



Oracle 데이터베이스 Enterprise applications

NetApp
May 03, 2024

목차

Oracle 데이터베이스	1
ONTAP 기반의 Oracle 데이터베이스	1
ONTAP 구성	1
데이터베이스 구성	11
호스트 구성	14
네트워크 구성	29
스토리지 구성	36
Oracle 데이터베이스 가상화	51
계층화	53
Oracle 데이터 보호	61
Oracle 재해 복구	82
Oracle 데이터베이스 마이그레이션	106
추가 참고 사항	223

Oracle 데이터베이스

ONTAP 기반의 Oracle 데이터베이스

ONTAP는 Oracle 데이터베이스를 위해 설계되었습니다. 지난 수십 년 동안 ONTAP은 관계형 데이터베이스 I/O의 고유한 요구사항에 맞게 최적화되었으며, Oracle 데이터베이스의 필요에 따라 여러 ONTAP 기능이 특별히 개발되었으며, 심지어 Oracle Inc. 자체의 요청도 있었습니다.



이 문서는 이전에 게시된 기술 보고서_TR-3633: ONTAP 기반 Oracle 데이터베이스, TR-4591: Oracle 데이터 보호: 백업, 복구, 복제, TR-4592: MetroCluster 기반 Oracle, TR-4534: NetApp 스토리지 시스템으로 Oracle 데이터베이스 마이그레이션 _을(를) 대체합니다

ONTAP이 데이터베이스 환경에 가치를 제공하는 다양한 방법 외에도 데이터베이스 규모, 성능 요구사항, 데이터 보호 요구사항 등 다양한 사용자 요구사항이 있습니다. 알려진 NetApp 스토리지 구축에는 Mware ESX에서 약 6,000개의 데이터베이스가 실행되는 가상 환경부터 현재 규모 996TB에서 계속 증가하는 단일 인스턴스 데이터 웨어하우스까지 모든 것이 포함됩니다. 따라서 NetApp 스토리지에 Oracle 데이터베이스를 구성하는 데는 명백한 몇 가지 모범 사례가 있습니다.

NetApp 스토리지에서 Oracle 데이터베이스 작동이라는 요구사항은 두 가지 방식으로 해결됩니다. 첫째, 확실한 모범 사례가 있을 경우 이를 구체적으로 언급합니다. 상위 레벨에서는 특정 비즈니스 요구사항에 따라 Oracle 스토리지 솔루션 설계자가 다루어야 할 여러 설계 고려사항에 대해 설명할 것입니다.

ONTAP 구성

RAID 및 Oracle 데이터베이스

RAID는 드라이브 손실로부터 데이터를 보호하기 위해 이중화를 사용하는 것을 말합니다.

Oracle 데이터베이스 및 기타 엔터프라이즈 애플리케이션에 사용되는 NetApp 스토리지를 구성할 때 RAID 레벨에 관한 질문이 가끔 제기됩니다. 스토리지 어레이 구성과 관련된 기존의 여러 Oracle 모범 사례에는 RAID 미러링 사용 및/또는 특정 유형의 RAID 회피에 대한 주의사항이 포함되어 있습니다. 그러나 이러한 소스는 유효한 점을 제시하지만 ONTAP에 사용되는 NetApp RAID DP 및 RAID-TEC 기술과 RAID 4에는 적용되지 않습니다.

RAID 4, RAID 5, RAID 6, RAID DP 및 RAID-TEC는 모두 패리티를 사용하여 드라이브 오류로 인해 데이터가 손실되지 않도록 합니다. 이들 RAID 옵션은 미러링에 비해 스토리지 사용률이 훨씬 뛰어나지만 대부분의 RAID 구현에서는 쓰기 작업이 영향을 받는다는 문제가 있습니다. 다른 RAID 구현에서 쓰기 작업을 완료하려면 다중 드라이브 읽기를 통해 패리티 데이터를 재생성해야 할 수 있으며, 이를 일반적으로 RAID 성능 저하라고 합니다.

그러나 ONTAP에서는 이 RAID 성능 저하가 발생하지 않습니다. 이는 NetApp WAFL(Write Anywhere File Layout)과 RAID 계층의 통합 때문입니다. 쓰기 작업은 RAM에 병합되며 패리티 생성을 포함하여 전체 RAID 스트라이프로 준비됩니다. ONTAP는 쓰기를 완료하기 위해 읽기를 수행할 필요가 없으며, 이는 ONTAP 및 WAFL에서 RAID 성능 저하가 일어나지 않는다는 뜻입니다. 로깅 재실행과 같이 지연 시간이 중요한 작업의 성능은 영향을 받지 않으며, 랜덤 데이터 파일 쓰기에는 패리티를 다시 생성해야 하므로 RAID 성능 저하가 발생하지 않습니다.

통계적 안정성을 고려하면 RAID DP도 RAID 미러링보다 우수한 보호 기능을 제공합니다. 기본적인 문제는 RAID 리빌드 중에 드라이브에서 발생하는 수요입니다. 미러링된 RAID 세트가 있으면 RAID 세트에서 파트너에 대해 리빌드를 하는 동안 드라이브 장애로 인해 데이터가 손실될 위험이 RAID DP 세트에서 삼중 드라이브 장애가 일어날 위험보다 훨씬 큽니다.

Oracle 데이터베이스 및 스토리지 용량 관리

예측 가능하고 관리가 용이한 고성능 엔터프라이즈 스토리지로 데이터베이스나 기타 엔터프라이즈 애플리케이션을 관리하려면 데이터 및 메타데이터 관리를 위해 드라이브에 여유 공간이 필요합니다. 필요한 여유 공간의 양은 사용하는 드라이브의 유형과 비즈니스 프로세스에 따라 다릅니다.

여유 공간은 실제 데이터에 사용되지 않는 모든 공간을 의미하며 애그리게이트 자체의 미할당 공간과 구성 볼륨 내의 미사용 공간을 포함합니다. 씬 프로비저닝도 고려해야 합니다. 예를 들어, 볼륨에 1TB LUN이 포함되어 있는데 그 중 50%만 실제 데이터에 의해 사용되고 있습니다. 씬 프로비저닝된 환경에서는 500GB의 공간이 사용되는 것으로 올바르게 표시되지만 완전히 프로비저닝된 환경에서는 1TB의 전체 용량이 사용 중인 것으로 표시되고 할당되지 않은 500GB의 공간은 숨겨집니다. 이 공간은 실제 데이터에 의해 사용되지 않으므로 총 여유 공간 계산 시 포함되어야 합니다.

엔터프라이즈 애플리케이션에 사용되는 스토리지 시스템에 적용되는 NetApp 권장사항은 다음과 같습니다.

AFF 시스템을 포함한, SSD 애그리게이트



* NetApp는 * 최소 10%의 여유 공간을 권장합니다. 여기에는 애그리게이트 또는 볼륨 내 여유 공간은 물론, 완전 프로비저닝 사용으로 인해 할당은 되었으나 실제 데이터에 의해 사용되지 않고 있는 여유 공간을 포함해 모든 미사용 공간이 포함됩니다. 논리적 공간은 중요하지 않습니다. 문제는 데이터 스토리지에 사용할 수 있는 실제 사용 가능한 물리적 공간의 양입니다.

10%의 여유 공간은 보수적인 권장사항입니다. SSD 애그리게이트는 활용률 수준이 높을 때에도 성능에 영향을 주지 않고 워크로드를 지원할 수 있습니다. 하지만 활용률을 신중하게 모니터링하지 않으면 애그리게이트의 활용률이 높아질수록 공간이 소진될 위험도 높아집니다. 또한 99% 용량으로 시스템을 실행하는 경우 성능 저하가 발생하지 않지만, 추가 하드웨어를 주문하는 동안 시스템이 완전히 채워지지 않도록 관리 노력이 필요하며, 추가 드라이브를 구입하고 설치하는 데 시간이 걸릴 수 있습니다.

Flash Pool 애그리게이트를 포함한, HDD 애그리게이트



*NetApp는 회전식 드라이브를 사용할 때 최소 15%의 여유 공간을 사용할 것을 권장합니다. 여기에는 애그리게이트 또는 볼륨 내 여유 공간은 물론, 완전 프로비저닝 사용으로 인해 할당은 되었으나 실제 데이터에 의해 사용되지 않고 있는 여유 공간을 포함해 모든 미사용 공간이 포함됩니다. 10%에 달하는 자유 음성이 성능에 영향을 미칩니다.

Oracle 데이터베이스 및 스토리지 가상 머신

Oracle 데이터베이스 스토리지 관리가 SVM(Storage Virtual Machine)에서 중앙 집중화됨

ONTAP CLI에서 SVM은 스토리지의 기본 기능 단위로서, SVM을 VMware ESX 서버의 게스트에 비교하면 유용합니다.

ESX는 처음 설치했을 때 게스트 운영 체제의 호스팅이나 최종 사용자 애플리케이션 지원 같은 사전 구성된 기능을 제공하지 않으며 가상 머신(VM)이 정의될 때까지는 빈 컨테이너입니다. ONTAP도 비슷합니다. ONTAP를 처음 설치했을 때는 SVM이 생성될 때까지 데이터 제공 기능이 없습니다. 데이터 서비스를 정의하는 것은 SVM의 특성입니다.

스토리지 아키텍처의 다른 측면과 마찬가지로 SVM 및 논리 인터페이스(LIF) 설계를 위한 최고의 옵션은 확장 요구사항과 비즈니스 요구사항에 크게 의존합니다.

SVM

ONTAP용 SVM 프로비저닝을 위한 공식적인 모범 사례는 없습니다. 올바른 접근 방식은 관리 및 보안 요구 사항에 따라 달라집니다.

대부분의 고객은 대부분의 일상적인 요구사항을 위해 하나의 1차 SVM을 작동하고, 이후 특별한 요구사항을 위해 소수의 SVM을 생성합니다. 예를 들어 다음을 만들 수 있습니다.

- 전문가 팀이 관리하는 주요 비즈니스 데이터베이스를 위한 SVM
- 자체 스토리지를 독립적으로 관리할 수 있는 전체 제어 권한을 부여받은 개발 그룹을 위한 SVM
- 관리 팀에 액세스 제한이 적용되어야 할 인사 관리 또는 재무 보고 데이터 등 민감한 비즈니스 데이터를 위한 SVM

멀티 테넌트 환경에서는 각 테넌트의 데이터에 전용 SVM이 제공될 수 있습니다. 클러스터당 SVM 및 LIF 수의 제한은 사용되는 프로토콜, 노드 모델 및 ONTAP 버전에 따라 달라집니다. 을 참조하십시오 ["NetApp Hardware Universe를 참조하십시오"](#) 이 제한 사항에 대해 알아보겠습니다.

ONTAP QoS를 사용하여 Oracle 데이터베이스 성능을 관리합니다

여러 Oracle 데이터베이스를 안전하고 효율적으로 관리하려면 효과적인 QoS 전략이 필요합니다. 그 이유는 최신 스토리지 시스템의 성능 용량이 계속 늘어나고 있기 때문입니다.

특히, All-Flash 스토리지의 채택이 늘어나면서 워크로드를 통합할 수 있었습니다. 회전식 미디어를 사용하는 스토리지 어레이는 구식 회전식 드라이브 기술로 인해 IOPS 용량에 제한이 있어 소수의 I/O 집약적인 워크로드만 지원하는 경향이 있었습니다. 고수준의 액티브 데이터베이스 1개 또는 2개로 인해 스토리지 컨트롤러가 한계에 도달하기 훨씬 전에 디스크가 한계에 다다른 것 같았지만, 이런 현상이 바뀌었습니다. 상대적으로 적은 수의 SSD 드라이브가 가진 성능 용량이 가장 강력한 스토리지 컨트롤러도 포화시킬 수 있습니다. 이는 회전식 미디어 지연 시간이 급증할 때 갑작스러운 성능 하락을 걱정할 일 없이 컨트롤러의 최대 용량을 활용할 수 있다는 뜻입니다.

참조할 만한 예를 들어보면, 단순한 2노드 HA AFF A800 시스템은 지연 시간이 1밀리초를 넘는 수준으로 증가하기 전에 최대 1백만 랜덤 IOPS를 충족할 수 있습니다. 이러한 수준에 도달하는 단일 워크로드도 거의 없을 것입니다. 이 AFF A800 시스템 어레이를 최대한 활용하려면 여러 워크로드를 호스팅하고 안전하게 수행하는 동시에 예측 가능성을 보장하기 위해 QoS 제어가 필요합니다.

ONTAP에는 IOPS와 대역폭이라는 두 가지 서비스 품질(QoS) 유형이 있습니다. QoS 제어는 SVM, 볼륨, LUN, 파일에 적용됩니다.

IOPS QoS입니다

IOPS QoS 제어는 확실히 특정 리소스의 총 IOPS가 기준이 되지만 IOPS QoS는 직관적이지 않은 여러 측면을 가지고 있습니다. IOPS 임계값에 도달했을 때 지연 시간이 뚜렷이 증가되는 현상에 몇몇 고객은 초기에 혼란을 느꼈습니다. 지연 시간 증가는 IOPS 제한으로 인한 당연한 결과입니다. 논리적으로는 토큰 시스템과 비슷하게 작동합니다. 예를 들어, 특정 볼륨에 10K IOPS 제한이 있는 데이터 파일이 포함된 경우 유입되는 각 I/O에서 처리가 계속되려면 먼저 토큰이 수신되어야 합니다. 초당 10K 이상의 토큰이 소비되지 않는 한 지연은 없습니다. 입출력 작업이 토큰을 수신하기 위해 대기해야 하는 경우 이 대기는 추가 지연 시간으로 나타납니다. 워크로드가 QoS 제한에 근접할수록 각 IO는 해당 회전이 처리될 때까지 대기열에서 대기해야 하는 시간이 길어집니다. 이는 사용자에게 더 높은 지연 시간으로 나타납니다.



데이터베이스 트랜잭션/재실행 로그 데이터에 QoS 제어를 적용할 때는 주의하십시오. 로깅 재실행의 성능 요구사항은 일반적으로 데이터 파일보다 훨씬 낮지만, 재실행 로그 활동은 폭주합니다. 입출력은 짧은 펄스로 이루어지며, 평균 REDO 입출력 레벨에 적합한 QoS 제한은 실제 요구 사항에 비해 너무 낮을 수 있습니다. 그 결과 QoS가 각 재실행 로그 버스트에 적용되므로 심각한 성능 제한이 발생할 수 있습니다. 일반적으로 재실행 및 아카이브 로깅은 QoS에 의해 제한되지 않습니다.

대역폭 QoS

모든 I/O 크기가 같지는 않습니다. 예를 들어, 데이터베이스가 여러 개의 작은 블록 읽기를 수행하여 IOPS 임계값에 도달할 수 있고 또한 데이터베이스는 매우 적은 수의 대규모 블록 읽기로 구성된 전체 테이블 스캔 작업을 수행할 수도 있으며, 이는 대역폭은 많이, IOPS는 상대적으로 적게 소비합니다.

마찬가지로 VMware 환경은 부팅 중에 매우 많은 수의 랜덤 IOPS를 구동할 수 있지만 외부 백업 중에는 적은 수의 입출력을 수행하지만 더 큰 입출력을 수행할 수 있습니다.

성능을 효과적으로 관리하려면 IOPS 또는 대역폭 QoS 제한이나 둘 다 필요합니다.

최소/QoS 보장

많은 고객들이 QoS 보장이 포함된 솔루션을 모색하는데, 이는 생각보다 달성하기 어렵고 낭비가 될 가능성이 있습니다. 예를 들어 10개의 데이터베이스를 10K IOPS 보장으로 배치하려면 10개의 데이터베이스 모두가 동시에 10K IOPS로 실행되는 시나리오의 경우 총 100,000에 대해 시스템을 사이징해야 합니다.

최소 QoS 제어를 위한 최선의 사용은 중요 워크로드를 보호하는 것입니다. 예를 들어 최대 IOPS가 500K이고 운영 워크로드와 개발 워크로드를 혼합할 수 있는 ONTAP 컨트롤러를 가정해 보겠습니다. 특정 데이터베이스가 컨트롤러를 독점하지 못하도록 개발 워크로드에 최대 QoS 정책을 적용해야 합니다. 그런 다음 운영 워크로드에 최소 QoS 정책을 적용하여 필요할 때 항상 필요한 IOPS를 사용할 수 있도록 합니다.

적응형 QoS

적응형 QoS는 QoS 제한이 스토리지 오브젝트의 용량을 기반으로 하는 ONTAP 기능을 가리킵니다. 일반적으로 데이터베이스 크기와 성능 요구사항 간에는 어떠한 링크도 없기 때문에 데이터베이스에는 거의 사용되지 않습니다. 대규모 데이터베이스는 거의 불활성 상태가 될 수 있지만, 작은 데이터베이스는 가장 IOPS를 많이 포함할 수 있습니다.

적응형 QoS는 이러한 데이터 세트의 IOPS 요구사항이 데이터베이스의 총 크기와 상관하는 경향이 있기 때문에 가상화 데이터 저장소와 함께 매우 유용할 수 있습니다. 1TB의 VMDK 파일이 포함된 최신 데이터 저장소에는 2TB 데이터 저장소보다 절반의 성능이 필요할 수 있습니다. 적응형 QoS를 사용하면 데이터 저장소가 데이터로 채워질 때 QoS 제한을 자동으로 늘릴 수 있습니다.

Oracle 데이터베이스 및 ONTAP 효율성 기능

ONTAP 공간 효율성 기능은 Oracle 데이터베이스에 최적화되어 있습니다. 거의 모든 경우에 최상의 접근 방식은 모든 효율성 기능을 활성화한 상태에서 기본값을 그대로 유지하는 것입니다.

압축, 컴팩션, 중복제거와 같은 공간 효율성 기능은 지정된 양의 물리적 스토리지에 적합한 논리적 데이터의 양을 늘리기 위해 설계되었습니다. 결과적으로 비용과 관리 부담이 줄어듭니다.

상위 수준에서 압축은 수학적 프로세스이며, 그 패턴은 공간 요구사항을 감소시키는 방식으로 데이터를 감지하고 인코딩합니다. 이와 반대로, 중복제거는 실제로 반복되는 데이터 블록을 감지하여 불필요한 복사본을 제거합니다. 컴팩션을 사용하면 여러 개의 논리적 데이터 블록이 미디어에서 동일한 물리적 블록을 공유할 수 있습니다.



스토리지 효율성과 부분 예약 간의 상호 작용에 대한 설명은 아래의 씬 프로비저닝에 대한 섹션을 참조하십시오.

압축

All-Flash 스토리지 시스템을 사용할 수 이전에는 어레이 기반 압축의 값이 제한되었습니다. 대부분의 I/O 집약적인 워크로드에는 허용되는 성능을 제공하기 위해 매우 많은 수의 스피들이 필요했기 때문입니다. 스토리지 시스템에는 항상 많은 수의 드라이브가 부작용으로 필요한 것보다 훨씬 많은 용량이 포함되어 있습니다. 그러나 솔리드 스테이트 스토리지가 부상하면서 상황이 달라졌습니다. 이제 우수한 성능을 얻기 위해 드라이브를 엄청나게 오버 프로비저닝하지 않아도 됩니다. 스토리지 시스템의 드라이브 공간은 실제 용량 요구 사항과 일치할 수 있습니다.

솔리드 스테이트 드라이브(SSD)의 IOPS 용량이 증가하면 대개 회전식 드라이브에 비해 비용이 절감되며 압축 덕분에 솔리드 스테이트 미디어의 실제 용량이 늘어나 추가 절감을 달성할 수 있습니다.

데이터를 압축하는 방법에는 여러 가지가 있습니다. 대부분의 데이터베이스에는 자체 압축 기능이 포함되어 있지만 고객 환경에서는 이런 일이 거의 발생하지 않습니다. 그 이유는 일반적으로 압축된 데이터로 * 변경 * 시 성능 저하가 발생하며 일부 애플리케이션의 경우 데이터베이스 수준 압축에 대한 라이선스 비용이 많이 듭니다. 마지막으로, 데이터베이스 작업의 전반적인 성능에 영향을 미칩니다. 실제 데이터베이스 작업이 아닌 데이터 압축과 압축 해제를 수행하는 CPU를 위해 CPU당 라이선스 비용으로 높은 금액을 지불하는 것은 합리적이지 않습니다. 더 좋은 옵션은 압축 작업을 스토리지 시스템으로 오프로드하는 것입니다.

적응형 압축

적응형 압축은 지연 시간이 마이크로초 단위로 측정되는 All-Flash 환경에서조차 성능에 미치는 영향 없이 엔터프라이즈 워크로드에 철저히 테스트되었습니다. 심지어 일부 고객은 데이터가 캐시에 압축된 상태로 남아 있으므로 압축을 사용하여 성능이 향상되었다고 보고했습니다. 따라서 컨트롤러에서 가용 캐시의 양이 실질적으로 증가하기 때문입니다.

ONTAP는 4KB 유닛의 물리적 블록을 관리하며 적응형 압축은 기본 압축 블록 크기 8KB를 사용하며, 이는 데이터가 8KB 단위로 압축된다는 것을 의미합니다. 이 크기는 관계형 데이터베이스에서 가장 많이 사용되는 8KB 블록 크기와 일치합니다. 압축 알고리즘은 단일 유닛으로 더 많은 데이터가 압축되므로 효율성이 더욱 향상됩니다. 32KB의 압축 블록 크기는 8KB 압축 블록 유닛보다 더 공간 효율적입니다. 이는 기본 8KB 블록 크기를 사용하는 적응형 압축을 사용하면 효율성이 약간 낮지만 압축 블록 크기를 더 작게 만들면 큰 이점이 있습니다. 데이터베이스 워크로드에는 많은 양의 덮어쓰기 활동이 포함됩니다. 압축된 32KB 데이터 블록의 8KB를 덮어쓰려면 전체 32KB의 논리적 데이터를 다시 읽고, 압축을 풀고, 필요한 8KB 영역을 업데이트하고, 재압축을 수행한 다음 전체 32KB를 드라이브에 다시 써야 합니다. 이는 스토리지 시스템의 경우 매우 많은 비용이 드는 작업이며, 이로 인해 압축 블록 크기가 더 큰 경쟁 스토리지 어레이에서도 데이터베이스 워크로드의 성능이 크게 저하될 수 있습니다.



적응형 압축에서 사용되는 블록 크기는 32KB까지 늘릴 수 있습니다. 이렇게 하면 스토리지 효율성이 향상될 수 있으며, 스토리지에 상당한 양의 데이터가 저장될 경우 트랜잭션 로그 및 백업 파일과 같은 대기 상태의 파일에 대해 고려해야 합니다. 경우에 따라 16KB 또는 32KB 블록 크기를 사용하는 액티브 데이터베이스가 이에 맞춰 적응형 압축의 블록 크기를 늘렸을 수도 있습니다. NetApp 또는 파트너 담당자에게 문의하여 이 솔루션이 현재 워크로드에 적합한지 여부를 확인하십시오.



8KB보다 큰 압축 블록 크기는 스트리밍 백업 대상에서 중복제거와 함께 사용해서는 안 됩니다. 백업된 데이터의 작은 변화가 32KB 압축 기간에 영향을 미치기 때문입니다. 기간이 바뀌면 그에 따라 파일 전체에서 압축된 데이터가 달라집니다. 압축 후 중복제거가 발생하며, 이는 중복제거 엔진이 압축된 각 백업을 다르게 간주한다는 의미입니다. 스트리밍 백업의 중복제거가 필요한 경우 8KB 블록 적응형 압축만 사용해야 합니다. 더 작은 블록 크기를 사용할 수 있고 중복제거 효율성을 방해하지 않기 때문에 적응형 압축이 더 낫습니다. 유사한 이유로 호스트 측 압축도 중복제거 효율성에 지장을 줍니다.

압축 정렬

데이터베이스 환경에서 적응형 압축을 수행할 때는 압축 블록 정렬과 관련된 몇 가지 사항을 고려해야 합니다. 이는 특정 블록의 랜덤 덮어쓰기가 데이터에 적용되는 경우만 해당합니다. 이 접근 방식은 파일 시스템의 시작이 4K 디바이스 경계에 맞춰 정렬되어야 하고 파일 시스템의 블록 크기가 4K의 배수여야 하는 전체 파일 시스템 정렬과 개념이 비슷합니다.

예를 들어, 파일에 대한 8KB 쓰기는 파일 시스템 자체 내에서 8KB 경계와 일치하는 경우에만 압축됩니다. 즉 파일의 첫 번째 8KB에, 두 번째 8KB 에 그리고 그 이후로도 동일하게 포함되어야 합니다. 올바른 정렬을 보장하는 가장 간단한 방법은 올바른 LUN 유형을 사용하는 것입니다. 생성된 모든 파티션은 8K의 배수인 디바이스의 시작 부분에서 오프셋을 가지며 데이터베이스 블록 크기의 배수인 파일 시스템 블록 크기를 사용해야 합니다.

백업이나 트랜잭션 로그 같은 데이터는 압축된 여러 블록을 확장하는 순차적 쓰기 작업이며 따라서 정렬을 고려할 필요가 없습니다. I/O 패턴에서 고려해야 할 한 가지는 파일의 랜덤 덮어쓰기입니다.

데이터 컴팩션

데이터 컴팩션은 압축 효율성을 향상하는 기술입니다. 앞서 설명한 것처럼, 적응형 압축만 사용했을 때는 절감 비율이 최대 2:1입니다. 4KB WAFL 블록에 8KB I/O를 저장하도록 제한되어 있기 때문입니다. 블록 크기가 더 큰 압축 방법을 통해 효율성이 향상됩니다. 그러나 이러한 복사본은 작은 블록 덮어쓰기가 적용되는 데이터에는 적합하지 않습니다. 32KB 단위 데이터의 압축 해제, 8KB 부분 업데이트, 재압축, 드라이브에 다시 쓰기 작업은 오버헤드를 발생시킵니다.

데이터 컴팩션은 여러 논리적 블록이 물리적 블록 내에 저장될 수 있게 합니다. 예를 들어, 텍스트 또는 부분 전체 블록과 같이 고도로 압축 가능한 데이터가 포함된 데이터베이스는 8KB에서 1KB로 압축될 수 있습니다. 컴팩션을 적용하지 않으면 이 1KB 데이터는 여전히 4KB 블록 전체를 점유할 것입니다. 인라인 데이터 컴팩션에서는 압축된 데이터 1KB를 다른 압축된 데이터와 함께 단 1KB의 물리적 공간에 저장할 수 있습니다. 이 방식은 압축 기술이 아니라 그저 드라이브의 공간을 더 효율적으로 할당하는 것이며 감지할 수 있는 성능 영향을 발생시키지 않습니다.

이로써 얻어지는 절감의 수준은 다양합니다. 이미 압축되었거나 암호화된 데이터는 일반적으로 더 압축할 수 없기 때문에 이 데이터 세트는 컴팩션의 이점을 얻지 못합니다. 반면 제로와 블록 메타데이터보다 조금 더 많이 포함하고 있으며 새롭게 초기화된 데이터 파일의 경우 80:1까지 압축합니다.

온도에 민감한 스토리지 효율성

TSSE(Temperature Sensitive Storage Efficiency)는 블록 액세스 히트 맵을 사용하여 자주 액세스하지 않는 블록을 식별하고 보다 효율적으로 압축하는 ONTAP 9.8 이상에서 사용할 수 있습니다.

중복 제거

중복 제거는 데이터 세트에서 중복된 블록 크기가 제거됩니다. 예를 들어, 동일한 4KB 블록이 10개 파일에 존재하면 중복제거는 파일 10개 전체에서 해당 4KB 블록을 동일한 4KB 물리적 블록으로 리디렉션합니다. 그 결과 데이터의 효율성이 10:1로 향상됩니다.

VMware 게스트 부팅 LUN과 같은 데이터는 동일한 운영 체제 파일의 여러 복사본으로 구성되어 있기 때문에 중복 제거가 매우 용이합니다. 100:1 이상의 효율성이 관찰되었습니다.

일부 데이터에 중복 데이터가 없습니다. 예를 들어, Oracle 블록에는 데이터베이스에 관해 전역적으로 고유한 헤더와 거의 고유한 트레일러가 포함되어 있습니다. 따라서 Oracle 데이터베이스의 중복 제거 기능을 사용하면 1%를 넘는 비용을 절감하는 경우는 거의 없습니다. MS SQL 데이터베이스의 중복 제거는 약간 더 낮지만 블록 수준에서 고유한 메타데이터는 여전히 제한 사항입니다.

일부 경우 16KB 및 대형 블록 크기의 데이터베이스에서 공간이 최대 15% 절약되었습니다. 각 블록의 처음 4KB에는 전역적으로 고유한 헤더가 포함되어 있고 마지막 4KB 블록에는 거의 고유한 트레일러가 포함되어 있습니다. 실제로는 거의 전적으로 제로화 데이터의 중복제거에 기인하지만 내부 블록은 중복제거 후보입니다.

많은 경쟁업체의 어레이는 데이터베이스가 여러 차례 복사된다는 추정을 기반으로 데이터베이스의 중복제거 기능을 내세웁니다. 이런 측면에서 NetApp 중복제거도 사용할 수 있지만 ONTAP은 더 나은 옵션인 NetApp FlexClone 기술을 제공합니다. 최종 결과는 같으며 기본 물리적 블록의 대부분을 공유하는 데이터베이스의 복사본이 여러 개 생성됩니다. FlexClone은 시간을 들여 데이터베이스 파일을 복사한 다음 중복제거하는 것보다 훨씬 더 효율적입니다. 실제로 이는 중복제거가 아니라 비중복이라 할 수 있습니다. 애초에 중복을 생성하지 않기 때문입니다.

효율성 및 씬 프로비저닝

효율성 기능은 씬 프로비저닝의 한 형태입니다. 예를 들어, 100GB 볼륨을 점유하는 100GB LUN은 50GB까지 압축할 수 있을 것이고 볼륨은 여전히 100GB이기 때문에 실제로 절감이 실현되지는 않았습니다. 먼저 볼륨의 크기를 줄여 절감된 공간을 시스템의 어느 곳에서든 사용할 수 있게 해야 합니다. 나중에 100GB LUN으로 변경하면 데이터 압축률이 줄어들어 LUN 크기가 커지고 볼륨을 가득 채울 수 있습니다.

씬 프로비저닝은 관리를 단순화하는 동시에 가용 용량을 크게 개선하면서 비용을 절감할 수 있기 때문에 적극 권장합니다. 단순한 데이터베이스 환경에서 많은 빈 공간, 많은 수의 볼륨 및 LUN, 압축 가능한 데이터가 포함되는 경우가 많습니다. 일반 프로비저닝은 언젠가 100% 채워지고 100% 압축할 수 없는 데이터가 포함될 경우에 대비해 볼륨 및 LUN에 대한 스토리지 공간을 예약합니다. 그런 일은 일어나지 않을 것입니다. 씬 프로비저닝을 사용하면 공간을 재확보하고 다른 위치에서 사용할 수 있으며 더 작은 볼륨 및 LUN이 아닌 스토리지 시스템 자체를 기반으로 용량을 관리할 수 있습니다.

일부 고객은 특정 워크로드에 대해 또는 일반적으로 확립된 운영 및 조달 사례를 기반으로 일반 프로비저닝을 사용하는 것을 선호합니다.

- 주의: * 볼륨이 일반 프로비저닝되면 압축 해제 및 를 사용한 중복 제거 제거를 포함하여 해당 볼륨에 대한 모든 효율성 기능을 완전히 비활성화하도록 주의해야 합니다 `sis undo` 명령. 볼륨은 에 나타나지 않아야 합니다 `volume efficiency show` 출력. 그렇지 않을 경우, 효율성 기능을 위해 볼륨이 부분적으로 구성됩니다. 결과적으로 덮어쓰기 보장은 서로 다르게 동작하므로 구성 과다 사용으로 인해 볼륨의 공간이 예기치 않게 부족해져서 데이터베이스 I/O 오류가 발생할 가능성이 높아집니다.

효율성 모범 사례

NetApp에서 권장하는 사항은 다음과 같습니다.

AFF 기본값

All-Flash AFF 시스템에서 실행되는 ONTAP에서 생성된 볼륨은 모든 인라인 효율성 기능을 사용하는 씬 프로비저닝됩니다. 일반적으로 데이터베이스에는 중복제거를 통해 이점을 얻을 수 없고 압축할 수 없는 데이터가 포함될 수 있지만 그럼에도 불구하고 기본 설정은 거의 모든 워크로드에 적합합니다. ONTAP는 절감 여부와 관계없이 모든 유형의 데이터와 I/O 패턴을 효율적으로 처리하도록 설계되었습니다. 원인을 완전히 이해하고 편차가 있는 경우에만 기본값을 변경해야 합니다.

일반 권장 사항

- 볼륨 및/또는 LUN이 씬 프로비저닝되지 않는 경우 모든 효율성 설정을 비활성화해야 합니다. 이러한 기능을 사용하면 절감 효과가 없고 일반 프로비저닝과 공간 효율성이 활성화된 조합을 통해 공간 부족 오류를 포함하여 예기치 않은 동작이 발생할 수 있기 때문입니다.
- 백업 또는 데이터베이스 트랜잭션 로그와 같이 데이터를 덮어쓰지 않는 경우 냉각 기간이 짧은 TSSE를 활성화하여 효율성을 높일 수 있습니다.
- 일부 파일에는 압축할 수 없는 많은 양의 데이터가 포함되어 있을 수 있습니다. 예를 들어 파일의 응용 프로그램 수준에서 압축이 이미 활성화되어 있는 경우 암호화됩니다. 이러한 시나리오가 적용되는 경우 압축 데이터를 포함하는 다른 볼륨에서 더 효율적으로 작업할 수 있도록 압축을 해제하는 것이 좋습니다.

- 데이터베이스 백업에 32KB 압축 및 중복제거를 모두 사용하지 마십시오. 섹션을 참조하십시오 [적응형 압축](#) 를 참조하십시오.

Oracle 데이터베이스를 사용한 씬 프로비저닝

Oracle 데이터베이스에 대한 씬 프로비저닝은 결국 실제로 사용할 수 있는 공간보다 더 많은 공간을 스토리지 시스템에 구성하게 되므로 신중한 계획이 필요합니다. 올바르게 수행하면 상당한 비용 절감 및 관리 효율성 향상을 얻을 수 있기 때문에 이러한 노력을 기울일 가치가 있습니다.

씬 프로비저닝은 다양한 형태로 제공되며 ONTAP이 엔터프라이즈 애플리케이션 환경에 제공하는 여러 기능 중 핵심이 됩니다. 또한 씬 프로비저닝은 효율성 기술과 밀접하게 관련되어 있습니다. 즉, 스토리지 시스템에서 기술적으로 있는 것보다 더 많은 논리적 데이터를 저장할 수 있다는 것입니다.

거의 모든 경우 스냅샷 사용에는 씬 프로비저닝이 수반됩니다. 예를 들어, NetApp 스토리지의 일반적인 10TB 데이터베이스에는 약 30일 스냅샷이 포함되어 있습니다. 이 방식을 통해 액티브 파일 시스템에서 약 10TB의 데이터가 표시되고 스냅샷 전용으로는 300TB가 표시됩니다. 보통 총 310TB의 스토리지가 12TB~15TB의 공간에 상주합니다. 원래 데이터의 변경사항만 저장되기 때문에 액티브 데이터베이스가 10TB를 소비하며 나머지 300TB 데이터에는 2TB~5TB의 공간만 필요합니다.

클로닝도 씬 프로비저닝의 한 예입니다. 주요 NetApp 고객은 개발에 사용하기 위해 80TB 데이터베이스의 클론 40개를 생성했습니다. 이 클론을 사용하는 40명 모두가 모든 데이터 파일의 모든 블록을 덮어쓴다면 3.2PB를 넘는 스토리지가 필요할 것입니다. 변경사항만 드라이브에 저장되기 때문에 실제로 턴오버가 낮으며 총체적인 공간 요구사항은 40TB에 가깝습니다.

공간 관리

데이터 변경률이 예기치 않게 증가할 수 있기 때문에 애플리케이션 환경에서 씬 프로비저닝을 수행할 때 주의를 기울여야 합니다. 예를 들어 데이터베이스 테이블을 다시 인덱싱하거나 VMware 게스트에 대규모 패치를 적용하면 스냅샷으로 인한 공간 소비가 빠르게 증가할 수 있습니다. 백업을 잘못 배치하면 매우 짧은 시간에 많은 양의 데이터를 쓸 수 있습니다. 마지막으로, 파일 시스템에 예기치 않게 사용 가능한 공간이 부족할 경우 일부 응용 프로그램을 복구하기가 어려울 수 있습니다.

하지만 을 신중하게 구성하면 이러한 위험을 해결할 수 있습니다 `volume-autogrow` 및 `snapshot-autodelete` 정책. 그 이름이 시사하듯 이들 옵션은 사용자가 스냅샷에 의해 소비되는 공간을 자동으로 삭제하거나 추가 데이터를 수용하기 위해 볼륨을 늘리는 정책을 수립하도록 지원합니다. 다양한 옵션을 사용할 수 있으며 요구사항은 고객에 따라 다릅니다.

를 참조하십시오 "[논리적 스토리지 관리 설명서](#)" 이러한 기능에 대한 자세한 내용은

부분 예약

부분 예약은 공간 효율성과 관련하여 볼륨에서의 LUN 동작을 칭합니다. 옵션을 선택합니다 `fractional-reserve` 100%로 설정하면 볼륨의 모든 데이터에서 볼륨의 공간이 소진되지 않으며 모든 데이터 패턴의 턴오버가 100%가 됩니다.

예를 들어 1TB 볼륨에서 단일 250GB LUN의 데이터베이스를 가정해 보겠습니다. 스냅샷을 생성하면 어떤 이유로든 볼륨이 공간을 소진하지 않도록 보장하기 위해 즉시 볼륨에서 250GB의 공간이 추가로 예약됩니다. 데이터베이스 볼륨의 모든 바이트에 덮어쓰기를 해야 하는 경우는 거의 없기 때문에 부분 예약 사용은 일반적으로 리소스를 낭비하는 것입니다. 일어나지 않을 경우를 위해 공간을 예약할 필요는 없습니다. 그럼에도, 고객이 스토리지 시스템에서 공간 소비를 모니터링할 수 없으며 공간이 절대 소진되지 않도록 할 경우에는 스냅샷을 사용하기 위해 100% 부분 예약이 필요할 것입니다.

압축 및 중복제거

압축과 중복제거는 둘 다 씬 프로비저닝의 한 형태입니다. 예를 들어, 50TB의 데이터가 30TB까지 압축될 수 있다면 20TB가 절약될 것입니다. 압축의 이점을 얻기 위해서는 이 20TB의 일부를 다른 데이터에 사용하거나 50TB보다 적은 용량의 스토리지 시스템을 구매해야 합니다. 그 결과 스토리지 시스템에서 기술적으로 지원되는 것보다 많은 데이터를 저장할 수 있게 됩니다. 데이터 관점에서 보면 드라이브에서 30TB만 점유되어 있어도 50TB의 데이터가 있는 것입니다.

데이터 세트의 압축 가능성은 언제든지 바뀔 수 있으며 이로 인해 실제 공간의 소비가 증가될 수 있습니다. 이렇게 소비가 증가하므로 모니터링과 사용 측면에서 압축을 다른 형태의 씬 프로비저닝과 마찬가지로 관리해야 합니다 volume-autogrow 및 snapshot-autodelete.

압축과 중복제거에 대한 자세한 내용은 [efficiency.html](#) 링크를 참조하십시오

압축 및 분할 예약

압축은 씬 프로비저닝의 한 형태입니다. 부분 예약은 압축 사용에 영향을 미치며, 스냅샷 생성 전에 공간을 예약한다는 점에 유의해야 합니다. 일반적으로 부분 예약은 스냅샷이 있는 경우에만 중요합니다. 스냅샷이 없는 경우 부분 예약은 중요하지 않습니다. 압축의 경우는 이와 다릅니다. 압축을 통해 볼륨에서 LUN이 생성되는 경우 ONTAP은 스냅샷을 수용하기 위해 공간을 보존합니다. 이 동작은 구성 단계에서 혼란을 줄 수 있지만 이는 예상할 수 있는 것입니다.

예를 들어, 스냅샷 없이 2.5GB까지 압축된 5GB LUN을 가지고 있는 10GB 볼륨을 생각해 보겠습니다. 두 가지 시나리오를 생각할 수 있습니다.

- 부분 예약 = 100, 결과: 7.5GB 활용률
- 부분 예약 = 0, 결과: 2.5GB 활용률

첫 번째 시나리오는 현재 데이터의 경우 2.5GB의 공간 소비와 스냅샷 사용을 예상하여 100% 소스 턴오버를 지원하기 위한 5GB의 공간이 포함됩니다. 두 번째 시나리오는 여분의 공간을 예약하지 않습니다.

혼란스러워 보이는 이 상황은 사실상 발생할 확률이 낮습니다. 압축은 씬 프로비저닝을 시사하며 LUN 환경에서 씬 프로비저닝을 수행하려면 부분 예약이 필요합니다. 압축할 수 없는 무언가로 인해 압축된 데이터를 항상 덮어쓸 수 있습니다. 즉, 압축을 위해서는 볼륨을 씬 프로비저닝해야 절약 효과를 거둘 수 있습니다.



- NetApp는 * 다음과 같은 예약 구성을 권장합니다.
- 설정 fractional-reserve 과 함께 기본 용량 모니터링이 있는 경우 0으로 설정합니다 volume-autogrow 및 snapshot-autodelete.
- 설정 fractional-reserve 모니터링 기능이 없거나 어떤 상황에서도 공간을 소진하는 것이 불가능한 경우 100까지.

여유 공간 및 LVM 공간 할당

파일 시스템 환경에서 액티브 LUN 씬 프로비저닝의 효율성은 데이터가 삭제되면 시간이 지남에 따라 손실될 수 있습니다. 삭제된 데이터가 0으로 덮어 쓰이지 않는 한(도 참조 "[아스크루주식회사](#)" 또는 TRIM/UNMAP 공간 재확보와 함께 공간이 해제되면 "지워진" 데이터는 파일 시스템에서 더 많은 할당되지 않은 공백을 차지합니다. 게다가 데이터 파일이 생성 당시의 파일 최대 크기로 초기화되기 때문에 많은 데이터베이스 환경에서는 액티브 LUN의 씬 프로비저닝의 사용이 제한적입니다.

LVM 구성을 신중하게 계획하면 효율성이 향상되고 스토리지 프로비저닝 및 LUN 리사이징에 대한 필요성이 최소화됩니다. Veritas VxVM 또는 Oracle ASM 같은 LVM을 사용할 때 기본 LUN은 익스텐트로 분할되며 이는 필요할 때만 사용됩니다. 예를 들어, 데이터 세트가 2TB 크기에서 시작되어 차차 10TB까지 증가할 수 있는 경우, 이 데이터 세트는 LVM 디스크 그룹으로 구성되어 씬 프로비저닝된 LUN 10TB에 배치될 수 있습니다. 이는 생성 당시에 2TB의

공간만 점유하며 데이터 증가를 수용하기 위해 익스텐트가 할당됨에 따라 추가 공간을 요구할 수 있습니다. 이 프로세스는 공간이 모니터링되는 한 안전합니다.

Oracle 데이터베이스 및 ONTAP 컨트롤러 페일오버/스위치오버

테이크오버와 스위치오버 작업이 Oracle 데이터베이스 작업에 방해가 되지 않도록 하려면 스토리지 테이크오버와 스위치오버 기능을 이해해야 합니다. 또한 테이크오버 및 스위치오버 작업에서 사용되는 인수를 잘못 사용할 경우 데이터 무결성에 영향을 미칠 수 있습니다.

- 정상적인 조건에서 특정 컨트롤러에 쓰기가 수신되면 파트너에게 동시에 미러링됩니다. NetApp MetroCluster 환경에서는 쓰기가 원격 컨트롤러에도 미러링됩니다. 쓰기가 모든 위치에서 비휘발성 미디어에 저장되기 전까지는 호스트 애플리케이션에서 인지되지 않습니다.
- 쓰기 데이터가 저장된 미디어를 비휘발성 메모리 또는 NVMEM이라 합니다. 이는 NVRAM(Nonvolatile Random-Access Memory)으로도 불리며, 저널 기능을 하지만 쓰기 캐시로 간주할 수 있습니다. 정상 작동 중에는 NVMEM에서 데이터를 읽을 수 없으며 소프트웨어나 하드웨어 장애 발생 시 데이터를 보호하는 데에만 사용됩니다. 드라이브에 데이터를 쓸 때 데이터는 NVMEM이 아니라 시스템의 RAM에서 전송됩니다.
- 테이크오버 작동 중에 고가용성(HA) 쌍의 노드 1개가 파트너로부터 작업을 넘겨받습니다. 스위치오버도 기본적으로는 같으나 MetroCluster 구성에 적용되며 여기서 원격 노드가 로컬 노드의 기능을 넘겨받습니다.

정기적인 유지보수 작업 중에 스토리지 테이크오버 또는 스위치오버 작동은 네트워크 경로가 변경될 때 작동이 잠깐 정지되는 경우를 제외하고 투명해야 합니다. 그러나 네트워킹은 복잡할 수 있고 오류가 발생하기 쉽기 때문에 NetApp 스토리지 시스템을 운영 환경에 구축하기 전에 테이크오버와 스위치오버 작업을 철저히 테스트하는 것이 좋습니다. 이것이 모든 네트워크 경로가 올바르게 구성되도록 하는 유일한 방법입니다. SAN 환경에서는 명령의 출력을 신중하게 확인하십시오 `sanlun lun show -p` 필요한 모든 기본 및 보조 경로를 사용할 수 있는지 확인합니다.

강제 적용 테이크오버나 스위치오버를 실행할 때는 주의해야 합니다. 이 옵션으로 스토리지 구성을 변경하면 드라이브가 속한 컨트롤러의 상태를 무시하고 대체 노드가 강제로 드라이브를 제어하게 됩니다. 강제 적용 테이크오버를 잘못 적용하면 데이터 손실 또는 손상을 야기할 수 있는데, 강제 적용 테이크오버 또는 스위치오버로 인해 NVMEM의 콘텐츠가 폐기될 수 있기 때문입니다. 테이크오버 또는 스위치오버가 완료된 후 데이터가 손실되면 드라이브에 저장된 데이터가 데이터베이스를 확인하는 시점보다 약간 더 과거의 상태로 되돌아갈 수 있습니다.

일반 HA 쌍의 강제 적용 테이크오버는 거의 필요하지 않습니다. 거의 모든 장애 시나리오에서 노드가 종료되고 자동 페일오버가 수행되도록 파트너에게 알립니다. 노드 간 인터커넥트가 손실된 후 컨트롤러 하나가 손실되는 롤링 장애와 같이 극단적인 상황이 발생하는 경우에는 강제 적용 테이크오버가 필요합니다. 이런 상황에서 노드 간 미러링은 컨트롤러 장애가 발생하기 전에 손실되며, 정상적인 컨트롤러는 쓰기 작업이 진행되고 있는 복사본을 더 이상 가지고 있지 않게 됩니다. 이때 테이크오버가 강제 적용되어야 하며, 이렇게 하면 데이터가 손실될 수 있습니다.

MetroCluster 스위치오버에도 동일한 논리가 적용됩니다. 정상 조건에서는 스위치오버가 거의 투명합니다. 그러나 재해로 인해 정상적인 사이트와 재해 사이트 간에 연결이 손실될 수 있습니다. 정상적인 사이트의 관점에서는 사이트 간 연결 중단 문제에 불과한 것으로 인식될 것이며 원래의 사이트는 여전히 데이터를 처리하고 있을 것입니다. 노드가 1차 컨트롤러의 상태를 확인할 수 없는 경우 강제 적용 스위치오버만 가능합니다.



- NetApp는 * 다음 주의사항을 준수할 것을 권장합니다.
- 테이크오버나 스위치오버를 실수로 강제 적용하지 않도록 특별히 주의하십시오. 일반적으로 강제 적용은 필요하지 않으며 변경을 강제 적용할 시 데이터가 손실될 수 있습니다.
- 강제 적용 테이크오버 또는 스위치오버가 필요한 경우 애플리케이션이 종료되고 모든 파일 시스템이 마운트 해제되며 논리적 볼륨 관리자(LVM) 볼륨 그룹이 varyoffed인지 확인합니다. ASM 디스크 그룹은 마운트 해제해야 합니다.
- 강제 적용 MetroCluster 스위치오버가 수행될 때에는 정상적인 스토리지 리소스로부터 장애 노드를 분리하십시오. 자세한 내용은 관련 ONTAP 버전에 대한 MetroCluster 관리 및 재해 복구 가이드를 참조하십시오.

MetroCluster 및 다중 애그리게이트

MetroCluster는 연결이 중단되면 비동기식 모드로 전환되는 동기식 복제 기술입니다. 이는 고객이 가장 일반적으로 요청하는 사항으로, 동기식 복제가 보장되면 사이트 연결이 중단되었을 때 데이터베이스 I/O가 완전히 지연되고 데이터베이스의 작동이 멈추기 때문입니다.

MetroCluster는 연결이 복원되면 애그리게이트를 신속하게 재동기화합니다. 다른 스토리지 기술과 달리 MetroCluster에서는 사이트 장애 후에 전체 재미러링이 전혀 필요하지 않으며 델타 변경사항만 전달되면 됩니다.

