762b47594c315861 FORMER
 0 0 /dev/mapper/3600a098038303537762b47594c315862 FORMER
NEWDATA_0005 1 10240 /dev/mapper/3600a098038303537762b47594c315869 MEMBER
NEWDATA_0007 1 10240 /dev/mapper/3600a098038303537762b47594c31586a MEMBER
NEWDATA_0004 1 10240 /dev/mapper/3600a098038303537762b47594c31586b MEMBER
NEWDATA_0006 1 10240 /dev/mapper/3600a098038303537762b47594c315858 MEMBER
8 rows selected.
```

LVM 移轉

此處介紹的程序顯示了以 LVM 為基礎的磁碟區群組移轉原則、稱為 `datavg`。這些範例來自 Linux LVM 、但這些原則同樣適用於 AIX 、 HP-UX 和 VxVM 。精確命令可能會有所不同。

1. 識別目前在中的 LUN `datavg` Volume 群組。

```
[root@host1 ~]# pvdisplay -C | grep datavg
  /dev/mapper/3600a098038303537762b47594c31582f datavg lvm2 a--  10.00g
10.00g
  /dev/mapper/3600a098038303537762b47594c31585a datavg lvm2 a--  10.00g
10.00g
  /dev/mapper/3600a098038303537762b47594c315859 datavg lvm2 a--  10.00g
10.00g
  /dev/mapper/3600a098038303537762b47594c31586c datavg lvm2 a--  10.00g
10.00g
```

2. 建立相同或稍大實體大小的新 LUN 、並將其定義為實體磁碟區。

```
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315864
  Physical volume "/dev/mapper/3600a098038303537762b47594c315864"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315863
  Physical volume "/dev/mapper/3600a098038303537762b47594c315863"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315862
  Physical volume "/dev/mapper/3600a098038303537762b47594c315862"
successfully created
[root@host1 ~]# pvcreate /dev/mapper/3600a098038303537762b47594c315861
  Physical volume "/dev/mapper/3600a098038303537762b47594c315861"
successfully created
```

3. 將新的磁碟區新增至磁碟區群組。

```
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315864
  Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315863
  Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315862
  Volume group "datavg" successfully extended
[root@host1 tmp]# vgextend datavg
/dev/mapper/3600a098038303537762b47594c315861
  Volume group "datavg" successfully extended
```

4. 發行 `pvmove` 命令將每個目前 LUN 的範圍重新放置到新 LUN 。。 `- i [seconds]` 引數會監控作業的進度。

```
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31582f
/dev/mapper/3600a098038303537762b47594c315864
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 14.2%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 28.4%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 42.5%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 57.1%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 72.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 87.3%
  /dev/mapper/3600a098038303537762b47594c31582f: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31585a
/dev/mapper/3600a098038303537762b47594c315863
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 14.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 29.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 44.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 60.1%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 75.8%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 90.9%
  /dev/mapper/3600a098038303537762b47594c31585a: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c315859
/dev/mapper/3600a098038303537762b47594c315862
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 14.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 29.8%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 45.5%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 61.1%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 76.6%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 91.7%
  /dev/mapper/3600a098038303537762b47594c315859: Moved: 100.0%
[root@host1 tmp]# pvmove -i 10
/dev/mapper/3600a098038303537762b47594c31586c
/dev/mapper/3600a098038303537762b47594c315861
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 0.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 15.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 30.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 46.0%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 61.4%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 77.2%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 92.3%
  /dev/mapper/3600a098038303537762b47594c31586c: Moved: 100.0%
```

5. 完成此程序後、請使用從磁碟區群組中刪除舊的 LUN `vgreduce` 命令。如果成功、現在即可安全地從系統移除 LUN 。

```
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31582f
Removed "/dev/mapper/3600a098038303537762b47594c31582f" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31585a
  Removed "/dev/mapper/3600a098038303537762b47594c31585a" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c315859
  Removed "/dev/mapper/3600a098038303537762b47594c315859" from volume
group "datavg"
[root@host1 tmp]# vgreduce datavg
/dev/mapper/3600a098038303537762b47594c31586c
  Removed "/dev/mapper/3600a098038303537762b47594c31586c" from volume
group "datavg"
```

外部 **LUN** 匯入

使用 **FLI** 進行 **Oracle** 移轉：規劃

NetApp 中記錄了使用 FLI 移轉 SAN 資源的程序 "TR-4380 ：使用外部 LUN Import 進行 SAN 移轉"。

從資料庫和主機的觀點來看、不需要採取任何特殊步驟。更新 FC 區域並在 ONTAP 上提供 LUN 之後、 LVM 應該能夠從 LUN 讀取 LVM 中繼資料。此外、這些磁碟區群組也可以開始使用、無需進一步的組態步驟。在極少數情況下、環境可能會包含硬編碼的組態檔案、其中包含先前儲存陣列的參考資料。例如、內含的 Linux 系統 `/etc/multipath.conf` 參考指定裝置 WWN 的規則必須更新、以反映 FLI 所做的變更。

如需支援組態的相關資訊、請參閱 NetApp 相容性對照表。如果您的環境未包含在內、請聯絡 NetApp 代表以取得協助。

此範例顯示 Linux 伺服器上代管的 ASM 和 LVM LUN 移轉。其他作業系統支援 FLI 、雖然主機端命令可能不同、但原則相同、 ONTAP 程序相同。

識別 **LVM LUN**

準備的第一步是識別要移轉的 LUN 。在此所示範例中、會在裝載兩個 SAN 型檔案系統 `/orabin` 和 `/backups`。

```
[root@host1 ~]# df -k
Filesystem                          1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root                52403200   8811464  43591736  17% /
devtmpfs                             65882776         0  65882776   0% /dev
...
fas8060-nfs-public:/install         199229440 119368128  79861312  60%
/install
/dev/mapper/sanvg-lvorabin           20961280  12348476   8612804  59%
/orabin
/dev/mapper/sanvg-lvbackups          73364480  62947536  10416944  86%
/backups
```

Volume 群組的名稱可以從裝置名稱中擷取、該名稱使用格式（ Volume 群組名稱） -（邏輯磁碟區名稱）。在這種情況下、會呼叫 Volume 群組 `sanvg`。

。 `pvdisplay` 命令可用於識別支援此 Volume 群組的 LUN 、如下所示。在這種情況下、共有 10 個 LUN 組成 `sanvg` Volume 群組。

```
[root@host1 ~]# pvdisplay -C -o pv_name,pv_size,pv_fmt,vg_name
  PV                                                    PSize   VG
  /dev/mapper/3600a0980383030445424487556574266  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574267  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574268  10.00g sanvg
  /dev/mapper/3600a0980383030445424487556574269  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426a  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426b  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426c  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426d  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426e  10.00g sanvg
  /dev/mapper/3600a098038303044542448755657426f  10.00g sanvg
  /dev/sda2                                             278.38g rhel
```

識別 **ASM LUN**

ASM LUN 也必須移轉。若要以 sysasm 使用者的身分從 sqlplus 取得 LUN 和 LUN 路徑的數目、請執行下列命令：

```
SQL> select path||' '||os_mb from v$asm_disk;
PATH||''||OS_MB
--------------------------------------------------------------------------------
------
/dev/oracleasm/disks/ASM0 10240
/dev/oracleasm/disks/ASM9 10240
/dev/oracleasm/disks/ASM8 10240
/dev/oracleasm/disks/ASM7 10240
/dev/oracleasm/disks/ASM6 10240
/dev/oracleasm/disks/ASM5 10240
/dev/oracleasm/disks/ASM4 10240
/dev/oracleasm/disks/ASM1 10240
/dev/oracleasm/disks/ASM3 10240
/dev/oracleasm/disks/ASM2 10240
10 rows selected.
SQL>
```

FC 網路變更

目前環境包含 20 個要移轉的 LUN 。更新目前的 SAN 、讓 ONTAP 能夠存取目前的 LUN 。資料尚未移轉、但 ONTAP 必須從目前的 LUN 讀取組態資訊、才能為該資料建立新的主目錄。

AFF/FAS 系統上至少必須將一個 HBA 連接埠設定為啟動器連接埠。此外、必須更新 FC 區域、讓 ONTAP 能夠存取外部儲存陣列上的 LUN 。某些儲存陣列已設定 LUN 遮罩、限制哪些 WWN 可以存取指定的 LUN 。在這種情況下、 LUN 遮罩也必須更新、才能授予 ONTAP WWN 存取權。

完成此步驟後、 ONTAP 應能使用檢視外部儲存陣列 `storage array show` 命令。它傳回的關鍵欄位是用來識別系統上外部 LUN 的首碼。在以下範例中、為外部陣列上的 LUN `FOREIGN_1` 在 ONTAP 中使用前置碼顯示 `FOR-1`。

識別外部陣列

