



Oracle 資料庫移轉

Enterprise applications

NetApp
May 09, 2024

目錄

Oracle 資料庫移轉	1
將 Oracle 資料庫移轉至 ONTAP 儲存系統	1
Oracle 資料庫移轉規劃	1
程序	3
Oracle 移轉程序範例指令碼	101

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。

[錯誤：缺少圖形影像]

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額外設定、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

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率嚴重限制。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為遷移的數據庫重新創建控制文件。

```
SQL> alter database backup controlfile to trace as '/tmp/waffle.ctl';
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.ctl /tmp/
oracle@jfsc1's password:
waffle.ctl                                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/control02.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@jpsc3 arch]$ scp jpsc1:/logs/WAFFLE/arch/* ./
oracle@jpsc1'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@jfscl arch]# cat /etc/rsyncd.conf
[waffle.arch]
  uid=oracle
  gid=dba
  path=/logs/WAFFLE/arch
  read only = true
[root@jfscl 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/lgr6c161_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/lhr6c164_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 `/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@jpsc2 ~]$ mkdir /oradata/pancake
[oracle@jpsc2 ~]$ mkdir /logs/pancake
[oracle@jpsc2 ~]$ cd /orabin/admin
[oracle@jpsc2 admin]$ mkdir PANCAKE
[oracle@jpsc2 admin]$ cd PANCAKE
[oracle@jpsc2 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@jfscl 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@jfscl 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說的指令碼 ["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@jpsc2 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@jpsc2 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,wsiz=65536,addr=172.20.101.10)
ontap-nfs1:/host1_logs on /logs type nfs
(ro,bg,vers=3,rsize=65536,wsiz=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,rsiz=65536,wsiz=65536,addr=172.20.101.10)
ontap-nfs1:/host1_oradata_new on /oradata type nfs
(rw,bg,vers=3,rsiz=65536,wsiz=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
ASMCMDB> umount DATA
ASMCMDB>

```

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@jfscl ~]$ cd $ORACLE_HOME/dbs  
[oracle@jfscl dbs]$ cat initTOAST.ora  
SPFILE='+ASM0/TOAST/spfileTOAST.ora'  
[oracle@jfscl dbs]$ mv initTOAST.ora initTOAST.ora.prev  
[oracle@jfscl 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.ctrl
output file name=/logs/TOAST/redo/control02.ctrl
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@jfscl 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@jfscl 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$HuVWBW
FOR-1.4   800DT$HuVWBV
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$HuVWBh
FOR-1.14  800DT$HuVWBh
FOR-1.15  800DT$HuVWBj
FOR-1.16  800DT$HuVWBk
FOR-1.17  800DT$HuVWBm
FOR-1.18  800DT$HuVWBn
FOR-1.19  800DT$HuVWBn
FOR-1.20  800DT$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$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -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 為 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$HuVWBX /vol/new_asm/LUN1 6
vserver1   800DT$HuVWBZ /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$HuVWBh /vol/new_lvm/LUN2 4
vserver1   800DT$HuVWBh /vol/new_lvm/LUN3 3
vserver1   800DT$HuVWBj /vol/new_lvm/LUN4 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN6 4
vserver1   800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1   800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1   800DT$HuVWBn /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 fpCzdLTuKfy2xDZjailNliJh3TjLUBiT: 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$HuVWBX /vol/new_asm/LUN1  completed
vserver1  800DT$HuVWBZ /vol/new_asm/LUN2  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$HuVWBW} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVVBX} -is
-foreign false
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVVBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$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.e17
```

Dependency Updated:

```
iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.e17
```

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。因為它編輯了 /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,wsiz=65536
0 0
cluster01:/vol_logs /logs nfs rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_newlogs /newlogs nfs rw,bg,vers=3,rsize=65536,wsiz=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@jpsc3 scripts]# ./swap.fs.pl /oradata,/neworadata
/neworadata unmounted
/oradata unmounted
Mounted updated /oradata
[root@jpsc3 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,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_neworadata /neworadata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_neworadata /oradata nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_logs /logs nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
#Removed by swap script cluster01:/vol_newlogs /newlogs nfs
rw,bg,vers=3,rsize=65536,wsiz=65536 0 0
cluster01:/vol_newlogs /logs nfs rw,bg,vers=3,rsize=65536,wsiz=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;

```

版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

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

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

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

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

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

商標資訊

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