애그리게이트가 확장되는 데이터 세트의 경우 롤링 재해 시나리오에서 추가 데이터 복구 단계가 필요할 수 있어 약간의 위험이 뒤따릅니다. 특히, (a) 사이트 간 연결이 중단되고 (b) 연결이 복원되고 (c) 애그리게이트가 일부는 동기화되고 일부는 동기화되지 않은 상태에 도달한 경우 그런 다음 (d) 기본 사이트가 손실되고, 결과적으로 애그리게이트가 서로 동기화되지 않는 정상적인 사이트가 됩니다. 이 경우 데이터 세트의 각 부분이 서로 동기화되며 복구가 수행되지 않으면 애플리케이션, 데이터베이스 또는 데이터 저장소를 가져올 수 없습니다. 데이터 세트에서 애그리게이트를 확장하는 경우 NetApp, 사용할 수 있는 여러 도구 중 하나와 함께 스냅샷 기반 백업을 활용하여 이 비정상적인 시나리오에서 빠른 복구 기능을 확인하는 것이 좋습니다.

데이터베이스 구성

Oracle 데이터베이스 블록 크기

ONTAP에서는 내부적으로 가변 블록 크기를 사용하므로 원하는 블록 크기로 Oracle 데이터베이스를 구성할 수 있습니다. 그러나 파일 시스템 블록 크기가 성능에 영향을 미칠 수 있으며 경우에 따라 재실행 블록 크기가 클수록 성능이 향상될 수 있습니다.

데이터 파일 블록 크기

일부 운영 체제에서는 파일 시스템 블록 크기를 선택할 수 있습니다. Oracle 데이터 파일을 지원하는 파일 시스템의 경우 압축을 사용했을 때 블록 크기는 8KB가 됩니다. 압축이 필요하지 않은 경우 8KB 또는 4KB의 블록 크기를 사용할 수 있습니다.

데이터 파일이 512바이트 블록의 파일 시스템에 배치되면 파일 정렬 불량이 발생할 수 있습니다. LUN과 파일 시스템은 NetApp 권장 사항에 따라 적절히 정렬되었으나 파일 I/O는 정렬 불량이 될 것입니다. 이러한 정렬 불량은 심각한 성능 문제를 초래합니다.

재실행 로그를 지원하는 파일 시스템은 재실행 블록 크기의 배수인 블록 크기를 사용해야 합니다. 일반적으로 재실행 로그 파일 시스템과 재실행 로그 자체는 512바이트의 블록 크기를 사용해야 합니다.

블럭 크기 재실행

재실행 속도가 빠르면 더 간소화되고 효율적인 작업으로 I/O를 수행할 수 있기 때문에 매우 빠른 재실행 속도에서 4KB 블럭 크기의 성능이 향상될 수 있습니다. 재실행 속도가 50MBps보다 빠른 경우 4KB 블럭 크기를 테스트하는 것을 고려해 보십시오.

고객 사례에서 4KB 블럭 크기와 매우 작은 트랜잭션이 많이 수행되는 파일 시스템에서 512바이트 블럭 크기의 재실행 로그를 사용하는 데이터베이스가 몇 가지 문제를 발생시키는 것으로 나타났습니다. 단일 4KB 파일 시스템 블럭에 여러 512바이트 변경사항을 적용할 때 발생하는 오버헤드는 성능 문제를 야기했으며 이 문제는 512바이트의 블럭 크기를 사용하도록 파일 시스템을 변경하여 해결되었습니다.



* NetApp은 관련 고객 지원 팀 또는 전문 서비스 조직에서 변경하라고 조언했거나 공식 제품 설명서에 따른 것이 아닌 한 재실행 블럭 크기를 변경하지 않을 것을 권장합니다.

Oracle 데이터베이스 매개 변수: `db_file_multiblock_read_count`

를 클릭합니다 `db_file_multiblock_read_count` 매개 변수는 Oracle이 순차적 I/O 중에 단일 작업으로 읽는 Oracle 데이터베이스 블럭의 최대 수를 제어합니다

하지만 이 매개 변수는 모든 읽기 작업 중에 Oracle이 읽는 블럭의 수나 랜덤 I/O에 영향을 미치지 않으며 순차적 I/O의 블럭 크기에만 영향을 미칩니다.

Oracle은 이 매개 변수를 설정 해제 상태로 둘 것을 권장합니다. 이렇게 하면 데이터베이스 소프트웨어가 최적의 값을 자동으로 설정할 수 있습니다. 이는 일반적으로 이 매개 변수가 1MB의 I/O 크기를 생성하는 값으로 설정된다는 뜻입니다. 예를 들어, 8KB 블럭의 1MB 읽기에서는 읽을 블럭이 128개 필요하며 이에 따라 이 매개 변수의 기본값은 128이 됩니다.

NetApp이 고객 사이트에서 관찰한 데이터베이스 성능 문제는 대부분 이 매개 변수의 잘못된 설정과 관련이 있습니다. Oracle 버전 8 및 9에서 이 값을 변경할 타당한 이유가 있었습니다. 그 결과, 매개 변수가 모르게 에 있을 수 있습니다 `init.ora` 데이터베이스가 Oracle 10 이상으로 업그레이드되었기 때문입니다. 128이라는 기본값에 비해 8 또는 16의 기존 설정은 순차적 I/O 성능을 크게 저하합니다.



* NetApp는 * 를 설정할 것을 권장합니다 `db_file_multiblock_read_count` 매개 변수는 에 있어서는 안 됩니다 `init.ora` 파일. NetApp이 관찰한 결과 이 매개 변수를 변경하여 성능이 개선된 경우는 없었으나 순차적 I/O 처리량 저하가 사라진 사례는 많이 있습니다.

Oracle 데이터베이스 매개 변수: `filesystemio_options`

Oracle 초기화 매개 변수 `filesystemio_options` 비동기식 및 직접 입출력의 사용을 제어합니다

일반적인 인식과 달리 비동기식 및 직접 I/O는 상호 배타적이지 않습니다. NetApp은 이 매개 변수가 고객 환경에서 종종 잘못 구성되며 이 잘못된 구성이 여러 성능 문제의 직접적인 원인이 된다는 것을 목격했습니다.

비동기식 I/O에서는 Oracle I/O 작업을 병렬화할 수 있습니다. 다양한 운영 체제에서 비동기식 I/O를 사용할 수 있게 되기 전에 사용자는 수많은 `dbwriter` 프로세스를 구성했고 서버 프로세스 구성을 변경했습니다. 비동기식 I/O를 통해 운영 체제는 매우 효율적인 병렬 방식으로 데이터베이스 소프트웨어 대신 I/O를 수행합니다. 이 프로세스는 데이터를 위험에 처하게 하지 않으며 Oracle 로깅 재실행 같은 중요한 작업은 여전히 동기식으로 수행됩니다.

직접 I/O는 운영 체제 버퍼 캐시를 우회합니다. UNIX 시스템의 I/O는 일반적으로 운영 체제 버퍼 캐시를 통해 흐릅니다. 이는 내부 캐시에서 유지되지 않는 애플리케이션에 유용하지만 Oracle은 SGA 내부에 자체 버퍼 캐시가 있습니다. 거의

모든 경우 운영 체제 버퍼 캐시를 사용하는 것보다 직접 I/O를 활성화하고 서버 RAM을 SGA에 할당하는 것이 더 낫습니다. Oracle SGA는 메모리를 더 효율적으로 사용하며 I/O가 운영 체제 버퍼를 통해 흐를 때 지연 시간을 증가시키는 추가 처리가 적용됩니다. 증가한 지연 시간은 짧은 지연 시간이 중요한 요구사항일 때 쓰기 집약적 I/O에서 특히 뚜렷하게 나타납니다.

의 옵션 `filesystemio_options` 다음과 같습니다.

- * 비동기 * Oracle은 처리를 위해 I/O 요청을 OS에 제출합니다. 이 프로세스는 Oracle I/O가 완료되기를 기다리면서 I/O 병렬화를 증가시키는 것이 아니라 Oracle이 다른 작업을 수행할 수 있도록 합니다.
- * directio. * Oracle은 호스트 OS 캐시를 통해 I/O를 라우팅하지 않고 물리적 파일에 직접 I/O를 수행합니다.
- * 없음. * Oracle은 동기 및 버퍼링된 I/O를 사용합니다 이 구성에서는 공유 서버와 전용 서버 프로세스 중 무엇을 선택할지 그리고 dbwriter의 개수가 어떻게 되는지가 더 중요합니다.
- * SetAll. * Oracle은 비동기 I/O와 직접 I/O를 모두 사용합니다 거의 모든 경우에 를 사용합니다 `setall` 최적의 상태.



를 클릭합니다 `filesystemio_options` 매개 변수는 DNFS 및 ASM 환경에 어떤 영향도 미치지 않습니다. DNFS 또는 ASM가 자동으로 사용되면 비동기식 및 직접 I/O 둘 다 사용됩니다

과거에는 특히 Red Hat Enterprise Linux 4(RHEL4) 릴리즈에서 비동기식 I/O 문제에 직면하는 고객이 있었습니다. 일부 오래된 인터넷 조언은 오래된 정보로 인해 비동기 IO를 피하는 것을 제안합니다. 비동기식 I/O는 현재의 모든 운영 체제에서 안정적입니다. OS에 알려진 버그가 없고 비활성화할 이유가 없습니다.

데이터베이스가 버퍼링된 I/O를 사용해온 경우 직접 I/O로 전환하면 SGA 크기 변경도 보장됩니다. 버퍼링된 I/O를 비활성화하면 호스트 운영 체제 캐시가 데이터베이스에 제공하는 성능 이점이 없어집니다. RAM을 SGA에 다시 추가하면 이 문제가 해결되며 결과적으로 I/O 성능이 개선될 것입니다.

거의 모든 경우에 Oracle SGA에 운영 체제 버퍼 캐싱보다 RAM을 사용하는 것이 더 낫지만 최고의 가치를 결정하는 것은 불가능할 수 있습니다. 예를 들어, 간헐적인 액티브 Oracle 인스턴스가 있는 데이터베이스 서버에는 아주 작은 SGA 크기의 버퍼링된 I/O를 사용하는 것이 더 나을 것입니다. 이렇게 하면 실행 중인 모든 데이터베이스 인스턴스를 통해 운영 체제의 나머지 가용 RAM을 유연하게 사용할 수 있습니다. 이는 매우 드문 상황이지만 일부 고객 사이트에서 관찰된 바 있습니다.



* NetApp는 * 설정을 권장합니다 `filesystemio_options` 를 선택합니다 `'setall'`는 어떤 상황에서는 호스트 버퍼 캐시의 손실로 인해 Oracle SGA를 늘려야 할 수 있음을 유념하십시오.

Oracle RAC(Real Application Clusters) 시간 초과

Oracle RAC는 클러스터 상태를 모니터링하는 여러 유형의 내부 하트비트 프로세스를 갖춘 클러스터웨어 제품입니다.



의 정보 "[잘못 장착되었습니다](#)" 섹션에는 네트워크 스토리지를 사용하는 Oracle RAC 환경에 대한 중요 정보가 포함되어 있으며, 대부분의 경우 RAC 클러스터가 네트워크 경로 변경 및 스토리지 페일오버 /스위치오버 작업을 지속할 수 있도록 기본 Oracle RAC 설정을 변경해야 합니다.

디스크 시간 초과

은 운영 스토리지와 관련된 주요 RAC 매개 변수입니다 `disktimeout`. 이 매개 변수는 voting 파일 I/O가 완료되어야 하는 임계값을 제어합니다. 를 누릅니다 `disktimeout` 매개 변수가 초과되면 클러스터에서 RAC 노드가 제거됩니다. 이 매개 변수의 기본값은 200이며 이 값은 표준 스토리지 테이크오버 및 반환 절차를 진행하는 데 충분합니다.

많은 요소가 테이크오버나 반환에 영향을 미치기 때문에 RAC 구성을 운영 환경에 배치하기 전에 철저히 테스트하는 것이 좋습니다. NetApp 스토리지 페일오버가 완료되는 데 필요한 시간 외에 링크 통합 제어 프로토콜(LACP) 변경사항을 전달하기 위한 추가 시간도 필요합니다. 또한 SAN 다중 경로 소프트웨어는 I/O 시간 초과를 감지하고 대체 경로를 다시 시도해야 합니다. 데이터베이스가 매우 활성화된 경우 보팅 디스크 I/O가 처리되기 전에 많은 양의 I/O를 대기열에 넣고 다시 시도해야 합니다.

실제 스토리지 테이크오버 또는 반환을 수행할 수 없을 경우 데이터베이스 서버에서 케이블을 빼는 테스트를 통해 그 영향을 시뮬레이션할 수 있습니다.



- NetApp는 * 다음을 권장합니다.
- `disktimeout` 기본값 200의 매개 변수입니다.
- 항상 RAC 구성을 철저히 테스트하십시오.

잘못 장착되었습니다

`misscount` 매개 변수는 일반적으로 RAC 노드 간 네트워크 하트비트에만 영향을 미치며 기본값은 30초입니다. 그리드 바이너리가 스토리지 어레이에 있거나 운영 체제 부트 드라이브가 로컬에 있지 않은 경우 이 매개 변수의 중요성이 커집니다. 부트 드라이브가 FC SAN에 있는 호스트, NFS 부팅 운영 체제 그리고 VMDK 파일처럼 부트 드라이브가 가상화 데이터 저장소에 있는 호스트가 여기에 포함됩니다.

스토리지 테이크오버 또는 반환에 의해 부트 드라이브에 대한 액세스가 중단되면 그리드 바이너리 위치 또는 전체 운영 체제가 일시적으로 멈출 수 있습니다. ONTAP에서 스토리지 작업이 완료되고 운영 체제에서 경로가 변경되고 I/O가 재개되는 데 필요한 시간이 을 초과할 수 있습니다 `misscount` 임계값. 노드는 부트 LUN 또는 그리드 바이너리에 대한 연결이 복원되는 즉시 제거됩니다. 대부분의 경우 재부팅 이유를 나타내는 로깅 메시지 없이 제거 및 후속 재부팅 동작이 수행합니다. 모든 구성이 영향을 받는 것은 아니므로 부트 드라이브에 대한 통신이 중단되는 경우 RAC가 안정된 상태를 유지하도록 RAC 환경에서 SAN 부팅, NFS 부팅 또는 데이터 저장소 기반 호스트를 테스트하십시오.

비로컬 부팅 드라이브 또는 비로컬 파일 시스템 호스팅의 경우 `grid` 바이너리 `misscount` 일치시키려면 변경해야 합니다 `disktimeout`. 이 매개 변수가 변경되면 추가 테스트를 수행하여 노드 페일오버 시간 등 RAC 동작에 미치는 영향도 파악하십시오.



- NetApp는 * 다음을 권장합니다.
- 를 그대로 둡니다 `misscount` 다음 조건 중 하나가 적용되지 않는 한 매개 변수 기본값은 30입니다.
 - `grid` 바이너리는 NFS, iSCSI, FC 및 데이터 저장소 기반 드라이브를 포함하여 네트워크 연결 드라이브에 있습니다.
 - 운영 체제가 SAN 환경에서 부팅됩니다.
- 이러한 경우 OS 또는 에 대한 액세스에 영향을 미치는 네트워크 중단의 영향을 평가합니다. `GRID_HOME` 파일 시스템 경우에 따라 이러한 중단으로 인해 Oracle RAC 데몬에 지연이 발생하며 이로 인해 로 이어질 수 있습니다 `misscount`` 시간 초과 및 제거 기반. 시간 초과 기본값은 27초이며 이 값은 입니다 ``misscount` 빠기 `reboottime`. 이 경우 을 증가시킵니다 `misscount` 200을 선택하십시오 `disktimeout`.

호스트 구성

IBM AIX를 사용하는 Oracle 데이터베이스

ONTAP 기반 IBM AIX 기반 Oracle 데이터베이스에 대한 구성 항목

동시 I/O

IBM AIX에서 최적의 성능을 달성하려면 동시 I/O를 사용해야 합니다. 동시 I/O를 사용하지 않으면 AIX에서 직렬화된 원자 I/O가 수행되어 성능이 저하될 수 있으며 이에 따라 상당한 오버헤드가 발생하게 됩니다.

원래 NetApp는 `rw`를 사용할 것을 권장합니다. `cio` 파일 시스템에 동시 I/O 사용을 강제 적용하는 마운트 옵션이지만 이 프로세스에는 결점이 있어 더 이상 필요하지 않습니다. AIX 5.2 및 Oracle 10gR1의 도입 이후로, AIX 기반 Oracle에서는 전체 파일 시스템에 동시 I/O를 강제 적용하는 것이 아니라 동시 IO를 위해 개별 파일을 열 수 있습니다.

동시 I/O 활성화를 위한 최상의 방법은 `rw`를 설정하는 것입니다. `init.ora` 매개 변수 `filesystemio_options`를 선택합니다. `setall`. 이렇게 설정하면 Oracle은 동시 I/O와 함께 사용할 특정 파일을 열 수 있습니다.

사용 `cio` 마운트 옵션은 동시 I/O 사용을 강제 적용되며 이는 부정적인 결과를 초래할 수 있습니다. 예를 들어 동시 I/O를 적용하면 파일 시스템에서 미리 읽기를 비활성화하므로 파일 복사, 테이프 백업 수행과 같이 Oracle 데이터베이스 소프트웨어 외부에서 발생하는 I/O 성능이 손상될 수 있습니다. 또한, Oracle GoldenGate와 SAP BR * Tools 같은 제품은 `rw`를 사용하는 것과 호환되지 않습니다. `cio` Oracle의 특정 버전에서 마운트 옵션.



- NetApp는 * 다음을 권장합니다.
- `rw`를 사용하지 마십시오. `cio` 파일 시스템 레벨에서 마운트 옵션입니다. `rw`를 사용하여 동시 I/O를 활성화하십시오. `filesystemio_options=setall`.
- 만 사용하십시오. `cio`를 설정할 수 없는 경우 마운트 옵션을 설정해야 합니다. `filesystemio_options=setall`.

AIX NFS 마운트 옵션

다음 표에는 Oracle 단일 인스턴스 데이터베이스에 대한 AIX NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsiz=262144,wsiz=262144</code>
제어 파일 데이터 파일 다시 실행 로그	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsiz=262144,wsiz=262144</code>
ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsiz=262144,wsiz=262144,intr</code>

다음 표에는 RAC에 대한 AIX NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsiz=262144,wsiz=262144</code>

파일 형식	마운트 옵션
제어 파일 데이터 파일 다시 실행 로그	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac
CRS/Voting	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac
전용 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144
공유됨 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr

단일 인스턴스와 RAC 마운트 옵션 간의 주된 차이점은 을 추가한다는 것입니다 noac 마운트 옵션으로 이동합니다. 이렇게 추가되면 호스트 운영 체제 캐싱이 비활성화되어 RAC 클러스터의 모든 인스턴스에서 데이터 상태에 관한 일관된 뷰를 얻게 됩니다.

를 사용하는 경우에도 마찬가지입니다 cio 마운트 옵션 및 init.ora 매개 변수 filesystemio_options=setall 호스트 캐싱이 비활성화되는 것과 동일한 효과가 있으며, 그것도 사용해야 합니다 noac. noac 은(는) 공유에 필요합니다 ORACLE_HOME 구축을 통해 Oracle 암호 파일 및 과 같은 파일의 일관성을 지원합니다 spfile 매개 변수 파일 RAC 클러스터의 각 인스턴스에 전용 이 있는 경우 `ORACLE_HOME`이 매개 변수는 필요하지 않습니다.

AIX jfs/jfs2 마운트 옵션

다음 표에는 AIX jfs/jfs2 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	기본값
제어 파일 데이터 파일 다시 실행 로그	기본값
ORACLE_HOME	기본값

AIX를 사용하기 전에 hdisk 데이터베이스를 포함하여 모든 환경에서 매개 변수를 확인합니다 queue_depth. 이 매개 변수는 호스트 버스 어댑터 큐 길이가 아니며, 개인의 SCSI 큐 길이와 관련이 있습니다 hdisk device. Depending on how the LUNs are configured, the value for `queue_depth` 성능이 뛰어나기 때문에 너무 낮을 수 있습니다. 테스트 결과 최적의 값은 64였습니다.

HP-UX를 사용하는 Oracle 데이터베이스

ONTAP를 사용하는 HP-UX에 대한 Oracle 데이터베이스의 구성 항목

HP-UX NFS 마운트 옵션

다음 표에는 단일 인스턴스에 대한 HP-UX NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,suid
제어 파일 데이터 파일 다시 실행 로그	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,forcedirectio,nointr,suid
ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,suid

다음 표에는 RAC에 대한 HP-UX NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,noac,suid
제어 파일 데이터 파일 다시 실행 로그	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,forcedirectio,suid
CRS/투표	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,forcedirectio,suid
전용 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,suid
공유됨 ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,suid

단일 인스턴스와 RAC 마운트 옵션 간의 주된 차이점은 을 추가한다는 것입니다 noac 및 forcedirectio 마운트 옵션으로 이동합니다. 이렇게 추가되면 호스트 운영 체제 캐싱이 비활성화되어 RAC 클러스터의 모든 인스턴스에서 데이터 상태에 관한 일관된 뷰를 얻게 됩니다. 를 사용하는 경우에도 마찬가지입니다 init.ora 매개 변수 filesystemio_options=setall 호스트 캐싱이 비활성화되는 것과 동일한 효과가 있으며, 그것도 사용해야 합니다 noac 및 forcedirectio.

그 이유는 noac 은(는) 공유에 필요합니다 ORACLE_HOME 배포는 Oracle 암호 파일과 spfile 같은 파일의 일관성을 지원하기 위한 것입니다. RAC 클러스터의 각 인스턴스에 전용 이 있는 경우 ORACLE_HOME, 이 매개 변수는 필요하지 않습니다.

HP-UX VxFS 마운트 옵션

Oracle 바이너리를 호스팅하는 파일 시스템에 다음 마운트 옵션을 사용하십시오.

```
delaylog,nodatainlog
```

HP-UX 버전이 동시 I/O를 지원하지 않으며 데이터 파일, 재실행 로그, 아카이브 로그, 제어 파일이 포함된 파일 시스템에 다음 마운트 옵션을 사용하십시오.

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

동시 I/O가 지원되는 경우(VxFS 5.0.1 이상 또는 ServiceGuard 스토리지 관리 제품군) 데이터 파일, 재실행 로그, 아카이브 로그, 제어 파일이 포함된 파일 시스템에 다음 마운트 옵션을 사용하십시오.

```
delaylog,cio
```



매개 변수입니다 db_file_multiblock_read_count 특히 VxFS 환경에서 중요합니다. 달리 명시되지 않는 한 Oracle 10g R1 이상에서는 이 매개 변수를 설정 해제 상태로 유지하는 것이 좋습니다. Oracle 8KB 블록 크기의 기본값은 128입니다. 이 매개 변수의 값이 16 이하로 강제 적용되면 를 제거합니다 convosync=direct 순차적 I/O 성능이 저하될 수 있으므로 마운트 옵션을 제공합니다. 이 단계는 성능의 다른 측면을 저하시키므로 의 가치가 있는 경우에만 수행해야 합니다 db_file_multiblock_read_count 기본값에서 변경해야 합니다.

Oracle 데이터베이스 및 Linux

Linux 운영 체제에만 해당되는 구성 항목

Linux NFSv3 TCP 슬롯 테이블

TCP 슬롯 테이블은 호스트 버스 어댑터(HBA) 큐 길이(queue depth)와 동등한 NFSv3의 기능입니다. 이들 테이블은 한 번에 수행될 수 있는 최대 NFS 작업의 수를 제어합니다. 기본값은 보통 16이며 최적의 성능을 발휘하기에 너무 낮습니다. 최신 Linux 커널에서는 반대의 문제가 발생하는데, 요청을 통해 NFS 서버를 포화시키는 수준까지 TCP 슬롯 테이블의 한계를 자동으로 높일 수 있습니다.

최적의 성능을 얻으면서 성능 문제를 방지하려면 TCP 슬롯 테이블을 제어하는 커널 매개 변수를 조정하십시오.

를 실행합니다 `sysctl -a | grep tcp.*.slot_table` 명령을 실행하고 다음 매개 변수를 관찰합니다.

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

모든 Linux 시스템에는 가 포함되어 있습니다 sunrpc.tcp_slot_table_entries`그러나 일부에만 가 포함됩니다 `sunrpc.tcp_max_slot_table_entries. 둘 다 128로 설정해야 합니다.

주의

이러한 매개 변수를 설정하지 않으면 성능에 상당한 영향을 미칠 수 있습니다. 경우에 따라 Linux 운영 체제가 충분한 I/O를 실행하지 않기 때문에 성능이 제한됩니다 또는 Linux 운영 체제가 처리할 수 있는 것보다 더 많은 I/O를 발급하려고 하면 I/O 지연 시간이 늘어납니다.

Linux NFS 마운트 옵션

다음 표에는 단일 인스턴스의 Linux NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144</code>
제어 파일 데이터 파일 다시 실행 로그	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr</code>
ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr</code>

다음 표에는 RAC에 대한 Linux NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,actimeo=0</code>
제어 파일 데이터 파일 다시 실행 로그	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,actimeo=0</code>
CRS/투표	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,actimeo=0</code>
전용 ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144</code>
공유됨 ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,actimeo=0</code>

단일 인스턴스와 RAC 마운트 옵션 간의 주된 차이점은 `actimeo=0` 마운트 옵션으로 이동합니다. 이렇게 추가되면 호스트 운영 체제 캐싱이 비활성화되어 RAC 클러스터의 모든 인스턴스에서 데이터 상태에 관한 일관된 뷰를 얻게 됩니다. 를 사용하는 경우에도 마찬가지입니다 `init.ora` 매개 변수 `filesystemio_options=setall` 호스트 캐싱이 비활성화되는 것과 동일한 효과가 있으며, 그것도 사용해야 합니다 `actimeo=0`.

그 이유는 `actimeo=0` 은(는) 공유에 필요합니다 ORACLE_HOME 배포는 Oracle 암호 파일과 `spfile` 같은 파일의 일관성을 지원하기 위한 것입니다. RAC 클러스터의 각 인스턴스에 전용 이 있는 경우 'ORACLE_HOME'이 매개 변수는 필요하지 않습니다.

특정 애플리케이션에서는 요구사항이 다를 수 있지만 일반적으로 데이터베이스가 아닌 파일은 단일 인스턴스 데이터 파일에 사용되는 것과 같은 옵션으로 마운트해야 합니다. 마운트 옵션을 사용하지 마십시오 `noac` 및 `actimeo=0` 이러한 옵션은 파일 시스템 레벨의 미리 읽기 및 버퍼링을 비활성화하기 때문에 가능한 경우 추출, 변환, 로딩 같은 프로세스에 심각한 성능 문제가 초래될 수 있습니다.

ACCESS 및 GETATTR

ACCESS 및 GETATTR 같은 극히 높은 레벨의 기타 IOPS가 워크로드를 장악할 수 있다는 사실을 깨달은 고객들이 있습니다. 극단적인 경우 그러한 읽기 및 쓰기 작업이 전체 중 10% 남짓할 수 있습니다. 이는 `rl` 포함하는 모든 데이터베이스에서 정상적인 동작입니다 `actimeo=0` 및/또는 `noac` 이러한 옵션 덕분에 Linux 운영 체제가 스토리지 시스템에서 파일 메타데이터를 지속적으로 다시 로드하게 되기 때문입니다. ACCESS 및 GETATTR 같은 작업은 데이터베이스 환경에서 ONTAP 캐시로부터 서비스되며 그 영향이 크지 않습니다. 스토리지 시스템에 대한 진정한 수요를 창출하는 읽기 및 쓰기 같은 진정한 IOPS로 간주해서는 안 됩니다. 이러한 기타 IOPS는 특히 RAC 환경에서 약간의 로드를 생성합니다. 이러한 상황을 해결하려면 DNFS를 활성화하여 운영 체제 버퍼 캐시를 우회하고 이 불필요한 메타데이터 작업을 피하십시오.

Linux 직접 NFS

한 가지 추가 마운트 옵션이라고 합니다 `nosharecache``는 (a) DNFS가 활성화되고 (b) 소스 볼륨이 단일 서버에 두 번 이상 마운트되며 (c) 중첩된 NFS 마운트가 있는 경우에 필요합니다. 이 구성은 SAP 애플리케이션을 지원하는 환경에서 주로 볼 수 있습니다. 예를 들어, NetApp 시스템의 단일 볼륨의 경우 `에 디렉토리가 있을 수 있습니다` /vol/oracle/base` 그리고 에서 두 번째에도 /vol/oracle/home. If(경우 /vol/oracle/base`에 탑재됩니다` /oracle` 및 /vol/oracle/home`에 탑재됩니다` /oracle/home`이 경우 같은 소스에서 NFS 마운트가 중첩됩니다.`

운영 체제는 바로 그 사실을 감지할 수 있습니다 `/oracle` 및 /oracle/home` 동일한 소스 파일 시스템인 동일한 볼륨에 상주합니다. 같은 장치를 사용하여 데이터에 대한 액세스를 처리합니다. 이렇게 되면 운영 체제 캐싱 사용 및 기타 특정 작업이 개선되나 DNFS에는 지장을 줍니다. DNFS가 같은 파일에 액세스해야 하는 경우 spfile, 커짐 /oracle/home` 잘못된 데이터 경로를 사용하려고 잘못 시도할 수 있습니다. 그 결과 I/O 작업이 실패합니다. 이 설정에서 rl` 추가합니다` `nosharecache` 소스 FlexVol 볼륨을 해당 호스트의 다른 NFS 파일 시스템과 공유하는 NFS 파일 시스템에 마운트 옵션 그러면 Linux 운영 체제에서 해당 파일 시스템을 처리할 독립적 장치를 할당하게 됩니다.`

Linux Direct NFS 및 Oracle RAC

Linux에는 노드 전체에 걸친 일관성을 위해 RAC에 필요한 직접 I/O를 강제 적용할 방법이 없기 때문에 DNFS를 사용하면 Linux 운영 체제 기반 Oracle RAC에서 특별한 성능 이점을 얻을 수 있습니다. 이를 해결하려면 Linux에서 `rl` 사용해야 합니다` actimeo=0` 마운트 옵션을 사용하면 운영 체제 캐시에서 파일 데이터가 즉시 만료됩니다` 그러면 이 옵션은 Linux NFS 클라이언트가 특성 데이터를 계속 다시 읽게 강제 적용합니다. 이는 지연 시간을 저하시키고 스토리지 컨트롤러의 로드를 증가시킵니다.`

DNFS를 활성화하면 호스트 NFS 클라이언트가 우회되어 이러한 저하가 일어나지 않습니다. 여러 고객의 보고에 따르면, DNFS를 활성화했을 때 RAC 클러스터에서의 성능이 상당히 개선되었고 특히 IOPS와 관련하여 ONTAP 로드가 크게 감소하였다고 합니다.

Linux Direct NFS 및 `oranfstab` 파일`

다중 경로 옵션으로 Linux에서 DNFS를 사용할 때는 여러 서브넷을 사용해야 합니다. 그 외 운영 체제에서는 `rl` 사용하여 여러 DNFS 채널을 설정할 수 있습니다` LOCAL 및 DONROUTE` 단일 서브넷에 여러 DNFS 채널을 구성하는 옵션입니다. 하지만 이 방법은 Linux에서 제대로 작동하지 않을 수 있고 예기치 않은 성능 문제가 초래될 수 있습니다. Linux에서는 DNFS 트래픽을 위해 사용되는 각 NIC가 다른 서브넷에 있어야 합니다.`

I/O 스케줄러

Linux 커널은 블록 장치의 I/O를 스케줄링하여 낮은 레벨의 제어를 허용합니다. Linux의 다양한 배포 방식에서는 여러 가지의 기본값을 사용할 수 있습니다. 테스트에 의하면 Deadline이 보통 최상의 결과를 제공하지만 NOOP에서 성능이 더 높은 경우도 간혹 있었습니다. 성능 차이는 크지 않으나 데이터베이스 구성에서 최대한의 성능을 끌어내려면 필요 시 두 옵션을 모두 테스트하십시오. 대다수 구성은 CFQ를 기본값으로 하며, 데이터베이스 워크로드에서 심각한 성능 문제가 발생한다는 사실이 입증되었습니다.

I/O 스케줄러 구성에 관한 관련 Linux 공급업체 설명서의 지침을 참조하십시오.

다중 경로

어떤 고객은 다중 경로 데몬이 시스템에서 실행되지 않아 네트워크 중단 시 충돌이 발생하였습니다. Linux 최신 버전에서는 운영 체제 설치 프로세스와 다중 경로 데몬에 의해 운영 체제가 이 문제에 취약해질 수 있습니다. 패키지는 올바르게 설치되거나 재부팅 후 자동 시동되도록 구성되지 않습니다.

예를 들어, RHEL5.5에서 다중 경로 데몬의 기본값은 다음과 같이 나타낼 수 있습니다.

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

다음과 같은 명령으로 이를 수정할 수 있습니다.

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

ASM 미러링

ASM 미러링은 ASM이 문제를 인식하고 대체 장애 그룹으로 전환할 수 있도록 Linux 다중 경로 설정을 변경해야 할 수 있습니다. ONTAP의 대다수 ASM 구성은 외부 이중화를 사용하는데, 이는 외부 어레이를 통해 데이터가 보호되고 ASM은 데이터를 미러링하지 않는다는 뜻입니다. 일부 사이트는 ASM에서 일반적인 수준의 이중화를 사용하며 일반적으로 여러 사이트에 걸쳐 양방향 미러링을 제공합니다.

에 나와 있는 Linux 설정입니다 "[NetApp Host Utilities 설명서](#)" I/O의 무한 대기 를 야기하는 다중 경로 매개 변수를 포함하십시오 즉, 액티브 경로가 없는 LUN 장치의 I/O가 I/O가 완료될 때까지 큐에서 대기합니다. Linux 호스트가 SAN 경로 변경이 완료될 때까지, FC 스위치가 재부팅될 때까지, 또는 스토리지 시스템의 파일오버가 완료될 때까지 대기하기 때문에 이는 일반적으로 바람직한 방식입니다.

무제한 큐잉 동작은 ASM 미러링에 문제를 발생시키는데, 대체 LUN에서 I/O를 재시도하려면 ASM이 I/O 장애를 수신해야 하기 때문입니다.

Linux에서 다음 매개 변수를 설정합니다 `multipath.conf` ASM 미러링과 함께 사용되는 ASM LUN용 파일:

```
polling_interval 5
no_path_retry 24
```

이들 설정은 ASM 장치의 시간 초과 값을 120초로 만듭니다. 시간 초과는 로 계산됩니다 `polling_interval * no_path_retry` 초 단위로 표시합니다. 정확한 값을 위해 조정이 필요할 때도 있지만 대부분의 경우에는 120초 시간 초과로 충분합니다. 특히, 장애 그룹을 오프라인 상태로 만들어버리는 I/O를 생성하지 않고 120초 동안 컨트롤러가 테이크오버 또는 반환을 수행할 수 있어야 합니다.

더 낮아졌습니다 `no_path_retry` 값을 지정하면 ASM이 대체 장애 그룹으로 전환하는 데 필요한 시간을 단축할 수 있지만 이렇게 하면 컨트롤러 테이크오버 같은 유지보수 활동 중에 원치 않는 페일오버 위험이 증가합니다. 이러한 위험은 ASM 미러링 상태를 주의 깊게 모니터링하여 완화할 수 있습니다. 원치 않는 페일오버가 발생한 경우에도 재동기화가 상대적으로 빠르게 수행된다면 미러링이 신속하게 재동기화됩니다. 추가 정보는 사용 중인 Oracle 소프트웨어 버전의 ASM 빠른 미러 재동기화에 관한 Oracle 설명서를 참조하십시오.

Linux xfs, ext3 및 ext4 마운트 옵션



* NetApp는 기본 마운트 옵션을 사용하여 * 를 권장합니다.

ASMLib/AFD를 사용하는 Oracle 데이터베이스(ASM 필터 드라이버)

AFD 및 ASMLib를 사용하는 Linux 운영 체제에만 해당되는 구성 항목

ASMLib 블록 크기

ASMLib는 ASM 관리 라이브러리 및 관련 유틸리티 옵션입니다. 이 옵션은 LUN 또는 NFS 기반 파일을 사람이 읽을 수 있는 라벨이 포함된 ASM 리소스로 스탬핑할 수 있어 유용합니다.

ASMLib의 최신 버전은 LBPPBE(Logical Blocks Per Physical Block Exponent)라는 LUN 매개 변수를 감지합니다. 이 값은 최근까지 ONTAP SCSI 타겟에 의해 보고되지 않았습니다. 여기서 반환되는 값은 4KB 블록 크기가 낫다는 사실을 나타냅니다. 이는 블록 크기의 정의가 아니며 LBPPBE 값을 사용하여 특정 크기의 I/O를 더 효율적으로 처리할 수 있게 만드는 애플리케이션을 암시합니다. 그러나 ASMLib는 LBPPBE를 블록 크기로 해석하며 ASM 장치가 생성될 때 ASM 헤더를 지속적으로 스탬핑합니다.

이 프로세스는 같은 ASM 디스크 그룹에서 ASMLib 장치를 여러 블록 크기와 혼합할 수 없기 때문에 업그레이드와 마이그레이션에서 다양한 문제를 일으킬 수 있습니다.

예를 들어, 기존의 어레이는 일반적으로 LBPPBE 값을 0으로 보고하거나 값을 아예 보고하지 않았습니다. ASMLib는 이를 512바이트 블록 크기로 해석합니다. 최신 어레이의 경우에는 블록 크기가 4KB인 것으로 해석될 것입니다. 같은 ASM 디스크 그룹에서 512바이트와 4KB 장치를 혼합하는 것이 불가능하지는 않습니다. 그러나 이렇게 하면 사용자가 두 어레이의 LUN을 사용하는 ASM 디스크 그룹의 크기를 늘리거나 ASM을 마이그레이션 툴로 활용할 수 없게 됩니다. RMAN이 512바이트 블록 크기의 ASM 디스크 그룹과 4KB 블록 크기의 ASM 디스크 그룹 간 파일 복사를 허용하지 않을 수 있습니다.

적절한 해결책은 ASMLib를 패치하는 것입니다. Oracle 버그 ID는 13999609이며 패치는 `oracleasm-support-2.1.8-1` 이후 버전에 있습니다. 이 패치는 사용자가 매개 변수를 설정할 수 있습니다

`ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 를 선택합니다 `true` 에 있습니다 `/etc/sysconfig/oracleasm` 구성 파일. 이렇게 하면 ASMLib가 LBPPBE 매개 변수를 사용할 수 없으며 새로운 어레이의 LUN이 512바이트 블록 장치로 인식됩니다.



이 옵션은 ASMLib에 의해 이전에 스탬핑된 LUN의 블록 크기를 변경하지 않습니다. 예를 들어, 512바이트 블록의 ASM 디스크 그룹은 4KB 블록을 보고하는 새로운 스토리지 시스템으로 마이그레이션해야 하는 경우 옵션 `ORACLEASM_USE_LOGICAL_BLOCK_SIZE` 새로운 LUN은 ASMLib에 스탬핑되기 전에 설정해야 합니다. 장치가 이미 `oracleasm`에 스탬프로 찍힌 경우 새 블록 크기로 재정비되기 전에 다시 포맷해야 합니다. 먼저 로 장치의 그림을 해제합니다 `oracleasm deletedisk``를 누른 다음 로 장치의 첫 1GB를 지웁니다 ``dd if=/dev/zero of=/dev/mapper/device bs=1048576 count=1024`. 마지막으로, 이전에 디바이스를 파티션한 경우 를 사용합니다 `kpartx` 오래된 파티션을 제거하거나 운영 체제를 재부팅하는 명령입니다.

ASMLib를 패칭할 수 없을 경우 구성에서 ASMLib를 제거할 수 있습니다. 이는 매우 큰 변경이므로 ASM 디스크의 스탬핑을 해제하고 가 확실히 변경되어야 합니다 `asm_diskstring` 매개 변수가 올바르게 설정되었습니다. 하지만 이 변경에 따라 데이터를 마이그레이션할 필요는 없습니다.

AFD(ASM 필터 드라이브) 블록 크기

AFD는 ASMLib를 대체하는 선택적 ASM 관리 라이브러리입니다. 스토리지 관점에서는 ASMLib와 매우 유사하지만 Oracle이 아닌 I/O를 차단하여 데이터 손상을 일으킬 수 있는 사용자 또는 애플리케이션 오류를 줄일 수 있는 기능과 같은 추가 기능이 포함되어 있습니다.

장치 블록 크기

ASMLib와 마찬가지로 AFD는 LUN 매개 변수 LBPPBE(Logical Blocks Per Physical Block Exponent)를 읽으며 기본적으로 논리적 블록 크기가 아닌 물리적 블록 크기를 사용합니다.

ASM 디바이스가 이미 512바이트 블록 디바이스로 포맷된 기존 구성에 AFD가 추가되면 문제가 발생할 수 있습니다. AFD 드라이버는 LUN을 4K 디바이스로 인식하며 ASM 레이블과 물리적 디바이스 간의 불일치로 인해 액세스가 차단됩니다. 마찬가지로, 동일한 ASM 디스크 그룹에서 512바이트와 4KB 장치를 혼합하는 것이 불가능하기 때문에 마이그레이션이 영향을 받습니다. 그러나 이렇게 하면 사용자가 두 어레이의 LUN을 사용하는 ASM 디스크 그룹의 크기를 늘리거나 ASM을 마이그레이션 톨로 활용할 수 없게 됩니다. RMAN이 512바이트 블록 크기의 ASM 디스크 그룹과 4KB 블록 크기의 ASM 디스크 그룹 간 파일 복사를 허용하지 않을 수 있습니다.

솔루션은 간단합니다. AFD에는 논리적 블록 크기인지 물리적 블록 크기를 사용할지 여부를 제어하는 매개 변수가 포함됩니다. 시스템의 모든 장치에 영향을 주는 전역 매개 변수입니다. AFD에서 논리 블록 크기를 사용하도록 강제 설정하려면 를 설정합니다 `options oracleafd oracleafd_use_logical_block_size=1` 에 있습니다 `/etc/modprobe.d/oracleafd.conf` 파일.

다중 경로 전송 크기

최근 Linux 커널 변경 사항은 다중 경로 디바이스에 전송되는 입출력 크기 제한을 적용하며 AFD는 이러한 제한을 준수하지 않습니다. 그런 다음 I/O가 거부되어 LUN 경로가 오프라인 상태가 됩니다. 그 결과 Oracle Grid를 설치하거나 ASM을 구성하거나 데이터베이스를 생성할 수 없게 됩니다.

해결책은 ONTAP LUN에 대해 `multipath.conf` 파일에서 최대 전송 길이를 수동으로 지정하는 것입니다.

```

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

```



현재 문제가 없는 경우에도 향후 Linux 업그레이드로 인해 예기치 않게 문제가 발생하지 않도록 AFD를 사용하는 경우 이 매개 변수를 설정해야 합니다.

Microsoft Windows를 사용하는 Oracle 데이터베이스

ONTAP가 있는 Microsoft Windows의 Oracle 데이터베이스에 대한 구성 항목

NFS 를 참조하십시오

Oracle은 Microsoft Windows를 direct NFS 클라이언트와 함께 사용할 수 있도록 지원합니다. 이 기능을 통해 환경 전반의 파일을 확인하고, 볼륨 크기를 동적으로 조정하며 비용이 적게 드는 IP 프로토콜을 활용하는 것 등 NFS 관리에서 이점을 얻을 수 있습니다. DNFS를 사용하여 Microsoft Windows에 데이터베이스를 설치하고 구성하는 방법은 공식 Oracle 설명서를 참조하십시오. 특별한 모범 사례는 없습니다.

산

최적의 압축 효율성을 위해 NTFS 파일 시스템이 8K 이상의 할당 유닛을 사용하는지 확인하십시오. 일반적으로 기본값인 4K 할당 유닛을 사용하면 압축 효율성에 부정적인 영향을 줄 수 있습니다.

Solaris를 사용한 Oracle 데이터베이스

Solaris OS에만 해당하는 구성 항목

Solaris NFS 마운트 옵션

다음 표에는 단일 인스턴스의 Solaris NFS 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	rw,bg,hard,[vers=3,vers=4.1], roto=tcp, timeo=600, rsize=262144, wsize=262144
제어 파일 데이터 파일 다시 실행 로그	rw,bg,hard,[vers=3,vers=4.1], proto=tcp, timeo=600, rsize=262144, wsize=262144, nointr, llock, suid
ORACLE_HOME	rw,bg,hard,[vers=3,vers=4.1], proto=tcp, timeo=600, rsize=262144, wsize=262144, suid

의 사용 llock 스토리지 시스템의 잠금 획득 및 해제와 관련된 지연 시간이 사라져 고객 환경에서 성능이 대폭

개선된다는 사실이 입증되었습니다. 여러 서버가 같은 파일 시스템을 마운트하도록 구성된 환경에서 이 옵션을 사용할 때는 주의를 기울여야 하며, Oracle은 이러한 데이터베이스를 마운트하도록 구성되어 있습니다. 이러한 구성은 흔치 않으며 소수의 고객들만 사용하고 있습니다. Oracle은 외부 서버에 있는 잠금 파일을 감지할 수 없기 때문에 인스턴스가 우발적으로 재시작되는 경우 데이터 손상이 발생할 수 있습니다. NFS 버전 3처럼 NFS 잠금을 통한 보호 기능은 제공되지 않으며 이는 권고사항일 뿐입니다.

왜냐하면 `llock` 및 `forcedirectio` 매개 변수는 상호 배타적이기 때문에 가 중요합니다 `filesystemio_options=setall` 가 있습니다 `init.ora` 파일을 클릭합니다 `directio` 사용됩니다. 이 매개 변수가 없으면 호스트 운영 체제 버퍼 캐싱이 사용되어 성능에 악영향을 미칠 수 있습니다.

다음 표에는 Solaris NFS RAC 마운트 옵션이 나와 있습니다.

파일 형식	마운트 옵션
ADR 홈	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,noac</code>
제어 파일 데이터 파일 다시 실행 로그	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,forcedirectio</code>
CRS/투표	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,forcedirectio</code>
전용 ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,suid</code>
공유됨 ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,suid</code>

단일 인스턴스와 RAC 마운트 옵션 간의 주된 차이점은 을 추가한다는 것입니다 `noac` 및 `forcedirectio` 마운트 옵션으로 이동합니다. 이렇게 추가되면 호스트 운영 체제 캐싱이 비활성화되어 RAC 클러스터의 모든 인스턴스에서 데이터 상태에 관한 일관된 뷰를 얻게 됩니다. 를 사용하는 경우에도 마찬가지입니다 `init.ora` 매개 변수 `filesystemio_options=setall` 호스트 캐싱이 비활성화되는 것과 동일한 효과가 있으며, 그것도 사용해야 합니다 `noac` 및 `forcedirectio`.

그 이유는 `actimeo=0` 은(는) 공유에 필요합니다 ORACLE_HOME 배포는 Oracle 암호 파일과 `spfile` 같은 파일의 일관성을 지원하기 위한 것입니다. RAC 클러스터의 각 인스턴스에 전용 이 있는 경우 ORACLE_HOME, 이 매개 변수는 필요하지 않습니다.

Solaris UFS 마운트 옵션

NetApp은 Solaris 호스트가 충돌했거나 FC 연결이 중단된 경우 데이터 무결성을 보존할 수 있도록 로깅 마운트 옵션을 사용하는 것이 좋습니다. 로깅 마운트 옵션은 스냅샷 백업의 사용 편의성도 보존해줍니다.

Solaris ZFS입니다

Solaris ZFS에서 최적의 성능을 얻으려면 신중하게 설치하고 구성해야 합니다.

mvector입니다

Solaris 11에서는 대규모 I/O 작업을 처리하는 방식이 변경되었으며 이로 인해 SAN 스토리지 어레이에 심각한 성능 문제가 초래될 수 있습니다. 이 문제는 NetApp 버그 보고서 630173, 'Solaris 11 ZFS 성능 퇴보'에서 상세히 다루고 있습니다. "해결책은 라는 OS 매개변수를 변경하는 것입니다 `zfs_mvector_max_size`.

다음 명령을 루트로 실행합니다.

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

이 변경으로 인해 예기치 않은 문제가 발생하는 경우 다음 명령을 루트로 실행하면 쉽게 되돌릴 수 있습니다.

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

커널

안정적인 ZFS 성능을 얻으려면 LUN 정렬 문제를 방지하는 패치가 적용된 Solaris 커널이 필요합니다. 이 픽스는 Solaris 10의 패치 147440-19와 SRU 10.5 for Solaris 11에 도입되었습니다. ZFS에는 Solaris 10 이후 버전만 사용하십시오.

LUN 구성

LUN을 구성하려면 다음 단계를 완료하십시오.