```
Cluster01::> storage array show -fields name,prefix
name          prefix
------------- ------
FOREIGN_1     FOR-1
Cluster01::>
```

識別外部 **LUN**

可以通過傳送來列出 LUN `array-name` 至 `storage disk show` 命令。移轉程序期間會多次參照傳回的資料。

```
Cluster01::> storage disk show -array-name FOREIGN_1 -fields disk,serial
disk     serial-number
-------- -------------
FOR-1.1  800DT$HuVWBX
FOR-1.2  800DT$HuVWBZ
FOR-1.3  800DT$HuVWBW
FOR-1.4  800DT$HuVWBY
FOR-1.5  800DT$HuVWB/
FOR-1.6  800DT$HuVWBa
FOR-1.7  800DT$HuVWBd
FOR-1.8  800DT$HuVWBb
FOR-1.9  800DT$HuVWBc
FOR-1.10 800DT$HuVWBe
FOR-1.11 800DT$HuVWBf
FOR-1.12 800DT$HuVWBg
FOR-1.13 800DT$HuVWBi
FOR-1.14 800DT$HuVWBh
FOR-1.15 800DT$HuVWBj
FOR-1.16 800DT$HuVWBk
FOR-1.17 800DT$HuVWBm
FOR-1.18 800DT$HuVWBl
FOR-1.19 800DT$HuVWBo
FOR-1.20 800DT$HuVWBn
20 entries were displayed.
Cluster01::>
```

將外部陣列 **LUN** 登錄為匯入候選項目

外部 LUN 一開始會歸類為任何特定的 LUN 類型。在匯入資料之前、必須將 LUN 標記為外部、因此是匯入程序的候選項目。將序號傳送至即可完成此步驟 `storage disk modify` 命令、如下列範例所示。請注意、此程序只會將 LUN 標記為 ONTAP 中的外部。不會將任何資料寫入外部 LUN 本身。

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign true
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign true
Cluster01::*>
```

建立磁碟區以裝載移轉的 **LUN**

需要一個磁碟區來裝載移轉的 LUN 。確切的 Volume 組態取決於運用 ONTAP 功能的整體計畫。在此範例中、ASM LUN 會放置在一個磁碟區中、而 LVM LUN 則放置在第二個磁碟區中。這樣做可讓您將 LUN 當作個別群組來管理、例如分層、建立快照或設定 QoS 控制。

設定 `snapshot-policy `to `none`。移轉程序可能包括大量資料流動。因此、如果快照是意外建立的、可能會大幅增加空間使用量、因為快照中會擷取不需要的資料。

```
Cluster01::> volume create -volume new_asm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1152] Job succeeded: Successful
Cluster01::> volume create -volume new_lvm -aggregate data_02 -size 120G
-snapshot-policy none
[Job 1153] Job succeeded: Successful
Cluster01::>
```

建立 **ONTAP LUN**

建立磁碟區之後、必須建立新的 LUN 。一般而言、建立 LUN 需要使用者指定 LUN 大小之類的資訊、但在此情況下、外部磁碟引數會傳遞給命令。因此、 ONTAP 會從指定的序號複寫目前的 LUN 組態資料。它也會使用 LUN 幾何資料和分割表格資料來調整 LUN 對齊、並建立最佳效能。

在此步驟中、序號必須與外部陣列交叉參照、以確保正確的外部 LUN 與正確的新 LUN 相符。

```
Cluster01::*> lun create -vserver vserver1 -path /vol/new_asm/LUN0 -ostype
linux -foreign-disk 800DT$HuVWBW
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_asm/LUN1 -ostype
linux -foreign-disk 800DT$HuVWBX
Created a LUN of size 10g (10737418240)
...
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_lvm/LUN8 -ostype
linux -foreign-disk 800DT$HuVWBn
Created a LUN of size 10g (10737418240)
Cluster01::*> lun create -vserver vserver1 -path /vol/new_lvm/LUN9 -ostype
linux -foreign-disk 800DT$HuVWBo
Created a LUN of size 10g (10737418240)
```

建立匯入關係

LUN 現已建立、但尚未設定為複寫目的地。在執行此步驟之前、必須先將 LUN 離線。這項額外步驟旨在保護資料不受使用者錯誤影響。如果 ONTAP 允許在線上 LUN 上執行移轉、可能會造成打字錯誤、導致覆寫作用中資料。強制使用者先將 LUN 離線的額外步驟、有助於確認使用正確的目標 LUN 做為移轉目的地。

```
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN0
Warning: This command will take LUN "/vol/new_asm/LUN0" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_asm/LUN1
Warning: This command will take LUN "/vol/new_asm/LUN1" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
...
Warning: This command will take LUN "/vol/new_lvm/LUN8" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
Cluster01::*> lun offline -vserver vserver1 -path /vol/new_lvm/LUN9
Warning: This command will take LUN "/vol/new_lvm/LUN9" in Vserver
         "vserver1" offline.
Do you want to continue? {y|n}: y
```

LUN 離線後、您可以將外部 LUN 序號傳送至、以建立匯入關係 `lun import create` 命令。

```
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN0
-foreign-disk 800DT$HuVWBW
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_asm/LUN1
-foreign-disk 800DT$HuVWBX
...
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN8
-foreign-disk 800DT$HuVWBn
Cluster01::*> lun import create -vserver vserver1 -path /vol/new_lvm/LUN9
-foreign-disk 800DT$HuVWBo
Cluster01::*>
```

建立所有匯入關係之後、即可將 LUN 重新上線。

```
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun online -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun online -vserver vserver1 -path /vol/new_lvm/LUN9
Cluster01::*>
```

建立啟動器群組

啟動器群組（ igroup ）是 ONTAP LUN 遮罩架構的一部分。除非先授予主機存取權、否則無法存取新建立的 LUN 。這是透過建立一個 igroup 、列出應授予存取權的 FC WWN 或 iSCSI 啟動器名稱來完成。在撰寫本報告

時、僅 FC LUN 支援 FLI 。不過、轉換為 iSCSI 後移轉是一項簡單的工作、如所示 "傳輸協定轉換"。

在此範例中、會建立一個 igroup 、其中包含兩個 WWN 、對應於主機 HBA 上可用的兩個連接埠。

```
Cluster01::*> igroup create linuxhost -protocol fcp -ostype linux
-initiator 21:00:00:0e:1e:16:63:50 21:00:00:0e:1e:16:63:51
```

將新 **LUN** 對應至主機

在建立 igroup 之後、 LUN 會對應至定義的 igroup 。這些 LUN 僅適用於此 igroup 中包含的 WWN 。NetApp 假設移轉程序目前階段主機尚未分區至 ONTAP 。這一點很重要、因為如果主機同時分區到外部陣列和新的 ONTAP 系統、則可能會在每個陣列上發現具有相同序號的 LUN 。這種情況可能導致多重路徑故障或資料受損。

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
Cluster01::*>
```

使用 **FLI** 進行 **Oracle** 移轉：轉換

由於需要變更 FC 網路組態、因此無法避免在外部 LUN 匯入期間發生中斷。不過、中斷時間不一定比重新啟動資料庫環境和更新 FC 分區所需的時間長、以便將主機 FC 連線能力從外部 LUN 切換至 ONTAP 。

此程序可歸納如下：

1. 在外部 LUN 上執行所有 LUN 活動。
2. 將主機 FC 連線重新導向至新的 ONTAP 系統。
3. 觸發匯入程序。
4. 重新探索 LUN 。
5. 重新啟動資料庫。

您不需要等待移轉程序完成。一旦開始移轉給定的 LUN 、就可以在 ONTAP 上使用、並在資料複製程序繼續進行時提供資料。所有讀取都會傳送到外部 LUN 、而且所有寫入都會同步寫入兩個陣列。複製作業非常快速、重新導向 FC 流量的負荷也很小、因此對效能的任何影響都應該是暫時性的、而且最小的。如果有疑慮、您可以延遲重新啟動環境、直到移轉程序完成、匯入關係已刪除為止。

關閉資料庫

在本範例中、停止環境的第一步是關閉資料庫。

```
[oracle@host1 bin]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base remains unchanged with value /orabin
[oracle@host1 bin]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Automatic Storage Management, OLAP, Advanced
Analytics
and Real Application Testing options
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

關閉網格服務

其中一個要移轉的 SAN 型檔案系統也包含 Oracle ASM 服務。若要停止基礎 LUN 、則需要卸除檔案系統、這也意味著在此檔案系統上停止任何開啟檔案的處理程序。

```
[oracle@host1 bin]$ ./crsctl stop has -f
CRS-2791: Starting shutdown of Oracle High Availability Services-managed
resources on 'host1'
CRS-2673: Attempting to stop 'ora.evmd' on 'host1'
CRS-2673: Attempting to stop 'ora.DATA.dg' on 'host1'
CRS-2673: Attempting to stop 'ora.LISTENER.lsnr' on 'host1'
CRS-2677: Stop of 'ora.DATA.dg' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.asm' on 'host1'
CRS-2677: Stop of 'ora.LISTENER.lsnr' on 'host1' succeeded
CRS-2677: Stop of 'ora.evmd' on 'host1' succeeded
CRS-2677: Stop of 'ora.asm' on 'host1' succeeded
CRS-2673: Attempting to stop 'ora.cssd' on 'host1'
CRS-2677: Stop of 'ora.cssd' on 'host1' succeeded
CRS-2793: Shutdown of Oracle High Availability Services-managed resources
on 'host1' has completed
CRS-4133: Oracle High Availability Services has been stopped.
[oracle@host1 bin]$
```

卸除檔案系統

如果所有程序都關閉、 umount 作業就會成功。如果權限遭拒、檔案系統上必須有鎖定的程序。。 `fuser` 命令可協助識別這些程序。

```
[root@host1 ~]# umount /orabin
[root@host1 ~]# umount /backups
```

停用 **Volume** 群組

卸除指定 Volume 群組中的所有檔案系統後、即可停用該 Volume 群組。

```
[root@host1 ~]# vgchange --activate n sanvg
  0 logical volume(s) in volume group "sanvg" now active
[root@host1 ~]#
```

FC 網路變更

現在可以更新 FC 區域、以移除主機對外部陣列的所有存取權、並建立對 ONTAP 的存取權。

開始匯入程序

若要啟動 LUN 匯入程序、請執行 `lun import start` 命令。

```
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN0
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_asm/LUN1
...
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN8
Cluster01::lun import*> lun import start -vserver vserver1 -path
/vol/new_lvm/LUN9
Cluster01::lun import*>
```

監控匯入進度

您可以使用監控匯入作業 `lun import show` 命令。如下所示、目前正在匯入所有 20 個 LUN 、這表示即使資料複製作業仍在進行中、仍可透過 ONTAP 存取資料。

```
Cluster01::lun import*> lun import show -fields path,percent-complete
vserver   foreign-disk path              percent-complete
--------- ------------ ----------------- ----------------
vserver1  800DT$HuVWB/ /vol/new_asm/LUN4 5
vserver1  800DT$HuVWBW /vol/new_asm/LUN0 5
vserver1  800DT$HuVWBX /vol/new_asm/LUN1 6
vserver1  800DT$HuVWBY /vol/new_asm/LUN2 6
vserver1  800DT$HuVWBZ /vol/new_asm/LUN3 5
vserver1  800DT$HuVWBa /vol/new_asm/LUN5 4
vserver1  800DT$HuVWBb /vol/new_asm/LUN6 4
vserver1  800DT$HuVWBc /vol/new_asm/LUN7 4
vserver1  800DT$HuVWBd /vol/new_asm/LUN8 4
vserver1  800DT$HuVWBe /vol/new_asm/LUN9 4
vserver1  800DT$HuVWBf /vol/new_lvm/LUN0 5
vserver1  800DT$HuVWBg /vol/new_lvm/LUN1 4
vserver1  800DT$HuVWBh /vol/new_lvm/LUN2 4
vserver1  800DT$HuVWBi /vol/new_lvm/LUN3 3
vserver1  800DT$HuVWBj /vol/new_lvm/LUN4 3
vserver1  800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1  800DT$HuVWBl /vol/new_lvm/LUN6 4
vserver1  800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1  800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1  800DT$HuVWBo /vol/new_lvm/LUN9 2
20 entries were displayed.
```

如果您需要離線程序、請延遲重新探索或重新啟動服務、直到 `lun import show` 命令表示所有移轉均已成功完成。接著您可以依照中所述、完成移轉程序 "外部 LUN 匯入：完成"。

如果您需要線上移轉、請繼續在新的主目錄中重新探索 LUN 、並啟動服務。

掃描 SCSI 裝置變更

在大多數情況下、重新探索新 LUN 最簡單的選項是重新啟動主機。這樣做會自動移除舊的過時裝置、正確探索所有新的 LUN 、並建置相關的裝置、例如多重路徑裝置。以下範例顯示出完全線上的示範程序。

注意：在重新啟動主機之前、請確定中的所有項目都已存在 `/etc/fstab` 這項參照移轉的 SAN 資源會被註解出來。如果未執行此操作、且 LUN 存取有問題、作業系統可能無法開機。這種情況不會損害資料。不過、開機進入救援模式或類似模式並修正可能非常不方便 `/etc/fstab` 如此一來、就能開機作業系統以進行疑難排解。

本範例所使用 Linux 版本上的 LUN 可與重新掃描 `rescan-scsi-bus.sh` 命令。如果命令成功、每個 LUN 路徑都會出現在輸出中。輸出可能很難解譯、但如果分區和 igroup 組態正確、許多 LUN 應該會顯示為包含 `NETAPP` 廠商字串。

```
[root@host1 /]# rescan-scsi-bus.sh
Scanning SCSI subsystem for new devices
Scanning host 0 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
 Scanning for device 0 2 0 0 ...
OLD: Host: scsi0 Channel: 02 Id: 00 Lun: 00
      Vendor: LSI      Model: RAID SAS 6G 0/1  Rev: 2.13
      Type:   Direct-Access                    ANSI SCSI revision: 05
Scanning host 1 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
 Scanning for device 1 0 0 0 ...
OLD: Host: scsi1 Channel: 00 Id: 00 Lun: 00
      Vendor: Optiarc  Model: DVD RW AD-7760H  Rev: 1.41
      Type:   CD-ROM                           ANSI SCSI revision: 05
Scanning host 2 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 3 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 4 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 5 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 6 for  SCSI target IDs  0 1 2 3 4 5 6 7, all LUNs
Scanning host 7 for  all SCSI target IDs, all LUNs
 Scanning for device 7 0 0 10 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 10
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 7 0 0 11 ...
OLD: Host: scsi7 Channel: 00 Id: 00 Lun: 11
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 7 0 0 12 ...
...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 18
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
 Scanning for device 9 0 1 19 ...
OLD: Host: scsi9 Channel: 00 Id: 01 Lun: 19
      Vendor: NETAPP   Model: LUN C-Mode       Rev: 8300
      Type:   Direct-Access                    ANSI SCSI revision: 05
0 new or changed device(s) found.
0 remapped or resized device(s) found.
0 device(s) removed.
```

檢查多重路徑裝置

LUN 探索程序也會觸發多重路徑裝置的重新開發、但已知 Linux 多重路徑驅動程式偶爾會發生問題。的輸出 `multipath -ll` 應檢查以驗證輸出是否如預期。例如、下列輸出顯示與相關的多重路徑裝置 `NETAPP` 廠商字串。每個裝置有四條路徑、其中兩條優先順序為 50 、兩條優先順序為 10 。雖然確切的輸出可能會因 Linux 的不同版本而有所不同、但此輸出的外觀與預期相同。

請參閱您用來驗證的 Linux 版本的主機公用程式文件 `/etc/multipath.conf` 設定正確。

```
[root@host1 /]# multipath -ll
3600a098038303558735d493762504b36 dm-5 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:4  sdat 66:208 active ready running
| `- 9:0:1:4  sdbn 68:16  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:4  sdf  8:80   active ready running
  `- 9:0:0:4  sdz  65:144 active ready running
3600a098038303558735d493762504b2d dm-10 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:8  sdax 67:16  active ready running
| `- 9:0:1:8  sdbr 68:80  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:8  sdj  8:144  active ready running
  `- 9:0:0:8  sdad 65:208 active ready running
...
3600a098038303558735d493762504b37 dm-8 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:5  sdau 66:224 active ready running
| `- 9:0:1:5  sdbo 68:32  active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:5  sdg  8:96   active ready running
  `- 9:0:0:5  sdaa 65:160 active ready running
3600a098038303558735d493762504b4b dm-22 NETAPP  ,LUN C-Mode
size=10G features='4 queue_if_no_path pg_init_retries 50
retain_attached_hw_handle' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 7:0:1:19 sdbi 67:192 active ready running
| `- 9:0:1:19 sdcc 69:0   active ready running
`-+- policy='service-time 0' prio=10 status=enabled
  |- 7:0:0:19 sdu  65:64  active ready running
  `- 9:0:0:19 sdao 66:128 active ready running
```

重新啟動 **LVM Volume** 群組

如果正確探索到 LVM LUN 、則會發現 `vgchange --activate y` 命令應該成功。這是邏輯 Volume Manager 的價值範例。由於磁碟區群組中繼資料是寫入 LUN 本身、因此 LUN 的 WWN 變更甚至是序列號都不重要。