1. 유형이 인 LUN을 생성합니다 `solaris`.
2. 에 지정된 적절한 호스트 유틸리티 키트(HUK)를 설치합니다 "[NetApp 상호 운용성 매트릭스 툴\(IMT\)](#)".
3. HUK의 지침을 그대로 따릅니다. 기본 단계는 아래에 설명되어 있지만 을 참조하십시오 "[최신 설명서](#)" 를 참조하십시오.
 - a. 를 실행합니다 `host_config` 를 업데이트하는 유틸리티입니다 `sd.conf/sdd.conf` 파일. 이렇게 하면 SCSI 드라이브가 ONTAP LUN을 정확하게 찾을 수 있습니다.
 - b. 에서 제공하는 지침을 따릅니다 `host_config` 다중 경로 I/O(MPIO)를 활성화하는 유틸리티.
 - c. 재부팅합니다. 이 단계는 시스템 전체에 변경사항을 적용하는 데 필요합니다.
4. LUN을 파티셔닝하고 제대로 정렬되었는지 확인합니다. 정렬을 직접 테스트하고 확인하는 방법은 "[부록 B: WAFL 정렬 확인](#)"을 참조하십시오.

zpool입니다

의 단계를 수행해야 `zpool`을 생성할 수 있습니다 "[LUN 구성](#)" 수행됩니다. 절차를 정확하게 수행하지 않으면 I/O 정렬로 인해 심각한 성능 저하가 야기될 수 있습니다. ONTAP에서 최적의 성능을 얻으려면 I/O를 드라이브의 4K 경계에 맞춰 정렬해야 합니다. `zpool`에 생성된 파일 시스템은 라는 매개 변수를 통해 제어되는 유효 블록 크기를 사용합니다 `ashift`, 명령을 실행하여 볼 수 있습니다 `zdb -C`.

의 값 `ashift` 기본값은 9이며, 이는 2(9) 또는 512바이트를 의미합니다. 최적의 성능을 위해 `ashift` 값은 12(2-12=4K)여야 합니다. 이 값은 `zpool`이 생성될 때 설정되며 변경할 수 없습니다. 즉, `zpool` 데이터의 경우 가 로 설정된다는 의미입니다 `ashift 12`가 아닌 경우 데이터를 새로 생성된 `zpool`으로 마이그레이션해야 합니다.

zpool을 생성한 후 의 값을 확인합니다 ashift 계속 진행하기 전에 값이 12가 아니면 LUN이 정확히 검색되지 않았다는 뜻입니다. zpool을 폐기하고 관련 호스트 유틸리티 설명서에 표시된 모든 단계가 정확히 수행되었는지 확인한 다음 zpool을 다시 생성하십시오.

zpool 및 Solaris LDOM

Solaris LDOM은 I/O 정렬의 정확성을 보장하기 위해 추가 요구사항을 생성합니다. LUN이 4K 장치로 제대로 검색되어도 LDOM의 가상 vdsk 장치는 I/O 도메인의 구성을 상속하지 않습니다. 이 LUN을 기반으로 하는 vdsk는 512바이트 블록으로 되돌아갑니다.

추가 구성 파일이 필요합니다. 먼저, 추가 구성 옵션을 활성화하려면 개별 LDOM에 Oracle 버그 15824910 패치를 적용해야 합니다. 이 패치는 현재 사용되는 모든 Solaris 버전에 이식되었습니다. LDOM에 패치를 적용했다면 다음과 같이 제대로 정렬된 새로운 LUN을 구성할 준비가 된 것입니다.

1. LUN이 새 zpool에서 사용되고 있는지 확인합니다. 이 예에서는 c2d1 장치입니다.

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

2. ZFS 풀에 사용할 장치의 vdc 인스턴스를 검색합니다.

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

3. 편집 /platform/sun4v/kernel/drv/vdc.conf:

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

이렇게 하면 장치 인스턴스 1이 블록 크기 4096에 할당됩니다.

다른 예로, vdisk 인스턴스 1~6을 4K 블록 크기 및 로 구성해야 한다고 가정합니다 /etc/path_to_inst 는 다음과 같습니다.

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

4. 결승선입니다 vdc.conf 파일에는 다음이 포함되어야 합니다.

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

주의

vdc.conf를 구성하고 vdisk를 생성한 후에 LDOM을 재부팅해야 합니다. 이 단계는 반드시 수행해야 합니다. 블록 크기 변경은 재부팅 후에 적용됩니다. 계속해서 zpool을 구성합니다. 앞서 설명한 것처럼 shift가 12로 설정되었는지 확인합니다.

ZFS Intent Log(ZIL)

일반적인 상황에서는 다른 장치에 ZIL(ZFS Intent Log)을 배치할 이유가 없습니다. 이 로그는 공간을 메인 풀과 공유할 수 있습니다. 개별 ZIL은 최신 스토리지 어레이에서 쓰기 캐싱 기능이 없는 물리적 드라이브를 사용할 때 주로 활용합니다.

로그 바이어스

를 설정합니다 logbias Oracle 데이터를 호스팅하는 ZFS 파일 시스템의 매개 변수입니다.

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

이 매개 변수를 사용하면 쓰기 레벨이 전체적으로 낮아집니다. 기본값으로 설정된 경우, 작성된 데이터는 먼저 ZIL에 할당된 다음 기본 스토리지 풀에 할당됩니다. 이 접근 방식은 기본 스토리지 풀을 위한 SSD 기반 ZIL 장치와 회전식 미디어가 포함된 일반적인 드라이브 구성에 적합합니다. 이는 사용 가능한 미디어의 단일 I/O 트랜잭션에서 커밋이 발생할 수 있도록 하기 때문입니다.

자체 캐싱 기능이 포함된 최신 스토리지 어레이를 사용할 때는 보통 이 접근 방식이 필요하지 않습니다. 매우 집약적이고 지연 시간에 민감한 랜덤 쓰기로 구성된 워크로드 등의 드문 상황에서 로그에 관한 단일 트랜잭션으로 쓰기를 커밋하는

것이 바람직할 때도 있습니다. 로깅된 데이터는 결국 기본 스토리지 풀에 작성되기 때문에 쓰기가 증폭되는 결과가 발생하며 이에 따라 쓰기 활동이 두 배로 늘어납니다.

직접 I/O

Oracle 제품을 포함한 다수의 애플리케이션이 직접 I/O를 활성화하여 호스트 버퍼 캐시를 우회할 수 있으나 ZFS 파일 시스템에서는 이 전략이 예상했던 효과를 발휘하지 않습니다. 호스트 버퍼 캐시를 우회하더라도 ZFS 자체가 계속하여 데이터를 캐싱하기 때문입니다. 이 동작으로 인해 fio 또는 sio 같은 툴을 사용할 때 잘못된 결과가 발생할 수 있는데, I/O가 스토리지 시스템에 도달했는지 또는 운영 체제 내에서 로컬 I/O 캐싱이 이뤄지고 있는지를 예측하기가 어렵기 때문입니다. 또한, 이 동작은 그러한 가상 테스트를 사용하여 ZFS 성능을 다른 파일 시스템과 비교하기 힘들게 만듭니다. 현실적으로 실제 사용자 워크로드에서 파일 시스템의 성능은 거의 차이가 없습니다.

다중 zpool

스냅샷 기반 백업, 복원, 클론, ZFS 기반 데이터 아카이빙은 zpool 레벨에서 수행되어야 하며 일반적으로 여러 개의 zpool이 필요합니다. zpool은 LVM 디스크 그룹과 비슷하며, 같은 규칙을 사용하여 구성해야 합니다. 예를 들어, 데이터베이스는 에 상주하는 데이터 파일과 함께 배치하는 것이 가장 좋습니다 zpool1 및 에 상주하는 아카이브 로그, 제어 파일 및 재실행 로그도 있습니다 zpool2. 이 접근 방식에서는 표준 핫 백업이 허용되어 데이터베이스가 핫 백업 모드로 전환되고 그 뒤에 의 스냅샷이 생성됩니다 zpool1. 그런 다음 핫 백업 모드에서 데이터베이스가 제거되고 로그 아카이브가 강제 적용되며 의 스냅샷이 생성됩니다 zpool2 이 생성됩니다. 복원 작업에서는 zfs 파일 시스템의 마운트를 해제하고 zpool 전체를 오프라인으로 만들어야 하며 그 다음으로 SnapRestore 복원 작업이 이뤄집니다. 이렇게 되면 zpool이 다시 온라인으로 전환될 수 있고 데이터베이스가 복구됩니다.

filesystemio_options 를 참조하십시오

Oracle 매개 변수 filesystemio_options ZFS와 다르게 작동합니다. If(경우 setall 또는 directio 사용되는 경우 쓰기 작업이 동기식이고 운영 체제 버퍼 캐시가 우회되며 ZFS에서 읽기 작업을 버퍼링합니다. 이 동작은 I/O를 가로채서 ZFS 캐시에 의해 서비스하는 경우가 있기 때문에 성능 분석에 어려움을 야기하며, 스토리지 지연 시간과 총 I/O가 실제보다 작아지기 때문입니다.

네트워크 구성

Oracle 데이터베이스용 논리 인터페이스 설계

Oracle 데이터베이스는 스토리지에 액세스해야 합니다. 논리 인터페이스(LIF)는 SVM(스토리지 가상 머신)을 네트워크에 연결하고 데이터베이스에 연결하는 네트워크 사업부입니다. 각 데이터베이스 워크로드에 충분한 대역폭을 보장하기 위해 올바른 LIF 설계가 필요하며, 페일오버로 인해 스토리지 서비스가 손실되지 않습니다.

이 섹션은 LIF 설계의 주요 원칙에 대해 간략하게 설명합니다. 보다 포괄적인 설명서는 를 참조하십시오 ["ONTAP 네트워크 관리 설명서"](#). 데이터베이스 아키텍처의 다른 측면과 마찬가지로, 스토리지 가상 머신(SVM, CLI의 SVM) 및 논리 인터페이스(LIF) 설계를 위한 최고의 옵션은 확장 요구사항과 비즈니스 요구사항에 크게 의존합니다.

LIF 전략을 구축할 때는 다음 주요 주제를 고려해야 합니다.

- * 성능. * 네트워크 대역폭은 충분한가?
- * 복구. * 설계에 단일 장애 지점이 있습니까?
- * 관리 효율성. * 네트워크를 중단 없이 확장할 수 있습니까?

이러한 주제는 호스트부터 스위치, 스토리지 시스템에 이르기까지 엔드 투 엔드 솔루션에 적용됩니다.

LIF 유형

LIF 유형에는 여러 가지가 있습니다. "[LIF 유형에 대한 ONTAP 설명서](#)" 이 주제에 대한 전체 정보를 제공하지만 기능적 관점에서 LIF를 다음 그룹으로 나눌 수 있습니다.

- * 클러스터 및 노드 관리 LIF. * 스토리지 클러스터 관리에 사용되는 LIF.
- * SVM 관리 LIF. * 스냅샷 생성 또는 볼륨 리사이징 같은 기능을 위해 REST API 또는 ONTAPI(ZAPI라고도 함)를 통해 SVM에 대한 액세스를 허용하는 인터페이스. SMO(SnapManager for Oracle)와 같은 제품은 SVM 관리 LIF에 대한 액세스 권한을 가지고 있어야 합니다.
- * Data LIF. * FC, iSCSI, NVMe/FC, NVMe/TCP, NFS용 인터페이스 또는 SMB/CIFS 데이터를 생성할 수 있습니다.



NFS 트래픽에 사용되는 데이터 LIF는 의 방화벽 정책을 에서 변경하여 관리에 사용할 수도 있습니다 data 를 선택합니다 mgmt 또는 HTTP, HTTPS 또는 SSH를 허용하는 또 다른 정책입니다. 이렇게 변경할 때는 NFS 데이터 LIF와 별도의 관리 LIF 둘 다에 대한 액세스를 위해 각 호스트를 구성할 필요가 없기 때문에 네트워크 구성을 단순화할 수 있습니다. 둘 다 IP 프로토콜을 사용함에도 불구하고 iSCSI와 관리 트래픽 모두를 위한 인터페이스를 구성하는 것은 불가능하며 iSCSI 환경에는 별도의 관리 LIF가 필요합니다.

SAN LIF 설계

SAN 환경에서 LIF 설계는 상대적으로 단순합니다. 바로 다중 경로를 사용할 수 있기 때문입니다. 모든 최신 SAN 구현에서는 클라이언트가 다중 독립형 네트워크 경로를 통해 데이터에 액세스하고 액세스를 위한 최상의 경로를 선택할 수 있습니다. 그 결과, SAN 클라이언트가 최상의 경로를 통과하여 자동으로 로드 밸런싱을 수행하기 때문에 LIF 설계에서 성능을 더 간단하게 해결할 수 있습니다.

최상의 경로를 사용할 수 없게 되면 클라이언트는 자동으로 다른 경로를 선택합니다. 이러한 설계 단순성 덕분에 일반적으로 SAN LIF의 관리가 더 용이해집니다. 그렇다고 SAN 환경의 관리가 항상 더 쉬운 것은 아닙니다. SAN 스토리지의 여러 다른 측면은 NFS보다 훨씬 더 복잡하며 SAN LIF는 설계만 더 쉽습니다.

성능

SAN 환경의 LIF 성능과 관련하여 가장 중요한 고려사항은 대역폭입니다. 예를 들어, 노드당 16Gb FC 포트가 2개인 2노드 ONTAP AFF 클러스터는 각 노드에서 최대 32Gb의 대역폭을 허용합니다.

복원력

SAN LIF는 AFF 스토리지 시스템에서 페일오버되지 않습니다. 컨트롤러 페일오버로 SAN LIF에서 장애가 발생하면 클라이언트의 다중 경로 소프트웨어가 경로 손실을 탐지하고 I/O를 다른 LIF로 리디렉션합니다. ASA 스토리지 시스템을 사용할 경우 짧은 지연 후 LIF가 페일오버되지만, 다른 컨트롤러에 이미 활성 경로가 있으므로 IO가 중단되지 않습니다. 페일오버 프로세스는 정의된 모든 포트에서 호스트 액세스를 복구하기 위해 수행됩니다.

관리 효율성

LIF 마이그레이션은 클러스터 주변의 볼륨 재배치와 관련된 경우가 많기 때문에 NFS 환경에서 일반적으로 수행됩니다. SAN 환경에서는 HA 쌍 내에 볼륨을 재배치할 때 LIF를 마이그레이션할 필요가 없습니다. 볼륨 이동이 완료된 후에 경로가 변경되면 ONTAP가 SAN에 알림을 전송하고 SAN 클라이언트가 자동으로 다시 최적화하기 때문입니다. SAN의 LIF 마이그레이션은 기본적으로 주요 물리적 하드웨어 변경과 관련되어 있습니다. 예를 들어, 컨트롤러의 무중단 업그레이드가 필요한 경우 SAN LIF는 새로운 하드웨어로 마이그레이션됩니다. FC 포트에서 오류가 발견되면 LIF를 미사용 포트로 마이그레이션할 수 있습니다.

NetApp는 다음과 같은 권장 사항을 제공합니다.

- 필요한 수보다 많은 경로를 생성하지 마십시오. 경로 수가 너무 많으면 전체적인 관리가 더 복잡해지며 일부 호스트에서 경로 페일오버 문제가 야기될 수 있습니다. 또한, 호스트에 SAN 부팅 같은 구성과 관련하여 예기치 않은 경로 한계가 있을 수도 있습니다.
- 극소수의 구성에서는 LUN에 대한 경로가 4개 이상 필요합니다. LUN과 그 HA 파트너가 속한 노드에 장애가 발생하는 경우 LUN을 호스팅하는 애그리게이트에 액세스할 수 없기 때문에 LUN에 3개 이상의 노드 보급 경로를 소유했을 때의 가치는 크지 않습니다. 이 상황에서 1차 HA 쌍 외에 노드에 대한 경로를 생성하는 것은 도움이 되지 않습니다.
- FC 존에 어떤 포트를 포함할 것인지 선택하여 가시적인 LUN 경로의 수를 관리할 수 있긴 하나 일반적으로 FC 존의 모든 잠재적 타겟 지점을 포함하고 ONTAP 수준에서 LUN 가시성을 제어하는 것이 더 간편합니다.
- ONTAP 8.3 이후 버전에서 선택적 LUN 매핑(SLM: Selective LUN Mapping) 기능은 기본값입니다. SLM 기능을 통해 새로운 LUN은 기본 애그리게이트와 노드의 HA 파트너가 속한 노드에서 자동으로 보급됩니다. 이러한 방식에서는 포트 세트를 생성하거나 포트 접근성을 제한하기 위해 조닝을 구성할 필요가 없습니다. 각 LUN은 최적의 성능과 복원력을 위해 필요한 최소 개수의 노드에서 동작합니다.
- LUN을 2개의 컨트롤러 외부로 마이그레이션해야 하는 경우 `lun mapping add-reporting-nodes` 명령을 사용하여 노드를 추가할 수 있습니다. 이렇게 하면 LUN 마이그레이션을 위해 LUN에 대한 추가 SAN 경로를 생성할 수 있습니다. 그러나 호스트는 새로운 경로를 사용하기 위한 검색 작업을 수행해야 합니다.
- 간접 트래픽을 너무 신경 쓰지 마십시오. 매우 I/O 집약적인 환경에서는 간접 트래픽을 피하는 것이 가장 좋으며 1마이크로초의 지연 시간도 중요하지만 일반적 워크로드의 성능에 미치는 가시적 영향은 미미합니다.

NFS LIF 설계

SAN 프로토콜과 대조적으로 NFS는 데이터에 대한 다중 경로를 정의하는 기능이 제한적입니다. NFSv4에 대한 병렬 NFS(pNFS) 확장이 이 한계를 해결해주긴 하나, 이더넷 속도가 100GB에 도달했기 때문에 추가 경로를 추가할 가치가 거의 없습니다.

성능 및 복원력

SAN LIF 성능을 측정할 때는 기본적으로 모든 1차 경로의 총 대역폭을 계산하는 것이 관건이지만 NFS LIF 성능을 결정하기 위해서는 정확한 네트워크 구성을 면밀히 살펴야 합니다. 예를 들어, 2개의 10Gb 포트를 원시 물리적 포트 구성하거나 링크 통합 제어 프로토콜(LACP) 인터페이스 그룹으로 구성할 수 있습니다. 이를 인터페이스 그룹으로 구성하는 경우, 트래픽이 스위칭될 때와 라우팅될 때 각각 다르게 작동하는 다중 로드 밸런싱 정책을 사용할 수 있습니다. 마지막으로, Oracle dNFS(Direct NFS)는 현재 어떤 OS NFS 클라이언트에도 존재하지 않는 로드 밸런싱 구성을 제공합니다.

SAN 프로토콜과 달리 NFS 파일 시스템은 프로토콜 계층의 복원력이 필요합니다. 예를 들어, LUN은 항상 다중 경로가 활성화되어 구성되므로 스토리지 시스템에 다중 이중화 채널이 제공되고 각각 FC 프로토콜을 사용하게 됩니다. 반면, NFS 파일 시스템은 물리적 계층에서만 보호되는 단일 TCP/IP 채널의 가용성에 따라 달라집니다. 포트 페일오버와 LACP 포트 애그리게이션 같은 옵션은 이 방식 때문에 존재하는 것입니다.

NFS 환경에서는 네트워크 프로토콜 계층에서 성능과 복원력 둘 다 제공되기 때문에 두 주제가 긴밀히 연관되어 있으며 함께 논의되어야 합니다.

포트 그룹에 LIF를 바인딩합니다

LIF를 포트 그룹에 바인딩하려면 LIF IP 주소를 물리적 포트 그룹에 연계합니다. 물리적 포트를 함께 애그리게이팅하는 주된 방법은 LACP입니다. LACP의 내결함성 기능은 상당히 단순한데, LACP 그룹의 각 포트를 모니터링하고 오작동이

발생하면 포트 그룹에서 제거하는 것입니다. 그러나 LACP의 성능과 관련하여 다음과 같이 많은 오해가 있습니다.

- LACP는 엔드포인트 매칭을 위해 스위치를 구성하지 않아도 됩니다. 예를 들어, ONTAP은 IP 기반 부하 분산을 사용하여 구성할 수 있고 스위치는 MAC 기반 부하 분산을 사용할 수 있습니다.
- LACP 연결을 사용하는 각 엔드포인트는 패킷 전송 포트를 독립적으로 선택할 수 있지만 수신에 사용할 포트는 선택할 수 없습니다. 즉, ONTAP에서 특정 대상으로 가는 트래픽이 특정 포트에 연관되어 있고 반환 트래픽은 다른 인터페이스에 도착할 수 있습니다. 하지만 이로 인해 문제가 발생하지는 않습니다.
- LACP가 트래픽을 언제나 균등하게 분산하지는 않습니다. 다수의 NFS 클라이언트가 있는 대규모 환경에서는 일반적으로 LACP 애그리게이션의 모든 포트가 균등하게 사용됩니다. 그러나 이 환경에서 모든 NFS 파일 시스템은 전체 애그리게이션이 아닌 단 1포트의 대역폭으로 제한됩니다.
- ONTAP에서 라운드 로빈 LACP 정책을 사용할 수 있지만 이들 정책은 스위치에서 호스트로의 연결을 다루지 않습니다. 예를 들어, 한 호스트에 4포트 LACP 트렁크가 있고 ONTAP에 4포트 LACP 트렁크가 있는 구성에서는 단일 포트를 사용하여 파일 시스템을 읽을 수만 있습니다. ONTAP은 4포트 모두를 통해 데이터를 전송할 수 있지만 현재 4포트 모두를 통해 스위치에서 호스트로 전송하는 데 사용할 수 있는 스위치 기술은 없으며 하나만 사용됩니다.

여러 데이터베이스 호스트로 구성된 대규모 환경에서 가장 일반적인 접근 방식은 IP 로드 밸런싱을 사용하여 적절한 수의 10Gb(또는 더 빠른) 인터페이스 LACP 애그리게이트를 구축하는 것입니다. 이 접근 방식에서는 클라이언트 수가 충분하다면 ONTAP에서 모든 포트를 사용할 수 있습니다. 구성에 있는 클라이언트 수가 더 적을 때는 LACP 트렁킹이 로드를 동적으로 재분산하지 않으므로 로드 밸런싱이 중단됩니다.

연결이 확립되면 특정 방향의 트래픽이 하나의 포트에만 배치됩니다. 예를 들어, 4포트 LACP 트렁크로 연결된 NFS 파일 시스템에 대해 전체 테이블 스캔을 수행하는 데이터베이스는 네트워크 인터페이스 카드(NIC)가 하나에 불과하지만 데이터를 읽습니다. 이러한 환경에 단 3개의 데이터베이스 서버가 있는 경우 3개 서버 모두 같은 포트에서 데이터를 읽을 가능성도 있으며 다른 3개의 포트는 유향 상태입니다.

LIF를 물리적 포트에 바인딩합니다

LIF를 물리적 포트에 바인딩하면 ONTAP 시스템의 특정 IP 주소가 한 번에 하나의 네트워크 포트에만 연계되기 때문에 네트워크 구성을 더 세부적으로 제어할 수 있습니다. 이렇게 하고 나면 파일오버 그룹 구성과 파일오버 정책을 통해 복원력을 실현할 수 있습니다.

파일오버 정책 및 파일오버 그룹

네트워크가 중단되었을 때 LIF의 동작은 파일오버 정책과 파일오버 그룹에 의해 제어됩니다. 구성 옵션은 ONTAP의 다른 버전에 따라 변경되었습니다. 을 참조하십시오 ["파일오버 그룹 및 정책에 대한 ONTAP 네트워크 관리 설명서"](#) 구축하고 있는 ONTAP 버전에 대한 세부 정보를 참조하십시오.

ONTAP 8.3 이상에서는 브로드캐스트 도메인 기반의 LIF 파일오버 관리를 허용합니다. 그러므로 관리자는 특정 서브넷에 대한 액세스 권한을 가진 모든 포트를 정의하여 ONTAP이 적절한 파일오버 LIF를 선택하도록 할 수 있습니다. 어떤 고객은 이 접근 방식을 사용할 수 있지만 예측 가능성이 부족하기 때문에 고속 스토리지 네트워크 환경에서는 한계가 있습니다. 예를 들어, 일반적인 파일 시스템 액세스를 위한 1Gb 포트와 데이터 파일 I/O를 위한 10Gb 포트 모두를 환경에 포함할 수 있습니다 두 유형의 포트가 같은 브로드캐스트 도메인에 존재하는 경우 LIF 파일오버는 데이터 파일 I/O를 10Gb 포트에서 1Gb 포트에 이동할 수 있습니다.

요약하자면, 다음과 같은 방식을 사용해 보십시오.

1. 사용자 정의대로 파일오버 그룹을 구성합니다.
2. 스토리지 파일오버 중에 LIF가 애그리게이트를 따르도록 스토리지 파일오버(SFO) 파트너 컨트롤러의 포트에 파일오버 그룹을 채웁니다. 그러면 간접 트래픽의 생성을 방지할 수 있습니다.
3. 성능 특성이 원래의 LIF와 일치하는 파일오버 포트를 사용합니다. 예를 들어, 하나의 물리적 10Gb 포트에 있는 LIF에는 단일 10Gb 포트의 파일오버 그룹이 포함되어야 합니다. 4포트 LACP LIF는 다른 4포트 LACP LIF로

페일오버해야 합니다. 이들 포트는 브로드캐스트 도메인에서 정의된 포트의 하위 세트가 될 것입니다.

4. SFO 파트너에 관한 페일오버 정책을 수립합니다. 이렇게 하면 페일오버 중에 LIF가 애그리게이트를 따르도록 할 수 있습니다.

자동 되돌리기

를 설정합니다 `auto-revert` 원하는 대로 매개 변수입니다. 대부분의 고객은 이 매개 변수를 `로` 설정하는 것을 선호합니다 `true` LIF가 홈 포트에 되돌아갑니다. 그러나 경우에 따라 LIF를 홈 포트에 반환하기 전에 예기치 않은 페일오버를 조사할 수 있다는 사실이 이를 `false`로 설정한 경우도 있습니다.

LIF-볼륨 비율

일반적인 오해는 볼륨과 NFS LIF 사이에 1:1 관계가 있어야 한다는 것입니다. 이 구성은 인터커넥트 트래픽을 추가로 생성하지 않고 클러스터의 어느 곳으로든 볼륨을 이동하기 위해 필요하기는 하나 절대적인 요구사항은 아닙니다. 인터클러스터 트래픽을 고려해야 하지만 단순히 인터클러스터 트래픽이 존재하는 것만으로 문제가 발생하지는 않습니다. ONTAP를 위해 수립되고 발표된 대다수의 벤치마크에는 대개 간접 I/O가 포함되어 있습니다

예를 들어, 성능이 중요한 데이터베이스가 상대적으로 적게 포함된 데이터베이스 프로젝트에서 LIF 전략에 대한 1:1 볼륨을 보장하기 위해 총 40개의 볼륨만 필요하다면 IP 주소는 40개가 필요합니다. 어떤 볼륨이든 연계된 LIF와 함께 클러스터 내 어느 곳으로든 이동할 수 있으며 트래픽이 항상 직접적이기 때문에 마이크로초 수준에서도 지연 시간의 소스를 모두 최소화합니다.

반대의 예를 들어 보면, 대규모 호스팅 환경은 고객과 LIF 간 1:1 관계를 더 쉽게 관리할 수 있습니다. 시간이 경과하면 볼륨을 다른 노드로 마이그레이션해야 할 수 있으며 이로 인해 간접 트래픽이 발생할 수 있습니다. 하지만 인터커넥트 스위치의 네트워크 포트가 포화 상태가 되지 않는 한 성능 영향을 감지할 수 없습니다. 우려가 된다면 새로운 LIF를 추가 노드에 설정할 수 있으며 다음 유지보수 윈도우에 호스트를 업데이트하여 구성에서 간접 트래픽을 제거할 수 있습니다.

Oracle 데이터베이스용 TCP/IP 및 이더넷 구성

많은 Oracle on ONTAP 고객은 이더넷, NFS, iSCSI, NVMe/TCP 및 특히 클라우드를 사용합니다.

호스트 OS 설정

대부분의 응용 프로그램 공급업체 설명서에는 응용 프로그램이 최적의 상태로 작동하도록 하기 위한 특정 TCP 및 이더넷 설정이 포함되어 있습니다. 일반적으로 이러한 설정은 최적의 IP 기반 스토리지 성능을 제공하기에 충분합니다.

이더넷 흐름 제어

이 기술을 통해 클라이언트는 발신자에게 데이터 전송을 일시적으로 중단할 것을 요청할 수 있습니다. 이 요청은 보통 수신자가 유입되는 데이터를 빠르게 처리할 수 없을 때 이루어집니다. 이와 동시에, 발신자에게 전송 중단을 요청할 때는 버퍼가 가득 차서 수신자가 패킷을 폐기할 때보다 중단 시간이 짧습니다. 이는 이제 오늘날 운영 체제에서 사용되는 TCP 스택의 사례가 아닙니다. 사실, 흐름 제어는 해결하는 문제보다 야기하는 문제가 더 많습니다.

최근 들어 이더넷 흐름 제어가 초래하는 성능 문제가 증가해 왔는데 이는 이더넷 흐름 제어가 물리적 계층에서 작동하기 때문입니다. 네트워크 구성에서 호스트 운영 체제에서 이더넷 흐름 제어 요청을 스토리지 시스템으로 전송하도록 허용하는 경우, 연결된 모든 클라이언트의 I/O가 중지됩니다. 단일 스토리지 컨트롤러가 처리하는 클라이언트의 수가 증가하면 이러한 클라이언트가 하나 이상의 흐름 제어 요청을 보낼 가능성이 높아집니다. 방대한 운영 체제 가상화를 구현한 고객 사이트에서 문제가 자주 관찰되었습니다.

NetApp 시스템의 NIC는 흐름 제어 요청을 수신하면 안 됩니다. 이러한 결과를 얻기 위해 사용하는 방법은 네트워크

스위치 제조업체에 따라 다릅니다. 대부분의 경우 이더넷 스위치의 흐름 제어는 `receive desired` 또는 `receive on` 다시 말해, 흐름 제어 요청이 스토리지 컨트롤러로 전달되지 않습니다. 스토리지 컨트롤러의 네트워크 연결이 흐름 제어 비활성화를 허용하지 않는 경우도 있습니다. 이 경우 호스트 서버 자체의 NIC 구성 또는 호스트 서버가 연결되는 스위치 포트를 변경하여 클라이언트가 흐름 제어 요청을 전송하지 않도록 구성해야 합니다.



* NetApp는 NetApp 스토리지 컨트롤러가 이더넷 흐름 제어 패킷을 수신하지 않도록 하는 것을 권장합니다. 일반적으로는 컨트롤러가 연결될 스위치 포트를 설정하면 되지만 일부 스위치 하드웨어에는 제약이 있어 클라이언트 측 변경이 필요할 수 있습니다.

MTU 크기

CPU와 네트워크 오버헤드를 줄여 1Gb 네트워크에서 성능을 개선하기 위해 점보 프레임이 사용되어온 것으로 나타났지만 대부분은 이점이 크지 않습니다.



*NetApp는 잠재적인 성능 이점을 실현하고 솔루션의 미래 경쟁력을 확보하기 위해 가능한 경우 점보 프레임을 구현하는 것을 권장합니다.

10Gb 네트워크에서는 점보 프레임을 거의 의무적으로 사용해야 합니다. 대부분의 10Gb 구현에서 10Gb 표시에 도달하기 전에 점보 프레임이 없이는 초당 패킷 한계에 도달하기 때문입니다. 점보 프레임을 사용하면 OS, 서버, NIC, 스토리지 시스템에서 처리되는 패킷의 수가 줄어들었지만 크기가 더 커질 수 있기 때문에 TCP/IP 프로세싱의 효율성이 개선됩니다. NIC에 따라 달라지기는 하지만 상당한 성능 개선이 이루어집니다.

점보 프레임 구현의 경우 연결된 모든 장치가 점보 프레임을 지원해야 하며 MTU 크기가 완전히 일치해야 한다는 잘못된 인식이 일반적입니다. 연결을 설정할 때 2개의 네트워크 엔드 포인트는 상호 허용되는 최대 프레임 크기를 협상합니다. 일반적인 환경에서 네트워크 스위치는 MTU 크기 9216으로 설정되며 NetApp 컨트롤러는 9000, 클라이언트는 9000과 1514의 혼합으로 설정됩니다. 9000의 MTU를 지원하는 클라이언트는 점보 프레임을 사용할 수 있으며 1514만 지원하는 클라이언트는 더 낮은 값을 협상할 수 있습니다.

이 방식의 문제는 완전히 스위칭된 환경에서는 자주 사용되지 않는다는 것입니다. 하지만 점보 프레임을 단편화하도록 강제 적용되는 중간 라우터가 없는 라우팅 환경에서는 주의하십시오.



- NetApp는 다음을 * 구성할 것을 권장합니다.
- 1Gb 이더넷(GbE)에서 점보 프레임을 사용하면 좋지만 필수는 아닙니다.
- 10GbE 이상의 속도로 최대 성능을 얻으려면 점보 프레임이 필요합니다.

TCP 매개 변수입니다

3가지 설정, 즉 TCP 타임스탬프, 선택적 승인(Selective Acknowledgment, SACK), TCP 윈도우 확장이 잘못 구성되는 경우가 많습니다. 인터넷상의 오래된 문서들은 이러한 매개 변수를 하나 이상 비활성화하여 성능을 개선할 것을 권장합니다. CPU 용량이 훨씬 낮았고 가능할 때마다 TCP 처리 오버헤드를 줄여 이점을 얻을 수 있었던 몇 해 전에는 이 권장사항이 어느 정도 가치가 있었습니다.

그러나 최신 운영 체제에서 이러한 TCP 기능을 비활성화하면 일반적으로 어떤 이점도 얻을 수 없는 반면 성능 저하가 야기될 수도 있습니다. 성능 저하는 특히 가상화된 네트워킹 환경에서 발생할 가능성이 높는데 이들 기능이 패킷 손실을 효율적으로 처리하고 네트워크 품질을 바꾸는 데 필요하기 때문입니다.



* NetApp는 호스트에서 TCP 타임스탬프, SACK 및 TCP 윈도우 확장을 활성화하는 것을 권장하며, 현재 모든 OS에서 이러한 매개 변수 세 개가 기본적으로 켜져 있어야 합니다.

Oracle 데이터베이스를 위한 FC 구성

Oracle 데이터베이스용 FC SAN 구성은 주로 일상적인 SAN 모범 사례를 따르는 것과 관련됩니다.

여기에는 호스트와 스토리지 시스템 사이에 SAN에 충분한 대역폭이 존재하는지 확인하고, FC 스위치 공급업체에서 요구하는 FC 포트 설정을 사용하여 모든 필수 디바이스 간에 모든 SAN 경로가 존재하는지 확인하는 등의 일반적인 계획 조치가 포함됩니다. ISL 경합 방지 그리고 적절한 SAN 패브릭 모니터링을 사용합니다.

조닝

FC 존은 하나 이상의 이니시에이터를 포함할 수 없습니다. 이 방식은 처음에는 효과가 있는 것처럼 보이지만 이니시에이터 간 혼선이 결국에는 성능과 안정성을 저하시킵니다.

여러 공급업체의 FC 타겟 포트 동작이 드물게 문제를 일으키기도 하지만 멀티타겟 존은 일반적으로 안전하다고 간주됩니다. 예를 들어, NetApp와 NetApp이 아닌 스토리지 어레이의 타겟 포트를 같은 존에 포함하지 마십시오. 또한, NetApp 스토리지 시스템과 테이프 장치를 같은 존에 배치하면 문제가 발생할 가능성이 더 높아집니다.

Oracle 데이터베이스 및 직접 연결 ONTAP 연결

스토리지 관리자는 구성에서 네트워크 스위치를 제거하여 인프라를 단순화하기를 원할 수도 있습니다. 일부 시나리오에서는 이 기능이 지원될 수 있습니다.

iSCSI 및 NVMe/TCP

iSCSI 또는 NVMe/TCP를 사용하는 호스트는 스토리지 시스템에 직접 연결하여 정상적으로 작동할 수 있습니다. 그 이유는 경로 지정입니다. 두 개의 서로 다른 스토리지 컨트롤러에 직접 연결되므로 데이터 흐름을 위한 두 개의 독립적 경로가 됩니다. 경로, 포트 또는 컨트롤러가 손실되어도 다른 경로가 사용되지 않습니다.

NFS 를 참조하십시오

직접 연결 NFS 스토리지를 사용할 수 있지만 중대한 제한 사항이 있는 경우 스크립팅의 상당한 노력 없이는 페일오버가 수행되지 않으며 고객의 책임입니다.

직접 연결 NFS 스토리지에서 무중단 페일오버가 복잡해지는 이유는 로컬 OS에서 발생하는 라우팅입니다. 예를 들어, 호스트의 IP 주소가 192.168.1.1/24이고 IP 주소가 192.168.1.50/24인 ONTAP 컨트롤러에 직접 연결되어 있다고 가정합니다. 장애 조치 중에 192.168.1.50 주소는 다른 컨트롤러로 장애 조치될 수 있으며 호스트에서 사용할 수 있지만 호스트는 어떻게 그 존재를 감지합니까? 원래 192.168.1.1 주소는 더 이상 운영 체제에 연결되지 않는 호스트 NIC에 계속 존재합니다. 192.168.1.50으로 향하는 트래픽은 작동하지 않는 네트워크 포트에 계속 전송됩니다.

두 번째 OS NIC를 19로 구성할 수 있습니다 2.168.1.2 및 은 192.168.1.50을 통해 실패한 주소와 통신할 수 있지만, 로컬 라우팅 테이블은 기본적으로 192.168.1.0/24 서브넷과 통신하는 데 하나의 * 및 하나의 * 주소만 사용합니다. sysadmin은 실패한 네트워크 연결을 감지하고 로컬 라우팅 테이블을 변경하거나 인터페이스를 가동 및 중지시키는 스크립팅 프레임워크를 생성할 수 있습니다. 정확한 절차는 사용 중인 운영 체제에 따라 다릅니다.

실제로 NetApp 고객은 직접 연결 NFS를 가지고 있지만 일반적으로 페일오버 중에 IO가 일시 중지되는 워크로드에만 해당됩니다. 하드 마운트를 사용하는 경우 이러한 일시 중지 중에는 입출력 오류가 발생하지 않아야 합니다. 호스트의 NIC 간에 IP 주소를 이동하는 장애 복구 또는 수동 작업으로 인해 서비스가 복구될 때까지 입출력이 중단되어야 합니다.

FC 직접 연결

호스트를 FC 프로토콜을 사용하여 ONTAP 스토리지 시스템에 직접 연결할 수는 없습니다. NPIV를 사용하기 때문입니다. FC 네트워크에 대한 ONTAP FC 포트를 식별하는 WWN은 NPIV라는 가상화 유형을 사용합니다. ONTAP 시스템에 연결된 모든 디바이스가 NPIV WWN을 인식할 수 있어야 합니다. 현재 NPIV 타겟을 지원할 수 있는 호스트에 설치할 수 있는 HBA를 제공하는 HBA 공급업체는 없습니다.

스토리지 구성

FC SAN

Oracle 데이터베이스 입출력에 대한 LUN 정렬

LUN 정렬은 기본 파일 시스템 레이아웃과 관련하여 I/O를 최적화하는 것을 칭합니다.

ONTAP 시스템에서 스토리지는 4KB 장치로 구성됩니다. 데이터베이스 또는 파일 시스템 8KB 블록은 정확히 2개의 4KB 블록에 매핑되어야 합니다. LUN 구성 오류로 인해 정렬이 어느 방향으로든 1KB 이동되면 각 8KB 블록은 2개가 아닌 3개의 서로 다른 4KB 스토리지 블록에 존재하게 됩니다. 이 방식은 스토리지 시스템 내에서 지연 시간이 늘어나고 추가적인 I/O가 수행되도록 합니다.

정렬은 LVM 아키텍처에도 영향을 미칩니다. 논리적 볼륨 그룹 내의 물리적 볼륨은 전체 드라이브 장치에서 정의되며 (생성되는 파티션 없음), LUN의 첫 번째 4KB 블록이 스토리지 시스템의 첫 번째 4KB 블록에 맞춰 정렬됩니다. 이것이 올바른 정렬입니다. 문제는 파티션에서 발생하는데, 운영 체제가 LUN을 사용하는 시작 위치를 파티션이 바꾸기 때문입니다. 오프셋이 4KB 단위로 이동하는 한 LUN은 정렬됩니다.

Linux 환경에서는 전체 드라이브 장치에 논리적 볼륨 그룹이 구축됩니다. 파티션이 필요하면 `lvm`를 실행하여 정렬을 점검합니다 `fdisk -u` 각 파티션의 시작이 8의 배수인지 확인합니다. 이는 8개의 512바이트 섹터를 곱한 값인 4KB에서 파티션이 시작된다는 뜻입니다.

또한 압축 블록 정렬에 대한 내용은 섹션을 참조하십시오 ["효율성"](#). 8KB 압축 블록 경계에 맞춰 정렬된 레이아웃은 4KB 경계에도 맞춰집니다.

정렬 불량 경고

데이터베이스 재실행/트랜잭션 로깅은 일반적으로 정렬되지 않은 I/O를 생성하여 ONTAP에서 정렬 불량 LUN과 관련된 경고를 야기할 수 있으며 이는 사용자를 호도할 수 있습니다.

로깅은 다양한 크기의 쓰기로 로그 파일의 순차적 쓰기를 수행합니다. 4KB 경계에 맞춰 정렬되지 않는 로그 쓰기 작업은 일반적으로 성능 문제를 야기하지 않는데 다음 로그 쓰기 작업에서 블록이 완료되기 때문입니다. 결과적으로 4KB 블록이 완료되면 ONTAP은 거의 모든 쓰기를 처리할 수 있으며, 이는 일부 4KB 블록의 데이터가 2개의 개별 작업으로 작성되었더라도 마찬가지입니다.

과 같은 유틸리티를 사용하여 정렬을 확인합니다 `sio` 또는 `dd` 이를 통해 정의된 블록 크기로 I/O를 생성할 수 있습니다. 스토리지 시스템의 I/O 정렬 통계는 `lvs`에서 확인할 수 있습니다 `stats` 명령. 을 참조하십시오 ["WAFL 정렬 확인"](#) 를 참조하십시오.

Solaris 환경에서는 정렬이 더 복잡합니다. 을 참조하십시오 ["ONTAP SAN 호스트 구성"](#) 를 참조하십시오.

주의

Solaris x86 환경에서는 대부분의 구성에 여러 개의 파티션 계층이 있기 때문에 정렬이 제대로 되는지 더 주의를 기울여야 합니다. Solaris x86 파티션 조각은 일반적으로 표준 마스터 부트 레코드 파티션 테이블의 상단에 존재합니다.

Oracle 데이터베이스 LUN 사이징 및 LUN 수

최적의 LUN 크기와 사용할 LUN 수를 선택하는 것은 Oracle 데이터베이스의 성능 및 관리 효율성을 최적화하는 데 매우 중요합니다.

LUN은 ONTAP의 가상화 오브젝트로서 호스팅 애그리게이트의 모든 드라이브에 존재합니다. 그 결과, LUN은 어떤 크기를 선택하든 애그리게이트의 잠재력을 최대한으로 활용하기 때문에 크기가 LUN의 성능에 영향을 미치지 않습니다.

고객이 편의상 특정 크기의 LUN을 사용하려 할 수 있습니다. 예를 들어, 데이터베이스가 각각 1TB인 2개의 LUN으로 구성된 LVM 또는 Oracle ASM 디스크 그룹에 구축되어 있는 경우 이 디스크 그룹은 1TB의 증분으로 확장되어야 합니다. 디스크 그룹이 더 적은 증분으로 증가할 수 있도록 각각 500GB인 8개의 LUN으로 디스크 그룹을 구축하는 것이 더 낫습니다.

범용 표준 LUN 크기를 설정하는 방법은 관리가 복잡할 수 있으므로 사용하지 않는 것이 좋습니다. 예를 들어, 100GB의 표준 LUN 크기는 1TB~2TB 범위의 데이터베이스에서는 효과적일 수 있지만 20TB 크기의 데이터베이스나 데이터 저장소에는 200개의 LUN이 필요할 것입니다. 이에 따라 서버 재부팅 시간이 길어지고, 다양한 UI에서 더 많은 오브젝트를 관리해야 하며, SnapCenter 같은 제품을 통해 여러 오브젝트를 탐색해야 합니다. LUN의 개수는 줄이고 크기는 늘리면 이 문제를 피할 수 있습니다.

- LUN 수가 LUN 크기보다 더 중요합니다.
- LUN 크기는 주로 LUN 개수 요구사항에 따라 결정됩니다.
- LUN을 필요한 것보다 많이 생성하지 마십시오.

LUN 수입니다

LUN 크기와 달리 LUN 개수는 성능에 영향을 미칩니다. 애플리케이션 성능은 SCSI 계층을 통해 병렬 I/O를 수행하는 능력에 따라 달라집니다. 그 결과 2개의 LUN이 1개의 LUN보다 나은 성능을 제공합니다. Veritas VxVM, Linux LVM2 또는 Oracle ASM 같은 LVM은 병렬 처리를 늘리는 가장 단순한 방법입니다.

부하가 매우 큰 랜덤 I/O의 100% SSD 환경 테스트에서 최대 64개 LUN이라는 추가 개선이 입증되었음에도 NetApp 고객은 일반적으로 LUN의 개수가 16을 초과하여 증가할 때 최소한의 이점을 경험합니다.

- NetApp는 * 다음을 권장합니다.



일반적으로 특정 데이터베이스 워크로드의 입출력 요구를 지원하는 데는 4~16개의 LUN으로 충분합니다. 4개 미만의 LUN은 호스트 SCSI 구현의 제한으로 인해 성능 제한이 발생할 수 있습니다.

Oracle 데이터베이스 LUN 배치

ONTAP 볼륨 내에 데이터베이스 LUN을 최적으로 배치하는 것은 주로 다양한 ONTAP 기능을 사용하는 방법에 따라 달라집니다.

볼륨

ONTAP를 처음 접하는 고객에게 혼동을 주는 한 가지 일반적인 점은 일반적으로 "볼륨"이라고 하는 FlexVol을 사용하는 것입니다.

볼륨이 LUN이 아닙니다. 이러한 용어는 클라우드 공급자를 포함한 다른 여러 공급업체 제품과 동일하게 사용됩니다. ONTAP 볼륨은 단순한 관리 컨테이너입니다. 이러한 애플리케이션은 자체적으로 데이터를 제공하지 않으며 공간을 점유하지 않습니다. 파일 또는 LUN용 컨테이너이며, 관리 효율성을 개선하고 단순화하기 위해 존재하며, 특히 규모에 맞게 사용됩니다.

볼륨 및 LUN

관련 LUN은 일반적으로 단일 볼륨에 함께 배치됩니다. 예를 들어, 10개의 LUN이 필요한 데이터베이스에서는 일반적으로 10개의 LUN이 모두 동일한 볼륨에 배치되어 있습니다.



- 볼륨에 대한 LUN의 1:1 비율을 사용하는 것은 볼륨당 하나의 LUN을 사용하는 것은 공식적인 모범 사례가 아닙니다.
- 대신, 볼륨을 워크로드 또는 데이터 세트의 컨테이너로 보아야 합니다. 볼륨당 하나의 LUN이 있거나 여러 LUN이 있을 수 있습니다. 정답은 관리 효율 요구사항에 따라 다릅니다.
- 불필요한 볼륨 수에 걸쳐 LUN을 분산시키면 스냅샷 작업, UI에 표시되는 개체 수가 너무 많아지고 LUN 제한에 도달하기 전에 플랫폼 볼륨 제한에 도달할 수 있는 등의 작업에 추가적인 오버헤드 및 스케줄링 문제가 발생할 수 있습니다.

볼륨, LUN 및 스냅샷

스냅샷 정책과 일정은 LUN이 아닌 볼륨에 배치됩니다. 10개의 LUN으로 구성된 데이터 세트에는 해당 LUN이 동일한 볼륨에 공존할 경우 단일 스냅샷 정책만 필요합니다.

또한 특정 데이터 세트에 대한 모든 관련 LUN을 단일 볼륨에 함께 배치하면 원자 단위 스냅샷 작업이 가능합니다. 예를 들어, 10개의 LUN에 상주하는 데이터베이스 또는 10개의 서로 다른 OS로 구성된 VMware 기반 애플리케이션 환경은 기본 LUN이 모두 단일 볼륨에 배치되어 있는 경우 일관된 단일 오브젝트로 보호할 수 있습니다. 서로 다른 볼륨에 배치되는 경우 동시에 예약된 경우에도 스냅샷이 100% 동기화될 수도 있고 그렇지 않을 수도 있습니다.

경우에 따라 복구 요구 사항으로 인해 관련 LUN 세트를 두 개의 다른 볼륨으로 분할해야 할 수도 있습니다. 예를 들어, 데이터베이스에 데이터 파일에는 LUN 4개와 로그에는 LUN 2개가 포함될 수 있습니다. 이 경우 4개의 LUN이 있는 데이터 파일 볼륨과 2개의 LUN이 포함된 로그 볼륨이 최상의 옵션이 될 수 있습니다. 그 이유는 독립적인 복구 기능입니다. 예를 들어, 데이터 파일 볼륨을 선택적으로 이전 상태로 복원할 수 있습니다. 즉, 4개의 LUN이 모두 스냅샷 상태로 되돌려지고 중요 데이터가 있는 로그 볼륨은 영향을 받지 않습니다.