作業系統掃描 LUN 、並發現 LUN 上寫入的少量資料、可將其識別為屬於的實體磁碟區 `sanvg volumegroup`。然後、它會建置所有必要的裝置。只需重新啟動 Volume 群組即可。

```
[root@host1 /]# vgchange --activate y sanvg
  Found duplicate PV fpCzdLTuKfy2xDZjai1NliJh3TjLUBiT: using
/dev/mapper/3600a098038303558735d493762504b46 not /dev/sdp
  Using duplicate PV /dev/mapper/3600a098038303558735d493762504b46 from
subsystem DM, ignoring /dev/sdp
  2 logical volume(s) in volume group "sanvg" now active
```

重新掛載檔案系統

磁碟區群組重新啟動後、檔案系統可以裝入、所有原始資料均完整無缺。如前所述、即使資料複寫仍在後端群組中作用中、檔案系統仍可完全運作。

```
[root@host1 /]# mount /orabin
[root@host1 /]# mount /backups
[root@host1 /]# df -k
Filesystem                        1K-blocks      Used Available Use%
Mounted on
/dev/mapper/rhel-root              52403200   8837100  43566100  17% /
devtmpfs                           65882776         0  65882776   0% /dev
tmpfs                               6291456        84   6291372   1%
/dev/shm
tmpfs                              65898668      9884  65888784   1% /run
tmpfs                              65898668         0  65898668   0%
/sys/fs/cgroup
/dev/sda1                            505580    224828    280752  45% /boot
fas8060-nfs-public:/install      199229440 119368256  79861184  60%
/install
fas8040-nfs-routable:/snapomatic   9961472     30528   9930944   1%
/snapomatic
tmpfs                              13179736        16  13179720   1%
/run/user/42
tmpfs                              13179736         0  13179736   0%
/run/user/0
/dev/mapper/sanvg-lvorabin         20961280  12357456   8603824  59%
/orabin
/dev/mapper/sanvg-lvbackups        73364480  62947536  10416944  86%
/backups
```

重新掃描 **ASM** 設備

重新掃描 SCSI 裝置時、應已重新探索 ASMLib 裝置。重新探索可透過重新啟動 ASMLib 、然後掃描磁碟來線上驗證。

此步驟僅與使用 ASMLib 的 ASM 組態相關。

注意：若未使用 ASMLib 、請使用 `/dev/mapper` 裝置應已自動重新建立。不過、權限可能不正確。在 ASMLib 不存在的情況下、您必須為基礎裝置設定特殊權限。這樣做通常是透過中的特殊項目來完成 `/etc/multipath.conf` 或 `udev` 規則、或可能同時在兩個規則集中。這些檔案可能需要更新、以反映環境中的 WWN 或序號變更、以確保 ASM 裝置仍擁有正確的權限。

在此範例中、重新啟動 ASMLib 並掃描磁碟時、會顯示與原始環境相同的 10 個 ASM LUN 。

```
[root@host1 /]# oracleasm exit
Unmounting ASMlib driver filesystem: /dev/oracleasm
Unloading module "oracleasm": oracleasm
[root@host1 /]# oracleasm init
Loading module "oracleasm": oracleasm
Configuring "oracleasm" to use device physical block size
Mounting ASMlib driver filesystem: /dev/oracleasm
[root@host1 /]# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
Instantiating disk "ASM0"
Instantiating disk "ASM1"
Instantiating disk "ASM2"
Instantiating disk "ASM3"
Instantiating disk "ASM4"
Instantiating disk "ASM5"
Instantiating disk "ASM6"
Instantiating disk "ASM7"
Instantiating disk "ASM8"
Instantiating disk "ASM9"
```

重新啟動網格服務

現在、 LVM 和 ASM 裝置已上線且可供使用、可以重新啟動網格服務。

```
[root@host1 /]# cd /orabin/product/12.1.0/grid/bin
[root@host1 bin]# ./crsctl start has
```

重新啟動資料庫

網格服務重新啟動後、即可啟動資料庫。在嘗試啟動資料庫之前、可能需要等待幾分鐘、 ASM 服務才能完全可用。

```
[root@host1 bin]# su - oracle
[oracle@host1 ~]$ . oraenv
ORACLE_SID = [oracle] ? FLIDB
The Oracle base has been set to /orabin
[oracle@host1 ~]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0
Copyright (c) 1982, 2014, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> startup
ORACLE instance started.
Total System Global Area 3221225472 bytes
Fixed Size                  4502416 bytes
Variable Size            1207962736 bytes
Database Buffers         1996488704 bytes
Redo Buffers               12271616 bytes
Database mounted.
Database opened.
SQL>
```

Oracle 移轉與 **FLI** ：完成

從主機的角度來看、移轉已完成、但在匯入關係刪除之前、仍會從外部陣列提供 I/O 。

刪除關係之前、您必須確認所有 LUN 的移轉程序已完成。

```
Cluster01::*> lun import show -vserver vserver1 -fields foreign-
disk,path,operational-state
vserver   foreign-disk path              operational-state
--------- ------------ ----------------- -----------------
vserver1 800DT$HuVWB/ /vol/new_asm/LUN4 completed
vserver1 800DT$HuVWBW /vol/new_asm/LUN0 completed
vserver1 800DT$HuVWBX /vol/new_asm/LUN1 completed
vserver1 800DT$HuVWBY /vol/new_asm/LUN2 completed
vserver1 800DT$HuVWBZ /vol/new_asm/LUN3 completed
vserver1 800DT$HuVWBa /vol/new_asm/LUN5 completed
vserver1 800DT$HuVWBb /vol/new_asm/LUN6 completed
vserver1 800DT$HuVWBc /vol/new_asm/LUN7 completed
vserver1 800DT$HuVWBd /vol/new_asm/LUN8 completed
vserver1 800DT$HuVWBe /vol/new_asm/LUN9 completed
vserver1 800DT$HuVWBf /vol/new_lvm/LUN0 completed
vserver1 800DT$HuVWBg /vol/new_lvm/LUN1 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWBi /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWBl /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWBo /vol/new_lvm/LUN9 completed
20 entries were displayed.
```

刪除匯入關係

移轉程序完成後、請刪除移轉關係。完成後、 I/O 將由 ONTAP 上的磁碟機獨家提供。

```
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN0
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_asm/LUN1
...
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN8
Cluster01::*> lun import delete -vserver vserver1 -path /vol/new_lvm/LUN9
```

取消註冊外部 **LUN**

最後、修改磁碟以移除 `is-foreign` 指定。

```
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign false
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign false
Cluster01::*>
```

使用 **FLI** 進行 **Oracle** 移轉：傳輸協定轉換

變更用於存取 LUN 的傳輸協定是常見的需求。

在某些情況下、它是將資料移轉至雲端的整體策略的一部分。TCP/IP 是雲端的傳輸協定、從 FC 變更為 iSCSI 可讓您更輕鬆地移轉至各種雲端環境。在其他情況下、 iSCSI 可能需要善用 IP SAN 降低的成本。有時候、移轉可能會使用不同的傳輸協定作為臨時措施。例如、如果外部陣列和 ONTAP 型 LUN 無法共存於同一個 HBA 上、您可以使用足夠長的 iSCSI LUN 、從舊陣列複製資料。然後、您可以在從系統移除舊 LUN 之後、將其轉換回 FC 。

下列程序示範從 FC 轉換至 iSCSI 的過程、但整體原則適用於從 iSCSI 轉換至 FC 的反轉過程。

安裝 **iSCSI** 啟動器

大多數作業系統預設都包含軟體 iSCSI 啟動器、但如果不包含軟體 iSCSI 啟動器、則可輕鬆安裝。

```
[root@host1 /]# yum install -y iscsi-initiator-utils
Loaded plugins: langpacks, product-id, search-disabled-repos,
subscription-
              : manager
Resolving Dependencies
--> Running transaction check
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.el7 will be
updated
--> Processing Dependency: iscsi-initiator-utils = 6.2.0.873-32.el7 for
package: iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7 will be
an update
--> Running transaction check
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.el7 will
be updated
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
will be an update
--> Finished Dependency Resolution
Dependencies Resolved
```

```
================================================================================
===
 Package                         Arch   Version              Repository
Size
================================================================================
===
Updating:
 iscsi-initiator-utils           x86_64 6.2.0.873-32.0.2.el7 ol7_latest 416
k
Updating for dependencies:
 iscsi-initiator-utils-iscsiuio x86_64 6.2.0.873-32.0.2.el7 ol7_latest  84
k
Transaction Summary
================================================================================
===
Upgrade  1 Package (+1 Dependent package)
Total download size: 501 k
Downloading packages:
No Presto metadata available for ol7_latest
(1/2): iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_6 | 416 kB   00:00
(2/2): iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2. |  84 kB   00:00
--------------------------------------------------------------------------------
---
Total                                           2.8 MB/s | 501 kB
00:00Cluster01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
1/4
  Updating   : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
2/4
  Cleanup    : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
  Cleanup    : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
4/4
rhel-7-server-eus-rpms/7Server/x86_64/productid       | 1.7 kB     00:00
rhel-7-server-rpms/7Server/x86_64/productid           | 1.7 kB     00:00
  Verifying  : iscsi-initiator-utils-6.2.0.873-32.0.2.el7.x86_64
1/4
  Verifying  : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.0.2.el7.x86
2/4
  Verifying  : iscsi-initiator-utils-iscsiuio-6.2.0.873-32.el7.x86_64
3/4
  Verifying  : iscsi-initiator-utils-6.2.0.873-32.el7.x86_64
```

```
4/4
Updated:
  iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.el7
Dependency Updated:
  iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.el7
Complete!
[root@host1 /]#
```