볼륨, LUN 및 SnapMirror를 지원합니다

SnapMirror 정책 및 작업은 스냅샷 작업과 마찬가지로 LUN이 아닌 볼륨에 대해 수행됩니다.

단일 볼륨에서 관련 LUN을 함께 배치하면 단일 SnapMirror 관계를 생성하고 포함된 모든 데이터를 단일 업데이트로 업데이트할 수 있습니다. 스냅샷과 마찬가지로 업데이트는 원자 단위 작업이기도 합니다. SnapMirror 대상은 소스 LUN의 단일 시점 복제본을 보유하도록 보장됩니다. LUN이 여러 볼륨에 분산되어 있는 경우 복제본이 서로 정합성이 보장되거나 일치하지 않을 수 있습니다.

볼륨, LUN 및 QoS를 지원합니다

QoS를 개별 LUN에 선택적으로 적용할 수 있지만 일반적으로 볼륨 수준에서 설정하기가 더 쉽습니다. 예를 들어, 특정 ESX 서버에서 게스트가 사용하는 모든 LUN을 단일 볼륨에 배치한 다음 ONTAP 적응형 QoS 정책을 적용할 수

있습니다. 그 결과 모든 LUN에 적용되는 TB당 자체 확장 IOPS 제한이 적용됩니다.

마찬가지로, 데이터베이스에 100K IOPS가 필요하고 10개의 LUN을 사용하는 경우 각 LUN에 하나씩 10개의 개별 10K IOPS 제한을 설정하는 것보다 단일 볼륨에 단일 100K IOPS 제한을 설정하는 것이 더 쉽습니다.

다중 볼륨 레이아웃

여러 볼륨에 LUN을 분산하는 것이 도움이 되는 경우도 있습니다. 주된 이유는 컨트롤러 스트라이핑입니다. 예를 들어, HA 스토리지 시스템은 각 컨트롤러의 완전한 처리와 캐싱 잠재성이 필요한 단일 데이터베이스를 호스팅하고 있을 수 있습니다. 이 경우 일반적인 설계에서는 LUN의 절반을 컨트롤러 1의 단일 볼륨에 배치하고 나머지 절반은 컨트롤러 2의 단일 볼륨에 배치하는 것입니다.

마찬가지로, 로드 밸런싱에 컨트롤러 스트라이핑을 사용할 수도 있습니다. 10개의 LUN으로 구성된 100개의 데이터베이스를 호스팅하는 HA 시스템은 각 데이터베이스가 2개의 컨트롤러 각각에서 5개의 LUN 볼륨을 받을 수 있도록 설계될 수 있습니다. 그 결과, 추가 데이터베이스를 프로비저닝할 때 각 컨트롤러를 대칭으로 로드할 수 있습니다.

단, 이 예에서는 볼륨 대 LUN 비율에 1:1이 관련되지 않습니다. 목표는 볼륨에 관련된 LUN을 함께 배치함으로써 관리 효율성을 최적화하는 것입니다.

1:1 LUN 대 볼륨 비율이 적합한 한 가지 예는 컨테이너화입니다. 각 LUN이 실제로 단일 워크로드를 나타내므로 각 워크로드를 개별적으로 관리해야 합니다. 이 경우 1:1 비율이 최적일 수 있습니다.

Oracle 데이터베이스 LUN 리사이징과 LVM 기반 리사이징

SAN 기반 파일 시스템이 용량 제한에 도달하면 사용 가능한 공간을 늘리기 위한 두 가지 옵션이 있습니다.

- LUN의 크기를 늘립니다
- 기존 볼륨 그룹에 LUN을 추가하고 포함된 논리적 볼륨을 늘립니다

LUN 리사이징이 용량을 증대하는 옵션이기는 하지만 Oracle ASM을 포함한 LVM을 사용하는 것이 대체로 더 낫습니다. LVM이 존재하는 주된 이유 중 하나는 LUN 리사이징이 필요하지 않도록 하는 것입니다. LVM은 여러 개의 LUN을 스토리지 가상 풀에 함께 바인딩합니다. 이 풀에서 생성된 논리적 볼륨은 LVM이 관리하며 쉽게 리사이징할 수 있습니다. 이 외에도 특정 논리적 볼륨을 가용 LUN 전반에 분산하여 특정 드라이브의 핫스팟을 피할 수 있다는 이점이 있습니다. 일반적으로 볼륨 관리자를 사용하여 투명한 마이그레이션을 수행함으로써 논리적 볼륨의 기본 익스텐트를 새 LUN에 재배치할 수 있습니다.

Oracle 데이터베이스를 사용한 LVM 스트라이핑

LVM 스트라이핑은 여러 LUN에 데이터를 배포하는 것을 의미합니다. 그 결과 많은 데이터베이스의 성능이 크게 향상되었습니다.

플래시 드라이브의 시대 이전에는 스트라이핑이 회전식 드라이브의 성능 제한을 극복하는 데 사용되었습니다. 예를 들어, OS에서 1MB 읽기 작업을 수행해야 하는 경우 1MB가 느리게 전송되기 때문에 단일 드라이브에서 1MB의 데이터를 읽으려면 많은 드라이브 헤드가 필요합니다. 이 1MB의 데이터가 8개의 LUN에 스트라이핑된 경우 운영 체제에서는 8개의 128K의 읽기 작업을 병렬로 실행하고 1MB 전송을 완료하는 데 필요한 시간을 줄일 수 있습니다.

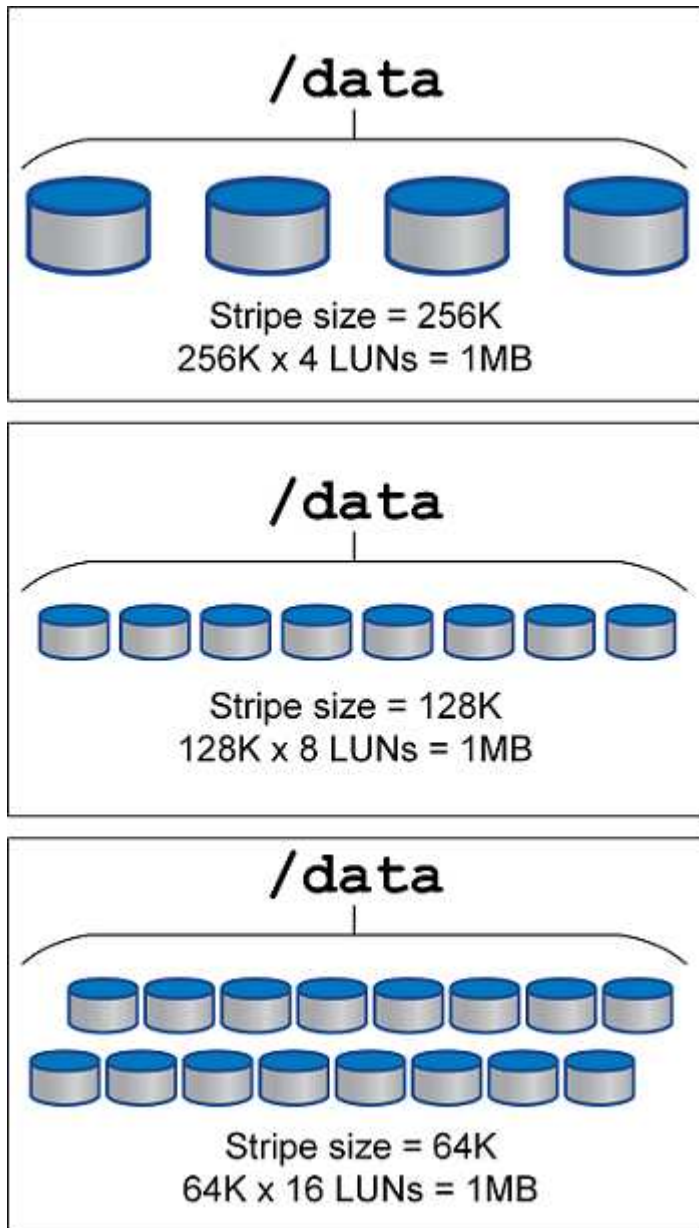
회전식 드라이브를 사용한 스트라이핑은 I/O 패턴을 사전에 알고 있어야 하므로 더 어려웠습니다. 스트라이핑이 실제 I/O 패턴에 맞게 올바르게 조정되지 않은 경우 스트라이핑된 구성이 성능을 저하시킬 수 있습니다. Oracle 데이터베이스, 특히 All-Flash 구성을 사용하면 스트라이핑이 훨씬 쉽게 구성되고 성능이 크게 향상된다는 사실이 입증되었습니다.

기본적으로 Oracle ASM 스트라이프와 같은 논리적 볼륨 관리자는 있지만 기본 OS LVM은 그렇지 않습니다. 이 중 일부는 여러 LUN을 연결된 장치로 연결하므로 하나의 LUN 디바이스와 하나의 LUN 디바이스에 데이터 파일이 존재합니다. 이로 인해 핫스팟이 발생합니다. 다른 LVM 구현은 기본적으로 분산 익스텐트로 설정됩니다. 이는 스트라이핑과 비슷하지만 더 거칠습니다. 볼륨 그룹의 LUN은 익스텐트라고 하는 큰 조각으로 분할되며 일반적으로 메가바이트 단위로 측정되며 논리적 볼륨은 이러한 익스텐트에 분산됩니다. 그 결과 파일에 대한 랜덤 I/O가 LUN 전체에 분산되어야 하지만 순차적 I/O 작업의 효율성이 최대한 높지는 않습니다.

높은 성능을 필요로 하는 애플리케이션 I/O는 거의 항상 (a) 기본 블록 크기 단위 또는 (b) 1MB입니다.

스트라이핑 구성의 기본적인 목표는 단일 파일 I/O를 단일 유닛으로 수행하고, 1MB 크기여야 하는 다중 블록 I/O를 스트라이핑 볼륨의 모든 LUN에 걸쳐 균등하게 병렬 처리할 수 있도록 지원하는 것입니다. 즉, 스트라이프 크기가 데이터베이스 블록 크기보다 작아서는 안 되며 스트라이프 크기를 LUN 수를 곱한 크기가 1MB여야 합니다.

다음 그림에서는 스트라이프 크기 및 폭 조정에 사용할 수 있는 세 가지 옵션을 보여 줍니다. 위에서 설명한 대로 성능 요구 사항을 충족하기 위해 LUN 수를 선택하지만 모든 경우에 단일 스트라이프의 총 데이터는 1MB입니다.



NFS 를 참조하십시오

Oracle 데이터베이스에 대한 NFS 구성

NetApp은 30년 이상 엔터프라이즈급 NFS 스토리지를 제공하고 있으며 단순성 덕분에 클라우드 기반 인프라로의 전환과 함께 사용 사례가 증가하고 있습니다.

NFS 프로토콜에는 다양한 요구사항이 있는 여러 버전이 포함되어 있습니다. ONTAP를 사용한 NFS 구성에 대한 자세한 내용은 을 참조하십시오 "[ONTAP 모범 사례 기반 TR-4067 NFS](#)". 다음 섹션에서는 보다 중요한 요구 사항 및 일반적인 사용자 오류에 대해 설명합니다.

NFS 버전

운영 체제 NFS 클라이언트는 NetApp에서 지원되어야 합니다.

- NFSv3은 NFSv3 표준 OS에서 지원됩니다.
- NFSv3은 Oracle dNFS 클라이언트에서 지원됩니다.
- NFSv4는 NFSv4 표준을 따르는 모든 OS에서 지원됩니다.
- NFSv4.1 및 NFSv4.2에는 특정 OS 지원이 필요합니다. 을 참조하십시오 "[NetApp IMT](#)" 지원되는 운영 체제.
- NFSv4.1에 대해 Oracle dNFS를 지원하려면 Oracle 12.2.0.2 이상이 필요합니다.



를 클릭합니다 "[NetApp 지원 매트릭스](#)" NFSv3 및 NFSv4의 경우 특정 운영 체제가 포함되지 않습니다. RFC를 준수하는 모든 OS는 일반적으로 지원됩니다. IMT에서 NFSv3 또는 NFSv4 지원을 찾을 때 일치하는 항목이 표시되지 않으므로 특정 OS를 선택하지 마십시오. 모든 OS는 일반 정책에 의해 암시적으로 지원됩니다.

Linux NFSv3 TCP 슬롯 테이블

TCP 슬롯 테이블은 호스트 버스 어댑터(HBA) 큐 길이(queue depth)와 동등한 NFSv3의 기능입니다. 이들 테이블은 한 번에 수행될 수 있는 최대 NFS 작업의 수를 제어합니다. 기본값은 보통 16이며 최적의 성능을 발휘하기에 너무 낮습니다. 최신 Linux 커널에서는 반대의 문제가 발생하는데, 요청을 통해 NFS 서버를 포화시키는 수준까지 TCP 슬롯 테이블의 한계를 자동으로 높일 수 있습니다.

최적의 성능을 얻으면서 성능 문제를 방지하려면 TCP 슬롯 테이블을 제어하는 커널 매개 변수를 조정하십시오.

를 실행합니다 `sysctl -a | grep tcp.*.slot_table` 명령을 실행하고 다음 매개 변수를 관찰합니다.

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

모든 Linux 시스템에는 가 포함되어 있습니다 `sunrpc.tcp_slot_table_entries` 그러나 일부에만 가 포함됩니다 `sunrpc.tcp_max_slot_table_entries`. 둘 다 128로 설정해야 합니다.

주의

이러한 매개 변수를 설정하지 않으면 성능에 상당한 영향을 미칠 수 있습니다. 경우에 따라 Linux 운영 체제가 충분한 I/O를 실행하지 않기 때문에 성능이 제한됩니다 또는 Linux 운영 체제가 처리할 수 있는 것보다 더 많은 I/O를 발급하려고 하면 I/O 지연 시간이 늘어납니다.

ADR과 NFS

에서 데이터와 관련하여 과도한 양의 I/O가 초래하는 성능 문제를 보고하는 고객이 있습니다 ADR 위치. 일반적으로, 성능 데이터가 대량으로 누적되기 전에는 이 문제가 발생하지 않습니다. 과도한 I/O의 이유는 알려지지 않았지만 이 문제는 Oracle 프로세스가 변경을 위해 타겟 디렉토리를 반복하여 스캐닝했기 때문인 것으로 추측됩니다.

의 제거 noac 및/또는 actimeo=0 마운트 옵션을 사용하면 호스트 운영 체제 캐싱이 수행되고 스토리지 I/O 레벨이 낮아집니다.



* NetApp는 배치하지 않을 것을 권장합니다 ADR 을 사용하는 파일 시스템 데이터 noac 또는 actimeo=0 성능 문제를 일으킬 수 있기 때문입니다. 분리 ADR 필요한 경우 데이터를 다른 마운트 지점으로 이동합니다.

NFS - rootonly 및 mount-rootonly

ONTAP에는 이라는 NFS 옵션이 포함되어 있습니다 nfs-rootonly 서버가 높은 포트의 NFS 트래픽 연결을 허용할지 여부를 제어합니다. 보안 조치로서 루트 사용자만 1024 미만의 소스 포트를 사용하여 TCP/IP 연결을 열 수 있습니다. 이러한 포트는 일반적으로 사용자 프로세스가 아닌 OS 사용을 위해 예약되기 때문입니다. 이러한 제한을 통해 NFS 트래픽이 NFS 클라이언트를 에뮬레이트하는 악의적인 프로세스가 아닌 실제 운영 체제 NFS 클라이언트에서 발생하도록 할 수 있습니다. Oracle dNFS 클라이언트는 사용자 공간 드라이버이지만 프로세스는 루트로 실행되므로 일반적으로 의 값을 변경할 필요가 없습니다 nfs-rootonly. 이 연결은 로우 포트에 이루어집니다.

를 클릭합니다 mount-rootonly 옵션은 NFSv3에만 적용됩니다. 1024 이상의 포트에서 RPC 마운트 호출을 수락할지 여부를 제어합니다. dNFS를 사용하면 클라이언트가 루트로 다시 실행되므로 1024 미만의 포트를 열 수 있습니다. 이 매개 변수는 효과가 없습니다.

프로세스 NFS 버전 4.0 이상을 통해 dNFS 연결을 열면 루트로 실행되지 않으므로 1024를 초과하는 포트가 필요합니다. 를 클릭합니다 nfs-rootonly dNFS가 연결을 완료하려면 매개 변수를 disabled로 설정해야 합니다.

if(경우 nfs-rootonly 이 활성화되면 dNFS 접속을 여는 마운트 단계 동안 작동이 중단됩니다. sqlplus 출력은 다음과 유사합니다.

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

파라미터는 다음과 같이 변경할 수 있다.

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



드문 경우지만 `nfs-rootonly`와 `mount-rootonly`를 모두 `disabled`로 변경해야 할 수도 있습니다. 서버가 매우 많은 수의 TCP 연결을 관리하는 경우 1024 미만의 포트를 사용할 수 없으며 OS가 더 높은 포트를 사용하도록 강제됩니다. 연결을 완료하려면 이 두 ONTAP 매개 변수를 변경해야 합니다.

NFS 내보내기 정책: `superuser` 및 `setuid`

Oracle 바이너리가 NFS 공유에 있는 경우 내보내기 정책에 슈퍼유저 및 `setuid` 권한이 포함되어야 합니다.

사용자 홈 디렉토리와 같은 일반 파일 서비스에 사용되는 공유 NFS 내보내기는 일반적으로 루트 사용자를 스쿼시 처리합니다. 즉, 파일 시스템을 마운트한 호스트에서 루트 사용자의 요청이 더 낮은 권한을 가진 다른 사용자로 다시 매핑됩니다. 이렇게 하면 특정 서버의 루트 사용자가 공유 서버의 데이터에 액세스하지 못하도록 하여 데이터를 보호할 수 있습니다. `setuid` 비트는 공유 환경에서 보안 위험이 될 수도 있습니다. `setuid` 비트를 사용하면 명령을 호출하는 사용자와 다른 사용자로 프로세스를 실행할 수 있습니다. 예를 들어, `setuid` 비트가 있는 루트가 소유한 쉘 스크립트는 루트로 실행됩니다. 다른 사용자가 쉘 스크립트를 변경할 수 있는 경우, 루트가 아닌 사용자는 스크립트를 업데이트하여 명령을 루트로 실행할 수 있습니다.

Oracle 바이너리는 루트가 소유한 파일을 포함하며 `setuid` 비트를 사용합니다. Oracle 바이너리가 NFS 공유에 설치된 경우 내보내기 정책에 적절한 슈퍼 사용자 및 `setuid` 권한이 포함되어야 합니다. 아래 예제에서는 규칙에 두 가지가 모두 포함되어 있습니다 `allow-suid` 있습니다 `superuser` 시스템 인증을 사용하여 NFS 클라이언트에 대한 (루트) 액세스.

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

NFSv4/4.1 구성

대부분의 애플리케이션에서 NFSv3과 NFSv4 사이에는 아주 작은 차이가 있습니다. 애플리케이션 입출력은 대개 매우 간단한 입출력이며 NFSv4에서 제공되는 일부 고급 기능의 이점은 크게 활용되지 않습니다. 더 높은 버전의 NFS는 데이터베이스 스토리지의 관점에서 "업그레이드"로 간주해서는 안 되며, 추가 기능이 포함된 NFS 버전으로 간주해야 합니다. 예를 들어 Kerberos 개인 정보 보호 모드(krb5p)의 엔드 투 엔드 보안이 필요한 경우 NFSv4가 필요합니다.



* NetApp는 NFSv4 기능이 필요한 경우 NFSv4.1을 사용할 것을 권장합니다. NFSv4.1에서는 NFSv4 프로토콜의 기능이 개선되어 특정 옛지 경우 복원력을 향상할 수 있습니다.

NFSv4로 전환하는 것은 마운트 옵션을 단순히 `vers=3`에서 `vers=4.1`로 변경하는 것보다 더 복잡합니다. 운영 체제 구성에 대한 지침을 포함하여 ONTAP를 사용한 NFSv4 구성에 대한 자세한 설명은 을 참조하십시오 ["ONTAP 모범 사례 기반 TR-4067 NFS"](#). 이 TR의 다음 섹션에서는 NFSv4 사용을 위한 몇 가지 기본 요구 사항에 대해 설명합니다.

NFSv4 도메인입니다

NFSv4/4.1 구성에 대한 자세한 설명은 이 문서의 범위를 벗어나지만 도메인 매핑이 일치하지 않는 문제가 흔히 발생합니다. `sysadmin` 관점에서 NFS 파일 시스템은 정상적으로 작동하는 것처럼 보이지만 애플리케이션이 특정 파일에 대한 권한 및/또는 `setuid`에 대한 오류를 보고합니다. 경우에 따라 관리자는 응용 프로그램 바이너리의 사용

권한이 손상되었다는 잘못된 결론을 내리고 실제 문제가 도메인 이름일 때 chown 또는 chmod 명령을 실행했습니다.

NFSv4 도메인 이름은 ONTAP SVM에 설정됩니다.

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

호스트의 NFSv4 도메인 이름은 에 설정되어 있습니다 /etc/idmap.cfg

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

도메인 이름이 일치해야 합니다. 그렇지 않은 경우 에 다음과 유사한 매핑 오류가 나타납니다 /var/log/messages:

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

Oracle 데이터베이스 바이너리와 같은 애플리케이션 바이너리에는 setuid 비트가 있는 루트가 소유한 파일이 포함됩니다. 즉, NFSv4 도메인 이름이 일치하지 않으면 Oracle 시작 시 오류가 발생하고 라는 파일의 소유권이나 사용 권한에 대한 경고가 발생합니다 oradism`에 있습니다 ` \$ORACLE_HOME/bin 디렉토리. 다음과 같이 나타납니다.

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

이 파일의 소유권이 아무도 없는 경우 NFSv4 도메인 매핑 문제가 있을 수 있습니다.

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

이 문제를 해결하려면 을 선택합니다 /etc/idmap.cfg ONTAP에서 v4-id-domain 설정을 기준으로 한 파일로, 일관성이 있는지 확인합니다. 그렇지 않은 경우 필요한 변경 작업을 수행하고 를 실행합니다 nfsidmap -c`를 클릭하고 변경 사항이 전파될 때까지 잠시 기다립니다. 그런 다음 파일 소유권이 루트로 올바르게 인식되어야 합니다. 사용자가 실행을 시도한 경우 `chown root NFS 도메인 구성이 수정되기 전에 이 파일에서 를 실행해야 할 수도 있습니다 chown root 다시?

Oracle directNFS를 참조하십시오

Oracle 데이터베이스는 두 가지 방법으로 NFS를 사용할 수 있습니다.

먼저 운영 체제의 일부인 네이티브 NFS 클라이언트를 사용하여 마운트된 파일 시스템을 사용할 수 있습니다. 이를 커널 NFS 또는 kNFS라고도 합니다. NFS 파일 시스템은 다른 애플리케이션이 NFS 파일 시스템을 사용하는 것과 정확히 동일한 방식으로 Oracle 데이터베이스에 의해 마운트되고 사용됩니다.

두 번째 방법은 Oracle Direct NFS(dNFS)입니다. Oracle 데이터베이스 소프트웨어 내에서 NFS 표준을 구현한 것입니다. Oracle 데이터베이스가 DBA에 의해 구성되거나 관리되는 방식은 변경되지 않습니다. 스토리지 시스템 자체에 올바른 설정이 있는 한 dNFS의 사용은 DBA 팀과 최종 사용자에게 투명해야 합니다.

dNFS 기능이 활성화된 데이터베이스에는 여전히 일반적인 NFS 파일 시스템이 마운트되어 있습니다. 데이터베이스가 열리면 Oracle 데이터베이스가 일련의 TCP/IP 세션을 열고 NFS 작업을 직접 수행합니다.

직접 NFS

Oracle Direct NFS의 기본 가치는 호스트 NFS 클라이언트를 우회하고 NFS 파일 작업을 NFS 서버에서 직접 수행하는 것입니다. 이를 활성화하려면 ODM(Oracle Disk Manager) 라이브러리를 변경하기만 하면 됩니다. 이 프로세스 관련 지침은 Oracle 설명서에 제공되어 있습니다.

dNFS를 사용하면 입출력이 가능한 가장 효율적인 방식으로 수행되기 때문에 입출력 성능이 크게 향상되고 호스트와 스토리지 시스템의 부하가 감소합니다.

또한 Oracle dNFS에는 네트워크 인터페이스 다중 경로 및 내결함성을 위한 * 옵션 * 이 포함되어 있습니다. 예를 들어, 2개의 10Gb 인터페이스를 서로 바인딩하여 20Gb 대역폭을 제공할 수 있습니다. 한 인터페이스에 장애가 발생하면 다른 인터페이스에서 I/O가 재시도됩니다. 전반적인 작동 방식은 FC 다중 경로와 매우 비슷합니다. 다중 경로는 1Gb 이더넷이 가장 일반적인 표준이었던 수년 전에 일반적이었습니다. 대부분의 Oracle 워크로드에는 10Gb NIC로 충분하지만 더 많은 작업이 필요한 경우 10Gb NIC를 연결할 수 있습니다.

dNFS를 사용할 때는 Oracle Doc 1495104.1에 설명된 모든 패치를 설치하는 것이 중요합니다. 패치를 설치할 수 없는 경우 환경을 평가하여 문서에 설명된 버그가 문제를 유발하지 않는지 확인해야 합니다. 필요한 패치를 설치할 수 없어 dNFS를 사용할 수 없는 경우도 있습니다.

DNS, DDNS, NIS 또는 기타 방법을 포함하여 어떤 유형의 라운드 로빈 이름 확인과도 dNFS를 함께 사용하지 마십시오. ONTAP에서 사용할 수 있는 DNS 로드 밸런싱 기능도 여기에 포함됩니다. dNFS를 사용하는 Oracle 데이터베이스가 호스트 이름을 IP 주소로 확인하면 이후 조회에서 호스트 이름을 변경해서는 안 됩니다. 변경할 경우 Oracle 데이터베이스 충돌과 잠재적 데이터 손상을 야기할 수 있습니다.

Direct NFS 및 호스트 파일 시스템 액세스

dNFS를 사용하면 호스트에 마운트된 가시적인 파일 시스템을 사용하는 애플리케이션이나 사용자 작업에 문제가 발생할 수 있습니다. dNFS 클라이언트가 호스트 OS에서 대역 외 파일 시스템에 액세스하기 때문입니다. dNFS 클라이언트는 운영 체제에 대한 지식 없이도 파일을 생성, 삭제 및 수정할 수 있습니다.

단일 인스턴스 데이터베이스의 마운트 옵션을 사용할 때 이들 옵션은 파일과 디렉토리 특성의 캐싱을 활성화할 수 있으며 이에 따라 디렉토리의 콘텐츠가 캐싱됩니다. 따라서 dNFS는 파일을 생성할 수 있으며 OS가 디렉토리 콘텐츠를 다시 읽고 파일이 사용자에게 표시되기 전에 짧은 지연이 발생합니다. 일반적으로는 이것이 문제가 되지 않지만 드물게 SAP BR * Tools 같은 유틸리티에 문제가 발생할 수 있습니다. 이런 일이 발생하면 Oracle RAC 관련 권장사항을 사용하기 위한 마운트 옵션을 변경하여 문제를 해결하십시오. 이렇게 변경하면 모든 호스트 캐싱이 비활성화됩니다.

(a) dNFS가 사용되고 (b) 파일 가시성의 지연으로 인해 문제가 발생하는 경우에만 마운트 옵션을 변경하십시오. dNFS를 사용하지 않는 경우 단일 인스턴스 데이터베이스에서 Oracle RAC 마운트 옵션을 사용하면 성능이

저하됩니다.



에 대한 참고 사항을 참조하십시오 nosharecache 인치 "Linux NFS 마운트 옵션" 비정상적인 결과를 초래할 수 있는 Linux 관련 dNFS 문제의 경우

Oracle 데이터베이스 및 NFS 임대 및 잠금

NFSv3은 상태를 저장하지 않습니다. 이는 사실상 NFS 서버(ONTAP)가 어떤 파일 시스템이 마운트되었는지, 누가 어떤 잠금이 설정되어 있는지 추적하지 않는다는 것을 의미합니다.

ONTAP에는 마운트 시도를 기록하는 몇 가지 기능이 있으므로 어떤 클라이언트가 데이터에 액세스할 수 있는지 알 수 있고, 권고 잠금이 있을 수 있지만 이 정보가 100% 완전하지는 않을 수 있습니다. NFS 클라이언트 상태 추적은 NFSv3 표준의 일부가 아니므로 완료할 수 없습니다.

NFSv4 상태

이와 반대로 NFSv4는 상태 저장 방식입니다. NFSv4 서버는 어떤 클라이언트가 어떤 파일 시스템을 사용하고 있는지, 어떤 파일이 있는지, 어떤 파일 및/또는 파일 영역이 잠겨 있는지 등을 추적합니다 즉, 상태 데이터를 최신 상태로 유지하려면 NFSv4 서버 간에 정기적인 통신이 필요합니다.

NFS 서버에서 관리하는 가장 중요한 상태는 NFSv4 잠금 및 NFSv4 리스 상태이며, 서로 매우 밀접하게 연관되어 있습니다. 당신은 각각의 작동 방식 및 서로 어떻게 관련되는지 이해해야 합니다.

NFSv4 잠금

NFSv3에서는 잠금은 권장됩니다. NFS 클라이언트는 "잠긴" 파일을 계속 수정하거나 삭제할 수 있습니다. NFSv3 잠금은 단독으로 만료되지 않고 제거되어야 합니다. 이로 인해 문제가 발생합니다. 예를 들어, NFSv3 잠금을 생성하는 클러스터 애플리케이션이 있는데 노드 중 하나에 장애가 발생하면 어떻게 해야 하나? 나머지 노드에서 애플리케이션을 코딩하여 잠금을 제거할 수 있지만 안전한지 어떻게 알 수 있습니까? "장애가 발생한" 노드가 작동 중이지만 나머지 클러스터와 통신하지 않는 것 같습니까?

NFSv4에서는 잠금 기간이 제한됩니다. 잠금을 보유한 클라이언트가 NFSv4 서버에 계속 체크인하는 한 다른 클라이언트는 이러한 잠금을 가져올 수 없습니다. 클라이언트가 NFSv4를 통해 체크인하지 못하면 서버에서 잠금이 취소되고 다른 클라이언트가 잠금을 요청하고 받을 수 있습니다.

NFSv4 임대

NFSv4 잠금은 NFSv4 임대와 연결됩니다. NFSv4 클라이언트가 NFSv4 서버와 연결을 설정하면 임대됩니다. 클라이언트가 잠금을 얻는 경우(여러 유형의 잠금이 있음) 잠금이 임대와 연결됩니다.

이 임대에 정의된 시간 초과가 있습니다. 기본적으로 ONTAP는 시간 초과 값을 30초로 설정합니다.

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

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

즉, NFSv4 클라이언트는 30초마다 NFSv4 서버에 체크인해야 리스를 갱신할 수 있습니다.

리스는 모든 활동에 의해 자동으로 갱신되므로 클라이언트가 작업을 수행하는 경우 추가 작업을 수행할 필요가 없습니다. 응용 프로그램이 조용해지고 실제 작업을 수행하지 않는 경우 대신 연결 유지 작업(시퀀스라고 함)을 수행해야 합니다. "아직 여기 있어요. 리스를 새로 고치세요."라고 말하는 것입니다.

***Question:** What happens if you lose network connectivity for 31 seconds?
NFSv3은 상태를 저장하지 않습니다. 클라이언트로부터 통신을 기대하지 않습니다. NFSv4는 상태 저장이며 임대 기간이 경과하면 임대가 만료되고 잠금이 해제되고 잠긴 파일을 다른 클라이언트에서 사용할 수 있게 됩니다.

NFSv3을 사용하면 네트워크 케이블을 이동할 수 있고, 네트워크 스위치를 재부팅하고, 구성을 변경하며, 나쁜 부분이 전혀 발생하지 않도록 매우 확신할 수 있습니다. 일반적으로 응용 프로그램은 네트워크 연결이 다시 작동할 때까지 인내심을 가지고 기다립니다.

NFSv4를 사용하면 30초 안에 작업을 완료할 수 있습니다(ONTAP 내에서 이 매개 변수의 값을 늘리지 않는 한). 이 값을 초과하면 리스가 시간 초과됩니다. 일반적으로 이 경우 응용 프로그램이 충돌합니다.

예를 들어, Oracle 데이터베이스가 있는데 네트워크 연결 손실("네트워크 파티션"이라고도 함)이 임대 시간 초과를 초과하면 데이터베이스가 충돌합니다.

이 경우 Oracle 경고 로그에서 발생하는 작업의 예는 다음과 같습니다.

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

syslogs를 보면 다음과 같은 몇 가지 오류가 표시됩니다.

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

로그 메시지는 일반적으로 응용 프로그램 정지 이외의 문제의 첫 번째 징후입니다. 일반적으로 프로세스와 운영 체제

자체가 NFS 파일 시스템에 대한 액세스 시도를 차단하기 때문에 네트워크가 중단되는 동안에는 아무 것도 표시되지 않습니다.

네트워크가 다시 작동하면 오류가 나타납니다. 위의 예에서 연결이 다시 설정되면 OS가 잠금을 다시 가져오려고 시도했지만 너무 늦었습니다. 임대가 만료되었고 잠금이 제거되었습니다. 이로 인해 오류가 Oracle 계층까지 전파되고 경고 로그에 메시지가 표시됩니다. 데이터베이스의 버전 및 구성에 따라 이러한 패턴의 변형이 표시될 수 있습니다.

요약하면 NFSv3은 네트워크 중단은 허용하지만 NFSv4는 더 민감하며 정의된 임대 기간을 사용합니다.

30초 시간 초과가 허용되지 않는 경우에는 어떻게 해야 합니까? 스위치가 재부팅되거나 케이블이 재배치되어 간헐적인 네트워크 중단이 발생하는 동적으로 변화하는 네트워크를 관리하는 경우 어떻게 해야 합니까? 임대 기간을 연장하도록 선택할 수 있지만, 이 작업을 수행하려면 NFSv4 유예 기간에 대한 설명이 필요합니다.

NFSv4 유예 기간

NFSv3 서버를 재부팅하면 거의 즉시 IO를 처리할 수 있습니다. 그것은 고객에 대한 어떤 종류의 상태를 유지하지 않았다. 그 결과 ONTAP 테이크오버 작업이 거의 즉각적으로 발생하는 것으로 보이는 경우가 많습니다. 컨트롤러가 데이터 제공을 시작할 준비가 되면 토폴로지 변경을 알리는 ARP를 네트워크에 보냅니다. 일반적으로 클라이언트는 이를 거의 즉각적으로 감지하여 데이터가 다시 흐릅니다.

하지만 NFSv4는 잠시 일시 중지됩니다. NFSv4의 작동 방식 중 일부에 불과합니다.

NFSv4 서버는 임대, 잠금 및 누가 어떤 데이터를 사용하는지 추적해야 합니다. NFS 서버가 패닉 및 재부팅되거나 잠시 전원이 꺼지거나 유지 보수 작업 중에 다시 시작되면 임대/잠금이 발생하고 다른 클라이언트 정보가 손실됩니다. 서버에서 작업을 재개하기 전에 어떤 클라이언트가 어떤 데이터를 사용 중인지 파악해야 합니다. 유예 기간이 시작되는 곳입니다.

NFSv4 서버의 전원을 갑자기 껐다 켜는 경우 다시 백업되면 입출력을 재개하려는 클라이언트는 기본적으로 "리스/잠금 정보가 손실되었습니다. 잠금 장치를 다시 등록하시겠습니까?" 이것이 유예 기간의 시작입니다. ONTAP에서는 기본적으로 45초로 설정됩니다.

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

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

그 결과, 재시작 후 모든 클라이언트가 임대를 재확보하고 잠금을 재설정하는 동안 컨트롤러는 IO를 일시 중지합니다. 유예 기간이 종료되면 서버가 입출력 작업을 재개합니다.

리스 시간 초과 대 유예 기간

유예 기간 및 임대 기간이 연결되었습니다. 위에서 언급한 것처럼 기본 임대 시간 초과는 30초입니다. 즉, NFSv4 클라이언트는 30초마다 서버에 확인해야 합니다. 그렇지 않으면 리스와 잠금이 손실됩니다. NFS 서버가 임대/잠금 데이터를 재구축할 수 있는 유예 기간이 있으며 기본값은 45초입니다. ONTAP의 유예 기간은 임대 기간보다 15초 더 길어야 합니다. 이를 통해 최소 30초마다 리스를 갱신하도록 설계된 NFS 클라이언트 환경에서는 재시작 후 서버를 통해 확인할 수 있습니다. 45초의 유예 기간은 최소 30초마다 리스를 갱신할 모든 고객이 확실히 그렇게 할 기회를 갖도록 합니다.

30초의 시간 초과가 허용되지 않는 경우 임대 기간을 연장할 수 있습니다. 60초의 네트워크 중단을 견디기 위해 리스 시간 제한을 60초로 늘리려면 유예 기간을 최소 75초로 늘려야 합니다. ONTAP에서는 임대 기간보다 15초 더 높게

설정해야 합니다. 이는 컨트롤러 페일오버 중 IO 일시 중단이 더 길다는 것을 의미합니다.

이것은 일반적으로 문제가 되지 않습니다. 일반 사용자는 연간 1~2회 ONTAP 컨트롤러를 업데이트하며, 하드웨어 장애로 인한 계획되지 않은 페일오버는 매우 드물게 발생합니다. 또한 60초 네트워크 중단이 발생할 가능성이 있는 네트워크가 있고 임대 시간 초과가 60초로 필요한 경우 드물게 발생하는 스토리지 시스템 장애 조치를 거부하여 75초 동안 일시 중지되지 않을 수 있습니다. 이미 60초 이상 일시 중지된 네트워크가 있음을 확인했습니다.

Oracle 데이터베이스를 통한 NFS 캐싱

다음과 같은 마운트 옵션이 있으면 호스트 캐싱이 비활성화됩니다.

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

이러한 설정은 소프트웨어 설치, 패치 적용 및 백업/복원 작업 속도에 심각한 악영향을 미칠 수 있습니다. 경우에 따라, 특히 클러스터형 애플리케이션의 경우 클러스터의 모든 노드에 캐시 정합성을 보장해야 하기 때문에 이러한 옵션이 불가피하게 필요하게 되었습니다. 고객이 이러한 매개 변수를 잘못 사용하면 불필요한 성능 저하가 발생하게 됩니다.

많은 고객들이 애플리케이션 바이너리의 설치 또는 패칭 중에 이러한 마운트 옵션을 임시로 제거합니다. 설치 또는 패칭 프로세스 중에 타겟 디렉토리를 적극적으로 사용하는 여타 프로세스가 없음이 확인되면 제거를 안전하게 수행할 수 있습니다.

Oracle 데이터베이스의 NFS 전송 크기

기본적으로 ONTAP에서는 NFS I/O 크기를 64K로 제한합니다.

대부분의 애플리케이션 및 데이터베이스에서 랜덤 I/O는 최대 64K 미만으로 훨씬 작은 블록 크기를 사용합니다. 대규모 블록 I/O는 일반적으로 병렬화되므로 최대 64K 역시 최대 대역폭을 확보하는 데 제한이 없습니다.

일부 워크로드는 최대 64K로 인해 제한이 발생합니다. 특히, 데이터베이스가 적은 수의 입출력을 수행할 수 있는 경우 백업 또는 복구 작업 또는 데이터베이스 전체 테이블 스캔과 같은 단일 스레드 작업이 보다 빠르고 효율적으로 실행됩니다. ONTAP의 최적의 I/O 처리 크기는 256K입니다.

특정 ONTAP SVM의 최대 전송 크기를 다음과 같이 변경할 수 있습니다.

```
Cluster01::> set advanced
Warning: These advanced commands are potentially dangerous; use them only
when directed to do so by NetApp personnel.
Do you want to continue? {y|n}: y
Cluster01::*> nfs server modify -vserver vserver1 -tcp-max-xfer-size
262144
Cluster01::*>
```

주의

ONTAP에서 허용되는 최대 전송 크기를 현재 마운트된 NFS 파일 시스템의 rsize/wsize 값보다 작게 줄이지 마십시오. 이로 인해 일부 운영 체제에서 중단되거나 심지어 데이터 손상이 발생할 수 있습니다. 예를 들어, NFS 클라이언트가 현재 rsize/wsize 65536으로 설정되어 있는 경우 클라이언트 자체가 제한되기 때문에 ONTAP 최대 전송 크기를 65536에서 1048576 사이에서 아무 영향 없이 조정할 수 있습니다. 최대 전송 크기를 65536 미만으로 줄이면 가용성 또는 데이터가 손상될 수 있습니다.

Oracle 데이터베이스 및 NVFAIL

NVFAIL은 대규모 페일오버 시나리오 중에 무결성을 보장하는 ONTAP의 기능입니다.

데이터베이스는 대규모 내부 캐시를 유지하기 때문에 스토리지 페일오버 중에 손상에 취약합니다. 재앙적인 이벤트에서 ONTAP 페일오버를 강제 적용하거나 MetroCluster 스위치오버를 강제 적용해야 하는 경우, 전체 구성 상태와 관계없이 이전에 승인된 변경 사항을 효과적으로 삭제할 수 있기 때문입니다. 스토리지 어레이의 콘텐츠가 시간 뒤로 이동하며, 데이터베이스 캐시의 상태는 디스크에 있는 데이터의 상태를 더 이상 반영하지 않습니다. 이러한 불일치로 인해 데이터가 손상됩니다.

캐싱은 애플리케이션이나 서버 계층에서 발생할 수 있습니다. 예를 들어, 운영 사이트와 원격 사이트 모두에서 서버가 활성 상태인 Oracle RAC(Real Application Cluster) 구성은 Oracle SGA 내의 데이터를 캐시합니다. 강제 전환 작업으로 인해 데이터가 손실되면 SGA에 저장된 블록이 디스크의 블록과 일치하지 않을 수도 있기 때문에 데이터베이스가 손상될 위험에 처하게 됩니다.

운영 체제 파일 시스템 계층에서의 캐싱 사용은 이보다 덜 명확합니다. 마운트된 NFS 파일 시스템의 블록이 운영 체제에 캐싱될 수 있습니다. 또는 운영 사이트에 있는 LUN 기반의 클러스터된 파일 시스템을 원격 사이트의 서버에 마운트하여 데이터를 다시 캐시할 수 있습니다. 이러한 상황에서 NVRAM 장애 또는 강제 적용 테이크오버 또는 강제 적용 스위치오버로 인해 파일 시스템이 손상될 수 있습니다.

ONTAP는 NVFAIL 및 관련 설정을 통해 이 시나리오로부터 데이터베이스와 운영 체제를 보호합니다.

ASM 재생 유틸리티 및 ONTAP 제로 블록 감지

ONTAP은 인라인 압축이 활성화되었을 때 파일 또는 LUN에 기록되는 제로 블록을 효율적으로 제거합니다. Oracle ASRU(ASM Reclamation Utility)와 같은 유틸리티는 사용되지 않는 ASM 익스텐트에 0을 쓰는 방식으로 작동합니다.

따라서 DBA는 데이터가 삭제된 후 스토리지 시스템의 공간을 재확보할 수 있습니다. ONTAP는 0을 가로채고 LUN의 공간을 재할당합니다. 스토리지 시스템 내에 데이터가 기록되지 않으므로 재확보 프로세스가 매우 빠릅니다.

데이터베이스 관점에서 보면 ASM 디스크 그룹은 0을 포함하고 LUN의 해당 영역을 읽으면 스트림이 0이 되지만 ONTAP는 드라이브에 0을 저장하지 않습니다. 그 대신, 내부적으로 LUN의 제로화된 영역을 모든 데이터가 비어 있는 상태로 표시하는 간단한 메타데이터 변경이 이루어집니다.

마찬가지로 0의 블록이 실제로 스토리지 시스템 내에서 쓰기로 처리되지 않기 때문에 제로화된 데이터를 포함하는 성능 테스트는 유효하지 않습니다.



ASRU를 사용하는 경우 모든 Oracle 권장 패치가 설치되어 있는지 확인합니다.

Oracle 데이터베이스 가상화

VMware, Oracle OLVM 또는 KVM을 통한 데이터베이스 가상화는 미션 크리티컬한 데이터베이스도 가상화를 선택한 NetApp 고객에게 점점 더 많이 사용되고 있습니다.

지원 가능성

가상화를 위한 Oracle 지원 정책, 특히 VMware 제품 관련 정책에 관해 많은 오해가 있습니다. Oracle Induight가 가상화를 지원하지 않는다는 말은 흔히 들립니다. 이는 틀린 생각이고, 가상화의 이점을 얻을 수 있는 기회를 놓치게 되는 결과를 낳습니다. Oracle Doc ID 249212.1은 실제 요구사항을 설명하며 고객이 관심사로 간주하는 경우는 거의 없습니다.

가상화된 서버에서 문제가 발생했고 이전에 Oracle Support에서 이 문제를 알려지지 않은 경우 고객은 물리적 하드웨어에서 문제를 재현하도록 요청받을 수 있습니다. 제품의 최첨단 버전을 실행하는 Oracle 고객은 지원 가능성 문제 때문에 가상화를 사용하려 하지 않을 수 있지만 일반적으로 제공되는 Oracle 제품 버전을 사용하는 가상화 고객 경우에는 현실이 아니었습니다.

스토리지 프레젠테이션

데이터베이스 가상화를 고려하는 고객은 비즈니스 요구사항을 근거로 스토리지 관련 결정을 내려야 합니다. 이는 모든 IT 결정에 있어 일반적으로 적용되는 사항이지만 요구 사항의 규모와 범위가 상당히 다르기 때문에 데이터베이스 프로젝트에서 특히 중요합니다.

스토리지 프레젠테이션에는 다음과 같은 세 가지 기본 옵션이 있습니다.

- 하이퍼바이저 데이터 저장소의 가상화된 LUN
- 하이퍼바이저가 아닌 VM의 iSCSI 이니시에이터에 의해 관리되는 iSCSI LUN
- VM에 의해 마운트되는 NFS 파일 시스템(NFS 기반 데이터 저장소가 아님)
- 직접 장치 매핑 VMware RDM은 고객이 선호하지 않지만 물리적 디바이스는 KVM 및 OLVM 가상화와 유사하게 직접 매핑되는 경우가 많습니다.

성능

가상 게스트에 스토리지를 제공하는 방법은 일반적으로 성능에 영향을 미치지 않습니다. 호스트 OS, 가상화 네트워크 드라이버, 하이퍼바이저 데이터 저장소 구현은 모두 고도로 최적화되어 있으며 기본 모범 사례를 따르는 한 하이퍼바이저와 스토리지 시스템 간에 사용 가능한 FC 또는 IP 네트워크 대역폭을 모두 사용할 수 있습니다. 어떤 경우에는 다른 스토리지 프레젠테이션 방식에 비해 한 가지 스토리지 프레젠테이션 방식을 사용하면 최적의 성능을 얻는 것이 약간 더 쉬울 수 있지만 결과는 비슷해야 합니다.

관리 효율성

가상화된 게스트에 스토리지를 제공하는 방법을 결정하는 핵심 요소는 관리 능력입니다. 올바른 방법이나 잘못된 방법이 없습니다. 최상의 접근 방식은 IT 운영 요구사항, 기술 및 선호도에 따라 달라집니다.

고려해야 할 요소는 다음과 같습니다.

- * 투명성. * VM이 파일 시스템을 관리할 때 데이터베이스 관리자나 시스템 관리자가 데이터에 대한 파일 시스템의 소스를 쉽게 식별할 수 있습니다. 파일 시스템 및 LUN은 물리적 서버와 달리 액세스됩니다.
- * 정합성 보장. * VM이 파일 시스템을 소유하는 경우 하이퍼바이저 계층을 사용하거나 사용하지 않는 것은 관리

용이성에 영향을 미칩니다. 가상화 및 비가상화 환경 모두를 포함한 전체 자산에 걸쳐 프로비저닝, 모니터링, 데이터 보호 등에 같은 절차를 사용할 수 있습니다.

반면에 100% 가상화된 데이터 센터라면 앞에서 언급한 것처럼 전체 설치 공간 전반에서 데이터 저장소 기반 스토리지를 사용하는 것이 더 나을 것입니다. 일관성 - 프로비저닝, 보호, 조정 및 데이터 보호에 대해서도 동일한 절차를 사용할 수 있는 능력입니다.

- * 안정성 및 문제 해결. * VM이 파일 시스템을 소유할 때 VM에 전체 스토리지 스택이 있기 때문에 우수하고 안정적인 성능 및 문제 해결이 간단해집니다. 여기서 하이퍼바이저는 FC 또는 IP 프레임워크를 운반하는 역할만 합니다. 데이터 저장소가 구성에 포함된 경우 시간 초과, 매개 변수, 로그 파일, 잠재적 버그 문제가 발생하여 구성이 복잡해집니다.
- * 이식성. * VM이 파일 시스템을 소유하면 Oracle 환경을 이동하는 프로세스가 훨씬 간단해집니다. 가상화 게스트와 비가상화 게스트 간에 파일 시스템을 쉽게 이동할 수 있습니다.
- * 공급업체 종속. * 데이터 저장소에 데이터를 배치한 후에는 다른 하이퍼바이저를 사용하거나 가상화된 환경에서 데이터를 완전히 가져오는 것이 어려워집니다.
- * Snapshot Enablement. * 상대적으로 제한된 대역폭으로 인해 가상화 환경의 기존 백업 절차가 문제가 될 수 있습니다. 예를 들어, 4포트 10GbE 트렁크는 여러 가상화 데이터베이스의 일상적인 성능 요구를 충분히 지원할 수 있지만 RMAN 또는 데이터의 전체 크기 복사본 스트리밍이 필요한 기타 백업 제품을 사용하여 백업을 수행하기에는 불충분합니다. 그 결과 점점 더 통합되는 가상화 환경에서 스토리지 스냅샷을 통해 백업을 수행해야 합니다. 따라서 백업 윈도우에서 순수하게 대역폭과 CPU 요구사항을 지원하기 위해 하이퍼바이저 구성을 오버 빌드할 필요가 없습니다.

게스트 소유 파일 시스템을 사용하면 보호가 필요한 스토리지 오브젝트를 더 쉽게 타겟팅할 수 있기 때문에 스냅샷 기반 백업과 복원을 활용하기가 어려울 수 있습니다. 그러나 데이터 저장소 및 스냅샷과 잘 통합되는 가상화 데이터 보호 제품이 점점 더 많아지고 있습니다. 가상화된 호스트에 스토리지를 제공하는 방법을 결정하기 전에 백업 전략을 완전히 고려해야 합니다.

반가상화 드라이버

최적의 성능을 얻으려면 반가상화 네트워크 드라이버를 사용해야 합니다. 데이터 저장소를 사용할 때는 반가상화 SCSI 드라이버가 필요합니다. 에뮬레이션된 드라이버가 하이퍼바이저가 물리적 하드웨어의 동작을 모방하면서 많은 CPU 시간을 사용하는 것과 대조적으로, 반가상화 장치 드라이버는 게스트가 하이퍼바이저에 더 긴밀히 통합될 수 있게 합니다.

RAM 오버 커밋

RAM 오버 커밋이란 물리적 하드웨어에 있는 것보다 더 많은 가상화 RAM을 다양한 호스트에 구성하는 것을 의미합니다. 오버 커밋은 예기치 않은 성능 문제를 초래할 수 있습니다. 데이터베이스를 가상화할 때 Oracle SGA의 기본 블록이 하이퍼바이저에 의해 스토리지로 교체되면 안 됩니다. 교체할 경우 성능 결과가 매우 불안정해집니다.

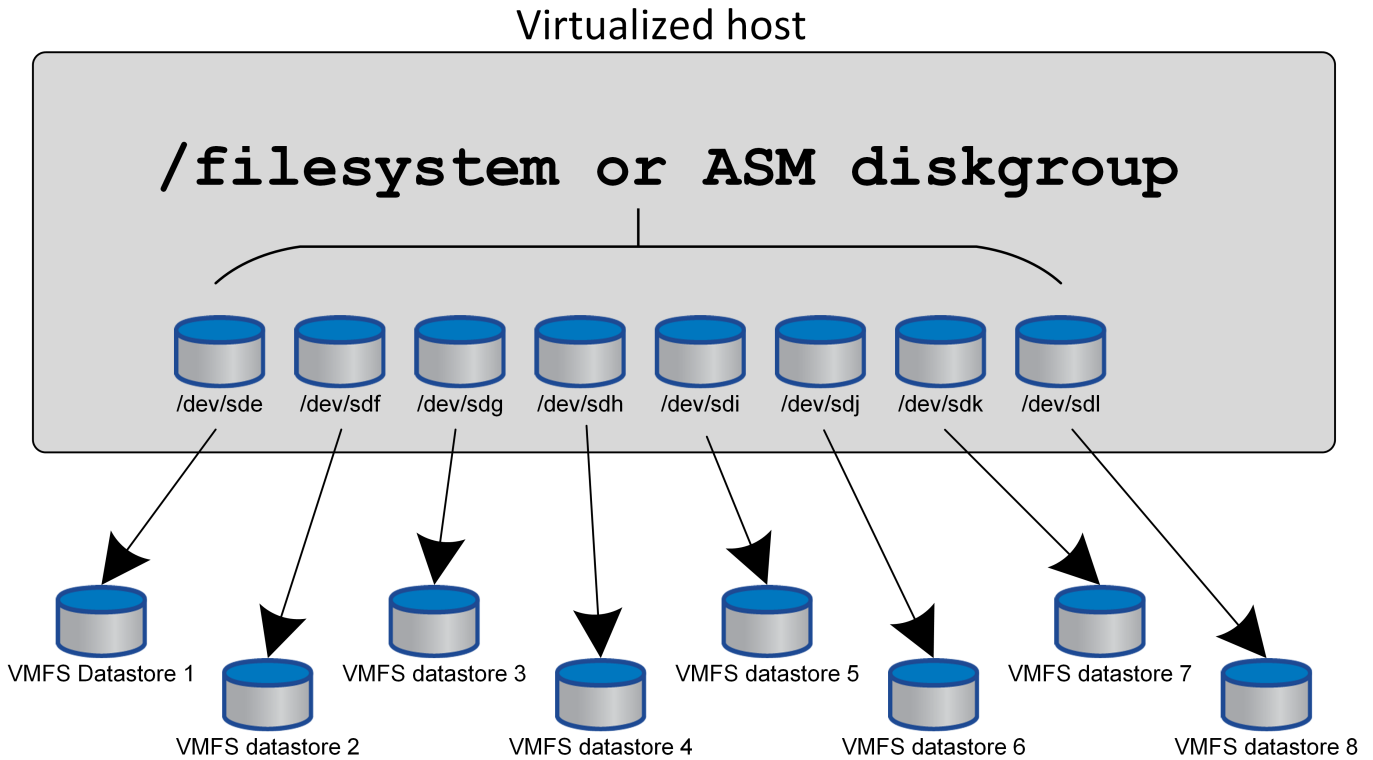
데이터 저장소 스트라이핑

데이터 저장소와 함께 데이터베이스를 사용할 때는 성능 스트라이핑과 관련하여 고려해야 할 중요한 한 가지 요소가 있습니다.

VMFS와 같은 데이터 저장소 기술은 여러 LUN을 확장할 수 있지만 스트라이핑된 디바이스는 아닙니다. LUN이 연결됩니다. 결과적으로 LUN 핫스팟이 될 수 있습니다. 예를 들어, 일반적인 Oracle 데이터베이스의 경우 8-LUN ASM 디스크 그룹이 있을 수 있습니다. 8개의 가상화된 LUN을 모두 8개의 LUN VMFS 데이터 저장소에 프로비저닝할 수 있지만 데이터가 상주할 LUN은 보장할 수 없습니다. 그 결과 VMFS 데이터 저장소 내에서 단일 LUN을 차지하는 8개의 가상화된 LUN이 모두 구성될 수 있습니다. 이로 인해 성능 병목 현상이 발생합니다.

일반적으로 스트라이핑이 필요합니다. KVM을 비롯한 일부 하이퍼바이저에서는 설명된 대로 LVM 스트라이핑을 사용하여 데이터 저장소를 구축할 수 있습니다 "여기". VMware를 사용하면 아키텍처가 조금 다르게 보입니다. 각 가상화된 LUN은 서로 다른 VMFS 데이터 저장소에 배치해야 합니다.

예를 들면 다음과 같습니다.



이 접근 방식의 주요 동인은 ONTAP가 아닙니다. 단일 VM 또는 하이퍼바이저 LUN이 병렬로 처리할 수 있는 작업 수의 내재적 제한 때문입니다. 일반적으로 단일 ONTAP LUN은 호스트가 요청할 수 있는 것보다 훨씬 더 많은 IOPS를 지원할 수 있습니다. 단일 LUN 성능 제한은 거의 보편적으로 호스트 운영 체제의 결과입니다. 그 결과 대부분의 데이터베이스에서 성능 요구를 충족하기 위해 4~8개의 LUN이 필요합니다.

VMware 아키텍처는 데이터 저장소 및/또는 LUN 경로 최대화가 이 접근 방식에서 발생하지 않도록 아키텍처를 신중하게 계획해야 합니다. 또한 모든 데이터베이스에 대해 고유한 VMFS 데이터 저장소 집합이 필요하지 않습니다. 각 호스트에 가상 LUN에서 스토리지 시스템 자체의 백엔드 LUN으로 연결되는 4-8개의 깨끗한 입출력 경로 세트가 있어야 합니다. 드문 경우지만 더 많은 데이터베이스가 진정한 성능 요구 사항에 도움이 될 수 있지만 일반적으로 전체 데이터베이스의 95%에 대해 4-8개의 LUN으로 충분합니다. 8개의 LUN이 포함된 단일 ONTAP 볼륨은 일반적인 OS/ONTAP/네트워크 구성에서 최대 250,000개의 랜덤 Oracle 블록 IOPS를 지원할 수 있습니다.

계층화

Oracle 데이터베이스 FabricPool 계층화 개요

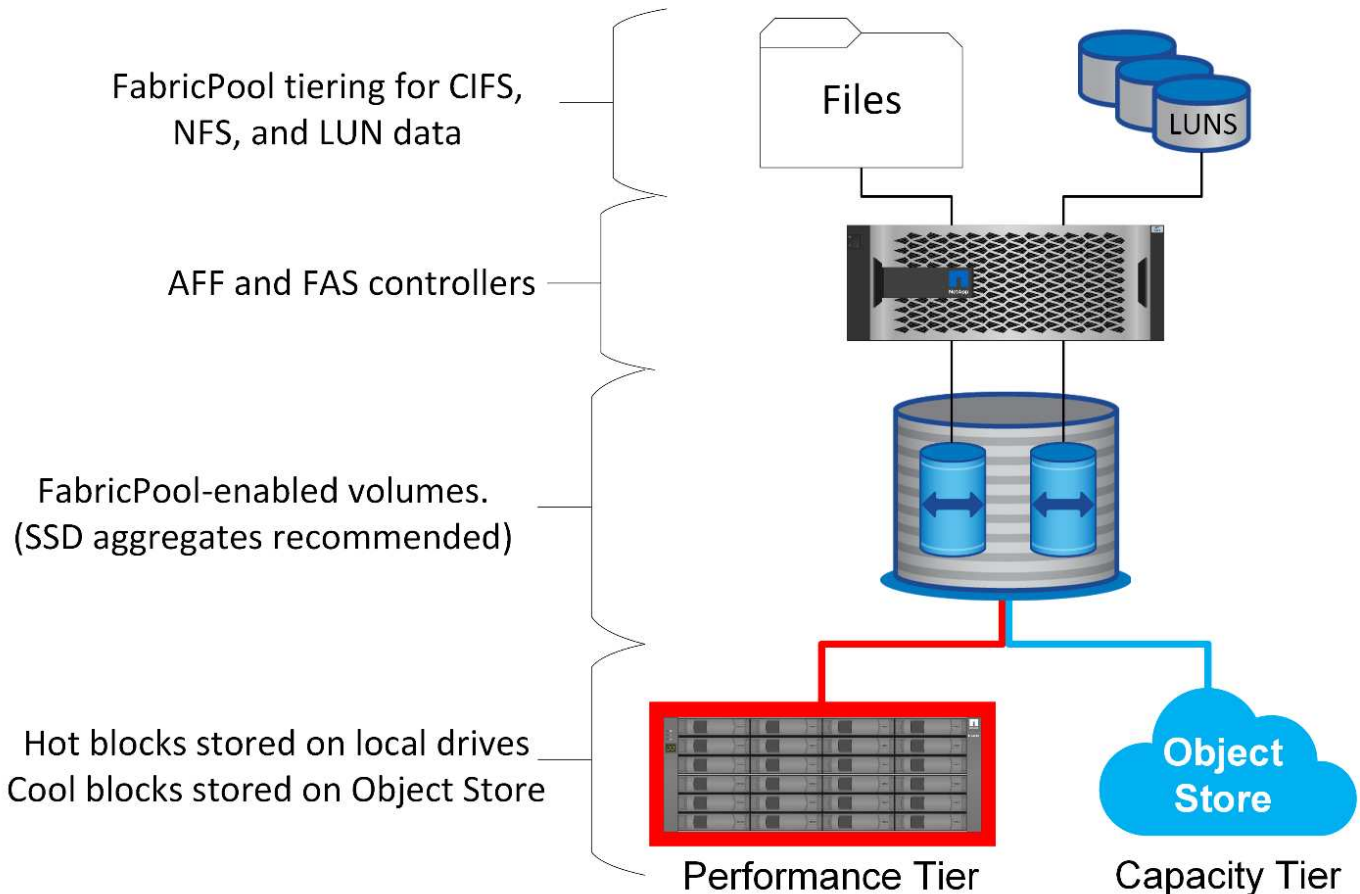
FabricPool 계층화가 Oracle 및 다른 데이터베이스에 미치는 영향을 이해하려면 낮은 수준의 FabricPool 아키텍처에 대한 이해가 필요합니다.

있습니다

FabricPool는 블록을 핫 또는 쿨 블록으로 분류하여 가장 적절한 스토리지 계층에 배치하는 계층화 기술입니다. 성능

계층은 가장 일반적으로 SSD 스토리지에 배치되며 핫 데이터 블록을 호스팅합니다. 용량 계층은 오브젝트 저장소에 있으며 쿨 데이터 블록을 호스팅합니다. 오브젝트 스토리지 지원에는 NetApp StorageGRID, ONTAP S3, Microsoft Azure Blob 스토리지, Alibaba 클라우드 오브젝트 스토리지 서비스, IBM 클라우드 오브젝트 스토리지, Google Cloud 스토리지, Amazon AWS S3가 포함됩니다.

여러 계층화 정책을 사용하여 블록의 핫 또는 쿨 분류 방식을 제어하고, 볼륨별로 정책을 설정하고 필요에 따라 변경할 수 있습니다. 성능 계층과 용량 계층 간에는 데이터 블록만 이동됩니다. LUN 및 파일 시스템 구조를 정의하는 메타데이터는 항상 성능 계층에 유지됩니다. 결과적으로 관리가 ONTAP에서 중앙 집중화됩니다. 파일 및 LUN은 다른 ONTAP 구성에 저장된 데이터와 다르지 않습니다. NetApp AFF 또는 FAS 컨트롤러는 정의된 정책을 적용하여 데이터를 적절한 계층으로 이동합니다.



오브젝트 저장소 공급자

오브젝트 스토리지 프로토콜은 다수의 데이터 오브젝트를 저장하기 위해 간단한 HTTP 또는 HTTPS 요청을 사용합니다. ONTAP에서 데이터에 액세스하는 것은 요청을 즉시 처리하는 데 좌우되므로 오브젝트 스토리지에 대한 액세스는 안정적이어야 합니다. 옵션에는 Amazon S3 Standard 및 빈도가 낮은 액세스 옵션과 Microsoft Azure Hot and Cool Blob Storage, IBM Cloud 및 Google Cloud가 있습니다. Amazon Glacier 및 Amazon Archive와 같은 아카이브 옵션은 데이터를 검색하는 데 필요한 시간이 호스트 운영 체제 및 애플리케이션의 허용 한도를 초과할 수 있기 때문에 지원되지 않습니다.

NetApp StorageGRID는 또한 지원되며 최적의 엔터프라이즈급 솔루션입니다. 확장성이 뛰어나고 매우 안전한 오브젝트 스토리지 시스템으로, FabricPool 데이터와 엔터프라이즈 애플리케이션 환경에 포함될 가능성이 더 높은 기타 오브젝트 저장소 애플리케이션에 지리적 이중화를 제공할 수 있습니다.

또한 StorageGRID은 많은 퍼블릭 클라우드 공급자가 데이터를 서비스에서 다시 읽으면서 부과하는 이그레스 비용을 방지하여 비용을 절감할 수 있습니다.

데이터 및 메타데이터

여기서 "데이터"라는 용어는 메타데이터가 아닌 실제 데이터 블록에 적용됩니다. 데이터 블록만 계층화되며 메타데이터는 성능 계층에 유지됩니다. 또한 블록의 상태는 실제 데이터 블록을 읽기만 하면 핫 또는 쿨링으로 표시됩니다. 파일의 이름, 타임스탬프 또는 소유권 메타데이터를 읽기만 해도 기본 데이터 블록의 위치에 영향을 주지 않습니다.

백업

FabricPool는 스토리지 설치 공간을 크게 줄일 수 있지만 이에 국한되지 않는 백업 솔루션입니다. NetApp WAFL 메타데이터는 항상 성능 계층에 유지됩니다. 심각한 재해로 인해 성능 계층이 폐기되는 경우 WAFL 메타데이터가 없으므로 용량 계층의 데이터를 사용하여 새로운 환경을 생성할 수 없습니다.

하지만 FabricPool는 백업 전략의 일부가 될 수 있습니다. 예를 들어, FabricPool를 NetApp SnapMirror 복제 기술로 구성할 수 있습니다. 미러의 각 절반은 오브젝트 스토리지 타겟에 자체적으로 연결될 수 있습니다. 결과는 데이터의 독립적인 복사본 두 개가 됩니다. 운영 복제본은 성능 계층의 블록과 용량 계층의 관련 블록으로 구성되며 복제본은 두 번째 성능 및 용량 블록 집합입니다.

계층화 정책

Oracle 데이터베이스 FabricPool 계층화 정책

ONTAP에서는 성능 계층의 Oracle 데이터가 용량 계층으로 재배치되는 방식을 제어하는 4가지 정책을 사용할 수 있습니다.

스냅샷 전용

를 클릭합니다 `snapshot-only tiering-policy` 액티브 파일 시스템과 공유되지 않는 블록에만 적용됩니다. 기본적으로 데이터베이스 백업의 계층화가 이루어집니다. 스냅샷이 생성되고 블록이 덮어쓰지면 블록이 계층화 대상이 되어 스냅샷 내에만 존재하는 블록이 됩니다. 이전 지연 시간 `snapshot-only` 블록은 냉각된 것으로 간주됩니다 `tiering-minimum-cooling-days` 볼륨 설정입니다. ONTAP 9.8의 범위는 2-183일입니다.

많은 데이터 세트의 변경률이 낮기 때문에 이 정책에서의 절약 효과가 최소화됩니다. 예를 들어 ONTAP에서 관찰되는 일반적인 데이터베이스의 변경률은 주당 5% 미만입니다. 데이터베이스 아카이브 로그는 광범위한 공간을 차지할 수 있지만 일반적으로 활성 파일 시스템에 계속 존재하므로 이 정책에 따라 계층화할 대상이 아닙니다.

자동

를 클릭합니다 `auto` 계층화 정책을 통해 액티브 파일 시스템 내의 블록뿐만 아니라 스냅샷별 블록까지 계층화를 확장할 수 있습니다. 블록이 냉각된 것으로 간주되기 전의 지연은 에 의해 제어됩니다 `tiering-minimum-cooling-days` 볼륨 설정입니다. ONTAP 9.8의 범위는 2-183일입니다.

이 접근 방식을 사용하면 에서 제공되지 않는 계층화 옵션을 사용할 수 있습니다 `snapshot-only` 정책. 예를 들어, 데이터 보호 정책에서는 특정 로그 파일을 90일간 보존해야 할 수 있습니다. 냉각 기간을 3일로 설정하면 3일이 지난 로그 파일이 성능 계층에서 계층화됩니다. 이렇게 하면 성능 계층에서 상당한 공간을 확보하면서 90일 분량의 데이터를 확인하고 관리할 수 있습니다.

없음

를 클릭합니다 `none` 계층화 정책을 통해 추가 블록이 스토리지 계층에서 계층화되지 않게 하고, 용량 계층에 있는 데이터는 읽을 때까지 용량 계층에 유지됩니다. 블록이 읽히면 해당 블록을 다시 가져와 성능 계층에 배치합니다.

을 사용하는 주된 이유 `none` 계층화 정책은 블록이 계층화되지 않도록 하는 것이지만 시간이 지남에 따라 정책을 변경하는 것이 유용할 수 있습니다. 예를 들어, 특정 데이터 세트는 용량 계층에 광범위하게 계층화되지만, 전체 성능 기능에 대한 예기치 않은 요구사항이 발생한다고 가정해 보겠습니다. 추가 계층화를 방지하고 입출력이 증가함에 따라 다시 읽히는 모든 블록이 성능 계층에 유지되는지 확인하도록 정책을 변경할 수 있습니다.

모두

를 클릭합니다 `all` 계층화 정책이 을 대체합니다 `backup` ONTAP 9.6 기준 정책. 를 클릭합니다 `backup` 정책은 데이터 보호 볼륨에만 적용됩니다. 즉, `SnapMirror` 또는 `NetApp SnapVault` 대상을 의미합니다. 를 클릭합니다 `all` 정책은 동일하게 작동하지만 데이터 보호 볼륨으로 제한되지 않습니다.

이 정책을 사용하면 블록이 즉시 쿨 상태로 간주되어 용량 계층에 즉시 계층화할 수 있습니다.

이 정책은 장기 백업에 특히 적합합니다. HSM(Hierarchical Storage Management)의 한 형태로도 사용할 수 있습니다. 이전에는 HSM을 사용하여 파일 자체를 파일 시스템에 표시하면서 파일의 데이터 블록을 테이프에 계층화했습니다. 를 포함하는 `FabricPool` 볼륨 `all` 정책을 사용하면 로컬 스토리지 계층에서 공간을 거의 사용하지 않으면서 표시 가능하고 관리 가능한 상태로 파일을 저장할 수 있습니다.

Oracle 데이터베이스 및 `FabricPool` 검색 정책

계층화 정책은 성능 계층에서 용량 계층으로 계층화하는 Oracle 데이터베이스 블록을 제어합니다. 검색 정책은 계층화된 블록을 읽을 때 수행되는 작업을 제어합니다.

기본값

모든 `FabricPool` 볼륨은 처음에 에서 설정됩니다 `default` 즉, 클라우드 검색 정책에 의해 동작이 제어됩니다. 정확한 동작은 사용된 계층화 정책에 따라 다릅니다.

- `auto`- 임의로 읽은 데이터만 검색합니다
- `snapshot-only`- 순차 또는 임의로 읽은 데이터를 모두 검색합니다
- `none`- 순차 또는 임의로 읽은 데이터를 모두 검색합니다
- `all`-용량 계층에서 데이터를 검색하지 마십시오

온리드

설정 `cloud-retrieval-policy` `On-read`를 선택하면 기본 동작이 재정의되므로 계층화된 데이터를 읽으면 해당 데이터가 성능 계층으로 반환됩니다.

예를 들어, 볼륨을 오랫동안 가볍게 사용한 적이 있을 수 있습니다 `auto` 계층화 정책과 대부분의 블록이 이제 계층화됩니다.

예상치 못한 비즈니스 요구 사항이 변화함에 따라 특정 보고서를 준비하기 위해 일부 데이터를 반복적으로 스캔해야 하는 경우 를 변경하는 것이 바람직할 수 있습니다 `cloud-retrieval-policy` 를 선택합니다 `on-read` 순차 및 랜덤 읽기 데이터를 모두 포함하여 읽히는 모든 데이터가 성능 계층으로 반환되도록 합니다. 이렇게 하면 볼륨에 대한 순차적 I/O 성능이 향상됩니다.

승격

상향 이동 정책의 동작은 계층화 정책에 따라 다릅니다. 계층화 정책이 인 경우 `auto` `를 선택한 다음 를 설정합니다 `cloud-retrieval-policy`to`promote` 다음 계층화 스캔 시 용량 계층의 모든 블록을 되돌립니다.

계층화 정책이 인 경우 `snapshot-only`` 그리고 반환되는 유일한 블록은 액티브 파일 시스템과 연결된 블록입니다. 일반적으로 에서 계층화된 유일한 블록은 영향을 미치지 않습니다 `snapshot-only` 정책은 스냅샷과 독립적으로 연결된 블록입니다. 액티브 파일 시스템에는 계층화된 블록이 없습니다.

하지만 볼륨의 SnapRestore 또는 스냅샷의 파일 클론 작업에 의해 볼륨의 데이터가 복원된 경우 스냅샷에만 연결되기 때문에 계층화된 일부 블록이 이제 액티브 파일 시스템에 필요할 수 있습니다. 를 일시적으로 변경하는 것이 바람직할 수 있습니다 `cloud-retrieval-policy` 정책 적용 대상 `promote` 로컬에 필요한 모든 블록을 신속하게 검색

안 함

용량 계층에서 블록을 검색하지 마십시오.

계층화 전략

Oracle 데이터베이스 전체 파일 FabricPool 계층화

FabricPool 계층화는 블록 레벨에서 작동하지만 파일 레벨 계층화를 제공하는 데 사용될 수도 있습니다.

대부분의 애플리케이션 데이터 세트는 날짜별로 구성되며 이러한 데이터가 대개 노후화되면 데이터에 액세스할 가능성이 더 적습니다. 예를 들어 은행에는 5년간의 고객 명세서가 포함된 PDF 파일 저장소가 있을 수 있지만 최근 몇 달만 활성 상태입니다. FabricPool를 사용하여 기존 데이터 파일을 용량 계층으로 재배치할 수 있습니다. 14일의 냉각 기간을 사용하면 더 최근의 14일 PDF 파일을 성능 계층에 유지할 수 있습니다. 또한 최소 14일마다 읽히는 파일은 핫 상태로 유지되므로 성능 계층에 유지됩니다.

정책

파일 기반 계층화 방식을 구현하려면 기록되고 이후에 수정되지 않는 파일이 있어야 합니다. 를 클릭합니다 `tiering-minimum-cooling-days` 필요한 파일이 성능 계층에 유지되도록 정책을 충분히 높게 설정해야 합니다. 예를 들어, 최적의 성능을 위해 최근 60일간의 데이터 세트가 필요한 경우 로 설정된 것입니다 `tiering-minimum-cooling-days` 60까지. 파일 액세스 패턴을 기준으로 비슷한 결과를 얻을 수도 있습니다. 예를 들어, 최근 90일 동안의 데이터가 필요하고 애플리케이션에서 90일 동안의 데이터에 액세스하는 경우 데이터는 성능 계층에 유지됩니다. 를 설정합니다 `tiering-minimum-cooling-days` 마침표 2를 사용하면 데이터의 활성 상태가 감소된 후에 곧바로 계층화할 수 있습니다.

를 클릭합니다 `auto`에만 해당되기 때문에 이러한 블록을 계층화하는 데 정책이 필요합니다 `auto` 정책은 액티브 파일 시스템에 있는 블록에 영향을 줍니다.



데이터에 대한 모든 액세스 유형은 열 지도 데이터를 재설정합니다. 바이러스 검사, 인덱싱, 심지어 소스 파일을 읽는 백업 활동으로 인해 계층화가 방지됩니다 `tiering-minimum-cooling-days` 임계값에 도달하지 않았습니다.

Oracle 부분 파일 FabricPool 계층화

FabricPool가 블록 레벨에서 작동하므로 변경이 필요한 파일은 부분적으로 오브젝트 스토리지로 계층화할 수 있고 성능 계층에 부분적으로 유지할 수도 있습니다.

이는 데이터베이스에 공통적으로 적용됩니다. 비활성 블록을 포함하는 것으로 알려진 데이터베이스도 FabricPool 계층화의 후보입니다. 예를 들어 공급망 관리 데이터베이스에는 필요 시 사용할 수 있어야 하지만 정상 작업 중에는 액세스할 수 없는 내역 정보가 포함될 수 있습니다. FabricPool를 사용하여 비활성 블록을 선택적으로 재배치할 수 있습니다.

예를 들어, 으로 FabricPool 볼륨에서 실행되는 데이터 파일이 여기에 해당합니다 `tiering-minimum-cooling-days 90` 일간 지난 90일 동안 액세스된 모든 블록을 성능 계층에서 유지합니다. 그러나 90일 동안 액세스하지 않은 모든 데이터는 용량 계층으로 재배치됩니다. 다른 경우에는 정상적인 애플리케이션 작업이 올바른 계층에서 올바른 블록을 보존합니다. 예를 들어 데이터베이스가 일반적으로 이전 60일 동안의 데이터를 정기적으로 처리하는 데 사용되는 경우 훨씬 더 낮습니다 `tiering-minimum-cooling-days` 애플리케이션의 자연적 활동으로 인해 블록이 조기에 재배치되지 않도록 하기 때문에 기간을 설정할 수 있다.

를 클릭합니다 `auto` 정책은 데이터베이스에 주의하여 사용해야 합니다. 많은 데이터베이스에는 분기말 프로세스 또는 재인덱싱 작업 같은 주기적인 활동이 있습니다. 이 작업의 기간이 보다 큰 경우 `tiering-minimum-cooling-days` 성능 문제가 발생할 수 있습니다. 예를 들어, 분기말 처리에는 영향을 받지 않은 1TB의 데이터가 필요한 경우 해당 데이터가 용량 계층에 존재할 수 있습니다. 용량 계층에서 읽는 속도는 매우 빠르며 성능 문제를 일으키지 않지만 정확한 결과는 오브젝트 저장소 구성에 따라 달라집니다.

정책

를 클릭합니다 `tiering-minimum-cooling-days` 정책은 성능 계층에 필요할 수 있는 파일을 보존할 수 있을 만큼 높게 설정해야 합니다. 예를 들어, 최적의 성능을 얻으려면 가장 최근 60일간의 데이터가 필요할 수 있는 데이터베이스가 필요할 경우 을 설정해야 합니다 `tiering-minimum-cooling-days 60` 일까지입니다. 파일의 액세스 패턴을 기준으로 비슷한 결과를 얻을 수도 있습니다. 예를 들어, 최근 90일 동안의 데이터가 필요하고 애플리케이션에서 90일 동안의 데이터에 액세스하는 경우 데이터는 성능 계층에 유지됩니다. 를 설정합니다 `tiering-minimum-cooling-days` 기간 - 2일 동안 데이터의 사용 빈도가 낮아지면 데이터를 즉시 계층화합니다.

를 클릭합니다 `auto`에만 해당되기 때문에 이러한 블록을 계층화하는 데 정책이 필요합니다 `auto` 정책은 액티브 파일 시스템에 있는 블록에 영향을 줍니다.



데이터에 대한 모든 액세스 유형은 열 지도 데이터를 재설정합니다. 따라서 데이터베이스 전체 테이블 검사 및 소스 파일을 읽는 백업 작업까지 필요할 때 계층화를 수행할 수 없습니다 `tiering-minimum-cooling-days` 임계값에 도달하지 않았습니다.

Oracle 데이터베이스 아카이브 로그 계층화

FabricPool의 가장 중요한 용도는 데이터베이스 트랜잭션 로그와 같은 알려진 콜드 데이터의 효율성을 개선하는 것입니다.

대부분의 관계형 데이터베이스는 트랜잭션 로그 보관 모드에서 작동하여 시점 복구를 제공합니다. 데이터베이스에 대한 변경 내용은 트랜잭션 로그에 변경 내용을 기록하여 커밋되며 트랜잭션 로그는 덮어쓰지 않고 유지됩니다. 따라서 대량의 아카이빙된 트랜잭션 로그를 보존해야 할 수 있습니다. 유사한 예제가 존재하며, 보존해야 하지만 액세스할 가능성은 매우 낮은 데이터를 생성합니다.

FabricPool는 통합 계층화를 통한 단일 솔루션을 제공하여 이러한 문제를 해결합니다. 파일이 저장되어 일반적인 위치에 계속 액세스할 수 있지만 운영 스토리지의 공간을 거의 차지하지 않습니다.

정책

를 사용합니다 `tiering-minimum-cooling-days` 며칠 정책을 적용하면 최근에 생성된 파일(단기간 내에 가장 필요할 가능성이 높은 파일)의 블록이 성능 계층에 보존됩니다. 그런 다음 이전 파일의 데이터 블록이 용량 계층으로 이동합니다.

를 클릭합니다 `auto` 로그가 삭제되었거나 운영 파일 시스템에 계속 존재하는지 여부에 관계없이 냉각 임계값에 도달하면 프롬프트 계층화를 적용합니다. 액티브 파일 시스템의 단일 위치에 잠재적으로 필요한 모든 로그를 저장하면 관리가 간편해집니다. 복원해야 하는 파일을 찾기 위해 스냅샷을 검색할 이유가 없습니다.

Microsoft SQL Server와 같은 일부 응용 프로그램은 백업 작업 중에 트랜잭션 로그 파일을 잘라서 로그가 더 이상 활성 파일 시스템에 없도록 합니다. 을 사용하여 용량을 절약할 수 있습니다 snapshot-only 계층화 정책만 지원하고 auto 활성 파일 시스템에 로그 데이터가 냉각되는 경우는 거의 없기 때문에 로그 데이터에는 정책이 유용하지 않습니다.

Oracle with FabricPool 스냅샷 계층화

FabricPool의 초기 릴리즈는 백업 활용 사례를 대상으로 합니다. 계층화할 수 있는 유일한 블록 유형은 액티브 파일 시스템의 데이터와 더 이상 관련되지 않은 블록이었습니다. 따라서 스냅샷 데이터 블록만 용량 계층으로 이동할 수 있습니다. 이는 성능에 영향을 미치지 않도록 해야 할 때 가장 안전한 계층화 옵션 중 하나로 남아 있습니다.

정책 - 로컬 스냅샷

비활성 스냅샷 블록을 용량 계층으로 계층화하는 두 가지 옵션이 있습니다. 첫째, 입니다 snapshot-only 정책은 스냅샷 블록만 타겟으로 합니다. 하지만 auto 정책에는 가 포함됩니다 snapshot-only 블록 또한 액티브 파일 시스템의 블록을 계층화합니다. 이것은 바람직하지 않을 수 있습니다.

를 클릭합니다 tiering-minimum-cooling-days 값을 성능 계층에서 복원 중에 필요할 수 있는 데이터를 사용할 수 있도록 하는 기간으로 설정해야 합니다. 예를 들어 중요한 운영 데이터베이스의 대부분의 복원 시나리오에는 이전 며칠 동안의 특정 시점의 복원 지점이 포함됩니다. 설정 A tiering-minimum-cooling-days 값을 3으로 설정하면 파일을 복원하면 즉시 최대 성능을 제공하는 파일이 만들어집니다. 활성 파일의 모든 블록은 용량 계층에서 복구할 필요 없이 고속 스토리지에 존재합니다.

정책 - 복제된 스냅샷

SnapMirror 또는 SnapVault로 복제된 스냅샷으로, 복구에만 사용되는 스냅샷은 일반적으로 FabricPool을 사용해야 합니다 all 정책. 이 정책을 사용하면 메타데이터가 복제되지만 모든 데이터 블록이 용량 계층으로 즉시 전송되어 최고의 성능을 낼 수 있습니다. 대부분의 복구 프로세스에는 본질적으로 효율적인 순차적 I/O가 포함됩니다. 객체 저장소 대상으로부터의 복구 시간을 평가해야 하지만, 잘 설계된 아키텍처에서는 이 복구 프로세스가 로컬 데이터에서 복구하는 것보다 훨씬 느릴 필요가 없습니다.

복제된 데이터도 클론 복제에 사용하도록 의도된 경우 는 를 참조하십시오 auto 을 사용하면 정책이 더 적절합니다 tiering-minimum-cooling-days 클론 복제 환경에서 정기적으로 사용될 것으로 예상되는 데이터를 포괄하는 값입니다. 예를 들어 데이터베이스의 활성 작업 집합에는 지난 3일 동안 읽거나 쓴 데이터가 포함될 수 있지만 6개월 동안의 기록 데이터도 포함될 수 있습니다. 그렇다면 를 클릭합니다 auto SnapMirror 대상에 대한 정책을 적용하면 성능 계층에서 작업 세트를 사용할 수 있습니다.

Oracle 데이터베이스 백업 계층화

기존 애플리케이션 백업에는 원래 데이터베이스의 위치 외부에서 파일 기반 백업을 생성하는 Oracle Recovery Manager 같은 제품이 포함됩니다.

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

를 클릭합니다 `auto` 정책은 백업 데이터에 가장 적합한 정책입니다. 이렇게 하면 파일이 삭제되었거나 운영 파일 시스템에 계속 존재하는지에 관계없이 냉각 임계값에 도달한 경우 프롬프트 계층화가 보장됩니다. 액티브 파일 시스템의 한 위치에 잠재적으로 필요한 모든 파일을 저장하면 관리가 간편해집니다. 복원해야 하는 파일을 찾기 위해 스냅샷을 검색할 이유가 없습니다.

를 클릭합니다 snapshot-only 정책을 적용할 수 있지만 이 정책은 더 이상 액티브 파일 시스템에 없는 블록에만 적용됩니다. 따라서 데이터를 계층화하기 전에 NFS 또는 SMB 공유의 파일을 먼저 삭제해야 합니다.

LUN에서 파일을 삭제하면 파일 시스템 메타데이터에서 파일 참조만 제거되기 때문에 이 정책은 LUN 구성에서는 훨씬 효율적입니다. LUN의 실제 블록은 덮어쓸 때까지 그대로 유지됩니다. 이 경우 파일이 삭제된 시간과 블록이 덮어써지고 계층화 대상이 되는 시간 사이에 긴 지연이 발생할 수 있습니다. 을(를) 옮기면 몇 가지 이점이 있습니다 snapshot-only 블록을 용량 계층으로 이동하지만 백업 데이터의 FabricPool 관리는 에서 가장 잘 작동합니다 auto 정책.



이러한 접근 방식은 사용자가 백업에 필요한 공간을 보다 효율적으로 관리하는 데 도움이 되지만 FabricPool 자체는 백업 기술이 아닙니다. 백업 파일을 오브젝트 저장소로 계층화하면 파일이 원래 스토리지 시스템에 계속 표시되지만 오브젝트 저장소 대상의 데이터 블록은 원본 스토리지 시스템에 따라 다르므로 관리가 간소화됩니다. 소스 볼륨이 손실되면 오브젝트 저장소 데이터를 더 이상 사용할 수 없습니다.

Oracle 데이터베이스 및 오브젝트 저장소 액세스 중단

FabricPool를 사용하여 데이터 세트를 계층화하면 운영 스토리지 어레이와 오브젝트 저장소 계층 간에 종속성이 발생합니다. 다양한 수준의 가용성을 제공하는 수많은 오브젝트 스토리지 옵션이 있습니다. 운영 스토리지 어레이와 오브젝트 스토리지 계층 간에 연결 손실의 영향을 이해하는 것이 중요합니다.

ONTAP로 발행된 I/O에 용량 계층의 데이터가 필요하고 ONTAP가 용량 계층에 연결하여 블록을 검색할 수 없는 경우 결국 I/O가 시간 초과됩니다. 이 시간 초과 효과는 사용된 프로토콜에 따라 다릅니다. NFS 환경에서 ONTAP는 프로토콜에 따라 EJUKEBOX 또는 EDELAY 응답으로 응답합니다. 일부 오래된 운영 체제에서는 이를 오류로 해석할 수 있지만 현재 운영 체제 및 Oracle Direct NFS 클라이언트의 현재 패치 수준에서는 이를 다시 시도 가능한 오류로 처리하고 I/O가 완료될 때까지 계속 기다립니다.

SAN 환경에 더 짧은 시간 초과가 적용됩니다. 객체 저장소 환경에서 블록이 필요하고 2분 동안 액세스할 수 없는 경우 읽기 오류가 호스트에 반환됩니다. ONTAP 볼륨과 LUN은 온라인 상태로 유지되지만 호스트 운영 체제에서 파일 시스템에 오류 상태가 플래그를 지정할 수 있습니다.

오브젝트 스토리지 연결 문제 snapshot-only 백업 데이터만 계층화되기 때문에 정책이 관심의 대상이 아닙니다. 통신 문제로 인해 데이터 복구 속도가 느려지지만 사용 중인 데이터에 영향을 주지 않습니다. 를 클릭합니다 auto 및 all 정책을 통해 활성 LUN에서 콜드 데이터를 계층화할 수 있으므로 오브젝트 저장소 데이터를 검색하는 동안 오류가 데이터베이스 가용성에 영향을 미칠 수 있습니다. 이러한 정책을 통한 SAN 구축은 고가용성을 위해 설계된 엔터프라이즈급 오브젝트 스토리지 및 네트워크 연결에서만 사용해야 합니다. NetApp StorageGRID는 탁월한 옵션입니다.

Oracle 데이터 보호

ONTAP을 사용한 Oracle 데이터 보호

NetApp은 가장 미션 크리티컬한 데이터가 데이터베이스에 있다는 것을 알고 있습니다.

데이터에 액세스하지 않고는 기업을 운영할 수 없으며 경우에 따라서는 데이터가 비즈니스를 정의하기도 합니다. 이러한 데이터는 보호해야 합니다. 그러나 데이터 보호는 사용 가능한 백업을 보장하는 것 이상의 의미를 갖습니다. 백업을 안전하게 저장하는 것은 물론 빠르고 안정적으로 수행하는 것입니다.

데이터 보호의 다른 측면은 데이터 복구입니다. 데이터에 액세스할 수 없으면 엔터프라이즈가 영향을 받으며 데이터를 복원하기 전까지 작동하지 않을 수 있습니다. 이 프로세스는 빠르고 안정적이어야 합니다. 마지막으로, 대부분의 데이터베이스는 재해로부터 보호해야 합니다. 즉, 데이터베이스의 복제본을 유지 관리해야 합니다. 복제본이 최신 상태여야 합니다. 또한 복제본을 완벽하게 작동하는 데이터베이스로 빠르고 간단하게 만들 수 있어야 합니다.



이 문서는 이전에 게시된 기술 보고서_TR-4591: Oracle 데이터 보호: 백업, 복구 및 복제 _를 대체합니다

계획 수립

적절한 엔터프라이즈 데이터 보호 아키텍처는 다양한 이벤트 중에 데이터 보존, 복구 기능 및 운영 중단에 대한 허용성과 관련된 비즈니스 요구 사항에 따라 달라집니다.

예를 들어, 적용 범위의 애플리케이션, 데이터베이스 및 중요 데이터 세트의 수를 예로 들어 보겠습니다. 관리할 객체가 많지 않기 때문에 단일 데이터 세트에 대해 일반적인 SLA를 준수하는 백업 전략을 구축하는 것은 매우 간단합니다. 데이터 세트의 수가 늘어나면 모니터링이 더 복잡해지고 관리자는 백업 장애를 해결하는 데 더 많은 시간을 소비해야 할 수도 있습니다. 환경이 클라우드에 도달하고 서비스 공급자가 확장됨에 따라 완전히 다른 접근 방식이 필요합니다.

데이터 세트 크기도 전략에 영향을 줍니다. 예를 들어, 데이터 세트가 매우 작기 때문에 100GB 데이터베이스를 사용한 백업 및 복구에 사용할 수 있는 옵션이 많이 있습니다. 기존 툴을 사용하여 백업 미디어에서 데이터를 복제하는 것만으로도 복구에 충분한 RTO를 얻을 수 있습니다. 일반적으로 100TB 데이터베이스에는 다일의 운영 중단이 허용되지 않는 한 100TB 데이터베이스에는 일반적으로 완전히 다른 전략이 필요합니다. 이 경우 기존의 복사본 기반 백업 및 복구 절차가 허용되는 경우가 아니라면 말입니다.

마지막으로, 백업 및 복구 프로세스 자체 이외의 요소가 있습니다. 예를 들어, 중요한 운영 작업을 지원하는 데이터베이스가 있어 숙련된 DBA만 수행하는 드문 이벤트로 간주되니까? 아니면 데이터베이스가 대규모 개발 환경에 있어서 복구가 자주 발생하고 일반 IT 팀이 관리하는 환경입니까?

Oracle 데이터베이스 RTO, RPO 및 SLA 계획 수립

ONTAP을 사용하면 Oracle 데이터베이스 데이터 보호 전략을 비즈니스 요구 사항에 맞게 쉽게 조정할 수 있습니다.

이러한 요구 사항에는 복구 속도, 허용되는 최대 데이터 손실 및 백업 보존 요구 사항 등의 요인이 포함됩니다. 데이터 보호 계획도 데이터 보존 및 복원에 대한 다양한 규정 요구 사항을 고려해야 합니다. 마지막으로 사용자 또는 애플리케이션 오류로 인해 발생하는 일반적인 복구 방법부터 사이트의 완전한 손실을 포함하는 재해 복구 시나리오에 이르기까지 다양한 데이터 복구 시나리오를 고려해야 합니다.

데이터 보호 및 복구 정책을 조금만 변경하면 스토리지, 백업 및 복구의 전체 아키텍처에 상당한 영향을 줄 수 있습니다. 데이터 보호 아키텍처의 복잡성을 방지하려면 설계 작업을 시작하기 전에 표준을 정의하고 문서화해야 합니다. 불필요한 기능이나 보호 수준은 불필요한 비용과 관리 부담을 초래하며, 초기에 간과한 요구 사항으로 인해 프로젝트가 잘못된

방향으로 진행되거나 최종 설계 변경이 필요할 수 있습니다.

복구 시간 목표

RTO(복구 시간 목표)는 서비스 복구에 허용되는 최대 시간을 정의합니다. 예를 들어 인사 데이터베이스의 RTO는 24시간이 될 수 있습니다. 왜냐하면 업무 중에 이 데이터에 액세스하지 못하는 것이 매우 불편함에도 불구하고 비즈니스가 계속 운영될 수 있기 때문입니다. 반면 은행의 총계정원장을 지원하는 데이터베이스에는 분 또는 초 단위로 측정된 RTO가 있습니다. 실제 서비스 중단과 네트워크 패킷 손실과 같은 일상적인 이벤트를 구분하는 방법이 있어야 하므로 RTO 0은 불가능합니다. 하지만 제로에 가까운 RTO는 일반적인 요구사항입니다.

복구 시점 목표

RPO(복구 지점 목표)는 허용되는 최대 데이터 손실을 정의합니다. 대부분의 경우 RPO는 스냅샷 또는 SnapMirror 업데이트 빈도에 의해서만 결정됩니다.

경우에 따라 RPO를 보다 적극적으로 설정하여 특정 데이터를 보다 자주 보호할 수 있습니다. 데이터베이스 컨텍스트에서 RPO는 일반적으로 특정 상황에서 손실될 수 있는 로그 데이터의 양이 어느 정도인지에 관한 문제입니다. 제품 버그 또는 사용자 오류로 인해 데이터베이스가 손상된 일반적인 복구 시나리오에서 RPO는 0이어야 합니다. 즉, 데이터 손실이 없어야 합니다. 복구 절차에서는 데이터베이스 파일의 이전 복사본을 복원한 다음 로그 파일을 재생하여 데이터베이스 상태를 원하는 시점으로 되돌리는 작업을 수행합니다. 이 작업에 필요한 로그 파일이 이미 원래 위치에 있어야 합니다.

비정상적인 시나리오에서는 로그 데이터가 손실될 수 있습니다. 예를 들어, 우발적 또는 악의적 경우입니다 `rm -rf *` 데이터베이스 파일의 경우 모든 데이터가 삭제될 수 있습니다. 유일한 옵션은 로그 파일을 포함하여 백업에서 복원하는 것이며, 일부 데이터가 손실될 수밖에 없습니다. 기존 백업 환경에서 RPO를 향상시킬 수 있는 유일한 옵션은 로그 데이터의 반복 백업을 수행하는 것입니다. 그러나 지속적인 데이터 이동과 백업 시스템을 지속적으로 실행하는 서비스로 유지 관리하기가 어렵기 때문에 이러한 문제에는 한계가 있습니다. 고급 스토리지 시스템의 이점 중 하나는 파일에 대한 우발적 또는 악의적 손상으로부터 데이터를 보호하여 데이터 이동 없이 더 높은 RPO를 제공하는 기능입니다.

재해 복구

재해 복구에는 물리적 재해 발생 시 서비스를 복구하는 데 필요한 IT 아키텍처, 정책 및 절차가 포함됩니다. 여기에는 홍수, 화재 또는 악의적이거나 과실로 행동하는 사람이 포함될 수 있습니다.

재해 복구는 단순한 복구 절차 그 이상입니다. 다양한 위험을 식별하고, 데이터 복구 및 서비스 연속성 요구 사항을 정의하고, 관련 절차에 따라 올바른 아키텍처를 제공하는 완전한 프로세스입니다.

데이터 보호 요구 사항을 설정할 때는 일반적인 RPO 및 RTO 요구 사항과 재해 복구에 필요한 RPO 및 RTO 요구 사항을 구분해야 합니다. 일부 애플리케이션 환경에서는 비교적 일반적인 사용자 오류부터 데이터 센터 파괴에 이르는 데이터 손실 상황에 대해 0의 RPO와 0에 가까운 RTO가 필요합니다. 그러나 이러한 높은 수준의 보호를 위해서는 비용과 관리 상의 문제가 발생합니다.

일반적으로 비재해 데이터 복구 요구사항은 두 가지 이유로 엄격해야 합니다. 첫째, 애플리케이션 버그와 사용자 오류로 인해 데이터가 손상되는 것은 거의 불가피한 시점까지 예상할 수 있습니다. 둘째, 스토리지 시스템이 폐기되지 않는 한 RPO 0과 낮은 RTO를 제공할 수 있는 백업 전략을 설계하는 것은 쉽지 않습니다. 쉽게 해결할 수 있는 심각한 위험을 해결하지 않을 이유가 없습니다. 따라서 로컬 복구에 대한 RPO 및 RTO 목표를 적극적으로 적용해야 합니다.