識別 **iSCSI** 啟動器名稱

在安裝過程中會產生唯一的 iSCSI 啟動器名稱。在 Linux 上、它位於 `/etc/iscsi/initiatorname.iscsi` 檔案：此名稱用於識別 IP SAN 上的主機。

```
[root@host1 /]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1992-05.com.redhat:497bd66ca0
```

建立新的啟動器群組

啟動器群組（ igroup ）是 ONTAP LUN 遮罩架構的一部分。除非先授予主機存取權、否則無法存取新建立的 LUN 。此步驟的完成方法是建立一個 igroup 、列出需要存取的 FC WWN 或 iSCSI 啟動器名稱。

在此範例中、會建立包含 Linux 主機 iSCSI 啟動器的 igroup 。

```
Cluster01::*> igroup create -igroup linuxiscsi -protocol iscsi -ostype
linux -initiator iqn.1994-05.com.redhat:497bd66ca0
```

關閉環境

變更 LUN 傳輸協定之前、必須完全禁用 LUN 。任何要轉換的 LUN 上的資料庫都必須關機、檔案系統必須卸載、而且必須停用磁碟區群組。使用 ASM 時、請確定已卸除 ASM 磁碟群組、並關閉所有網格服務。

從 **FC** 網路取消對應 **LUN**

LUN 完全禁用後、請從原始 FC igroup 移除對應。

```
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxhost
...
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxhost
Cluster01::*> lun unmap -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxhost
```

將 LUN 重新對應至 IP 網路

將每個 LUN 的存取權授予新的 iSCSI 型啟動器群組。

```
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN0 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_asm/LUN1 -igroup
linuxiscsi
...
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN8 -igroup
linuxiscsi
Cluster01::*> lun map -vserver vserver1 -path /vol/new_lvm/LUN9 -igroup
linuxiscsi
Cluster01::*>
```

探索 iSCSI 目標

iSCSI 探索分為兩個階段。第一是探索目標、這與探索 LUN 不同。。 `iscsiadm` 下列命令會探查指定的入口網站群組 `-p argument` 並儲存提供 iSCSI 服務的所有 IP 位址和連接埠清單。在這種情況下、預設連接埠 3260 上有四個 iSCSI 服務的 IP 位址。

如果無法到達任何目標 IP 位址、此命令可能需要幾分鐘的時間才能完成。

```
[root@host1 ~]# iscsiadm -m discovery -t st -p fas8060-iscsi-public1
10.63.147.197:3260,1033 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
10.63.147.198:3260,1034 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.203:3260,1030 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
172.20.108.202:3260,1029 iqn.1992-
08.com.netapp:sn.807615e9ef6111e5a5ae90e2ba5b9464:vs.3
```

探索 iSCSI LUN

發現 iSCSI 目標後、請重新啟動 iSCSI 服務以探索可用的 iSCSI LUN 、並建置相關裝置、例如多重路徑或 ASMLib 裝置。

```
[root@host1 ~]# service iscsi restart
Redirecting to /bin/systemctl restart  iscsi.service
```

重新啟動環境

重新啟動 Volume 群組、重新掛載檔案系統、重新啟動 RAC 服務等、以重新啟動環境。為了預防這種情況、

NetApp 建議您在轉換程序完成後重新啟動伺服器、以確保所有組態檔案都正確無誤、並移除所有過時的裝置。

注意：在重新啟動主機之前、請確定中的所有項目都已存在 `/etc/fstab` 這項參照移轉的 SAN 資源會被註解出來。如果未執行此步驟、且 LUN 存取有問題、則可能是無法開機的作業系統。此問題不會損壞資料。不過、開機進入救援模式或類似模式進行修正可能會非常不方便 `/etc/fstab` 這樣就能啟動作業系統、開始進行疑難排解工作。

Oracle 移轉程序範例指令碼

提供的指令碼是如何為各種作業系統和資料庫工作撰寫指令碼的範例。這些都是依現狀供應。如果特定程序需要支援、請聯絡 NetApp 或 NetApp 經銷商。

資料庫關機

下列 Perl 指令碼會採用 Oracle SID 的單一引數、並關閉資料庫。它可以以 Oracle 使用者或 root 身分執行。

```
#! /usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
77 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
'
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF4
sqlplus / as sysdba << EOF2
shutdown immediate;
EOF2
`;};
print @out;
if ("@out" =~ /ORACLE instance shut down/) {
print "$oraclesid shut down\n";
exit 0;}
elsif ("@out" =~ /Connected to an idle instance/) {
print "$oraclesid already shut down\n";
exit 0;}
else {
print "$oraclesid failed to shut down\n";
exit 1;}
```

資料庫啟動

下列 Perl 指令碼會採用 Oracle SID 的單一引數、並關閉資料庫。它可以以 Oracle 使用者或 root 身分執行。

```
#! /usr/bin/perl
use strict;
use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
my $uid=$<;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
'
`;}
else {
@out=`. oraenv << EOF3
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
startup;
EOF2
`;};
print @out;
if ("@out" =~ /Database opened/) {
print "$oraclesid started\n";
exit 0;}
elsif ("@out" =~ /cannot start already-running ORACLE/) {
print "$oraclesid already started\n";
exit 1;}
else {
78 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
print "$oraclesid failed to start\n";
exit 1;}
```

將檔案系統轉換為唯讀

下列指令碼會採用檔案系統引數、並嘗試將其卸除並重新掛載為唯讀。在移轉過程中、這樣做非常有用、因為必須將檔案系統保留在可複寫資料的位置、但必須保護其免於意外損壞。

```
#! /usr/bin/perl
use strict;
#use warnings;
my $filesystem=$ARGV[0];
my @out=`umount '$filesystem'`;
if ($? == 0) {
 print "$filesystem unmounted\n";
 @out = `mount -o ro '$filesystem'`;
 if ($? == 0) {
 print "$filesystem mounted read-only\n";
 exit 0;}}
else {
 print "Unable to unmount $filesystem\n";
 exit 1;}
print @out;
```

取代檔案系統

下列指令碼範例用於將一個檔案系統取代為另一個檔案系統。因爲它編輯了 /etc/fstab 文件，所以它必須以 root 身份運行。它接受新舊檔案系統的單一逗號分隔引數。

1. 若要取代檔案系統、請執行下列指令碼：

```
#! /usr/bin/perl
use strict;
#use warnings;
my $oldfs;
my $newfs;
my @oldfstab;
my @newfstab;
my $source;
my $mountpoint;
my $leftover;
my $oldfstabentry='';
my $newfstabentry='';
my $migratedfstabentry='';
($oldfs, $newfs) = split (',',$ARGV[0]);
open(my $filehandle, '<', '/etc/fstab') or die "Could not open
/etc/fstab\n";
while (my $line = <$filehandle>) {
 chomp $line;
 ($source, $mountpoint, $leftover) = split(/[ , ]/,$line, 3);
 if ($mountpoint eq $oldfs) {
 $oldfstabentry = "#Removed by swap script $source $oldfs $leftover";}
 elsif ($mountpoint eq $newfs) {
```

```
 $newfstabentry = "#Removed by swap script $source $newfs $leftover";
 $migratedfstabentry = "$source $oldfs $leftover";}
 else {
 push (@newfstab, "$line\n")}}
79 Migration of Oracle Databases to NetApp Storage Systems © 2021
NetApp, Inc. All rights reserved
push (@newfstab, "$oldfstabentry\n");
push (@newfstab, "$newfstabentry\n");
push (@newfstab, "$migratedfstabentry\n");
close($filehandle);
if ($oldfstabentry eq ''){
 die "Could not find $oldfs in /etc/fstab\n";}
if ($newfstabentry eq ''){
 die "Could not find $newfs in /etc/fstab\n";}
my @out=`umount '$newfs'`;
if ($? == 0) {
 print "$newfs unmounted\n";}
else {
 print "Unable to unmount $newfs\n";
 exit 1;}
@out=`umount '$oldfs'`;
if ($? == 0) {
 print "$oldfs unmounted\n";}
else {
 print "Unable to unmount $oldfs\n";
 exit 1;}
system("cp /etc/fstab /etc/fstab.bak");
open ($filehandle, ">", '/etc/fstab') or die "Could not open /etc/fstab
for writing\n";
for my $line (@newfstab) {
 print $filehandle $line;}
close($filehandle);
@out=`mount '$oldfs'`;
if ($? == 0) {
 print "Mounted updated $oldfs\n";
 exit 0;}
else{
 print "Unable to mount updated $oldfs\n";
 exit 1;}
exit 0;
```

以本指令碼的使用範例為例、假設中的資料 `/oradata` 移轉至 `/neworadata` 和 `/logs` 移轉至 `/newlogs`。執行此工作最簡單的方法之一、就是使用簡單的檔案複製作業、將新裝置重新放置回原始安裝點。

2. 假設舊的和新的檔案系統存在於中 `/etc/fstab` 檔案如下：

```
cluster01:/vol_oradata /oradata nfs rw,bg,vers=3,rsize=65536,wsize=65536
0 0
cluster01:/vol_logs /logs nfs rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_newlogs /newlogs nfs rw,bg,vers=3,rsize=65536,wsize=65536
0 0
```

3. 執行時、此指令碼會卸載目前的檔案系統、並以新的：

```
[root@jfsc3 scripts]# ./swap.fs.pl /oradata,/neworadata
/neworadata unmounted
/oradata unmounted
Mounted updated /oradata
[root@jfsc3 scripts]# ./swap.fs.pl /logs,/newlogs
/newlogs unmounted
/logs unmounted
Mounted updated /logs
```

4. 指令碼也會更新 `/etc/fstab` 請據此歸檔。在此處所示範例中、包含下列變更：

```
#Removed by swap script cluster01:/vol_oradata /oradata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_neworadata /oradata nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_logs /logs nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
#Removed by swap script cluster01:/vol_newlogs /newlogs nfs
rw,bg,vers=3,rsize=65536,wsize=65536 0 0
cluster01:/vol_newlogs /logs nfs rw,bg,vers=3,rsize=65536,wsize=65536 0
0
```

自動化資料庫移轉

此範例示範如何使用關機、啟動及檔案系統置換指令碼來完全自動化移轉。

```
#! /usr/bin/perl
use strict;
#use warnings;
my $oraclesid=$ARGV[0];
```

```
my @oldfs;
my @newfs;
my $x=1;
while ($x < scalar(@ARGV)) {
   ($oldfs[$x-1], $newfs[$x-1]) = split (',',$ARGV[$x]);
   $x+=1;}
my @out=`./dbshut.pl '$oraclesid'`;
print @out;
if ($? ne 0) {
   print "Failed to shut down database\n";
   exit 0;}
$x=0;
while ($x < scalar(@oldfs)) {
   my @out=`./mk.fs.readonly.pl '$oldfs[$x]'`;
   if ($? ne 0) {
      print "Failed to make filesystem $oldfs[$x] readonly\n";
      exit 0;}
   $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
   my @out=`rsync -rlpogt --stats --progress --exclude='.snapshot'
'$oldfs[$x]/' '/$newfs[$x]/'`;
   print @out;
   if ($? ne 0) {
      print "Failed to copy filesystem $oldfs[$x] to $newfs[$x]\n";
      exit 0;}
   else {
      print "Succesfully replicated filesystem $oldfs[$x] to
$newfs[$x]\n";}
   $x+=1;}
$x=0;
while ($x < scalar(@oldfs)) {
   print "swap $x $oldfs[$x] $newfs[$x]\n";
   my @out=`./swap.fs.pl '$oldfs[$x],$newfs[$x]'`;
   print @out;
   if ($? ne 0) {
      print "Failed to swap filesystem $oldfs[$x] for $newfs[$x]\n";
      exit 1;}
   else {
      print "Swapped filesystem $oldfs[$x] for $newfs[$x]\n";}
   $x+=1;}
my @out=`./dbstart.pl '$oraclesid'`;
print @out;
```

顯示檔案位置

此指令碼會收集許多重要的資料庫參數、並以易讀的格式列印。此指令碼在檢閱資料配置時非常實用。此外、指令碼也可以修改以供其他用途使用。

```
#! /usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
        my $command = @_[0];
        my @lines;
        my $uid=$<;
        if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
        `;}
        else {
        $command=~s/\\\\\\\/\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
        `;};
return @lines}
print "\n";
@out=dosql('select name from v\\\\\$datafile;');
print "$oraclesid datafiles:\n";
for $line (@out) {
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select member from v\\\\\$logfile;');
print "$oraclesid redo logs:\n";
for $line (@out) {
```

```
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name from v\\\\\$tempfile;');
print "$oraclesid temp datafiles:\n";
for $line (@out) {
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('show parameter spfile;');
print "$oraclesid spfile\n";
for $line (@out) {
        chomp($line);
        if (length($line)>0) {print "$line\n";}}
print "\n";
@out=dosql('select name||\' \'||value from v\\\\\$parameter where
isdefault=\'FALSE\';');
print "$oraclesid key parameters\n";
for $line (@out) {
        chomp($line);
        if ($line =~ /control_files/) {print "$line\n";}
        if ($line =~ /db_create/) {print "$line\n";}
        if ($line =~ /db_file_name_convert/) {print "$line\n";}
        if ($line =~ /log_archive_dest/) {print "$line\n";}}
        if ($line =~ /log_file_name_convert/) {print "$line\n";}
        if ($line =~ /pdb_file_name_convert/) {print "$line\n";}
        if ($line =~ /spfile/) {print "$line\n";}
print "\n";
```

ASM 移轉清理

```
#! /usr/bin/perl
#use strict;
#use warnings;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my @out;
sub dosql{
        my $command = @_[0];
        my @lines;
        my $uid=$<;
        if ($uid == 0) {
        @lines=`su - $oracleuser -c "export ORAENV_ASK=NO;export
ORACLE_SID=$oraclesid;. oraenv -s << EOF1
EOF1
```

```
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
"
        `;}
        else {
        $command=~s/\\\\\\\/\\/g;
        @lines=`export ORAENV_ASK=NO;export ORACLE_SID=$oraclesid;. oraenv
-s << EOF1
EOF1
sqlplus -S / as sysdba << EOF2
set heading off
$command
EOF2
        `;}
return @lines}
print "\n";
@out=dosql('select name from v\\\\\$datafile;');
print @out;
print "shutdown immediate;\n";
print "startup mount;\n";
print "\n";
for $line (@out) {
        if (length($line) > 1) {
                chomp($line);
                ($first, $second,$third,$fourth)=split('_',$line);
                $fourth =~ s/^TS-//;
                $newname=lc("$fourth.dbf");
                $path2file=$line;
                $path2file=~ /(^.*.\/)/;
                print "host mv $line $1$newname\n";}}
print "\n";
for $line (@out) {
        if (length($line) > 1) {
                chomp($line);
                ($first, $second,$third,$fourth)=split('_',$line);
                $fourth =~ s/^TS-//;
                $newname=lc("$fourth.dbf");
                $path2file=$line;
                $path2file=~ /(^.*.\/)/;
                print "alter database rename file '$line' to
'$1$newname';\n";}}
print "alter database open;\n";
print "\n";
```

ASM 至檔案系統名稱轉換

```
set serveroutput on;
set wrap off;
declare
    cursor df is select file#, name from v$datafile;
    cursor tf is select file#, name from v$tempfile;
    cursor lf is select member from v$logfile;
    firstline boolean := true;
begin
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('Parameters for log file conversion:');
    dbms_output.put_line(CHR(13));
    dbms_output.put('*.log_file_name_convert = ');
    for lfrec in lf loop
        if (firstline = true) then
            dbms_output.put('''' || lfrec.member || ''', ');
            dbms_output.put('''/NEW_PATH/' ||
regexp_replace(lfrec.member,'^.*/','') || '''');
        else
            dbms_output.put(',''' || lfrec.member || ''', ');
            dbms_output.put('''/NEW_PATH/' ||
regexp_replace(lfrec.member,'^.*/','') || '''');
        end if;
        firstline:=false;
    end loop;
    dbms_output.put_line(CHR(13));
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('rman duplication script:');
    dbms_output.put_line(CHR(13));
    dbms_output.put_line('run');
    dbms_output.put_line('{');
    for dfrec in df loop
        dbms_output.put_line('set newname for datafile ' ||
            dfrec.file# || ' to ''' || dfrec.name ||''';');
    end loop;
    for tfrec in tf loop
        dbms_output.put_line('set newname for tempfile ' ||
            tfrec.file# || ' to ''' || tfrec.name ||''';');
    end loop;
    dbms_output.put_line('duplicate target database for standby backup
location INSERT_PATH_HERE;');
    dbms_output.put_line('}');
end;
/
```