재해 복구 RTO 및 RPO 요구 사항은 재해 가능성 및 관련 데이터 손실 또는 비즈니스 중단 결과에 따라 더 크게 달라집니다. RPO 및 RTO 요구 사항은 일반적인 원칙이 아닌 실제 비즈니스 요구 사항을 기반으로 해야 합니다. 여러 논리적 및 물리적 재해 시나리오를 고려해야 합니다.

논리적 재해

논리적 재해에는 사용자, 애플리케이션 또는 OS 버그, 소프트웨어 오작동으로 인한 데이터 손상이 포함됩니다. 논리적 재해에는 바이러스 또는 웜이 있는 외부 사용자에게 의한 악의적인 공격이나 응용 프로그램 취약점을 악용하는 공격도 포함될 수 있습니다. 이러한 경우 물리적 인프라스트럭처는 손상되지 않지만 기본 데이터는 더 이상 유효하지 않습니다.

점점 더 일반적인 유형의 논리적 재해를 랜섬웨어라고 하며, 이를 공격 벡터가 데이터를 암호화하는 데 사용됩니다. 암호화는 데이터를 손상시키지 않지만 제3자에게 지불이 이루어질 때까지 데이터를 사용할 수 없게 합니다. 랜섬웨어 해킹을 특별히 도입한 기업이 점점 더 많아지고 있습니다. NetApp은 이러한 위협에 대해 변조 방지 스냅샷을 제공합니다. 따라서 스토리지 관리자도 구성된 만료일 전에 보호된 데이터를 변경할 수 없습니다.

물리적 재해

물리적 재해에는 인프라의 구성 요소가 중복성을 넘어 데이터 손실이나 서비스 손실로 이어지는 장애가 포함됩니다. 예를 들어 RAID 보호는 디스크 드라이브 이중화를 제공하며 HBA를 사용하면 FC 포트 및 FC 케이블 이중화를 제공할 수 있습니다. 이러한 구성 요소의 하드웨어 장애는 예측 가능하며 가용성에 영향을 미치지 않습니다.

엔터프라이즈 환경에서는 일반적으로 예측 가능한 유일한 물리적 재해 시나리오가 사이트의 완전한 손실인 시점까지 중복 구성 요소를 사용하여 전체 사이트의 인프라를 보호할 수 있습니다. 그런 다음 재해 복구 계획이 사이트 간 복제에 달려 있습니다.

동기식 및 비동기식 데이터 보호

이상적인 환경에서는 모든 데이터를 지리적으로 분산된 사이트에 걸쳐 동기식으로 복제합니다. 이러한 복제는 다음과 같은 몇 가지 이유로 항상 실현 가능하지 않거나 가능한 경우가 있습니다.

- 애플리케이션/데이터베이스가 처리를 진행하기 전에 모든 변경 사항을 두 위치에 복제해야 하기 때문에 동기식 복제는 불가피하게 쓰기 지연 시간을 증가시킵니다. 이로 인해 발생할 수 있는 성능 영향은 용납되지 않으므로 동기식 미러링 사용을 배제할 수 있습니다.
- 100% SSD 스토리지의 채택이 증가함에 따라 성능 기대치에는 수십만 IOPS와 1ms 미만의 지연 시간이 포함되므로 쓰기 지연 시간이 더 많이 발생하고 있습니다. 100% SSD 사용의 이점을 최대한 활용하려면 재해 복구 전략을 다시 세워야 할 수 있습니다.
- 데이터 세트는 바이트 측면에서 계속 증가하고 있기 때문에 동기 복제를 지속할 수 있는 충분한 대역폭을 확보하는 데 어려움이 있습니다.
- 또한 데이터 세트가 복잡해지면서 대규모 동기식 복제 관리에 따르는 문제가 발생합니다.
- 클라우드 기반 전략에서는 복제 거리와 지연 시간이 더욱 길어져 동기식 미러링을 사용하는 것이 오히려 사라지는 경우가 많습니다.

NetApp은 가장 까다로운 데이터 복구 요구사항을 위한 동기식 복제 솔루션과 향상된 성능과 유연성을 지원하는 비동기 솔루션을 제공합니다. 또한 NetApp 기술은 Oracle DataGuard와 같은 여러 타사 복제 솔루션과 원활하게 통합됩니다.

보존 시간

데이터 보호 전략의 마지막 측면은 데이터 보존 시간이며, 이는 크게 달라질 수 있습니다.

- 일반적으로 운영 사이트에서 14일 야간 백업을 수행하고 보조 사이트에 90일 동안 백업을 저장해야 합니다.
- 많은 고객이 서로 다른 미디어에 저장된 분기별 독립 실행형 아카이브를 생성합니다.
- 데이터베이스를 지속적으로 업데이트하면 기록 데이터가 필요하지 않을 수 있으며, 백업은 며칠 동안만 보존되어야 합니다.

- 규정 요구 사항에 따라 365일 기간 내에 임의의 트랜잭션 시점까지 복구가 필요할 수 있습니다.

ONTAP을 사용한 Oracle 데이터베이스 가용성

ONTAP는 최대한의 Oracle 데이터베이스 가용성을 제공하도록 설계되었습니다. ONTAP 고가용성 기능에 대한 전체 설명은 본 문서의 범위를 벗어나며 그러나 데이터 보호와 마찬가지로 데이터베이스 인프라를 설계할 때는 이 기능에 대한 기본적인 이해가 중요합니다.

HA 쌍

고가용성의 기본 단위는 HA 쌍입니다. 각 쌍에는 NVRAM에 데이터 복제를 지원하는 중복 링크가 포함되어 있습니다. NVRAM은 쓰기 캐시가 아닙니다. 컨트롤러 내의 RAM은 쓰기 캐시 역할을 합니다. NVRAM의 목적은 예기치 않은 시스템 장애를 방지하기 위해 일시적으로 데이터를 저널링하는 것입니다. 이 점에서 데이터베이스 재실행 로그와 유사합니다.

NVRAM과 데이터베이스 재실행 로그를 모두 사용하여 데이터를 빠르게 저장하므로 데이터의 변경사항을 최대한 빠르게 커밋할 수 있습니다. 드라이브(또는 데이터 파일)의 영구 데이터에 대한 업데이트는 ONTAP 플랫폼과 대부분의 데이터베이스 플랫폼 모두에서 체크포인트라는 프로세스 도중 늦게 수행되지 않습니다. 정상 작업 중에는 NVRAM 데이터나 데이터베이스 재실행 로그를 읽지 않습니다.

컨트롤러가 갑자기 실패할 경우 드라이브에 아직 기록되지 않은 NVRAM에 저장된 변경 사항이 보류 중일 수 있습니다. 파트너 컨트롤러는 장애를 감지하고 드라이브를 제어하며 NVRAM에 저장된 필수 변경 사항을 적용합니다.

테이크오버 및 반환

Takeover 및 Giveback은 HA 2노드의 노드 간에 스토리지 리소스에 대한 책임을 지는 프로세스를 의미합니다. Takeover와 반환에는 두 가지 측면이 있습니다.

- 드라이브에 액세스할 수 있는 네트워크 연결 관리
- 드라이브 자체 관리

CIFS 및 NFS 트래픽을 지원하는 네트워크 인터페이스는 홈 위치와 페일오버 위치 모두를 사용하여 구성됩니다. 테이크오버는 원래 위치와 동일한 서브넷에 있는 물리적 인터페이스에서 네트워크 인터페이스를 임시 홈으로 이동하는 것을 포함합니다. 반환에는 네트워크 인터페이스를 원래 위치로 이동하는 것도 포함됩니다. 정확한 동작은 필요에 따라 조정할 수 있습니다.

iSCSI 및 FC와 같은 SAN 블록 프로토콜을 지원하는 네트워크 인터페이스는 테이크오버 및 반환 중에 재배치되지 않습니다. 전체 HA 쌍이 포함된 경로를 사용하여 LUN을 프로비저닝해야 하므로 1차 경로와 2차 경로가 됩니다.



추가 컨트롤러를 위한 추가 경로를 대규모 클러스터에서 노드 간 데이터 재배치를 지원하도록 구성할 수도 있지만 이는 HA 프로세스에 포함되지 않습니다.

Takeover와 Giveback의 두 번째 측면은 디스크 소유권을 이전하는 것입니다. 정확한 프로세스는 Takeover/Giveback 이유 및 실행된 명령줄 옵션을 비롯한 여러 요인에 따라 달라집니다. 목표는 최대한 효율적으로 작업을 수행하는 것입니다. 전체 프로세스에는 몇 분이 필요한 것처럼 보이지만 드라이브 소유권이 노드에서 노드로 전환되는 실제 순간은 일반적으로 초 단위로 측정할 수 있습니다.

인수 시간

테이크오버 및 반환 작업 중에 호스트 I/O가 잠깐 정지되지만 올바르게 구성된 환경에서는 애플리케이션이 중단되어서는 안 됩니다. I/O가 지연되는 실제 전환 프로세스는 일반적으로 몇 초 내로 측정되지만, 호스트에서 데이터 경로의 변경을

인식하고 I/O 작업을 다시 제출하기 위해 추가 시간이 필요할 수 있습니다.

중단 특성은 프로토콜에 따라 다릅니다.

- NFS 및 CIFS 트래픽을 지원하는 네트워크 인터페이스는 새 물리적 위치로 전환한 후 네트워크에 ARP(Address Resolution Protocol) 요청을 발급합니다. 이로 인해 네트워크 스위치가 MAC(Media Access Control) 주소 테이블을 업데이트하고 I/O 처리를 재개합니다. 계획된 테이크오버와 반환의 경우 운영 중단은 일반적으로 몇 초 단위로 측정되며, 대부분의 경우 감지할 수 없는 경우가 많습니다. 일부 네트워크는 네트워크 경로의 변화를 완전히 인식하기 위해 더 느려질 수 있으며 일부 운영 체제는 재시도해야 하는 매우 짧은 시간 내에 많은 I/O를 대기시킬 수 있습니다. 이렇게 하면 입출력을 재개하는 데 필요한 시간이 길어질 수 있습니다.
- SAN 프로토콜을 지원하는 네트워크 인터페이스는 새 위치로 전환되지 않습니다. 호스트 운영 체제에서 사용 중인 경로를 변경해야 합니다. 호스트에서 관찰되는 입출력 일시 중지는 여러 요인에 따라 달라집니다. 스토리지 시스템 관점에서 볼 때 입출력을 처리할 수 없는 기간은 불과 몇 초입니다. 그러나 다른 호스트 운영 체제마다 재시도하기 전에 I/O가 시간 초과되도록 하려면 추가 시간이 필요할 수 있습니다. 최신 운영 체제는 경로 변경을 훨씬 더 빠르게 인식할 수 있지만, 기존 운영 체제에서는 일반적으로 변경 사항을 인식하는 데 최대 30초가 걸립니다.

아래 표에는 스토리지 시스템에서 애플리케이션 환경에 데이터를 제공할 수 없는 테이크오버가 예상되고 있습니다. 애플리케이션 환경에서 오류가 발생하지 않도록 하려면 테이크오버 대신 IO 처리 중에 잠깐 정지된 상태로 표시되어야 합니다.

	NFS 를 참조하십시오	AFF	ASA
계획된 테이크오버	15초	6-10초	2-3초
계획되지 않은 테이크오버	30초	6-10초	2-3초

체크섬 및 Oracle 데이터베이스 무결성

ONTAP 및 지원되는 프로토콜에는 유틸 데이터와 네트워크 네트워크를 통해 전송되는 데이터를 비롯하여 Oracle 데이터베이스 무결성을 보호하는 여러 기능이 포함되어 있습니다.

ONTAP 내의 논리적 데이터 보호는 다음과 같은 세 가지 주요 요구 사항으로 구성됩니다.

- 데이터가 손상되지 않도록 보호해야 합니다.
- 드라이브 장애로부터 데이터를 보호해야 합니다.
- 데이터 변경사항을 손실로부터 보호해야 합니다.

이 세 가지 요구 사항은 다음 섹션에서 설명합니다.

네트워크 손상: 체크섬

가장 기본적인 데이터 보호 수준은 데이터와 함께 저장되는 특별한 오류 감지 코드인 체크섬입니다. 네트워크 전송 중 데이터 손상은 체크섬을 사용하여 감지되며 경우에 따라 여러 체크섬을 사용할 수 있습니다.

예를 들어, FC 프레임에는 전송 중에 페이로드가 손상되지 않도록 CRC(Cyclic Redundancy Check)라는 체크섬 유형이 포함되어 있습니다. 송신기는 데이터와 데이터의 CRC를 모두 전송합니다. FC 프레임의 수신기는 수신된 데이터의 CRC를 다시 계산하여 전송된 CRC와 일치하는지 확인합니다. 새로 계산된 CRC가 프레임에 연결된 CRC와 일치하지 않으면 데이터가 손상되고 FC 프레임이 삭제되거나 거부됩니다. iSCSI I/O 작업에는 TCP/IP 및 이더넷 계층에 체크섬이 포함되며, 추가적인 보호를 위해 SCSI 계층에 선택 사항인 CRC 보호 기능이 포함될 수도 있습니다. 와이어의 모든 비트 손상은 TCP 계층 또는 IP 계층에서 감지되어 패킷의 재전송을 초래합니다. FC와 마찬가지로 SCSI CRC의 오류로 인해 작업이 삭제되거나 거부됩니다.

드라이브 손상: 체크섬

또한 체크섬을 사용하여 드라이브에 저장된 데이터의 무결성을 검증합니다. 드라이브에 기록된 데이터 블록은 원래 데이터와 연결된 예측 불가능한 수를 생성하는 체크섬 기능과 함께 저장됩니다. 드라이브에서 데이터를 읽으면 체크섬이 다시 계산되어 저장된 체크섬과 비교됩니다. 일치하지 않으면 데이터가 손상되어 RAID 계층에 의해 복구되어야 합니다.

데이터 손상: 쓰기 손실

감지하기 가장 어려운 유형의 손상 중 하나는 손실되거나 잘못 배치된 쓰기입니다. 쓰기가 확인되면 올바른 위치에 있는 미디어에 기록해야 합니다. 데이터 이동 없는 데이터 손상은 데이터와 함께 저장된 간단한 체크섬을 사용하여 비교적 쉽게 감지할 수 있습니다. 그러나 쓰기가 단순히 손실되는 경우 이전 버전의 데이터가 여전히 존재하고 체크섬이 정확할 수 있습니다. 쓰기가 잘못된 물리적 위치에 배치되면 쓰기가 다른 데이터를 제거했더라도 연결된 체크섬이 저장된 데이터에 대해 다시 한 번 유효합니다.

이 문제에 대한 해결책은 다음과 같습니다.

- 쓰기 작업에는 쓰기를 찾을 위치를 나타내는 메타데이터가 포함되어야 합니다.
- 쓰기 작업에는 버전 식별자가 일부 포함되어 있어야 합니다.

ONTAP에서 블록을 쓰면 해당 블록이 속한 위치에 대한 데이터가 포함됩니다. 후속 읽기에서 블록을 식별하지만 메타데이터에서 위치 456에서 찾은 위치 123에 해당 블록이 속해 있음을 나타내는 경우 쓰기가 잘못 배치된 것입니다.

완전히 손실된 쓰기를 감지하기가 더 어렵습니다. 이 설명은 매우 복잡하지만 기본적으로 ONTAP은 쓰기 작업으로 인해 드라이브의 서로 다른 두 위치에 업데이트가 이루어지도록 메타데이터를 저장하고 있습니다. 쓰기가 손실되면 후속 데이터 읽기와 관련 메타데이터를 읽으면 서로 다른 두 버전 ID가 표시됩니다. 이는 드라이브에서 쓰기가 완료되지 않았음을 나타냅니다.

손실되거나 잘못 배치된 쓰기 손상은 매우 드물지만 드라이브가 계속 증가하고 데이터 세트가 엑사바이트 규모로 증가함에 따라 위험이 증가합니다. 손실된 쓰기 감지 기능은 데이터베이스 워크로드를 지원하는 모든 스토리지 시스템에 포함되어야 합니다.

드라이브 장애: RAID, RAID DP 및 RAID-TEC

드라이브의 데이터 블록이 손상되었거나 전체 드라이브에 장애가 발생하여 완전히 사용할 수 없는 경우 데이터를 다시 구성해야 합니다. 이 작업은 ONTAP에서 패리티 드라이브를 사용하여 수행됩니다. 데이터가 여러 데이터 드라이브에 스트라이핑된 다음 패리티 데이터가 생성됩니다. 원본 데이터와 별도로 저장됩니다.

ONTAP에서는 원래 데이터 드라이브 그룹마다 단일 패리티 드라이브를 사용하는 RAID 4를 사용했습니다. 그 결과, 그룹의 드라이브 중 하나에 장애가 발생하여 데이터가 손실되지 않을 수 있었습니다. 패리티 드라이브에 장애가 발생하면 데이터가 손상되지 않았으며 새 패리티 드라이브를 구성할 수 있습니다. 단일 데이터 드라이브에 장애가 발생하면 나머지 드라이브를 패리티 드라이브와 함께 사용하여 누락된 데이터를 재생성할 수 있습니다.

드라이브 용량이 작을 경우 두 개의 드라이브가 동시에 실패할 가능성이 매우 낮았습니다. 드라이브 용량이 증가함에 따라 드라이브 장애 후 데이터를 재구성하는 데 많은 시간이 필요하게 되었습니다. 이로 인해 두 번째 드라이브 오류로 인해 데이터가 손실되는 기간이 늘어났습니다. 또한 리빌드 프로세스는 나머지 드라이브에서 많은 I/O를 추가로 생성합니다. 드라이브가 노후화되면 추가 로드로 인해 두 번째 드라이브 장애가 발생할 위험도 증가합니다. 마지막으로 RAID 4를 계속 사용하면 데이터 손실 위험이 증가하지 않더라도 데이터 손실의 결과는 더욱 심각해집니다. RAID 그룹 장애 시 손실될 데이터가 많을수록 데이터 복구에 시간이 더 오래 걸리기 때문에 비즈니스 운영이 중단됩니다.

이러한 문제로 인해 NetApp는 RAID 6의 변종인 NetApp RAID DP 기술을 개발하게 되었습니다. 이 솔루션에는 패리티 드라이브가 2개 포함되어 있으므로 RAID 그룹에 있는 두 드라이브가 데이터 손실없이 실패할 수 있습니다. 드라이브의 크기가 계속 커짐에 따라 NetApp는 NetApp RAID-TEC 기술을 개발하여 세 번째 패리티 드라이브를 도입했습니다.

일부 내역 데이터베이스 모범 사례에서는 스트라이프 미러링이라고도 하는 RAID-10을 사용할 것을 권장합니다. 여러 개의 2-디스크 장애 시나리오가 있기 때문에 RAID DP보다 데이터 보호 기능이 떨어지는 반면 RAID DP는 없습니다.

또한 성능 문제로 인해 RAID-10이 RAID-4/5/6 옵션보다 선호됨을 나타내는 몇 가지 과거 데이터베이스 모범 사례가 있습니다. 이러한 권장 사항은 종종 RAID 성능 저하와 관련이 있습니다. 이러한 권장 사항은 일반적으로 정확하지만 ONTAP 내에서 RAID를 구현하는 경우에는 적용되지 않습니다. 성능 문제는 패리티 재생과 관련이 있습니다. 기존 RAID 구현에서 데이터베이스에서 수행하는 일상적인 랜덤 쓰기를 처리하려면 패리티 데이터를 재생성하고 쓰기를 완료하기 위해 여러 디스크 읽기를 수행해야 합니다. 페널티는 쓰기 작업을 수행하는 데 필요한 추가 읽기 IOPS로 정의됩니다.

쓰기 작업은 패리티가 생성된 메모리에서 스테이징된 다음 단일 RAID 스트라이프로 디스크에 기록되므로 ONTAP에서 RAID 페널티가 발생하지 않습니다. 쓰기 작업을 완료하는 데 읽기 작업이 필요하지 않습니다.

요약하면 RAID DP 및 RAID-TEC는 RAID 10과 비교하여 훨씬 더 많은 가용 용량을 제공하고, 드라이브 장애를 방지하며, 성능에 영향을 주지 않습니다.

하드웨어 장애 방지: **NVRAM**

데이터베이스 워크로드를 처리하는 스토리지 어레이는 쓰기 작업을 최대한 빨리 처리해야 합니다. 또한 쓰기 작업은 전원 장애와 같은 예기치 않은 이벤트로 인한 손실로부터 보호해야 합니다. 즉, 쓰기 작업은 적어도 두 위치에 안전하게 저장해야 합니다.

AFF 및 FAS 시스템은 이러한 요구사항을 충족하기 위해 NVRAM을 사용합니다. 쓰기 프로세스는 다음과 같이 작동합니다.

1. 인바운드 쓰기 데이터는 RAM에 저장됩니다.
2. 디스크의 데이터에 필요한 변경 사항은 로컬 및 파트너 노드 모두의 NVRAM으로 저널링됩니다. NVRAM은 쓰기 캐시가 아니라 데이터베이스 재실행 로그와 유사한 저널입니다. 정상적인 상태에서는 읽지 않습니다. I/O 처리 중 전원 장애 발생 후와 같은 복구에만 사용됩니다.
3. 그러면 쓰기가 호스트에 인식됩니다.

이 단계의 쓰기 프로세스는 응용 프로그램 관점에서 완료되며 데이터는 서로 다른 두 위치에 저장되므로 손실로부터 보호됩니다. 최종적으로 변경 사항이 디스크에 기록되지만 이 프로세스는 쓰기 확인 후에 발생하므로 지연 시간에 영향을 주지 않기 때문에 애플리케이션 관점에서 대역 외로 처리됩니다. 이 프로세스는 데이터베이스 로깅과 다시 유사합니다. 데이터베이스 변경 사항은 가능한 한 빨리 재실행 로그에 기록되고 변경 사항은 커밋된 것으로 확인됩니다. 데이터 파일에 대한 업데이트는 훨씬 나중에 수행되며 처리 속도에 직접적인 영향을 주지 않습니다.

컨트롤러 장애가 발생할 경우 파트너 컨트롤러가 필요한 디스크의 소유권을 가져오고 NVRAM에 기록된 데이터를 재생하여 장애가 발생했을 때 전송 중이었던 I/O 작업을 복구합니다.

하드웨어 장애 보호: **NVFAIL**

앞서 설명한 것처럼, 쓰기는 하나 이상의 다른 컨트롤러에서 로컬 NVRAM 및 NVRAM에 로그인되기 전까지는 승인되지 않습니다. 이렇게 하면 하드웨어 장애나 정전이 발생해도 전송 중인 I/O가 손실되지 않습니다. 로컬 NVRAM에 장애가 발생하거나 HA 파트너에 대한 연결이 실패하면 전송 중인 이 데이터는 더 이상 미러링되지 않습니다.

로컬 NVRAM에 오류가 보고되면 노드가 종료됩니다. 이 종료를 통해 HA 파트너 컨트롤러로 페일오버됩니다. 오류가 발생한 컨트롤러가 쓰기 작업을 인식하지 못했기 때문에 데이터가 손실되지 않습니다.

페일오버가 강제 적용되지 않는 한 ONTAP는 데이터가 동기화되지 않을 때 페일오버를 허용하지 않습니다. 이러한 방식으로 조건을 강제로 변경하면 데이터가 원래 컨트롤러에 남겨질 수 있으며 데이터 손실이 허용되는 수준임을 알 수 있습니다.

데이터베이스는 디스크에 대규모 내부 데이터 캐시를 유지하기 때문에 페일오버가 강제 적용되는 경우 손상에 특히 취약합니다. 강제 적용 페일오버가 발생하면 이전에 승인되었던 변경사항이 효과적으로 폐기됩니다. 스토리지 어레이의 콘텐츠가 사실상 이전 시간으로 이동하며, 데이터베이스 캐시의 상태는 디스크에 있는 데이터의 상태를 더 이상 반영하지 않습니다.

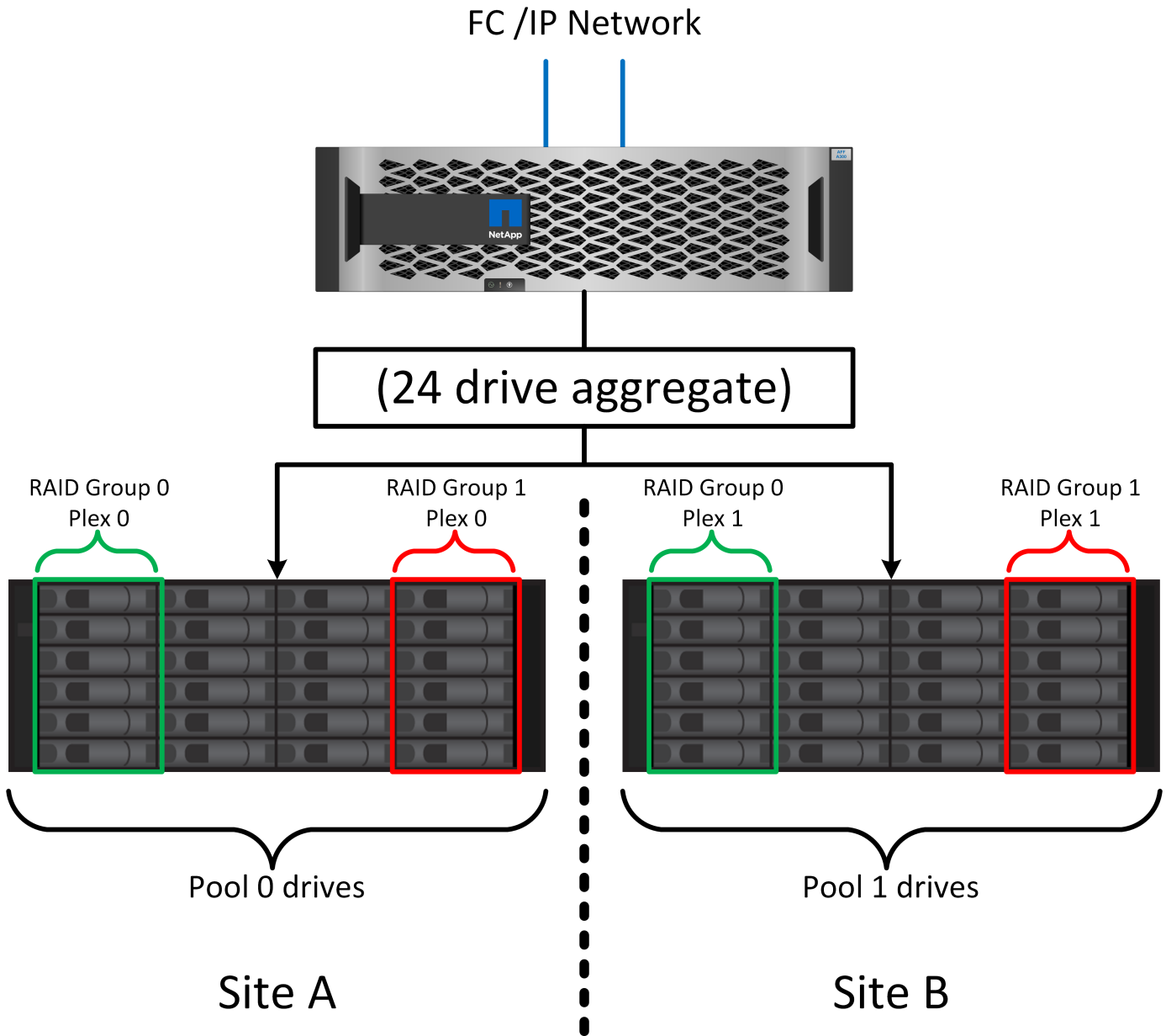
이 상황에서 데이터를 보호하기 위해 ONTAP에서는 NVRAM 장애에 대비하여 특별한 보호를 제공하도록 볼륨을 구성할 수 있습니다. 이 보호 메커니즘이 트리거되면 볼륨이 NVFAIL이라는 상태로 전환됩니다. 이 상태에서는 I/O 오류가 발생하여 오래된 데이터를 사용하지 않도록 애플리케이션이 종료됩니다. 확인된 쓰기가 스토리지 배열에 있어야 하므로 데이터가 손실되지 않아야 합니다.

일반적인 다음 단계는 관리자가 LUN 및 볼륨을 수동으로 다시 온라인 상태로 전환하기 전에 호스트를 완전히 종료하는 것입니다. 이러한 단계에는 일부 작업이 포함될 수 있지만 이 접근 방식은 데이터 무결성을 보장하는 가장 안전한 방법입니다. 모든 데이터에 이 보호가 필요한 것은 아니므로 NVFAIL 동작을 볼륨별로 구성할 수 있습니다.

사이트 및 셸프 장애 보호: **SyncMirror** 및 플렉스

SyncMirror는 RAID DP 또는 RAID-TEC를 향상하지만 대체하지는 않는 미러링 기술입니다. 2개의 독립적인 RAID 그룹의 콘텐츠를 미러링합니다. 논리적 구성은 다음과 같습니다.

- 드라이브는 위치에 따라 두 개의 풀로 구성됩니다. 하나의 풀은 사이트 A의 모든 드라이브로 구성되고, 두 번째 풀은 사이트 B의 모든 드라이브로 구성됩니다
- 그런 다음 애그리게이트라고 하는 공통 스토리지 풀이 RAID 그룹의 미러링된 세트를 기반으로 생성됩니다. 각 사이트에서 동일한 수의 드라이브가 그려집니다. 예를 들어, 20개 드라이브로 구성된 SyncMirror 애그리게이트는 사이트 A의 드라이브 10개와 사이트 B의 드라이브 10개로 구성됩니다
- 특정 사이트의 각 드라이브 세트는 미러링 사용과 관계없이 하나 이상의 완전히 이중화된 RAID-DP 또는 RAID-TEC 그룹으로 자동으로 구성됩니다. 따라서 사이트 손실 후에도 데이터를 지속적으로 보호할 수 있습니다.



위 그림은 SyncMirror 구성의 예를 보여 줍니다. 24-드라이브 애그리게이트가 사이트 A에 할당된 쉘프의 드라이브 12개와 사이트 B에 할당된 쉘프의 드라이브 12개로 컨트롤러에서 생성되었습니다. 드라이브는 두 개의 미러링된 RAID 그룹으로 그룹화되었습니다. RAID Group 0에는 사이트 B의 6개 드라이브 플렉스에 미러링된 사이트 A의 6개 드라이브 플렉스가 포함되어 있습니다. 마찬가지로, RAID 그룹 1에는 사이트 B의 6개 드라이브 플렉스에 미러링되는 사이트 A의 6개 드라이브 플렉스가 포함되어 있습니다.

SyncMirror는 일반적으로 각 사이트에 하나의 데이터 복사본으로 MetroCluster 시스템에 원격 미러링을 제공하는 데 사용됩니다. 경우에 따라 단일 시스템에서 추가 수준의 이중화를 제공하기 위해 사용되었습니다. 특히, 쉘프 레벨 이중화를 제공합니다. 드라이브 쉘프에는 이미 이중 전원 공급 장치와 컨트롤러가 포함되어 있으며 전반적으로 판금보다 조금 더 크지만, 경우에 따라 추가 보호가 필요할 수 있습니다. 예를 들어, 한 NetApp 고객은 자동차 테스트에 사용되는 모바일 실시간 분석 플랫폼용 SyncMirror를 구축했습니다. 시스템은 독립적인 UPS 시스템의 독립적인 전원 공급으로 공급되는 두 개의 물리적 랙으로 분리되었습니다.

체크섬

체크섬에 대한 주제는 Oracle RMAN 스트리밍 백업을 사용하는 데 익숙한 DBA가 스냅샷 기반 백업으로

마이그레이션하는 데 특히 유용합니다. RMAN은 백업 운영 중에 무결성 점검을 수행하는 기능을 가지고 있습니다. 이것이 유용하기는 하지만 이 기능의 주요 이점은 최신 스토리지 어레이에 사용되지 않는 데이터베이스를 위한 것입니다. Oracle 데이터베이스에 물리적 드라이브를 사용할 때 드라이브가 노후하면 결국 손상이 발생할 확률이 매우 높아지는데, 이 문제는 진정한 스토리지 어레이에서 어레이 기반 체크섬을 통해 해결됩니다.

진정한 스토리지 어레이는 여러 레벨에서 체크섬을 사용하여 데이터 무결성을 보호합니다. IP 기반 네트워크에서 데이터가 손상된 경우 TCP(Transmission Control Protocol) 계층은 패킷 데이터를 거부하고 재전송을 요청합니다. FC 프로토콜은 캡슐화된 SCSI 데이터처럼 체크섬을 포함하고 있습니다. 이것이 어레이에 배치되면 ONTAP에서 RAID 및 체크섬 보호 기능을 수행할 수 있습니다. 대부분의 엔터프라이즈 어레이에서 그렇듯 손상이 발생할 수도 있지만 감지하여 수정할 수 있습니다. 일반적으로 전체 드라이브에 장애가 발생하면 RAID 리빌드가 신속하게 이뤄지며 데이터베이스 무결성은 영향을 받지 않습니다. ONTAP가 드라이브의 데이터가 손상되었음을 의미하는 체크섬 오류를 감지하는 경우가 간혹 있습니다. 그러면 드라이브 작동이 중단되고 RAID 리빌드가 시작됩니다. 여기서도 데이터 무결성은 영향을 받지 않습니다.

또한, Oracle 데이터 파일 및 재실행 로그 아키텍처는 극단적인 환경에서도 최고 수준의 데이터 무결성을 제공하도록 설계되었습니다. 가장 기본적인 레벨에서 Oracle 블록은 거의 모든 I/O에 관한 체크섬과 기본 논리 점검을 포함합니다. Oracle이 충돌하거나 테이블스페이스를 오프라인으로 전환하지 않았다면 데이터는 온전한 상태입니다. 데이터 무결성 점검의 수준은 조정할 수 있으며 쓰기를 확인하도록 Oracle을 구성할 수도 있습니다. 결과적으로 거의 모든 충돌 및 장애 시나리오가 복구될 수 있으며, 극도로 드물긴 하나 복구가 불가능한 상황에서는 손상이 즉시 감지됩니다.

Oracle 데이터베이스를 사용하는 대부분의 NetApp 고객은 스냅샷 기반 백업으로 마이그레이션한 후에 RMAN 및 기타 백업 제품의 사용을 중단합니다. SnapCenter를 통한 블록 레벨 복구를 수행하기 위해 RMAN을 사용할 수 있는 옵션이 여전히 있습니다. 하지만, 일별 기준으로 보면 RMAN, NetBackup 및 기타 제품은 월별 또는 분기별 아카이빙 복사본을 생성하기 위해 가끔씩만 사용됩니다.

어떤 고객은 실행을 선택합니다 dbv 정기적으로 기존 데이터베이스에 대한 무결성 검사를 수행합니다. 하지만 NetApp에서는 불필요한 I/O 로드가 생성되기 때문에 이 방식은 권장되지 않습니다. 위에서 설명한 바와 같이 데이터베이스에 이전에 문제가 발생하지 않았다면 의 가능성이 높습니다 dbv 문제를 감지하는 것은 거의 0에 가까우며, 이 유틸리티는 네트워크 및 스토리지 시스템에 매우 높은 순차 I/O 로드를 생성합니다. 알려진 Oracle 버그에 관한 노출 같은 손상이 존재한다고 판단할 근거가 있지 않는 한 실행할 이유는 없습니다 dbv.

백업 및 복구 기초

Oracle 데이터베이스 및 스냅샷 기반 백업

ONTAP에서 Oracle 데이터베이스 데이터 보호의 기반은 NetApp Snapshot 기술입니다.

키 값은 다음과 같습니다.

- * Simplicity. * 스냅샷은 특정 시점의 데이터 컨테이너 내용의 읽기 전용 복사본입니다.
- * 효율성. * 스냅샷은 생성 시점에 공간이 필요하지 않습니다. 공간은 데이터가 변경될 때만 사용됩니다.
- * 관리 효율성. * 스냅샷이 스토리지 OS의 기본 부분이기 때문에 스냅샷을 기반으로 하는 백업 전략은 구성 및 관리가 용이합니다. 스토리지 시스템의 전원이 켜져 있으면 백업을 생성할 준비가 된 것입니다.
- * 확장성. * 파일 및 LUN의 단일 컨테이너에 대해 최대 1024개의 백업을 유지할 수 있습니다. 복잡한 데이터 세트의 경우 일관된 단일 스냅샷 세트로 여러 데이터 컨테이너를 보호할 수 있습니다.
- 볼륨에 1024개의 스냅샷이 포함되어 있는지 여부에 관계없이 성능에 영향을 주지 않습니다.

많은 스토리지 공급업체가 스냅샷 기술을 제공하지만 ONTAP 내 스냅샷 기술은 고유한 특성을 가지고 있으며 엔터프라이즈 애플리케이션 및 데이터베이스 환경에 큰 이점을 제공합니다.

- 스냅샷 복사본은 기본 WAFL(Write Anywhere File Layout)의 일부입니다. 이러한 기술은 애드온 또는 외부 기술이 아닙니다. 따라서 스토리지 시스템이 백업 시스템이므로 관리가 간소화됩니다.
- 스냅샷 복사본은 성능에 영향을 미치지 않습니다. 단, 기본 스토리지 시스템이 가득 찬 스냅샷에 많은 데이터가 저장되는 경우와 같이 일부 예외 사례는 예외입니다.
- "정합성 보장 그룹"이라는 용어는 일관된 데이터 모음으로 관리되는 스토리지 객체 그룹을 지칭하는 데 주로 사용됩니다. 특정 ONTAP 볼륨의 스냅샷은 정합성 보장 그룹 백업을 구성합니다.

ONTAP 스냅샷은 또한 경쟁 기술보다 더 효과적으로 확장할 수 있습니다. 고객은 성능에 영향을 주지 않고 5개, 50개 또는 500개의 스냅샷을 저장할 수 있습니다. 볼륨에 현재 허용되는 최대 스냅샷 수는 1024개입니다. 추가 스냅샷 보존이 필요한 경우 스냅샷을 추가 볼륨에 단계적으로 적용할 수 있는 옵션이 있습니다.

그 결과, ONTAP에서 호스팅되는 데이터 세트를 보호하는 것은 간단하며 확장성이 뛰어납니다. 백업에는 데이터를 이동할 필요가 없으므로 네트워크 전송 속도, 많은 수의 테이프 드라이브 또는 디스크 스테이징 영역의 제한이 아니라 비즈니스 요구에 맞게 백업 전략을 조정할 수 있습니다.

스냅샷이 백업입니까?

스냅샷을 데이터 보호 전략으로 사용하는 것에 대해 자주 묻는 질문 중 하나는 "실제" 데이터와 스냅샷 데이터가 동일한 드라이브에 있다는 사실입니다. 이러한 드라이브가 손실되면 기본 데이터와 백업이 모두 손실됩니다.

이는 유효한 문제입니다. 로컬 스냅샷은 일상적인 백업 및 복구 요구에 사용되며 이러한 측면에서 스냅샷은 백업입니다. NetApp 환경의 모든 복구 시나리오 중 거의 99%가 가장 공격적인 RTO 요구 사항까지도 충족하기 위해 스냅샷에 의존합니다.

하지만 로컬 스냅샷만 사용할 수 있는 백업 전략이 되어서는 안 됩니다. 그렇기 때문에 NetApp에서는 스냅샷을 독립 드라이브 세트에 빠르고 효율적으로 복제할 수 있는 SnapMirror 및 SnapVault 복제 같은 기술을 제공합니다. 스냅샷과 스냅샷 복제를 갖춘 제대로 설계된 솔루션을 사용하면 분기별 아카이브로 테이프 사용을 최소화하거나 완전히 제거할 수 있습니다.

스냅샷 기반 백업

ONTAP Snapshot 복사본을 사용하여 데이터를 보호하는 방법에는 여러 가지가 있으며, 스냅샷은 복제, 재해 복구, 클론 복제를 포함한 다른 ONTAP 기능의 기반입니다. 스냅샷 기술에 대한 전체 설명은 이 문서의 범위를 벗어나지만 다음 섹션에서는 일반적인 개요를 제공합니다.

데이터 세트의 스냅샷을 생성하는 방법에는 두 가지가 있습니다.

- 충돌 시에도 정합성 보장 백업
- 애플리케이션 정합성이 보장되는 백업

데이터 세트의 장애 발생 시 정합성이 보장되는 백업은 단일 시점의 전체 데이터 세트 구조를 캡처하는 것을 의미합니다. 데이터 세트가 단일 NetApp FlexVol 볼륨에 저장된 경우 프로세스는 간단하며 언제든지 스냅샷을 생성할 수 있습니다. 데이터 세트가 여러 볼륨으로 확장되는 경우에는 정합성 보장 그룹(CG) 스냅샷을 생성해야 합니다. CG 스냅샷을 생성하는 옵션에는 NetApp SnapCenter 소프트웨어, 기본 ONTAP 일관성 그룹 기능, 사용자 유지보수 스크립트 등 여러 가지가 있습니다.

충돌 시에도 정합성 보장 백업은 백업 지점 복구가 충분할 때 주로 사용됩니다. 더 세부적인 복구가 필요한 경우 일반적으로 애플리케이션 정합성이 보장되는 백업이 필요합니다.

"애플리케이션 정합성 보장"에서 "정합성 보장"이라는 단어가 잘못된 표현인 경우가 많습니다. 예를 들어 Oracle 데이터베이스를 백업 모드로 설정하는 것을 애플리케이션 정합성 보장 백업이라고 하지만 데이터가 어떤 식으로든 일관되지 않거나 정지되지 않습니다. 백업 내내 데이터가 계속 변경됩니다. 반면 대부분의 MySQL 및 Microsoft SQL

Server 백업은 실제로 백업을 실행하기 전에 데이터를 정지합니다. VMware는 특정 파일의 일관성을 유지할 수도 있고 그렇지 않을 수도 있습니다.

정합성 보장 그룹

"정합성 보장 그룹"이란 스토리지 배열이 여러 스토리지 리소스를 단일 이미지로 관리하는 기능을 의미합니다. 예를 들어, 데이터베이스는 10개의 LUN으로 구성될 수 있습니다. 스토리지에서 이러한 10개의 LUN을 일관된 방식으로 백업, 복원 및 복제할 수 있어야 합니다. LUN의 이미지가 백업 시점에 일치하지 않으면 복구할 수 없습니다. 이러한 10개의 LUN을 복제하려면 모든 복제본이 서로 완벽하게 동기화되어야 합니다.

ONTAP에 대해 설명할 때는 "일관성 그룹"이라는 용어가 자주 사용되지 않습니다. 왜냐하면 일관성이 항상 ONTAP 내의 볼륨 및 애그리게이트 아키텍처의 기본 기능이기 때문입니다. 다른 많은 스토리지 시스템은 LUN 또는 파일 시스템을 개별 유닛으로 관리합니다. 그런 다음 데이터 보호를 위해 "정합성 보장 그룹"으로 구성할 수도 있지만 이 작업은 구성에 있어 추가 단계입니다.

ONTAP는 항상 일관된 로컬 및 복제된 데이터 이미지를 캡처할 수 있었습니다. ONTAP 시스템의 다양한 볼륨이 일반적으로 공식적으로 일관성 그룹으로 설명되어 있는 것은 아니지만, 그것이 바로 그러한 볼륨입니다. 해당 볼륨의 스냅샷은 일관성 그룹 이미지이고, 해당 스냅샷에 대한 복원은 일관성 그룹 복원이며, SnapMirror 및 SnapVault에서 모두 일관성 그룹 복제를 제공합니다.

정합성 보장 그룹 스냅샷

일관성 그룹 스냅샷(CG-스냅샷)은 기본 ONTAP 스냅샷 기술의 확장입니다. 표준 스냅샷 작업은 단일 볼륨 내에서 모든 데이터의 일관된 이미지를 생성하지만 여러 볼륨 및 여러 스토리지 시스템에 걸쳐 일관된 스냅샷 세트를 생성해야 하는 경우도 있습니다. 그 결과 하나의 개별 볼륨의 스냅샷과 동일한 방식으로 사용할 수 있는 스냅샷 세트가 생성됩니다. 로컬 데이터 복구에 사용하거나, 재해 복구를 위해 복제하거나, 일관된 단일 유닛으로 복제할 수 있습니다.

CG-스냅샷의 가장 큰 용도는 12개의 컨트롤러를 포함하여 약 1PB의 데이터베이스 환경을 위한 것입니다. 이 시스템에서 생성된 CG 스냅샷이 백업, 복구 및 클론 생성에 사용되었습니다.

대부분의 경우 데이터 세트가 여러 볼륨에 걸쳐 있고 쓰기 순서를 보존해야 하는 경우 선택한 관리 소프트웨어에서 CG 스냅샷이 자동으로 사용됩니다. 이러한 경우 CG-스냅샷의 기술적 세부 사항을 이해할 필요가 없습니다. 그러나 복잡한 데이터 보호 요구 사항이 데이터 보호 및 복제 프로세스를 세부적으로 제어해야 하는 경우도 있습니다. 몇 가지 옵션은 자동화 워크플로우 또는 맞춤형 스크립트를 사용하여 CG-스냅샷 API를 호출하는 것입니다. 최상의 옵션과 CG-스냅샷의 역할을 이해하려면 기술에 대한 자세한 설명이 필요합니다.

CG 스냅샷 세트를 생성하는 프로세스는 2단계로 구성됩니다.

1. 모든 타겟 볼륨에 쓰기 펜싱을 설정합니다.
2. 펜싱된 상태에서 해당 볼륨의 스냅샷을 생성합니다.

쓰기 펜싱은 연속적으로 설정됩니다. 즉, 펜싱 프로세스가 여러 볼륨에 설정되면 나중에 나타나는 볼륨에 계속 커밋되기 때문에 쓰기 I/O가 시퀀스의 첫 번째 볼륨에 고정됩니다. 이는 처음에 쓰기 순서를 보존해야 하는 요구 사항을 위반하는 것처럼 보일 수 있지만, 이는 호스트에서 비동기식으로 실행되며 다른 쓰기에 의존하지 않는 입출력에만 적용됩니다.

예를 들어, 데이터베이스가 많은 비동기식 데이터 파일 업데이트를 발행하고 운영 체제가 I/O를 재주문하고 자체 스케줄러 구성에 따라 업데이트를 완료할 수 있습니다. 애플리케이션 및 운영 체제에서 쓰기 순서를 유지하기 위한 요구 사항을 이미 해제했기 때문에 이러한 유형의 입출력 순서를 보장할 수 없습니다.

반대의 예로 대부분의 데이터베이스 로깅 작업은 동기적입니다. 입출력이 확인되고 이러한 쓰기 순서가 유지되어야 데이터베이스가 더 이상 로그 쓰기를 진행하지 않습니다. 로그 입출력이 펜싱된 볼륨에 도착하면 로그 입출력이 확인되지 않고 애플리케이션이 추가 쓰기를 차단합니다. 마찬가지로 파일 시스템 메타데이터 I/O는 일반적으로 동기식입니다. 예를 들어 파일 삭제 작업은 손실되지 않아야 합니다. xfs 파일 시스템이 있는 운영 체제에서 파일 및 xfs

파일 시스템 메타데이터를 업데이트한 입출력이 펜싱된 볼륨에 있는 해당 파일에 대한 참조를 제거하기 위해 삭제된 경우 파일 시스템 작업이 일시 중지됩니다. 따라서 CG 스냅샷 작업 중에 파일 시스템의 무결성이 보장됩니다.

대상 볼륨에 쓰기 펜싱이 설정된 후에는 스냅샷을 생성할 준비가 됩니다. 볼륨의 상태가 종속 쓰기 관점에서 고정되므로 스냅샷을 정확하게 동시에 생성할 필요가 없습니다. CG-스냅샷을 생성하는 애플리케이션의 결함을 방지하기 위해 초기 쓰기 펜싱에는 구성 가능한 시간 초과가 포함되어 있습니다. 이 시간 초과는 ONTAP가 자동으로 펜싱을 해제하고 정의된 초 후에 쓰기 처리를 재개합니다. 시간 제한 기간이 만료되기 전에 모든 스냅샷이 생성되면 생성된 스냅샷 세트는 유효한 정합성 보장 그룹입니다.

종속 쓰기 순서입니다

기술적 관점에서 정합성 보장 그룹의 핵심은 쓰기 순서, 특히 종속 쓰기 순서를 유지하는 것입니다. 예를 들어, 10개의 LUN에 쓰는 데이터베이스는 이들 모두에 동시에 쓰입니다. 많은 쓰기가 비동기적으로 실행되므로 쓰기 작업이 완료되는 순서는 중요하지 않으며 실제 완료 순서는 운영 체제 및 네트워크 동작에 따라 다릅니다.

데이터베이스에서 추가 쓰기를 진행하려면 디스크에 일부 쓰기 작업이 있어야 합니다. 이러한 중요한 쓰기 작업을 종속 쓰기라고 합니다. 이후의 쓰기 입출력은 디스크에 이러한 쓰기가 있는지에 따라 달라집니다. 이러한 10개 LUN의 모든 스냅샷, 복구 또는 복제는 종속 쓰기 순서가 보장되도록 해야 합니다. 파일 시스템 업데이트는 쓰기 순서 종속 쓰기의 또 다른 예입니다. 파일 시스템 변경 순서를 보존해야 합니다. 그렇지 않으면 전체 파일 시스템이 손상될 수 있습니다.