在資料庫上重新播放記錄

此指令碼接受 Oracle SID 的單一引數、用於處於掛載模式的資料庫、並嘗試重新播放所有目前可用的歸檔記錄。

```
#! /usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
84 Migration of Oracle Databases to NetApp Storage Systems © 2021 NetApp,
Inc. All rights reserved
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
'
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover database until cancel;
auto
EOF2
`;
}
print @out;
```

在待命資料庫上重新播放記錄

此指令碼與前述指令碼相同、但其設計用於待命資料庫。

```
#! /usr/bin/perl
use strict;
my $oraclesid=$ARGV[0];
my $oracleuser='oracle';
my $uid = $<;
my @out;
if ($uid == 0) {
@out=`su - $oracleuser -c '. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
'
`;}
else {
@out=`. oraenv << EOF1
$oraclesid
EOF1
sqlplus / as sysdba << EOF2
recover standby database until cancel;
auto
EOF2
`;
}
print @out;
```

其他附註

Oracle 資料庫效能最佳化與基準測試程序

準確測試資料庫儲存效能是極為複雜的主題。需要瞭解下列問題：

- IOPS 與處理量
- 前景與背景 I/O 作業之間的差異
- 延遲對資料庫的影響
- 許多作業系統和網路設定也會影響儲存效能

此外、還有非儲存資料庫工作要考量。最佳化儲存效能並不會帶來實用效益、因為儲存效能不再是效能的限制因素。

大多數資料庫客戶現在都選擇 All Flash Array 、這會造成一些額外考量。例如、請考慮在雙節點 AFF A900 系統上進行效能測試：

- 有了 80/20 讀取 / 寫入比率、兩個 A900 節點可在延遲甚至超過 150 μ s 標記之前、提供超過 1M 的隨機資料庫 IOPS 。這遠遠超出了大多數資料庫目前的效能需求、很難預測預期的改善。儲存設備將會大幅清除、成為瓶頸。
- 網路頻寬是效能限制的常見來源。例如、旋轉式磁碟解決方案通常是資料庫效能的瓶頸、因為 I/O 延遲非常高。當 All Flash 陣列移除延遲限制時、障礙會頻繁移轉至網路。在虛擬化環境和刀鋒系統中、這一點尤其顯著、因為這些環境和刀鋒系統的真正網路連線能力難以視覺化。如果由於頻寬限制而無法充分利用儲存系統本身、這可能會使效能測試變得複雜。
- 由於 All Flash 陣列的延遲大幅改善、因此一般無法將 All Flash 陣列與含有旋轉磁碟的陣列進行效能比較。測試結果通常沒有意義。
- 將尖峰 IOPS 效能與 All Flash 陣列進行比較通常並不實用、因為資料庫不受儲存 I/O 限制例如、假設一個陣列可維持 500K 的隨機 IOPS 、而另一個陣列則可維持 300K 。如果資料庫花費 99% 的時間處理 CPU 、這種差異在現實世界中是不相關的。工作負載永遠不會使用儲存陣列的完整功能。相反地、尖峰 IOPS 功能在整合平台中可能非常重要、而在整合平台中、儲存陣列預期會載入至尖峰容量。
- 請務必在任何儲存測試中考慮延遲和 IOPS 。市面上許多儲存陣列都宣稱 IOPS 極高、但延遲卻使這些 IOPS 在這類層級上無法使用。使用 All Flash Array 的典型目標是 1 毫秒標記。更好的測試方法不是測量最大可能的 IOPS 、而是判斷儲存陣列在平均延遲大於 1 毫秒之前可以維持多少 IOPS 。

Oracle 自動工作負載儲存庫與基準測試

Oracle 效能比較的黃金標準是 Oracle 自動工作負載儲存庫（ AWR ）報告。

有多種類型的 AWR 報告。從儲存點來看、執行所產生的報告 `awrrpt.sql` Command 是最全面且最有價值的命令、因為它針對特定資料庫執行個體、並包含一些詳細的分佈圖、可根據延遲來中斷儲存 I/O 事件。

比較兩個效能陣列的理想方法是在每個陣列上執行相同的工作負載、並產生精確鎖定工作負載的 AWR 報告。在執行時間極長的工作負載中、可以使用包含開始和停止時間的單一 AWR 報告、但最好將 AWR 資料分成多份報告。例如、如果批次工作從午夜執行至上午 6 點、請建立一系列從午夜– 1 點、上午 1 點– 2 點開始的一小時 AWR 報告。

在其他情況下、應最佳化非常簡短的查詢。最佳選項是以查詢開始時建立的 AWR 快照為基礎的 AWR 報告、以及在查詢結束時建立的第二個 AWR 快照。否則資料庫伺服器應保持安靜、以將會使分析中查詢活動模糊的背景活動降至最低。

如果無法取得 AWR 報告、 Oracle 狀態報告是一個很好的替代方案。它們包含與 AWR 報告相同的大部分 I/O 統計資料。

Oracle AWR 與疑難排解

AWR 報告也是分析效能問題的最重要工具。

與基準測試一樣、效能疑難排解也需要您精確測量特定工作負載。如果可能、請在向 NetApp 支援中心回報效能問題、或與 NetApp 或合作夥伴客戶團隊合作、討論新解決方案時提供 AWR 資料。

提供 AWR 資料時、請考量下列需求：

- 執行 `awrrpt.sql` 產生報告的命令。輸出可以是文字或 HTML 。
- 如果使用 Oracle Real Application Clusters （ RAC ）、請為叢集中的每個執行個體產生 AWR 報告。
- 鎖定問題存在的特定時間。AWR 報告的最長可接受使用時間通常為一小時。如果問題持續數小時、或涉及多小時作業、例如批次工作、請提供多個一小時的 AWR 報告、涵蓋整個分析期間。

- 如有可能、請將 AWR 快照時間間隔調整為 15 分鐘。此設定可執行更詳細的分析。這也需要額外執行 `awrrpt.sql` 提供每 15 分鐘間隔的報告。
- 如果問題是非常短的執行查詢、請根據作業開始時建立的 AWR 快照、以及作業結束時建立的第二個 AWR 快照、提供 AWR 報告。否則、資料庫伺服器應保持安靜、以將會使分析中作業的活動受到影響的背景活動減至最低。
- 如果在特定時間回報效能問題、但未在其他時間回報、請提供額外的 AWR 資料、以展現良好的效能來進行比較。

calibr_IO

。 `calibrate_io` 絕對不可使用命令來測試、比較或基準測試儲存系統。如 Oracle 文件所述、本程序會校正儲存設備的 I/O 功能。

校準與基準測試不同。此命令的目的是發佈 I/O 、藉由最佳化發行給主機的 I/O 層級、協助校正資料庫作業並改善其效率。因為執行的 I/O 類型 `calibrate_io` 作業並不代表實際的資料庫使用者 I/O 、結果無法預測、而且經常無法重現。

SLOB2

SLOB2 是愚蠢的小 Oracle 基準測試工具、已成為評估資料庫效能的首選工具。這是由 Kevin Closson 開發的、可在取得 "https://kevinclosson.net/slob/"。安裝和設定需要幾分鐘的時間、它使用實際的 Oracle 資料庫來在使用者可定義的資料表空間上產生 I/O 模式。這是少數幾種可用的測試選項之一、可將全快閃陣列與 I/O 飽和它也有助於產生更低層級的 I/O 、以模擬低 IOPS 但對延遲敏感的儲存工作負載。

交換台工作台

交換基準台可用於測試資料庫效能、但使用交換基準台的方式會對儲存造成壓力、這是非常困難的。NetApp 尚未從 SwingWorkbench 中看到任何測試結果、這些測試產生足夠的 I/O 、使其在任何 AFF 陣列上都成為重大負載。在有限的情況下、訂單輸入測試（ OET ）可用於從延遲點評估儲存設備。這在資料庫對於特定查詢具有已知延遲相依性的情況下很有用。必須注意確保主機和網路已正確設定、以實現 All Flash 陣列的延遲潛力。

HammerDB

HammerDB 是一種資料庫測試工具、可模擬 TPC-C 和 TPC-H 基準測試等。建構足夠大的資料集以正確執行測試可能需要很長時間、但它可以是評估 OLTP 和資料倉儲應用程式效能的有效工具。