전략

스냅샷 기반 백업에는 다음과 같은 두 가지 기본 접근 방식이 있습니다.

- 충돌 시에도 정합성 보장 백업
- 스냅샷 보호 핫 백업

데이터베이스의 충돌 시에도 정합성 보장 백업은 데이터 파일, 재실행 로그, 제어 파일을 비롯한 전체 데이터베이스 구조를 단일 지점에서 캡처하는 것을 의미합니다. 데이터베이스를 단일 NetApp FlexVol 볼륨에 저장하면 프로세스가 단순해지며 언제든지 스냅샷을 생성할 수 있습니다. 데이터베이스가 여러 볼륨으로 확장되는 경우에는 일관성 그룹(CG) 스냅샷을 생성해야 합니다. CG 스냅샷을 생성하는 옵션에는 NetApp SnapCenter 소프트웨어, 기본 ONTAP 일관성 그룹 기능, 사용자 유지보수 스크립트 등 여러 가지가 있습니다.

스냅샷에서 충돌 시에도 정합성 보장 백업은 백업 지점 복구가 충분할 때 주로 사용됩니다. 경우에 따라 아카이브 로그를 적용할 수 있지만 더 세분화된 시점 복구가 필요한 경우에는 온라인 백업을 적용하는 것이 좋습니다.

스냅샷 기반 온라인 백업의 기본 절차는 다음과 같습니다.

1. 에 데이터베이스를 배치합니다 backup 모드를 선택합니다.
2. 데이터 파일을 호스팅하는 모든 볼륨의 스냅샷을 생성합니다.
3. Exit(종료) backup 모드를 선택합니다.
4. 명령을 실행합니다 alter system archive log current 로그 보관을 수행합니다.
5. 아카이브 로그를 호스팅하는 모든 볼륨의 스냅샷을 생성합니다.

이 절차를 따르면 백업 모드의 데이터 파일과 백업 모드 중에 생성된 주요 아카이브 로그가 포함된 스냅샷 세트가 만들어집니다. 데이터베이스를 복구하는 데에는 두 가지 요구사항이 있는데, 편의를 위해 제어 파일 같은 파일도 보호해야 하지만 데이터 파일과 아카이브 로그를 반드시 보호해야 합니다.

고객마다 전략은 다르겠지만 이 전략은 거의 모든 경우에 결국은 아래에 설명된 동일한 원칙에 기반을 두고 수립됩니다.

스냅샷 기반 복구

Oracle 데이터베이스를 위해 볼륨 레이아웃을 설계할 때 첫 번째 내려야 할 결정은 볼륨 기반 NetApp SnapRestore(VBSR) 기술을 사용할 것이냐입니다.

볼륨 기반 SnapRestore는 볼륨을 이전 시점으로 거의 즉시 되돌릴 수 있게 합니다. 볼륨의 모든 데이터를 되돌릴 수 있기 때문에 VBSR은 모든 사용 사례에는 적합하지 않을 수 있습니다. 예를 들어, 데이터 파일, 재실행 로그, 아카이브 로그를 비롯한 전체 데이터베이스가 단일 볼륨에 저장되고 이 볼륨이 VBSR을 통해 복원되는 경우 최신 아카이브 로그와 재실행 데이터가 삭제되기 때문에 데이터가 손실됩니다.

VBSR은 복원이 필요하지 않습니다. 대부분의 경우 파일을 기반으로 SFSR(Single File SnapRestore)을 사용하거나 스냅샷에서 액티브 파일 시스템으로 파일을 복사하여 데이터베이스를 복원할 수 있습니다.

VBSR은 데이터베이스가 대규모이거나 최대한 빨리 복구해야 할 경우에 적용하는 것이 좋으며 VBSR을 사용할 시 데이터 파일을 격리해야 합니다. NFS 환경에서는 다른 유형의 파일에 의해 손상되지 않은 전용 볼륨에 기존 데이터베이스의 데이터 파일을 저장해야 하며 SAN 환경에서는 전용 FlexVol 볼륨의 전용 LUN에 데이터 파일을 저장해야 합니다. Oracle 자동 스토리지 관리(ASM)와 같은 볼륨 관리자를 사용하는 경우 디스크 그룹도 데이터 파일 전용이어야 합니다.

이런 방식으로 데이터 파일을 격리하면 다른 파일 시스템을 손상시키지 않고 이전 상태로 되돌릴 수 있습니다.

스냅샷 예비 공간입니다

SAN 환경에 있는 Oracle 데이터의 각 볼륨에 대해 를 참조하십시오 `percent-snapshot-space` LUN 환경에서 스냅샷에 대한 공간을 예약하는 것은 유용하지 않으므로 0으로 설정해야 합니다. 부분 예약 공간이 100으로 설정된 경우 LUN이 있는 볼륨의 스냅샷은 전체 데이터의 100% 턴오버를 처리하기 위해 스냅샷 예약 공간을 제외하고 볼륨에서 충분한 여유 공간을 필요로 합니다. 부분 예약이 더 낮은 값으로 설정된 경우 이에 따라 더 적은 양의 여유 공간이 필요하지만 항상 스냅샷 예비 공간이 제외됩니다. 즉, LUN 환경에서 스냅샷 예약 공간이 낭비됩니다.

NFS 환경에는 다음 두 가지 옵션이 있습니다.

- 를 설정합니다 `percent-snapshot-space` 예상되는 스냅샷 공간 소비량을 기준으로 합니다.
- 를 설정합니다 `percent-snapshot-space` 활성 및 스냅샷 공간 소비를 총체적으로 제로화하고 관리합니다.

첫 번째 옵션으로 `percent-snapshot-space` 0이 아닌 값(일반적으로 약 20%)으로 설정됩니다. 그러면 이 공간이 사용자로부터 숨겨집니다. 하지만 이 값은 활용률의 한계를 생성하지 않습니다. 20%가 예약된 데이터베이스에서 턴오버가 30%인 경우 스냅샷 공간은 20% 예약이라는 경계를 넘어 확장할 수 있으며 미예약 공간을 점유할 수 있습니다.

예약을 20%와 같은 값으로 설정할 때 얻을 수 있는 가장 큰 이점은 일부 공간이 스냅샷에 항상 사용 가능한지 확인하는 것입니다. 예를 들어, 20%가 예약된 1TB 볼륨의 경우 데이터베이스 관리자(DBA)는 800GB의 데이터만 저장할 수 있을 것입니다. 이 구성은 스냅샷 소비를 위해 최소 200GB의 공간을 보장합니다.

시기 `percent-snapshot-space` 0으로 설정하면 볼륨의 모든 공간을 최종 사용자가 사용할 수 있어 가시성이 향상됩니다. DBA가 확인했을 때 스냅샷을 활용하는 볼륨이 1TB라면 이 1TB 공간이 액티브 데이터와 스냅샷 턴오버 간에 공유된다는 것을 알아야 합니다.

이 두 옵션 중 최종 사용자가 특별히 선호하는 것은 없습니다.

ONTAP 및 타사 스냅샷

Oracle Doc ID 604683.1은 타사 스냅샷 지원에 관련된 요구사항과 백업 및 복원 작업에 사용할 수 있는 여러 옵션을 설명합니다.

타사 공급업체는 회사의 스냅샷이 다음과 같은 요구 사항을 준수함을 보증해야 합니다.

- 스냅샷이 Oracle에서 권장하는 복원 및 복구 작업에 통합되어야 합니다.
- 스냅샷 지점에서 스냅샷의 데이터베이스 충돌이 일치해야 합니다.
- 쓰기 순서는 각 파일에 대해 스냅샷 내에 보존됩니다.

ONTAP 및 NetApp Oracle 관리 제품은 이러한 요구사항을 준수합니다.

SnapRestore을 사용한 신속한 **Oracle** 데이터베이스 복구

ONTAP의 빠른 스냅샷 데이터 복원은 NetApp SnapRestore 기술을 통해 수행됩니다.

중요 데이터 세트를 사용할 수 없으면 중요한 비즈니스 운영이 중단됩니다. 테이프는 작동 중지될 수 있으며 디스크 기반 백업에서 복원하는 경우 네트워크를 통해 전송되는 속도가 느려질 수 있습니다. SnapRestore은 데이터 세트를 거의 즉각적으로 복원하여 이러한 문제를 방지합니다. 페타바이트급 데이터베이스도 단 몇 분만에 완벽하게 복원할 수 있습니다.

SnapRestore 파일/LUN 기반과 볼륨 기반에는 두 가지 형식이 있습니다.

- 개별 파일 또는 LUN은 2TB LUN인지 4KB 파일인지에 관계없이 몇 초 이내에 복원할 수 있습니다.
- 파일 또는 LUN의 컨테이너는 10GB 또는 100TB의 데이터이든 몇 초 만에 복원할 수 있습니다.

"파일 또는 LUN 컨테이너"는 일반적으로 FlexVol 볼륨을 의미합니다. 예를 들어, 단일 볼륨에 LVM 디스크 그룹을 구성하는 10개의 LUN이 있거나 볼륨에 1000명의 사용자가 있는 NFS 홈 디렉토리를 저장할 수 있습니다. 각 개별 파일 또는 LUN에 대해 복구 작업을 실행하는 대신 전체 볼륨을 단일 작업으로 복구할 수 있습니다. 이 프로세스는 FlexGroup 또는 ONTAP 일관성 그룹과 같은 여러 볼륨이 포함된 스케일아웃 컨테이너에도 작동합니다.

SnapRestore이 빠르고 효율적으로 작동하는 이유는 기본적으로 특정 시점에 볼륨 콘텐츠에 대한 병렬 읽기 전용 뷰인 스냅샷의 특성 때문입니다. 활성 블록은 변경할 수 있는 실제 블록이지만 스냅샷은 스냅샷이 생성된 시점의 파일 및 LUN을 구성하는 블록 상태에 대한 읽기 전용 뷰입니다.

ONTAP에서는 스냅샷 데이터에 대한 읽기 전용 액세스만 허용하지만, SnapRestore를 사용하여 데이터를 다시 활성화할 수 있습니다. 스냅샷은 데이터의 읽기/쓰기 뷰로 다시 설정되며 데이터를 이전 상태로 되돌립니다. SnapRestore는 볼륨 또는 파일 레벨에서 작동할 수 있습니다. 이 기술은 본질적으로 같으며 몇 가지 사소한 행동 차이가 있습니다.

Volume SnapRestore를 참조하십시오

볼륨 기반 SnapRestore는 전체 데이터 볼륨을 이전 상태로 되돌립니다. 이 작업은 데이터를 이동할 필요가 없습니다. 즉, API 또는 CLI 작업을 처리하는 데 몇 초가 걸릴 수 있지만 복원 프로세스가 기본적으로 즉각적입니다. 1GB의 데이터를 복원하는 것은 1PB의 데이터를 복원하는 것보다 더 복잡하거나 시간이 많이 소요됩니다. 이 기능은 많은 엔터프라이즈 고객이 ONTAP 스토리지 시스템으로 마이그레이션하는 주된 이유입니다. 또한 가장 큰 데이터 세트에 대해 몇 초 단위의 RTO를 제공합니다.

볼륨 기반 SnapRestore의 한 가지 단점은 볼륨 내의 변경 사항이 시간 경과에 따라 누적된다는 사실에 의해 발생합니다. 따라서 각 스냅샷 및 활성 파일 데이터는 해당 시점까지 이어지는 변경 사항에 따라 달라집니다. 볼륨을 이전 상태로 되돌리면 데이터에 적용된 이후의 모든 변경 내용이 취소됩니다. 그러나 이는 이후에 생성된 스냅샷을 포함한다는 점은 그만큼 분명하지 않습니다. 이것이 항상 바람직한 것은 아닙니다.

예를 들어, 데이터 보존 SLA는 30일 야간 백업을 지정할 수 있습니다. 볼륨 SnapRestore를 사용하여 5일 전에 생성된 스냅샷으로 데이터 세트를 복구하면 지난 5일 동안 생성된 모든 스냅샷이 삭제되어 SLA를 위반하게 됩니다.

이 제한 사항을 해결하는 데 사용할 수 있는 여러 가지 옵션이 있습니다.

1. 전체 볼륨의 SnapRestore를 수행하는 것이 아니라 이전 스냅샷에서 데이터를 복사할 수 있습니다. 이 방법은 보다 작은 데이터 집합에 가장 적합합니다.
2. 스냅샷은 복구하지 않고 클론을 생성할 수 있습니다. 이 접근 방식의 제한 사항은 소스 스냅샷이 클론에 종속된다는 것입니다. 따라서 클론이 삭제되거나 독립 볼륨으로 분할되지 않는 한 삭제할 수 없습니다.
3. 파일 기반 SnapRestore 사용:

파일 SnapRestore

파일 기반 SnapRestore는 보다 세부적인 스냅샷 기반 복원 프로세스입니다. 전체 볼륨의 상태를 되돌리는 대신 개별 파일 또는 LUN의 상태를 되돌립니다. 스냅샷을 삭제할 필요가 없으며, 이 작업으로 이전 스냅샷에 대한 종속성이 생성되지 않습니다. 파일 또는 LUN을 활성 볼륨에서 즉시 사용할 수 있습니다.

파일 또는 LUN의 SnapRestore 복원 중에 데이터를 이동할 필요가 없습니다. 그러나 파일 또는 LUN의 기본 블록이 이제 스냅샷과 활성 볼륨 모두에 존재한다는 사실을 반영하려면 일부 내부 메타데이터를 업데이트해야 합니다. 성능에는 영향을 미치지 않겠지만 이 프로세스는 완료될 때까지 스냅샷 생성을 차단합니다. 처리 속도는 복원된 파일의 총 크기에 따라 약 5GBps(시간당 18TB)입니다.

Oracle 데이터베이스 온라인 백업

백업 모드에서 Oracle 데이터베이스를 보호하고 복구하려면 두 세트의 데이터가 필요합니다. 이것이 유일한 Oracle 백업 옵션은 아니지만 가장 일반적입니다.

- 백업 모드의 데이터 파일 스냅샷
- 데이터 파일이 백업 모드일 때 생성된 아카이브 로그입니다

커밋된 모든 트랜잭션을 포함하여 완전한 복구가 필요한 경우 세 번째 항목이 필요합니다.

- 현재 redo 로그 집합입니다

온라인 백업의 복구를 유도하는 방법에는 여러 가지가 있습니다. 많은 고객은 ONTAP CLI를 사용한 다음 Oracle RMAN 또는 sqlplus를 사용하여 복구를 완료함으로써 스냅샷을 복구합니다. 이러한 현상은 데이터베이스 복구 가능성과 빈도가 매우 낮고 숙련된 DBA가 복구 절차를 처리하는 대규모 운영 환경에서 특히 흔합니다. NetApp SnapCenter와 같은 솔루션에는 완벽한 자동화를 위해 명령줄 및 그래픽 인터페이스 모두를 지원하는 Oracle 플러그인이 포함되어 있습니다.

일부 대규모 고객은 예약된 스냅샷을 준비하는 과정에서 특정 시간에 데이터베이스를 백업 모드로 전환하도록 호스트에 기본 스크립트를 구성하여 보다 간단한 접근 방식을 취했습니다. 예를 들어, 명령을 예약합니다 alter database begin backup 23:58에 alter database end backup 00:02에 스냅샷을 예약한 다음 자정에 스토리지 시스템에서 직접 스냅샷을 예약합니다. 그 결과 외부 소프트웨어 또는 라이선스가 필요 없는 간단하고 확장성이 뛰어난 백업 전략을 구축할 수 있습니다.

데이터 레이아웃

가장 간단한 레이아웃은 데이터 파일을 하나 이상의 전용 볼륨으로 분리하는 것입니다. 다른 파일 형식으로 오염되지 않아야 합니다. 이는 중요한 재실행 로그, 제어 파일 또는 아카이브 로그를 삭제하지 않고 SnapRestore 작업을 통해 데이터 파일 볼륨을 신속하게 복원할 수 있도록 보장하기 위한 것입니다.

SAN은 전용 볼륨 내의 데이터 파일 격리에 대해서도 유사한 요구사항을 가지고 있습니다. Microsoft Windows 같은 운영 체제에서 단일 볼륨에는 각각 NTFS 파일 시스템이 포함된 여러 데이터 파일 LUN이 포함될 수 있습니다. 다른

운영 체제에는 일반적으로 논리적 볼륨 관리자가 있습니다. 예를 들어, Oracle ASM을 사용할 경우 가장 간단한 옵션은 ASM 디스크 그룹의 LUN을 하나의 볼륨으로 백업 및 복원할 수 있는 단일 볼륨으로 제한하는 것입니다. 성능 또는 용량 관리를 위해 추가 볼륨이 필요한 경우 새 볼륨에 추가 디스크 그룹을 생성하면 관리가 더 간단해집니다.

이러한 지침을 따를 경우 정합성 보장 그룹 스냅샷을 수행할 필요 없이 스토리지 시스템에서 스냅샷을 직접 예약할 수 있습니다. Oracle 백업에는 데이터 파일을 동시에 백업할 필요가 없기 때문입니다. 온라인 백업 절차는 몇 시간 내에 테이프를 천천히 스트리밍될 때 데이터 파일을 계속해서 업데이트할 수 있도록 설계되었습니다.

여러 볼륨에 분산된 ASM 디스크 그룹을 사용하는 것과 같은 상황에서는 문제가 발생합니다. 이 경우 ASM 메타데이터가 모든 구성 볼륨에서 일관되도록 CG-스냅샷을 수행해야 합니다.

- 주의: * ASM을 확인합니다 `spfile` 및 `passwd` 파일이 데이터 파일을 호스팅하는 디스크 그룹에 없습니다. 이로 인해 데이터 파일과 데이터 파일만 선택적으로 복원할 수 없습니다.

로컬 복구 절차 - NFS

이 절차는 수동으로 또는 SnapCenter와 같은 응용 프로그램을 통해 실행할 수 있습니다. 기본 절차는 다음과 같습니다.

1. 데이터베이스를 종료합니다.
2. 원하는 복원 지점 바로 전에 데이터 파일 볼륨을 스냅샷으로 복구합니다.
3. 원하는 지점으로 아카이브 로그를 재생합니다.
4. 전체 복구가 필요한 경우 현재 redo 로그를 재생합니다.

이 절차에서는 활성 파일 시스템에 원하는 아카이브 로그가 여전히 존재한다고 가정합니다. 그렇지 않은 경우 아카이브 로그를 복원해야 합니다. 그렇지 않으면 RMAN/sqlplus를 스냅샷 디렉토리의 데이터로 리디렉션할 수 있습니다.

또한 데이터베이스의 규모가 작은 경우 최종 사용자가 에서 직접 데이터 파일을 복구할 수 있습니다. `.snapshot` 자동화 툴 또는 스토리지 관리자의 도움 없이 디렉토리를 실행하여 `snapprestore` 명령.

로컬 복구 절차 - SAN

이 절차는 수동으로 또는 SnapCenter와 같은 응용 프로그램을 통해 실행할 수 있습니다. 기본 절차는 다음과 같습니다.

1. 데이터베이스를 종료합니다.
2. 데이터 파일을 호스팅하는 디스크 그룹을 중지합니다. 절차는 선택한 논리적 볼륨 관리자에 따라 다릅니다. ASM을 사용할 경우 이 프로세스에서는 디스크 그룹을 마운트 해제해야 합니다. Linux에서는 파일 시스템을 마운트 해제하고 논리적 볼륨 및 볼륨 그룹을 비활성화해야 합니다. 목표는 복구할 타겟 볼륨 그룹의 모든 업데이트를 중지하는 것입니다.
3. 원하는 복원 지점 바로 전에 데이터 파일 디스크 그룹을 스냅샷으로 복원합니다.
4. 새로 복구된 디스크 그룹을 다시 활성화합니다.
5. 원하는 지점으로 아카이브 로그를 재생합니다.
6. 전체 복구가 필요한 경우 모든 재실행 로그를 재생합니다.

이 절차에서는 활성 파일 시스템에 원하는 아카이브 로그가 여전히 존재한다고 가정합니다. 그렇지 않은 경우 아카이브 로그 LUN을 오프라인으로 전환하고 복원을 수행하여 아카이브 로그를 복원해야 합니다. 아카이브 로그를 전용 볼륨으로 분할하는 것이 유용한 예이기도 합니다. 아카이브 로그가 재실행 로그와 볼륨 그룹을 공유하는 경우 전체 LUN 세트를 복원하기 전에 재실행 로그를 다른 위치에 복사해야 합니다. 이 단계는 최종 기록된 트랜잭션의 손실을 방지합니다.

Oracle Database Storage Snapshot Optimized Backups의 약어입니다

데이터베이스를 핫 백업 모드로 설정할 필요가 없기 때문에 Oracle 12c가 출시되었을 때 스냅샷 기반 백업 및 복구가 더욱 간편해졌습니다. 그 결과 스토리지 시스템에서 직접 스냅샷 기반 백업을 예약하고 전체 또는 시점 복구를 수행하는 기능을 유지할 수 있습니다.

핫 백업 복구 절차는 DBA에게 더 익숙하지만 데이터베이스가 핫 백업 모드인 동안 생성되지 않은 스냅샷을 사용할 수 있는 것은 오래되었습니다. Oracle 10g 및 11g를 복구하는 동안 데이터베이스의 일관성을 유지하기 위해 추가 수동 단계가 필요했습니다. Oracle 12c를 사용하면 sqlplus 및 rman 핫 백업 모드에 있지 않은 데이터 파일 백업에서 아카이브 로그를 재생하는 추가 로직을 포함합니다.

앞에서 설명한 것처럼 스냅샷 기반 핫 백업을 복구하려면 두 가지 데이터 세트가 필요합니다.

- 백업 모드에서 생성된 데이터 파일의 스냅샷입니다
- 데이터 파일이 핫 백업 모드일 때 생성되는 아카이브 로그

복구 중에 데이터베이스는 데이터 파일에서 메타데이터를 읽어 복구에 필요한 아카이브 로그를 선택합니다.

스토리지 스냅샷으로 최적화된 복구에서는 동일한 결과를 얻기 위해 약간 다른 데이터 세트가 필요합니다.

- 데이터 파일의 스냅샷과 스냅샷이 생성된 시간을 식별하는 방법입니다
- 최신 데이터 파일 체크포인트 시점부터 스냅샷의 정확한 시간까지 로그를 아카이빙합니다

복구 중에 데이터베이스는 데이터 파일에서 메타데이터를 읽어 필요한 초기 아카이브 로그를 식별합니다. 전체 또는 특정 시점 복구를 수행할 수 있습니다. 시점 복구를 수행할 때는 데이터 파일의 스냅샷 시간을 알아야 합니다. 지정된 복구 지점은 스냅샷 생성 시간 이후여야 합니다. NetApp에서는 클럭 변동을 고려하여 스냅샷 시간에 최소 몇 분을 추가하는 것이 좋습니다.

자세한 내용은 Oracle 12c 설명서의 다양한 릴리스에서 제공되는 "Recovery using Storage Snapshot Optimization(스토리지 스냅샷 최적화를 사용한 복구)" 항목에 대한 Oracle 설명서를 참조하십시오. 또한 Oracle 타사 스냅샷 지원에 대해서는 Oracle Document ID Doc ID 604683.1을 참조하십시오.

데이터 레이아웃

가장 간단한 레이아웃은 데이터 파일을 하나 이상의 전용 볼륨으로 분리하는 것입니다. 다른 파일 형식으로 오염되지 않아야 합니다. 이는 중요한 redo 로그, 제어 파일 또는 아카이브 로그를 삭제하지 않고 SnapRestore 작업을 통해 데이터 파일 볼륨을 신속하게 복원할 수 있도록 하기 위한 것입니다.

SAN은 전용 볼륨 내의 데이터 파일 격리에 대해서도 유사한 요구사항을 가지고 있습니다. Microsoft Windows 같은 운영 체제에서 단일 볼륨에는 각각 NTFS 파일 시스템이 포함된 여러 데이터 파일 LUN이 포함될 수 있습니다. 다른 운영 체제에서는 일반적으로 논리적 볼륨 관리자도 있습니다. 예를 들어, Oracle ASM을 사용할 경우 가장 간단한 옵션은 디스크 그룹을 하나의 볼륨으로 백업 및 복원할 수 있는 단일 볼륨으로 제한하는 것입니다. 성능 또는 용량 관리를 위해 추가 볼륨이 필요한 경우 새 볼륨에 추가 디스크 그룹을 생성하면 관리가 더 쉬워집니다.

이러한 지침을 따를 경우 정합성 보장 그룹 스냅샷을 수행할 필요 없이 ONTAP에서 스냅샷을 직접 예약할 수 있습니다. 이유는 스냅샷 최적화 백업에는 데이터 파일을 동시에 백업할 필요가 없기 때문입니다.

여러 볼륨에 분산된 ASM 디스크 그룹과 같은 상황에서는 문제가 발생합니다. 이 경우 ASM 메타데이터가 모든 구성 볼륨에서 일관되도록 CG-스냅샷을 수행해야 합니다.

[참고] ASM spfile 및 passwd 파일이 데이터 파일을 호스팅하는 디스크 그룹에 있는지 확인합니다. 이로 인해 데이터 파일과 데이터 파일만 선택적으로 복원할 수 없습니다.

로컬 복구 절차 - NFS

이 절차는 수동으로 또는 SnapCenter와 같은 응용 프로그램을 통해 실행할 수 있습니다. 기본 절차는 다음과 같습니다.

1. 데이터베이스를 종료합니다.
2. 원하는 복원 지점 바로 전에 데이터 파일 볼륨을 스냅샷으로 복구합니다.
3. 원하는 지점으로 아카이브 로그를 재생합니다.

이 절차에서는 활성 파일 시스템에 원하는 아카이브 로그가 여전히 존재한다고 가정합니다. 그렇지 않은 경우, 또는 아카이브 로그를 복원해야 합니다 `rman` 또는 `sqlplus` 의 데이터로 이동할 수 있습니다 `.snapshot` 디렉토리.

또한 데이터베이스의 규모가 작은 경우 최종 사용자가 에서 직접 데이터 파일을 복구할 수 있습니다 `.snapshot` 디렉토리에는 자동화 툴 또는 스토리지 관리자의 도움 없이 `SnapRestore` 명령을 실행할 수 있습니다.

로컬 복구 절차 - SAN

이 절차는 수동으로 또는 SnapCenter와 같은 응용 프로그램을 통해 실행할 수 있습니다. 기본 절차는 다음과 같습니다.

1. 데이터베이스를 종료합니다.
2. 데이터 파일을 호스팅하는 디스크 그룹을 중지합니다. 절차는 선택한 논리적 볼륨 관리자에 따라 다릅니다. ASM을 사용할 경우 이 프로세스에서는 디스크 그룹을 마운트 해제해야 합니다. Linux에서는 파일 시스템을 마운트 해제해야 하며 논리적 볼륨 및 볼륨 그룹이 비활성화됩니다. 목표는 복구할 타겟 볼륨 그룹의 모든 업데이트를 중지하는 것입니다.
3. 원하는 복원 지점 바로 전에 데이터 파일 디스크 그룹을 스냅샷으로 복원합니다.
4. 새로 복구된 디스크 그룹을 다시 활성화합니다.
5. 원하는 지점으로 아카이브 로그를 재생합니다.

이 절차에서는 활성 파일 시스템에 원하는 아카이브 로그가 여전히 존재한다고 가정합니다. 그렇지 않은 경우 아카이브 로그 LUN을 오프라인으로 전환하고 복원을 수행하여 아카이브 로그를 복원해야 합니다. 아카이브 로그를 전용 볼륨으로 분할하는 것이 유용한 예이기도 합니다. 아카이브 로그가 재실행 로그와 볼륨 그룹을 공유하는 경우, 최종 기록된 트랜잭션이 손실되지 않도록 전체 LUN 세트를 복원하기 전에 재실행 로그를 다른 곳에 복사해야 합니다.

전체 복구 예

데이터 파일이 손상되었거나 제거되었으며 전체 복구가 필요한 것으로 가정합니다. 이렇게 하는 절차는 다음과 같습니다.

```

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

```

특정 시점 복구 예

전체 복구 절차는 단일 명령입니다. recover automatic.

시점 복구가 필요한 경우 스냅샷의 타임스탬프를 알고 있어야 하며 다음과 같이 식별할 수 있습니다.

```

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

```

스냅샷 생성 시간은 3월 9일 및 10:10:06으로 표시됩니다. 안전을 위해 스냅샷 시간에 1분이 추가됩니다.

```

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

```

이제 복구가 시작됩니다. 또한 스냅샷 시간을 10:11:00, 기록된 시간 1분 후 가능한 클럭 편차를 계산하고 목표 복구 시간을 10:44로 지정했습니다. 그런 다음 sqlplus는 원하는 복구 시간인 10:44에 도달하는 데 필요한 아카이브 로그를 요청합니다.

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



를 사용하여 스냅샷을 사용하여 데이터베이스 복구를 완료합니다 recover automatic 명령에는 특정 라이선스가 필요하지 않지만 를 사용하여 시점 복구가 필요합니다 snapshot time Oracle Advanced Compression 라이선스가 필요합니다.

Oracle 데이터베이스 관리 및 자동화 툴

Oracle 데이터베이스 환경에서 ONTAP의 기본적인 가치는 즉각적인 스냅샷 복사본, 간단한 SnapMirror 복제, 효율적인 FlexClone 볼륨 생성 등의 핵심 ONTAP 기술에서 나오는 것입니다.

이러한 핵심 기능을 ONTAP에서 직접 간단히 구성하여 요구사항을 충족하는 경우도 있지만, 더 복잡한 요구사항에는 오케스트레이션 계층이 필요합니다.

SnapCenter

SnapCenter은 NetApp의 대표적인 데이터 보호 제품입니다. 매우 낮은 수준에서 SnapManager 제품은 데이터베이스 백업을 수행하는 방법이라는 측면에서 볼 때 NetApp 제품과 비슷하지만, 처음부터 NetApp 스토리지 시스템의 데이터 보호 관리에 대한 단일 창 방식을 제공하도록 제작되었습니다.

SnapCenter에는 스냅샷 기반 백업 및 복원, SnapMirror 및 SnapVault 복제와 같은 기본 기능과 대기업에서 규모에 따라 운영하는 데 필요한 기타 기능이 포함되어 있습니다. 이러한 고급 기능에는 확장된 역할 기반 액세스 제어(RBAC) 기능, 타사 오케스트레이션 제품과 통합하기 위한 RESTful API, 데이터베이스 호스트에서 SnapCenter 플러그인의 무중단 중앙 관리, 클라우드급 환경에 맞게 설계된 사용자 인터페이스가 포함됩니다.

휴식

ONTAP에는 풍부한 RESTful API 세트도 포함되어 있습니다. 따라서 타사 공급업체에서는 ONTAP과 긴밀히 통합되는 데이터 보호 및 기타 관리 애플리케이션을 구축할 수 있습니다. 또한 RESTful API는 자체 자동화 워크플로우와 유틸리티를 생성하려는 고객이 손쉽게 사용할 수 있습니다.

Oracle 재해 복구

ONTAP을 통한 Oracle 데이터베이스 재해 복구

재해 복구는 스토리지 시스템 또는 전체 사이트를 파괴하는 화재와 같이 심각한 사고가 발생한 후 데이터 서비스를 복원하는 것을 의미합니다.



이 문서는 이전에 게시된 기술 보고서_TR-4591: Oracle Data Protection_and_TR-4592: MetroCluster 기반 Oracle_을(를) 대체합니다

재해 복구는 물론 많은 고객이 미러링된 복제본을 매시간마다 업데이트하는 방식으로 SnapMirror를 사용하여 데이터를 간단히 복제하여 수행할 수 있습니다.

대부분의 고객이 DR에는 데이터의 원격 복사본을 소유하는 것 이상의 것이 필요하며, 해당 데이터를 빠르게 사용할 수 있는 기능이 필요합니다. NetApp은 이러한 요구사항을 해결하는 두 가지 기술, 즉 MetroCluster와 SnapMirror Active Sync를 제공합니다

MetroCluster는 낮은 수준의 동기식 미러링 스토리지와 수많은 추가 기능을 포함하는 하드웨어 구성에서 ONTAP를 가리킵니다. MetroCluster와 같은 통합 솔루션은 오늘날의 복잡한 스케일아웃 데이터베이스, 애플리케이션 및 가상화 인프라를 단순화합니다. 여러 외부 데이터 보호 제품 및 전략을 하나의 단순한 중앙 스토리지 시스템으로 대체합니다. 또한 단일 클러스터 스토리지 시스템 내에서 통합 백업, 복구, 재해 복구 및 고가용성(HA)을 제공합니다.

SnapMirror 활성 동기화는 SnapMirror Synchronous에 기반을 둡니다. MetroCluster를 사용할 경우 각 ONTAP 컨트롤러는 드라이브 데이터를 원격 위치로 복제하는 작업을 담당합니다. SnapMirror가 활성 동기화에서는 기본적으로 LUN 데이터의 독립적인 복사본을 유지하지만 해당 LUN의 단일 인스턴스를 제공하기 위해 협력하는 두 개의 ONTAP 시스템이 있습니다. 호스트 관점에서 보면 단일 LUN 엔티티입니다.

SnapMirror Active Sync와 MetroCluster는 내부적으로 매우 다르게 작동하지만 호스트에 대한 결과는 매우 유사합니다. 주요 차이점은 세분성입니다. 동기적으로 복제할 워크로드만 선택하는 경우 SnapMirror 액티브 동기화가 더 나은 옵션입니다. 전체 환경이나 데이터 센터를 복제해야 하는 경우 MetroCluster가 더 나은 옵션입니다. 또한 SnapMirror 액티브 동기화는 현재 SAN, NFS, SMB를 포함한 멀티 프로토콜 MetroCluster이 있는 동안 SAN 전용입니다.

MetroCluster

MetroCluster 물리적 아키텍처 및 Oracle 데이터베이스

MetroCluster 환경에서 Oracle 데이터베이스가 작동하는 방식을 이해하려면 MetroCluster 시스템의 물리적 설계에 대해 몇 가지 설명이 필요합니다.



이 문서는 이전에 게시된 기술 보고서_TR-4592: MetroCluster 기반 Oracle._을(를) 대체합니다

MetroCluster는 3가지 구성으로 사용할 수 있습니다

- IP 연결이 포함된 HA 쌍
- FC 연결이 포함된 HA 쌍
- FC 연결이 포함된 단일 컨트롤러

[참고] '접속'이라는 용어는 사이트 간 복제에 사용되는 클러스터 접속을 의미합니다. 호스트 프로토콜을 참조하지 않습니다. 모든 호스트측 프로토콜은 클러스터 간 통신에 사용되는 연결 유형에 관계없이 MetroCluster 구성에서 평소와 같이 지원됩니다.

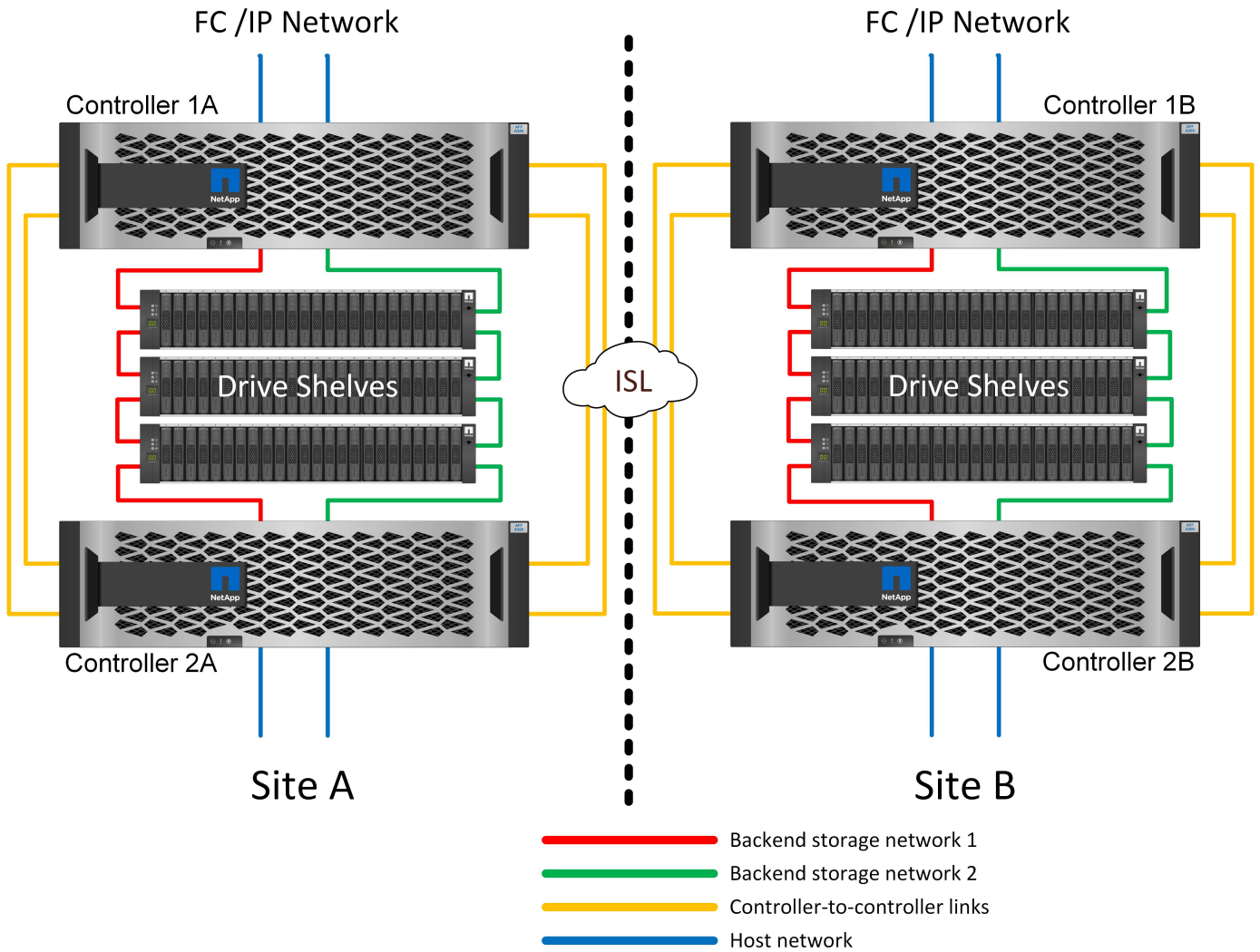
MetroCluster IP를 선택합니다

HA 쌍 MetroCluster IP 구성은 사이트당 2~4개의 노드를 사용합니다. 이 구성 옵션은 2노드 옵션에 비해 복잡성과 비용을 증가시키지만, 내부 중복이라는 중요한 이점을 제공합니다. 컨트롤러 장애가 간단하더라도 WAN을 통한 데이터 액세스가 필요하지 않습니다. 데이터 액세스는 대체 로컬 컨트롤러를 통해 로컬에 유지됩니다.

대부분의 고객은 인프라 요구 사항이 더 간단하기 때문에 IP 연결을 선택하고 있습니다. 과거에는 다크 파이버 및 FC 스위치를 사용하여 고속 사이트 간 연결을 제공하기가 일반적으로 더 쉬웠지만, 오늘날의 고속, 짧은 지연 시간 IP 회로는 보다 쉽게 사용할 수 있었습니다.

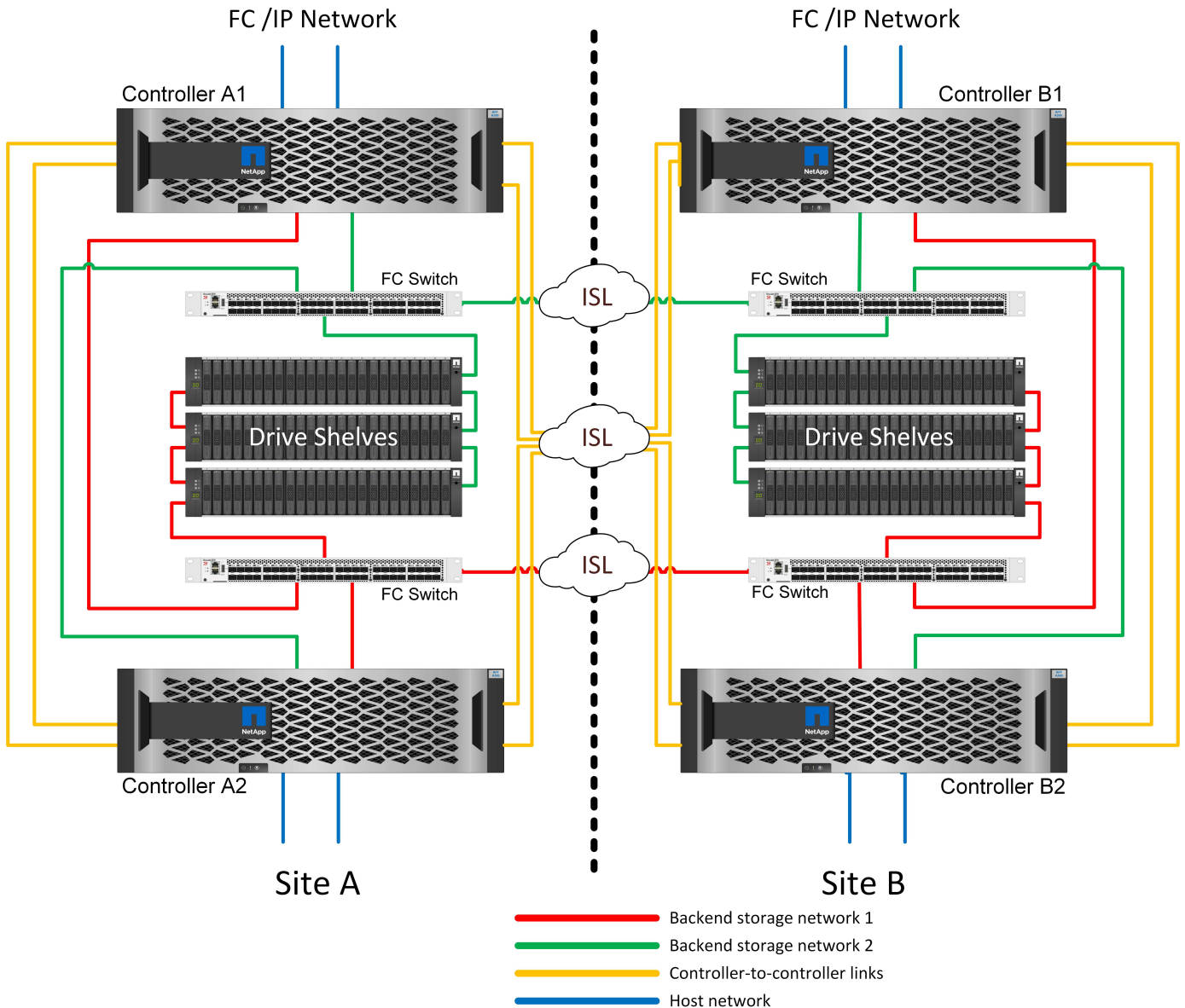
또한 사이트 간 연결만 컨트롤러를 위한 것이므로 아키텍처가 더욱 단순합니다. FC SAN 연결 MetroCluster에서 컨트롤러는 반대쪽 사이트의 드라이브에 직접 기록하므로 SAN 연결, 스위치 및 브리지가 추가로 필요합니다. 반면, IP 구성의 컨트롤러는 컨트롤러를 통해 반대쪽 드라이브에 씁니다.

자세한 내용은 공식 ONTAP 설명서 및 [를 참조하십시오 "MetroCluster IP 솔루션 아키텍처 및 설계"](#).



HA-쌍 FC SAN 연결 MetroCluster

HA 쌍 MetroCluster FC 구성은 사이트당 2개 또는 4개의 노드를 사용합니다. 이 구성 옵션은 2노드 옵션에 비해 복잡성과 비용을 증가시키지만, 내부 중복이라는 중요한 이점을 제공합니다. 컨트롤러 장애가 간단하더라도 WAN을 통한 데이터 액세스가 필요하지 않습니다. 데이터 액세스는 대체 로컬 컨트롤러를 통해 로컬에 유지됩니다.

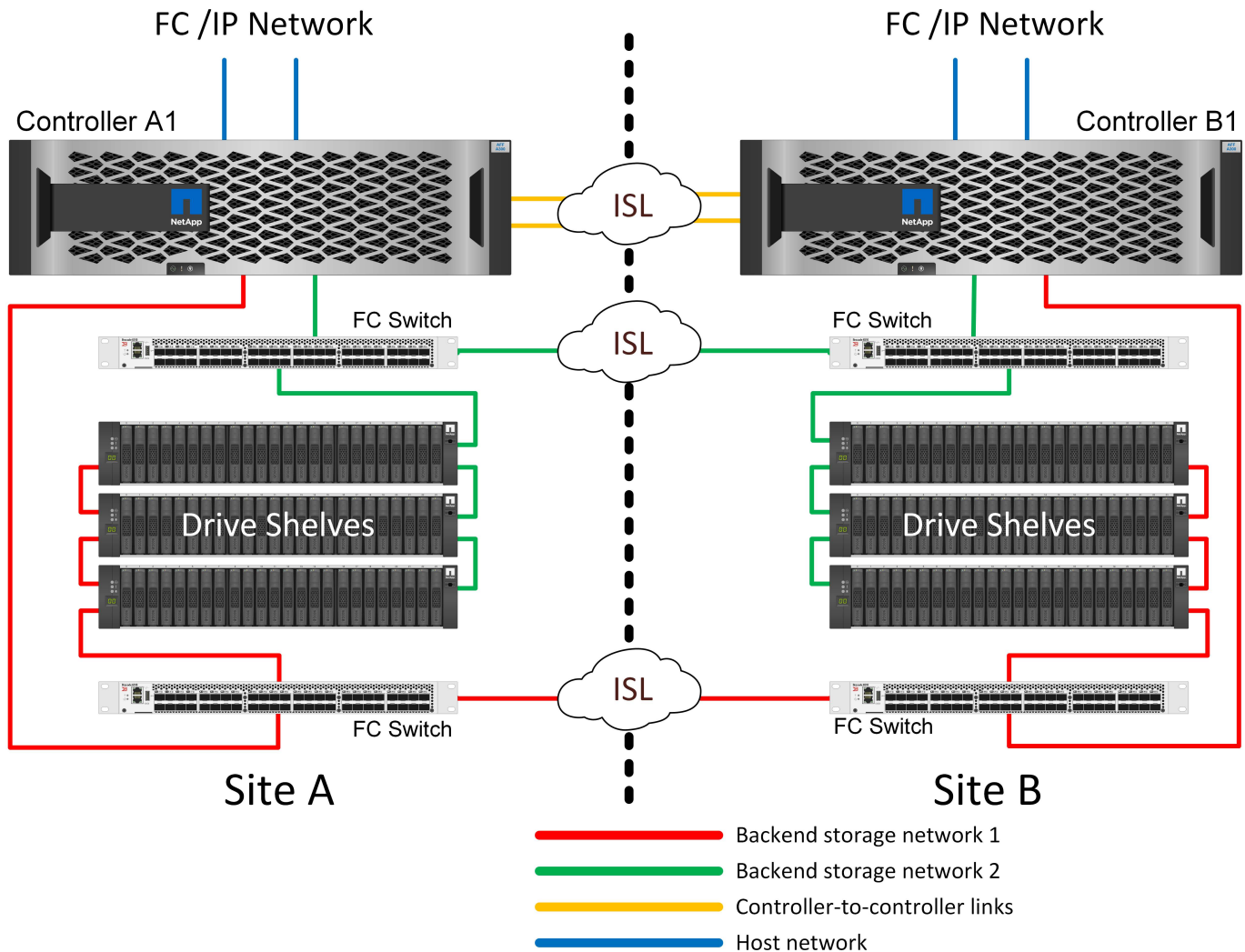


일부 멀티사이트 인프라는 액티브-액티브 운영을 위해 설계되지 않았지만 운영 사이트 및 재해 복구 사이트로 더 많이 사용됩니다. 이 상황에서 HA 쌍 MetroCluster 옵션이 일반적으로 다음과 같은 이유로 더 권장됩니다.

- 2노드 MetroCluster 클러스터는 HA 시스템이지만, 컨트롤러의 예상치 못한 장애나 계획된 유지 관리를 위해서는 반대쪽 사이트에서 데이터 서비스를 온라인으로 전환해야 합니다. 사이트 간 네트워크 연결이 필요한 대역폭을 지원할 수 없는 경우 성능이 영향을 받습니다. 유일한 옵션은 다양한 호스트 OS 및 관련 서비스를 대체 사이트로 페일오버하는 것입니다. HA Pair MetroCluster 클러스터는 동일한 사이트 내에서 단순한 페일오버가 발생하기 때문에 이 문제가 해소됩니다.
- 일부 네트워크 토폴로지는 사이트 간 액세스용으로 설계되지 않은 대신 서로 다른 서브넷이나 격리된 FC SAN을 사용합니다. 이런 경우 2노드 MetroCluster 클러스터는 다른 사이트의 서버에 데이터를 제공할 수 없기 때문에 더 이상 HA 시스템으로 작동하지 않습니다. 완벽한 이중화를 제공하려면 HA Pair MetroCluster 옵션이 필요합니다.
- 2개 사이트 인프라를 고가용성 단일 인프라로 간주하는 경우 2노드 MetroCluster 구성이 적합합니다. 하지만 사이트 장애 후 시스템이 오랫동안 작동해야 하는 경우에는 단일 사이트 내에서 HA를 계속 제공하기 때문에 HA 2노드가 선호됩니다.