Orion

Oracle Orion 工具通常與 Oracle 9 搭配使用、但並未加以維護、以確保與各種主機作業系統的變更相容。由於與作業系統和儲存組態不相容、因此很少與 Oracle 10 或 Oracle 11 搭配使用。

Oracle 重新編寫了該工具，默認情況下，該工具與 Oracle 12c 一起安裝。雖然本產品已經過改良、並使用許多與實際 Oracle 資料庫相同的呼叫、但它並未使用 Oracle 所使用的相同程式碼路徑或 I/O 行為。例如、大部分的 Oracle I/O 都是同步執行、這表示資料庫會暫停、直到 I/O 作業在前景完成為止。只是以隨機 I/O 淹沒儲存系統、並不是真正的 Oracle I/O 複製、也無法提供直接的儲存陣列比較方法、也無法測量組態變更的影響。

也就是說、 Orion 有一些使用案例、例如一般測量特定主機網路儲存組態的最大可能效能、或是測量儲存系統的健全狀況。仔細測試後、只要參數包括 IOPS 、處理量和延遲的考量、並嘗試忠實複寫真實的工作負載、就能設計出可用的 Orion 測試來比較儲存陣列或評估組態變更的影響。

過時的 NFSv3 鎖定和 Oracle 資料庫

如果 Oracle 資料庫伺服器當機、則重新啟動時可能會發生過時的 NFS 鎖定問題。請仔細注意伺服器上的名稱解析設定、以避免此問題。

產生此問題的原因是建立鎖定和清除鎖定會使用兩種稍微不同的名稱解析方法。涉及兩個過程： Network Lock Manager （ NLM ）和 NFS 用戶端。NLM 使用 `uname n` 以決定主機名稱、而 `rpc.statd` 程序用途 `gethostbyname()`。這些主機名稱必須相符、作業系統才能正確清除過時的鎖定。例如、主機可能正在尋找擁有的鎖定 `dbserver5`但鎖定已由主機登錄為 `dbserver5.mydomain.org`。如果 `gethostbyname()` 不會傳回與相同的值 `uname -a`，則鎖定釋放程序未成功。

下列範例指令碼會驗證名稱解析是否完全一致：

```
#! /usr/bin/perl
$uname=`uname -n`;
chomp($uname);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $uname;
print "uname -n yields: $uname\n";
print "gethostbyname yields: $name\n";
```

如果 `gethostbyname` 不符 `uname`、可能是過時的鎖定。例如、此結果顯示潛在問題：

```
uname -n yields: dbserver5
gethostbyname yields: dbserver5.mydomain.org
```

解決方案通常是透過變更主機出現在中的順序來找到 `/etc/hosts`。例如、假設 hosts 檔案包含下列項目：

```
10.156.110.201  dbserver5.mydomain.org dbserver5 loghost
```

若要解決此問題、請變更完整網域名稱和簡短主機名稱出現的順序：

```
10.156.110.201  dbserver5 dbserver5.mydomain.org loghost
```

`gethostbyname()` 現在傳回短 `dbserver5` 主機名稱、符合的輸出 `uname`。因此、鎖定會在伺服器當機後自動清除。

Oracle 資料庫的 WAFL 對齊驗證

正確的 WAFL 對齊對於良好的效能至關重要。雖然 ONTAP 以 4KB 單位管理區塊、但這並不表示 ONTAP 以 4KB 單位執行所有作業。事實上、 ONTAP 支援不同大小的區塊作業、但基礎會計是由 WAFL 以 4KB 為單位進行管理。

「對齊」一詞是指 Oracle I/O 與這些 4KB 單元的相對應方式。最佳效能要求 Oracle 8KB 區塊位於磁碟機上的

兩個 4KB WAFL 實體區塊上。如果區塊偏移 2KB 、則此區塊會位於一半的 4KB 區塊、一個獨立的完整 4KB 區塊、然後是第三個 4KB 區塊的一半。這種安排會導致效能降低。

對齊並不涉及 NAS 檔案系統。Oracle 資料檔案會根據 Oracle 區塊的大小、與檔案的開頭對齊。因此、 8KB 、 16KB 和 32KB 的區塊大小一律會對齊。所有區塊作業都會從檔案開頭偏移、單位為 4 KB 。

相反地、 LUN 在啟動時通常會包含某種驅動程式標頭或檔案系統中繼資料、以建立偏移。對齊在現代作業系統中很少是個問題、因為這些作業系統是專為可能使用原生 4KB 磁碟區的實體磁碟機所設計、因此也需要將 I/O 與 4KB 邊界對齊才能獲得最佳效能。

不過、有一些例外情況。資料庫可能已從未針對 4KB I/O 最佳化的舊版作業系統移轉、或是在建立分割區時發生使用者錯誤、可能導致偏移量、而大小單位不是 4KB 。

下列範例僅適用於 Linux 、但程序可適用於任何作業系統。

一致

以下範例顯示單一磁碟分割的單一 LUN 對齊檢查。

首先、建立使用磁碟機上所有可用分割區的分割區。

```
[root@host0 iscsi]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xb97f94c1.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240):
Using default value 10240
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
[root@host0 iscsi]#
```

您可以使用下列命令以數學方式檢查對齊方式：

```
[root@host0 iscsi]# fdisk -u -l /dev/sdb
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              32    20971519    10485744   83  Linux
```

輸出顯示單位為 512 位元組、且分割區的開頭為 32 個單位。總共 32 x 512 = 16 、 834 位元組、這是 4KB WAFL 區塊的整數倍數。此分割區已正確對齊。

若要驗證正確的對齊方式、請完成下列步驟：

1. 識別 LUN 的通用唯一識別碼（ UUID ）。

```
FAS8040SAP::> lun show -v /vol/jfs_luns/lun0
              Vserver Name: jfs
                  LUN UUID: ed95d953-1560-4f74-9006-85b352f58fcd
                    Mapped: mapped`                `
```

2. 進入 ONTAP 控制器上的節點 Shell 。

```
FAS8040SAP::> node run -node FAS8040SAP-02
Type 'exit' or 'Ctrl-D' to return to the CLI
FAS8040SAP-02> set advanced
set not found.  Type '?' for a list of commands
FAS8040SAP-02> priv set advanced
Warning: These advanced commands are potentially dangerous; use
         them only when directed to do so by NetApp
         personnel.
```

3. 在第一步中識別的目標 UUID 上開始收集統計資料。

```
FAS8040SAP-02*> stats start lun:ed95d953-1560-4f74-9006-85b352f58fcd
Stats identifier name is 'Ind0xffffff08b9536188'
FAS8040SAP-02*>
```

4. 執行一些 I/O請務必使用 `iflag` 用於確保 I/O 同步且無緩衝的引數。

> 請務必小心使用此命令。反轉 `if` 和 `of` 引數會破壞資料。

```
[root@host0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000
bs=4096
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. 停止統計資料並檢視對齊分佈圖。所有 I/O 都應位於 `.0` 貯體、表示 I/O 與 4KB 區塊邊界對齊。

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff08b9536188
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:186%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
```

未對齊

以下範例顯示 I/O 未對齊：

1. 建立不符合 4KB 邊界的分割區。這不是現代作業系統的預設行為。

```
[root@host0 iscsi]# fdisk -u /dev/sdb
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (32-20971519, default 32): 33
Last sector, +sectors or +size{K,M,G} (33-20971519, default 20971519):
Using default value 20971519
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

2. 已建立磁碟分割、並使用 33 磁區偏移值、而非預設的 32 。重複中所述的程序 "一致"。直方圖顯示如下：

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-
4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:136%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_partial_blocks:31%
```

錯誤的對齊是顯而易見的。I/O 大多落在 * 之中 *.1 符合預期偏移的貯體。建立分割區時、它會比最佳化的預設值更進一步移入 512 個位元組、這表示長條圖偏移 512 個位元組。

此外 `read_partial_blocks` 統計資料為非零、這表示執行的 I/O 並未填滿整個 4KB 區塊。

重作記錄

此處說明的程序適用於資料檔案。Oracle 重做記錄和歸檔記錄檔有不同的 I/O 模式。例如、重做記錄是單一檔案的循環覆寫。如果使用預設的 512 位元組區塊大小、寫入統計資料看起來會像這樣：

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-
9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.0:12%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.1:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.3:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.4:13%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.5:6%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.6:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.7:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_partial_blocks:85%
```

I/O 會分散到所有分佈式分佈區、但這並不是效能考量。不過、重做記錄率極高可能會因為使用 4KB 區塊大小而受惠。在這種情況下、最好確定重做記錄 LUN 已正確對齊。不過、這對於資料檔案對齊的良好效能並不重要。