2노드 FC SAN 연결 MetroCluster

2노드 MetroCluster 구성은 사이트당 하나의 노드만 사용합니다. 이 설계는 구성과 유지 관리가 필요한 구성 요소가 적기 때문에 HA 쌍 옵션보다 단순합니다. 또한 케이블 연결 및 FC 스위칭에 대한 인프라 요구도 줄었습니다. 마지막으로 비용을 절감할 수 있습니다.



이 설계의 분명한 영향은 단일 사이트에서 컨트롤러 장애가 발생하면 반대쪽 사이트에서 데이터를 사용할 수 있다는 것입니다. 이러한 제한이 반드시 문제가 되는 것은 아닙니다. 많은 기업은 기본적으로 단일 인프라로 작동하는 지연 시간이 짧은 확장된 고속 네트워크를 통해 멀티사이트 데이터 센터를 운영하고 있습니다. 이 경우 MetroCluster의 2노드 버전을 사용하는 것이 좋습니다. 현재 여러 서비스 공급자가 두 노드 시스템을 페타바이트 규모로 사용하고 있습니다.

MetroCluster 복원력 기능

MetroCluster 솔루션에는 단일 장애 지점이 없습니다.

- 각 컨트롤러에는 로컬 사이트의 드라이브 셸프에 대한 2개의 독립적 경로가 있습니다.
- 각 컨트롤러에는 원격 사이트의 드라이브 셸프에 대한 두 개의 독립적 경로가 있습니다.
- 각 컨트롤러에는 반대쪽 사이트에 있는 컨트롤러에 대한 독립적인 경로가 2개 있습니다.
- HA 쌍 구성에서 각 컨트롤러에는 로컬 파트너에 대한 두 가지 경로가 있습니다.

요약하면, MetroCluster의 데이터 제공 기능에 영향을 주지 않으면서 구성의 모든 구성 요소를 제거할 수 있습니다. 두 옵션 간의 복원력에서 유일한 차이점은 HA 쌍 버전이 사이트 장애 발생 후 전체 HA 스토리지 시스템이라는 점입니다.

MetroCluster 논리적 아키텍처와 Oracle 데이터베이스

MetroCluster 환경에서 Oracle 데이터베이스가 작동하는 방식을 이해하려면 MetroCluster 시스템의 논리적 기능에 대한 몇 가지 설명이 필요합니다.

사이트 장애 방지: NVRAM 및 MetroCluster

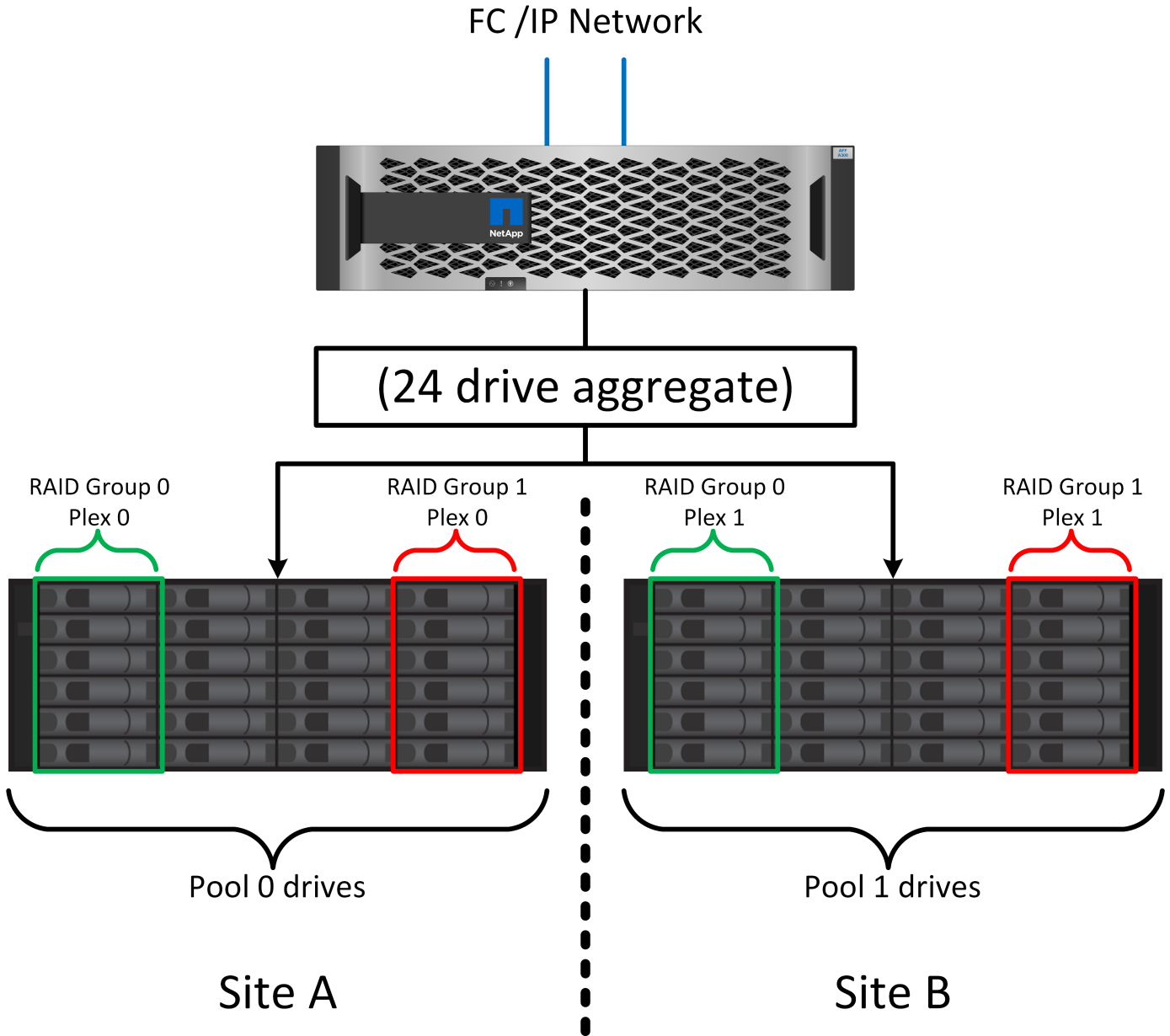
MetroCluster는 NVRAM 데이터 보호를 다음과 같은 방식으로 확장합니다.

- 2노드 구성에서는 ISL(Inter-Switch Link)을 사용하여 NVRAM 데이터를 원격 파트너에게 복제합니다.
- HA 쌍 구성에서는 NVRAM 데이터가 로컬 파트너와 원격 파트너 모두에 복제됩니다.
- 쓰기가 모든 파트너에게 복제될 때까지 확인되지 않습니다. 이 아키텍처는 NVRAM 데이터를 원격 파트너에게 복제하여 전송 중인 I/O를 사이트 장애로부터 보호합니다. 이 프로세스는 드라이브 수준 데이터 복제와 관련되지 않습니다. 애그리게이트를 소유한 컨트롤러는 애그리게이트의 두 플렉스에 쓰기를 통해 데이터 복제를 담당하지만, 사이트 손실 시 전송 중인 I/O 손실에 대한 보호가 여전히 필요합니다. 복제된 NVRAM 데이터는 파트너 컨트롤러가 장애가 발생한 컨트롤러를 인수해야 하는 경우에만 사용됩니다.

사이트 및 헬프 장애 보호: SyncMirror 및 플렉스

SyncMirror는 RAID DP 또는 RAID-TEC를 향상하지만 대체하지는 않는 미러링 기술입니다. 2개의 독립적인 RAID 그룹의 콘텐츠를 미러링합니다. 논리적 구성은 다음과 같습니다.

1. 드라이브는 위치에 따라 두 개의 풀로 구성됩니다. 하나의 풀은 사이트 A의 모든 드라이브로 구성되고, 두 번째 풀은 사이트 B의 모든 드라이브로 구성됩니다
2. 그런 다음 애그리게이트라고 하는 공통 스토리지 풀이 RAID 그룹의 미러링된 세트를 기반으로 생성됩니다. 각 사이트에서 동일한 수의 드라이브가 그려집니다. 예를 들어, 20개 드라이브로 구성된 SyncMirror 애그리게이트는 사이트 A의 드라이브 10개와 사이트 B의 드라이브 10개로 구성됩니다
3. 특정 사이트의 각 드라이브 세트는 미러링 사용과 상관없이 하나 이상의 완전히 이중화된 RAID DP 또는 RAID-TEC 그룹으로 자동으로 구성됩니다. 이와 같이 미러링에서 RAID를 사용하면 사이트 손실 후에도 데이터를 보호할 수 있습니다.



위 그림은 SyncMirror 구성의 예를 보여 줍니다. 24-드라이브 애그리게이트가 사이트 A에 할당된 쉘프의 드라이브 12개와 사이트 B에 할당된 쉘프의 드라이브 12개로 컨트롤러에서 생성되었습니다. 드라이브는 두 개의 미러링된 RAID 그룹으로 그룹화되었습니다. RAID 그룹 0에는 사이트 A의 6개 드라이브 플렉스가 사이트 B의 6개 드라이브 플렉스에 미러링됩니다. 마찬가지로, RAID 그룹 1에는 사이트 A의 6개 드라이브 플렉스가 사이트 B의 6개 드라이브 플렉스에 미러링됩니다.

SyncMirror는 일반적으로 각 사이트에 하나의 데이터 복사본으로 MetroCluster 시스템에 원격 미러링을 제공하는 데 사용됩니다. 경우에 따라 단일 시스템에서 추가 수준의 이중화를 제공하기 위해 사용되었습니다. 특히, 쉘프 레벨 이중화를 제공합니다. 드라이브 쉘프에는 이미 이중 전원 공급 장치와 컨트롤러가 포함되어 있으며 전반적으로 판금보다 조금 더 크지만, 경우에 따라 추가 보호가 필요할 수 있습니다. 예를 들어, 한 NetApp 고객은 자동차 테스트에 사용되는 모바일 실시간 분석 플랫폼용 SyncMirror를 구축했습니다. 시스템은 독립적인 전원 공급 장치와 독립적인 UPS 시스템이 함께 제공되는 두 개의 물리적 랙으로 분리되었습니다.

이중화 실패: NVFAIL

앞서 설명한 것처럼, 쓰기는 하나 이상의 다른 컨트롤러에서 로컬 NVRAM 및 NVRAM에 로그인되기 전까지는 승인되지 않습니다. 이렇게 하면 하드웨어 장애나 정전이 발생해도 전송 중인 I/O가 손실되지 않습니다. 로컬 NVRAM에 장애가

발생하거나 다른 노드에 대한 연결이 실패하면 데이터가 더 이상 미러링되지 않습니다.

로컬 NVRAM에 오류가 보고되면 노드가 종료됩니다. 이 종료를 통해 HA Pair를 사용할 경우 파트너 컨트롤러로 페일오버됩니다. MetroCluster를 사용할 경우 선택한 전체 구성에 따라 동작이 달라지지만 원격 메모로 자동 페일오버될 수 있습니다. 오류가 발생한 컨트롤러가 쓰기 작업을 인식하지 못했기 때문에 어떤 경우에도 데이터가 손실되지 않습니다.

사이트 간 연결 실패가 NVRAM 복제를 원격 노드로 차단하는 경우에 더 복잡한 상황이 됩니다. 쓰기가 더 이상 원격 노드에 복제되지 않으므로 컨트롤러에서 심각한 오류가 발생할 경우 데이터가 손실될 수 있습니다. 더 중요한 것은 이러한 상황에서 다른 노드로 페일오버하려고 하면 데이터가 손실된다는 것입니다.

제어 요소는 NVRAM의 동기화 여부입니다. NVRAM이 동기화되면 데이터 손실 위험 없이 노드 간 페일오버를 안전하게 수행할 수 있습니다. MetroCluster 구성에서 NVRAM 및 기본 애그리게이트 플렉스가 동기화되어 있는 경우 데이터 손실의 위험 없이 전환을 진행해도 안전합니다.

ONTAP는 페일오버 또는 스위치오버가 강제 적용되지 않는 한 데이터가 동기화되지 않을 때 페일오버 또는 스위치오버를 허용하지 않습니다. 이러한 방식으로 조건을 강제로 변경하면 데이터가 원래 컨트롤러에 남겨질 수 있으며 데이터 손실이 허용되는 수준임을 알 수 있습니다.

데이터베이스와 기타 애플리케이션은 디스크에 더 큰 내부 데이터 캐시를 유지하기 때문에 페일오버나 스위치오버가 강제 적용되는 경우 손상에 특히 취약합니다. 강제 적용 페일오버 또는 스위치오버가 발생하면 이전에 승인되었던 변경사항이 효과적으로 폐기됩니다. 스토리지 어레이의 콘텐츠가 사실상 이전 시간으로 이동하며, 캐시의 상태는 디스크에 있는 데이터의 상태를 더 이상 반영하지 않습니다.

이러한 상황을 방지하기 위해 ONTAP에서는 NVRAM 장애에 대한 특별 보호를 위해 볼륨을 구성할 수 있습니다. 이 보호 메커니즘이 트리거되면 볼륨이 NVFAIL이라는 상태로 전환됩니다. 이 상태에서는 애플리케이션 충돌을 일으키는 I/O 오류가 발생합니다. 이 충돌로 인해 애플리케이션이 종료되어 오래된 데이터를 사용하지 않습니다. 커밋된 트랜잭션 데이터가 로그에 있어야 하므로 데이터가 손실되지 않아야 합니다. 일반적인 다음 단계는 관리자가 LUN 및 볼륨을 수동으로 다시 온라인 상태로 전환하기 전에 호스트를 완전히 종료하는 것입니다. 이러한 단계에는 일부 작업이 포함될 수 있지만 이 접근 방식은 데이터 무결성을 보장하는 가장 안전한 방법입니다. 모든 데이터에 이 보호가 필요한 것은 아니므로 NVFAIL 동작을 볼륨별로 구성할 수 있습니다.

HA Pair 및 MetroCluster

MetroCluster는 2노드 및 HA 쌍의 2가지 구성으로 사용할 수 있습니다. 2노드 구성은 NVRAM에 관한 HA 쌍과 동일하게 동작합니다. 갑작스러운 장애가 발생하는 경우 파트너 노드는 NVRAM 데이터를 재생하여 드라이브의 일관성을 유지하고 확인된 쓰기가 손실되지 않도록 할 수 있습니다.

HA 쌍 구성은 NVRAM을 로컬 파트너 노드에 복제합니다. 컨트롤러 장애가 발생하면 MetroCluster 없이 독립 실행형 HA 쌍을 지원하는 경우처럼, 단순한 컨트롤러 장애가 파트너 노드에서 NVRAM이 재생됩니다. 갑작스러운 전체 사이트 손실이 발생하는 경우 원격 사이트에는 드라이브 일관성을 유지하고 데이터 제공을 시작하는 데 필요한 NVRAM이 있습니다.

MetroCluster의 한 가지 중요한 측면은 원격 노드가 정상적인 운영 조건에서는 파트너 데이터에 액세스할 수 없다는 것입니다. 각 사이트는 기본적으로 반대편의 사이트의 성격을 상정할 수 있는 독립적인 시스템으로서 기능한다. 이 프로세스를 스위치오버라고 하며 계획된 스위치오버를 포함합니다. 사이트 운영이 반대편 사이트로 중단 없이 마이그레이션됩니다. 또한, 재해 복구의 일부로 사이트가 손실되고 수동 또는 자동 전환이 필요한 계획되지 않은 상황이 포함됩니다.

전환 및 스위치백

스위치오버 및 스위치백이란 MetroCluster 구성에서 원격 컨트롤러 간에 볼륨을 전환하는 프로세스를 의미합니다. 이 프로세스는 원격 노드에만 적용됩니다. MetroCluster를 4 볼륨 구성으로 사용할 경우 로컬 노드 페일오버는 앞에서 설명한 테이크오버 및 반환 프로세스가 동일합니다.

계획된 전환 및 스위치백

계획된 스위치오버 또는 스위치백은 노드 간의 테이크오버 또는 반환과 비슷합니다. 이 프로세스에는 여러 단계가 있으며 몇 분이 소요되는 것처럼 보일 수 있지만 실제로 발생하는 것은 스토리지 및 네트워크 리소스의 다중 위상 원활한 전환입니다. 제어 전송이 전체 명령을 실행하는 데 필요한 시간보다 훨씬 빠르게 발생하는 순간입니다.

Takeover/Giveback과 스위치오버/스위치백 간의 주된 차이점은 FC SAN 연결에 영향을 미치는 것입니다. 로컬 테이크오버/반환을 사용하면 호스트에서 로컬 노드에 대한 모든 FC 경로가 손실되고 네이티브 MPIO에 의존하여 사용 가능한 대체 경로로 변경됩니다. 포트는 재배치되지 않습니다. 스위치오버 및 스위치백을 사용하면 컨트롤러의 가상 FC 타겟 포트가 다른 사이트로 전환됩니다. 이러한 애플리케이션은 사실상 SAN에 잠시 존재하지 않게 된 다음 대체 컨트롤러에 다시 나타납니다.

SyncMirror 시간 초과

SyncMirror는 헬프 장애로부터 보호하는 ONTAP 미러링 기술입니다. 헬프가 거리를 두고 분리되면 데이터를 원격으로 보호할 수 있습니다.

SyncMirror는 범용 동기식 미러링을 제공하지 않습니다. 결과적으로 가용성이 향상됩니다. 일부 스토리지 시스템은 도미노 모드라고도 하는 일정한 전체 또는 무관 미러링을 사용합니다. 이러한 형태의 미러링은 원격 사이트에 대한 연결이 끊긴 경우 모든 쓰기 작업이 중단되어야 하므로 응용 프로그램에서 제한됩니다. 그렇지 않으면 한 사이트에 쓰기가 존재하지만 다른 사이트에는 쓰기가 존재하지 않습니다. 일반적으로 이러한 환경은 30초 이상 사이트와 사이트 간의 연결이 끊긴 경우 LUN을 오프라인 상태로 전환하도록 구성됩니다.

이 동작은 일부 환경의 하위 집합에 적합합니다. 그러나 대부분의 애플리케이션은 정상적인 작동 조건에서 동기식 복제를 보장하지만 복제를 일시 중지할 수 있는 솔루션이 필요합니다. 사이트 간 연결의 완전한 손실은 주로 재해에 가까운 상황으로 간주됩니다. 일반적으로 이러한 환경은 연결이 복구되거나 데이터 보호를 위해 환경을 종료하기로 결정할 때까지 온라인 상태로 유지되고 데이터를 제공합니다. 순수하게 원격 복제 실패로 인해 애플리케이션을 자동으로 종료해야 하는 요구사항은 특이합니다.

SyncMirror는 시간 초과 방식의 유연성으로 동기식 미러링 요구사항을 지원합니다. 조종기 및/또는 플렉스에 대한 연결이 끊기면 30초 타이머가 카운트 다운을 시작합니다. 카운터가 0에 도달하면 로컬 데이터를 사용하여 쓰기 입출력 처리가 재개됩니다. 데이터의 원격 복제본을 사용할 수 있지만 연결이 복원될 때까지 시간이 지나면 동결됩니다. 재동기화는 애그리게이트 레벨 스냅샷을 활용하여 가능한 한 빨리 시스템을 동기식 모드로 되돌립니다.

특히 대부분의 경우 이러한 종류의 범용 전체 또는 무관 도미노 모드 복제는 애플리케이션 계층에서 더 잘 구현됩니다. 예를 들어 Oracle DataGuard에는 모든 상황에서 장기 인스턴스 복제를 보장하는 최대 보호 모드가 포함되어 있습니다. 구성 가능한 시간 제한을 초과하는 기간 동안 복제 링크가 실패하면 데이터베이스가 종료됩니다.

패브릭 연결 MetroCluster를 통한 자동 무인 전환

자동 무인 전환(AUSO)은 크로스 사이트 HA의 형태를 제공하는 패브릭 연결 MetroCluster 기능입니다. 앞서 설명했듯이, MetroCluster는 각 사이트의 단일 컨트롤러 또는 각 사이트의 HA 쌍 두 가지로 사용할 수 있습니다. HA 옵션의 주요 이점은 계획되었거나 계획되지 않은 컨트롤러 종료를 통해 모든 I/O를 로컬에 둘 수 있다는 것입니다. 단일 노드 옵션의 이점은 비용, 복잡성 및 인프라의 감소입니다.

AUSO의 주요 가치는 Fabric Attached MetroCluster 시스템의 HA 기능을 개선하는 것입니다. 각 사이트가 반대 사이트의 상태를 모니터링하며, 데이터를 제공할 노드가 남아 있지 않으면 AUSO로 인해 빠른 전환이 발생합니다. 이 접근 방식은 가용성 측면에서 구성이 HA 쌍에 더 가깝게 배치되기 때문에 사이트당 단일 노드만을 사용하는 MetroCluster 구성에서 특히 유용합니다.

AUSO는 HA 쌍 수준에서 포괄적인 모니터링을 제공할 수 없습니다. HA 쌍은 노드 간 직접 통신을 위한 이중화 물리적 케이블 2개가 포함되어 있기 때문에 매우 높은 가용성을 제공할 수 있습니다. 또한 HA 쌍의 두 노드는 이중 루프의 동일한 디스크 세트에 액세스할 수 있어, 한 노드에서 다른 노드의 상태를 모니터링할 수 있는 또 다른 경로를

제공합니다.

MetroCluster 클러스터는 사이트 간 네트워크 연결을 통해 노드 간 통신과 디스크 액세스가 모두 필요한 사이트 전체에 존재합니다. 클러스터의 나머지 하트비트를 모니터링하는 기능은 제한되어 있습니다. AUSO는 네트워크 문제로 인해 다른 사이트가 사용할 수 없는 상황이 아니라 실제로 다운된 상황을 구분해야 합니다.

그 결과, HA 쌍의 컨트롤러에서 시스템 패닉 같은 특정 이유로 컨트롤러 장애를 감지하면 테이크오버를 프롬프트 상태가 될 수 있습니다. 또한 하트비트 손실이라고도 하는 연결이 완전히 끊긴 경우 Takeover를 프롬프트할 수도 있습니다.

MetroCluster 시스템은 원래 사이트에서 특정 장애가 감지되는 경우에만 자동 전환을 안전하게 수행할 수 있습니다. 또한 스토리지 시스템의 소유권을 가져오는 컨트롤러는 디스크 및 NVRAM 데이터의 동기화를 보장할 수 있어야 합니다. 컨트롤러는 여전히 작동 가능한 소스 사이트와의 접촉이 끊겼다는 이유로 스위치오버의 안전을 보장할 수 없습니다. 스위치오버 자동화를 위한 추가 옵션은 다음 섹션에서 MetroCluster Tiebreaker(MCTB) 솔루션에 관한 정보를 참조하십시오.

패브릭 연결 MetroCluster가 포함된 MetroCluster Tiebreaker

를 클릭합니다 ["NetApp MetroCluster Tiebreaker의 약어입니다"](#) 소프트웨어를 세 번째 사이트에서 실행하여 MetroCluster 환경의 상태를 모니터링하고, 알림을 보내고, 재해 상황에서 선택적으로 스위치오버를 수행할 수 있습니다. 타이브레이커에 대한 자세한 설명은 에서 확인할 수 있습니다 ["NetApp Support 사이트"](#) 하지만 MetroCluster Tiebreaker의 주요 목적은 사이트 손실을 감지하는 것입니다. 또한 사이트 손실과 연결 손실 간에 구분해야 합니다. 예를 들어, Tiebreaker가 운영 사이트에 연결할 수 없기 때문에 전환이 발생하지 않아야 합니다. 따라서 Tiebreaker는 원격 사이트의 운영 사이트 접속 기능을 모니터링합니다.

AUSO를 통한 자동 절체는 MCTB와도 호환됩니다. AUSO는 특정 장애 이벤트를 감지한 다음 NVRAM 및 SyncMirror 플렉스가 동기화되어 있는 경우에만 스위치오버를 호출하도록 설계되었기 때문에 매우 빠르게 대응합니다.

반대로 타이브레이커는 원격으로 위치하므로 타이머가 경과할 때까지 기다린 후 사이트를 비활성화해야 합니다. Tiebreaker는 결국 AUSO에 포함된 일종의 컨트롤러 장애를 감지하지만, 일반적으로 AUSO는 이미 전환을 시작하고 Tiebreaker가 작동하기 전에 전환을 완료했을 수 있습니다. Tiebreaker에서 생성된 두 번째 switchover 명령은 거부됩니다.

- 주의: * MCTB 소프트웨어는 전환을 강제 적용할 때 NVRAM이 동기화되었는지 또는 플렉스가 동기화되었는지 확인하지 않습니다. 자동 전환이 구성된 경우 유지 관리 활동 중에 NVRAM 또는 SyncMirror 플렉스의 동기화가 손실되는 것을 방지해야 합니다.

또한 MCTB는 지속적인 재해를 처리하지 못해 다음과 같은 일련의 이벤트가 발생할 수 있습니다.

1. 사이트 간 연결이 30초 이상 중단됩니다.
2. SyncMirror 복제 시간이 초과되고 운영 사이트에서 작업이 계속되어 원격 복제본이 오래된 상태로 남습니다.
3. 기본 사이트가 손실되어 기본 사이트에 복제되지 않은 변경 내용이 있습니다. 이렇게 하면 다음과 같은 여러 가지 이유로 전환이 바람직하지 않을 수 있습니다.
 - 기본 사이트에 중요 데이터가 있을 수 있으며 이 경우 해당 데이터를 복구할 수 있습니다. 애플리케이션의 지속적인 운영을 허용한 전환은 중요 데이터를 효과적으로 폐기합니다.
 - 사이트 손실 시 기본 사이트의 스토리지 리소스를 사용 중이던 정상적인 사이트의 애플리케이션이 데이터를 캐싱했을 수 있습니다. 스위치오버로 인해 캐시와 일치하지 않는 오래된 데이터가 생성됩니다.
 - 사이트 손실 시 기본 사이트의 스토리지 리소스를 사용하고 있었던 정상적인 사이트의 운영 체제에서는 데이터가 캐시되었을 수 있습니다. 스위치오버로 인해 캐시와 일치하지 않는 오래된 데이터가 생성됩니다. 가장 안전한 옵션은 사이트 장애가 감지되면 알림을 보내도록 Tiebreaker를 구성한 다음 사람이 전환을 강제 적용할 것인지 여부를 결정하도록 하는 것입니다. 캐시된 데이터를 지우려면 먼저 응용 프로그램 및/또는 운영 체제를 종료해야 할 수 있습니다. 또한 NVFAIL 설정을 사용하여 보호 기능을 추가하고 장애 조치 프로세스를 간소화할

수 있습니다.

MetroCluster IP를 사용하는 ONTAP 중재자

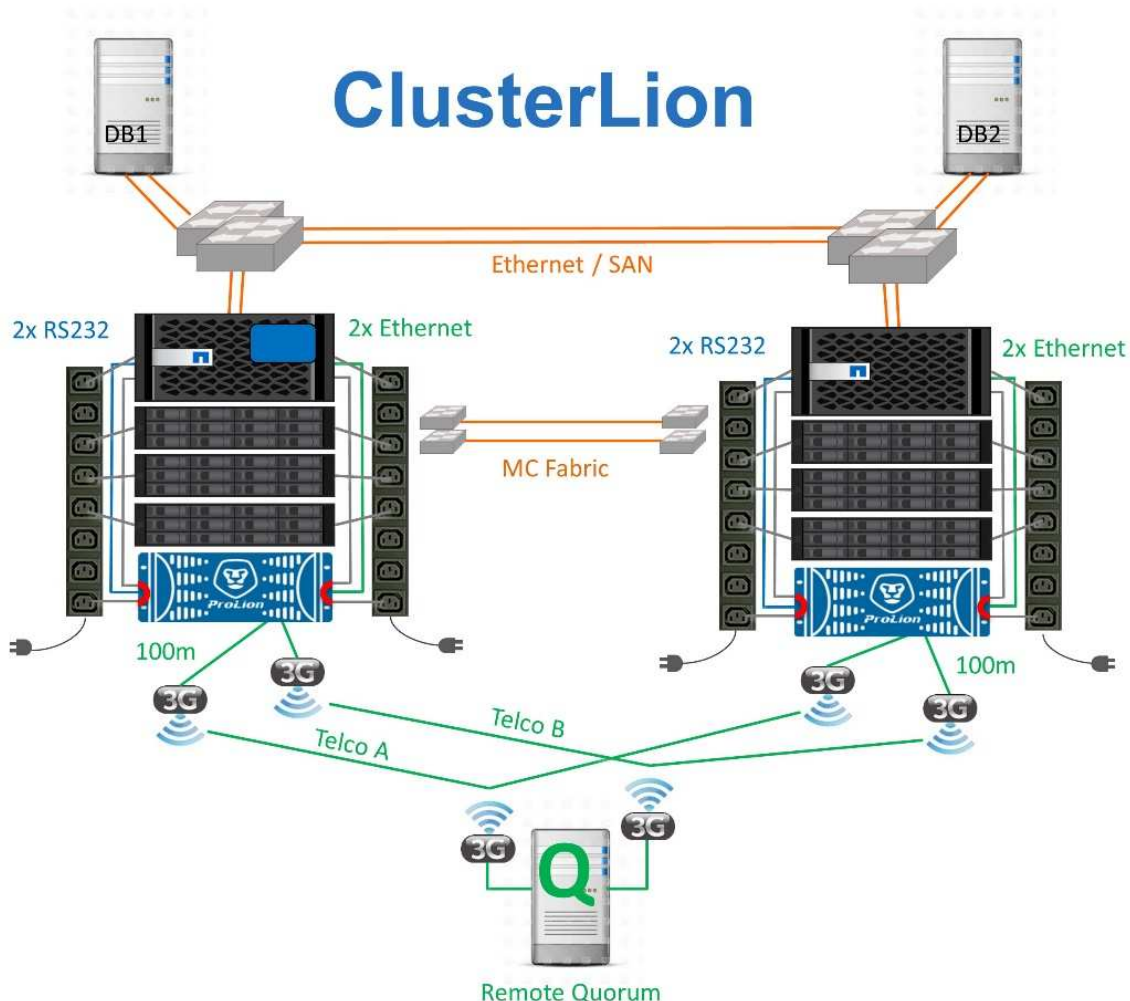
ONTAP mediator는 MetroCluster IP 및 기타 특정 ONTAP 솔루션과 함께 사용됩니다. 위에서 설명한 MetroCluster Tiebreaker 소프트웨어와 마찬가지로 기존 Tiebreaker 서비스 역할을 하지만 자동 자동 전환을 수행하는 중요한 기능도 포함되어 있습니다.

패브릭이 연결된 MetroCluster는 반대쪽 사이트의 스토리지 장치에 직접 액세스할 수 있습니다. 이를 통해 한 MetroCluster 컨트롤러가 드라이브에서 하트비트 데이터를 읽어 다른 컨트롤러의 상태를 모니터링할 수 있습니다. 이를 통해 한 컨트롤러가 다른 컨트롤러의 장애를 인식하고 전환을 수행할 수 있습니다.

반면, MetroCluster IP 아키텍처는 컨트롤러-컨트롤러 연결을 통해서만 모든 I/O를 라우팅하며, 원격 사이트의 스토리지 장치에 직접 액세스할 수 없습니다. 이로 인해 컨트롤러가 장애를 감지하고 스위치오버를 수행할 수 없게 됩니다. 따라서 사이트 손실을 감지하고 자동으로 전환을 수행하기 위한 Tiebreaker 장치로 ONTAP 중재자가 필요합니다.

ClusterLion이 포함된 가상 3번째 사이트

ClusterLion은 가상 3차 사이트로 작동하는 고급 MetroCluster 모니터링 어플라이언스입니다. 이 접근 방식을 통해 MetroCluster는 완전 자동화된 스위치오버 기능을 통해 2개 사이트 구성으로 안전하게 구축할 수 있습니다. 또한 ClusterLion은 추가 네트워크 수준 모니터를 수행하고 전환 후 작업을 실행할 수 있습니다. ProLion에서 전체 문서를 다운로드할 수 있습니다.



- ClusterLion 어플라이언스는 이더넷 및 직렬 케이블을 직접 연결하여 컨트롤러의 상태를 모니터링합니다.
- 이 두 장비는 이중화 3G 무선 연결을 통해 서로 연결됩니다.
- ONTAP 컨트롤러의 전원은 내부 릴레이를 통해 배선됩니다. 사이트 장애가 발생할 경우 내부 UPS 시스템이 포함된 ClusterLion은 전환을 호출하기 전에 전원 연결을 끊습니다. 이 과정을 통해 브레인 분할 상태가 발생하지 않도록 합니다.
- ClusterLion은 30초 SyncMirror 타임아웃 내에 전환을 수행하거나 전혀 전환하지 않습니다.
- NVRAM 및 SyncMirror 플렉스의 상태가 동기화되어 있지 않으면 ClusterLion은 전환을 수행하지 않습니다.
- ClusterLion은 MetroCluster가 완전히 동기화된 경우에만 전환을 수행하기 때문에 NVFAIL은 필요하지 않습니다. 이렇게 구성하면 확장된 Oracle RAC와 같은 사이트 확장 환경이 계획되지 않은 전환 중에도 온라인 상태를 유지할 수 있습니다.
- 여기에는 패브릭 연결 MetroCluster 및 MetroCluster IP가 모두 포함됩니다

Oracle 데이터베이스 및 SyncMirror

MetroCluster 시스템을 통한 Oracle 데이터 보호의 기반은 최대 성능 스케일아웃 동기식 미러링 기술인 SyncMirror입니다.

SyncMirror를 사용한 데이터 보호

가장 간단한 수준인 동기식 복제는 미러링된 스토리지의 양쪽에서 변경 사항이 확인되기 전에 수행되어야 함을 의미합니다. 예를 들어, 데이터베이스에서 로그를 작성하거나 VMware 게스트에 패치를 적용하는 경우 쓰기가 손실되지 않아야 합니다. 프로토콜 레벨에서 스토리지 시스템은 두 사이트의 비휘발성 미디어에 커밋될 때까지 쓰기를 인증해서는 안 됩니다. 그래야만 데이터 손실의 위험 없이 진행하는 것이 안전합니다.

동기식 복제 솔루션을 설계하고 관리하는 첫 번째 단계는 동기식 복제 기술을 사용하는 것입니다. 가장 중요한 고려 사항은 계획된 고장 시나리오와 예상치 못한 다양한 장애 시나리오 중에 발생할 수 있는 상황을 이해하는 것입니다. 모든 동기식 복제 솔루션이 동일한 기능을 제공하는 것은 아닙니다. 데이터 손실이 0인 복구 지점 목표(RPO)를 제공하는 솔루션이 필요한 경우 모든 장애 시나리오를 고려해야 합니다. 특히 사이트 간 연결 손실로 인해 복제가 불가능할 때 예상되는 결과는 무엇입니까?

SyncMirror 데이터 가용성

MetroCluster 복제는 NetApp SyncMirror 기술을 기반으로 하며 동기식 모드로 효율적으로 전환하거나 아웃하도록 설계되었습니다. 이 기능은 동기식 복제를 필요로 하지만 데이터 서비스를 위해고가용성이 필요한 고객의 요구사항을 충족합니다. 예를 들어 원격 사이트에 대한 연결이 끊어진 경우 일반적으로 스토리지 시스템이 복제되지 않은 상태로 계속 작동하도록 하는 것이 좋습니다.

대부분의 동기식 복제 솔루션은 동기식 모드에서만 작동할 수 있습니다. 이러한 유형의 모든 또는 무관 복제를 도미노 모드라고도 합니다. 이러한 스토리지 시스템은 데이터의 로컬 및 원격 복제본이 동기화되지 않도록 하는 대신 데이터 제공을 중지합니다. 복제가 강제로 중단되면 재동기화는 시간이 매우 오래 걸리고 미러링이 다시 설정되는 동안 고객이 완전한 데이터 손실에 노출되도록 할 수 있습니다.

원격 사이트에 연결할 수 없는 경우 SyncMirror가 동기식 모드를 원활하게 전환할 수 있을 뿐만 아니라 연결이 복원되면 RPO=0 상태로 빠르게 다시 동기화할 수 있습니다. 또한 재동기화 중에 원격 사이트의 오래된 데이터 복제본을 사용할 수 있으므로 데이터의 로컬 및 원격 복제본이 항상 존재합니다.

도미노 모드가 필요한 경우 NetApp은 SnapMirror Synchronous(SM-S)를 제공합니다. Oracle DataGuard 또는 호스트측 디스크 미러링의 시간 초과와 같은 애플리케이션 레벨 옵션도 있습니다. 자세한 정보와 옵션은 NetApp 또는 파트너 계정 팀에 문의하십시오.

MetroCluster을 사용한 Oracle 데이터베이스 페일오버

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

일반적인 모범 사례가 여전히 적용되므로 필요한 경우 RPO = 0 데이터 보호만 있으면 MetroCluster를 통해 이 요구사항을 충족할 수 있습니다. 하지만 대부분의 고객은 RPO=0 데이터 보호를 위해뿐만 아니라 재해 시나리오 중에 RTO를 개선하고 사이트 유지 관리 작업의 일부로 투명한 페일오버를 제공하기 위해 MetroCluster을 사용합니다.

사전 구성된 OS로 페일오버

SyncMirror은 재해 복구 사이트에서 데이터의 동기식 복사본을 제공하지만, 데이터를 사용하려면 운영 체제와 관련 애플리케이션이 필요합니다. 기본 자동화를 통해 전체 환경의 장애 조치 시간을 크게 개선할 수 있습니다. Oracle RAC, VCS(Veritas Cluster Server) 또는 VMware HA 같은 Clusterware 제품은 사이트 전체에 클러스터를 생성하는 데 자주 사용되며, 대부분의 경우 간단한 스크립트로 페일오버 프로세스를 구동할 수 있습니다.

운영 노드가 손실되면 대체 사이트에서 애플리케이션을 온라인으로 전환하도록 클러스터웨어(또는 스크립트)가 구성됩니다. 한 가지 옵션은 애플리케이션을 구성하는 NFS 또는 SAN 리소스에 대해 사전 구성된 대기 서버를 생성하는 것입니다. 운영 사이트에 장애가 발생하면 클러스터웨어 또는 스크립트된 대체 시스템이 다음과 유사한 일련의 작업을 수행합니다.

1. MetroCluster 강제 전환
2. FC LUN 검색 수행(SAN만 해당)
3. 파일 시스템을 마운트하는 중입니다
4. 응용 프로그램을 시작하는 중입니다

이 방법의 주요 요구 사항은 원격 사이트에서 실행 중인 OS입니다. 애플리케이션 바이너리로 사전 구성되어야 합니다. 즉, 패치와 같은 작업은 운영 및 대기 사이트에서 수행되어야 합니다. 또는 재해가 선언된 경우 애플리케이션 바이너리를 원격 사이트로 미러링하고 마운트할 수 있습니다.

실제 활성화 절차는 간단합니다. LUN 검색과 같은 명령은 FC 포트당 몇 개의 명령만 사용하면 됩니다. 파일 시스템 마운팅은 에 불과합니다 mount CLI에서 단일 명령으로 명령 및 데이터베이스와 ASM을 모두 시작하고 중지할 수 있습니다. 볼륨 및 파일 시스템이 전환 전 재해 복구 사이트에서 사용되지 않는 경우에는 설정할 필요가 없습니다 dr-force- nvfail On 볼륨.

가상화된 OS로 페일오버

데이터베이스 환경의 페일오버는 운영 체제 자체를 포함하도록 확장할 수 있습니다. 이론적으로 이 페일오버는 부팅 LUN에서 수행할 수 있지만 대부분의 경우 가상화된 OS에서 수행됩니다. 절차는 다음 단계와 유사합니다.

1. MetroCluster 강제 전환
2. 데이터베이스 서버 가상 머신을 호스팅하는 데이터 저장소를 마운트합니다
3. 가상 머신 시작
4. 데이터베이스를 수동으로 시작하거나 가상 시스템이 데이터베이스를 자동으로 시작하도록 구성합니다

예를 들어, ESX 클러스터가 사이트에 걸쳐 있을 수 있습니다. 재해 발생 시 전환 후 재해 복구 사이트에서 가상 시스템을 온라인으로 전환할 수 있습니다. 재해 발생 시 가상 데이터베이스 서버를 호스팅하는 데이터 저장소를 사용하지 않는 한 설정할 필요가 없습니다 `dr-force-nvfail` 연결된 볼륨에서.

모든 주요 **MetroCluster** 파이버 채널 인프라 환경을 지원합니다

NVFAIL은 데이터베이스를 통해 데이터 무결성 보호를 극대화하도록 설계된 ONTAP의 일반적인 데이터 무결성 기능입니다.



이 섹션에서는 MetroCluster 관련 주제를 다루는 기본 ONTAP NVFAIL에 대한 설명을 확장합니다.

MetroCluster를 사용할 경우, 쓰기가 하나 이상의 다른 컨트롤러의 로컬 NVRAM 및 NVRAM에 로그인되기 전까지는 승인되지 않습니다. 이렇게 하면 하드웨어 장애나 정전이 발생해도 전송 중인 I/O가 손실되지 않습니다 로컬 NVRAM에 장애가 발생하거나 다른 노드에 대한 연결이 실패하면 데이터가 더 이상 미러링되지 않습니다.

로컬 NVRAM에 오류가 보고되면 노드가 종료됩니다. 이 종료를 통해 HA Pair를 사용할 경우 파트너 컨트롤러로 페일오버됩니다. MetroCluster를 사용할 경우 선택한 전체 구성에 따라 동작이 달라지지만 원격 메모로 자동 페일오버될 수 있습니다. 오류가 발생한 컨트롤러가 쓰기 작업을 인식하지 못했기 때문에 어떤 경우에도 데이터가 손실되지 않습니다.

사이트 간 연결 실패가 NVRAM 복제를 원격 노드로 차단하는 경우에 더 복잡한 상황이 됩니다. 쓰기가 더 이상 원격 노드에 복제되지 않으므로 컨트롤러에서 심각한 오류가 발생할 경우 데이터가 손실될 수 있습니다. 더 중요한 것은 이러한 상황에서 다른 노드로 페일오버하려고 하면 데이터가 손실된다는 것입니다.

제어 요소는 NVRAM의 동기화 여부입니다. NVRAM이 동기화되면 데이터 손실 위험 없이 노드 간 페일오버를 안전하게 수행할 수 있습니다. MetroCluster 구성에서 NVRAM 및 기본 애그리게이트 플렉스가 동기화되어 있는 경우 데이터 손실 위험 없이 전환을 진행해도 안전합니다.

ONTAP는 페일오버 또는 스위치오버가 강제 적용되지 않는 한 데이터가 동기화되지 않을 때 페일오버 또는 스위치오버를 허용하지 않습니다. 이러한 방식으로 조건을 강제로 변경하면 데이터가 원래 컨트롤러에 남겨질 수 있으며 데이터 손실이 허용되는 수준임을 알 수 있습니다.

데이터베이스는 디스크에 더 큰 내부 데이터 캐시를 유지하기 때문에 페일오버나 스위치오버가 강제 적용되는 경우 손상에 특히 취약합니다. 강제 적용 페일오버 또는 스위치오버가 발생하면 이전에 승인되었던 변경사항이 효과적으로 폐기됩니다. 스토리지 어레이의 콘텐츠가 사실상 이전 시간으로 이동하며, 데이터베이스 캐시의 상태는 디스크에 있는 데이터의 상태를 더 이상 반영하지 않습니다.

이 상황에서 애플리케이션을 보호하기 위해 ONTAP에서는 NVRAM 장애에 대비하여 특별한 보호를 제공하도록 볼륨을 구성할 수 있습니다. 이 보호 메커니즘이 트리거되면 볼륨이 NVFAIL이라는 상태로 전환됩니다. 이 상태에서는 애플리케이션 종료가 I/O 오류가 발생하여 오래된 데이터를 사용하지 않습니다. 확인된 쓰기가 스토리지 시스템에 계속 존재하고 데이터베이스의 경우 커밋된 트랜잭션 데이터가 로그에 있어야 하므로 데이터가 손실되지 않아야 합니다.

일반적인 다음 단계는 관리자가 LUN 및 볼륨을 수동으로 다시 온라인 상태로 전환하기 전에 호스트를 완전히 종료하는 것입니다. 이러한 단계에는 일부 작업이 포함될 수 있지만 이 접근 방식은 데이터 무결성을 보장하는 가장 안전한 방법입니다. 모든 데이터에 이 보호가 필요한 것은 아니므로 NVFAIL 동작을 볼륨별로 구성할 수 있습니다.

수동으로 **NVFAIL**을 강제 적용합니다

사이트 전체에 분산된 애플리케이션 클러스터(VMware, Oracle RAC 등 포함)를 사용하여 강제 전환할 수 있는 가장 안전한 옵션은 `ul` 지정하는 것입니다 `-force-nvfail-all` 명령줄에 입력합니다. 이 옵션은 캐시된 모든 데이터를 플래시하기 위한 긴급 조치로 사용할 수 있습니다. 호스트에서 원래 재해 복구 사이트에 있는 스토리지 리소스를 사용하는 경우 입출력 오류 또는 오래된 파일 핸들이 발생합니다 (ESTALE) 오류. Oracle 데이터베이스가 충돌하고 파일

시스템이 완전히 오프라인 상태가 되거나 읽기 전용 모드로 전환됩니다.

전환이 완료된 후 `은` 을(를) 수행합니다 `in-nvfailed-state` 플래그를 지워야 하며 LUN을 온라인 상태로 설정해야 합니다. 이 작업이 완료되면 데이터베이스를 다시 시작할 수 있습니다. 이러한 작업을 자동화하여 RTO를 줄일 수 있습니다.

dr-force-nvfail입니다

일반적인 안전 조치로 `은` 설정합니다 `dr-force-nvfail` 정상 작업 중에 원격 사이트에서 액세스할 수 있는 모든 볼륨에 플래그를 표시하므로, 페일오버 전에 사용된 활동입니다. 이 설정의 결과로 선택한 원격 볼륨이 들어가면 사용할 수 없게 됩니다 `in-nvfailed-state` 스위치오버 중에 전환이 완료된 후 `은` 을(를) 수행합니다 `in-nvfailed-state` 플래그를 지워야 하며 LUN을 온라인 상태로 설정해야 합니다. 이러한 작업이 완료되면 응용 프로그램을 다시 시작할 수 있습니다. 이러한 작업을 자동화하여 RTO를 줄일 수 있습니다.

결과는 `은` 사용하는 것과 같습니다 `-force-nvfail-all` 수동 전환 플래그 그러나 영향을 받는 볼륨의 수는 오래된 캐시가 있는 애플리케이션이나 운영 체제에서 보호되어야 하는 볼륨으로만 제한될 수 있습니다.

`은` 사용하지 않는 환경에는 두 가지 중요한 요구사항이 있습니다 `dr-force-nvfail` 애플리케이션 볼륨에서:

- 강제 적용 스위치오버는 1차 사이트 손실 후 30초 이내여야 합니다.
- 유지보수 작업 중 또는 SyncMirror 플렉스 또는 NVRAM 복제가 동기화되지 않는 기타 조건에서는 전환이 발생하지 않아야 합니다. 사이트 장애 발생 후 30초 이내에 전환을 수행하도록 구성된 Tiebreaker 소프트웨어를 사용하여 첫 번째 요구사항을 충족할 수 있습니다. 이 요구사항이 사이트 장애 감지 후 30초 이내에 전환을 수행해야 함을 의미하는 것은 아닙니다. 즉, 사이트가 작동 가능으로 확인된 후 30초가 경과하면 강제로 전환을 수행하는 것이 더 이상 안전하지 않습니다.

MetroCluster 구성이 동기화되지 않은 것으로 알려진 경우 모든 자동 전환 기능을 비활성화하여 두 번째 요구 사항을 부분적으로 충족할 수 있습니다. 더 좋은 옵션은 NVRAM 복제 및 SyncMirror Plex의 상태를 모니터링할 수 있는 Tiebreaker 솔루션을 구축하는 것입니다. 클러스터가 완전히 동기화되지 않은 경우 Tiebreaker가 전환을 트리거해서는 안 됩니다.

NetApp MCTB 소프트웨어는 동기화 상태를 모니터링할 수 없으므로 어떤 이유로든 MetroCluster가 동기화되지 않은 경우 이 기능을 비활성화해야 합니다. ClusterLion에는 NVRAM 모니터링 및 플렉스 모니터링 기능이 포함되어 있으며, MetroCluster 시스템이 완전히 동기화되는 것으로 확인되지 않는 한 전환을 트리거하지 않도록 구성할 수 있습니다.

MetroCluster에 있는 **Oracle** 단일 인스턴스

앞서 설명한 것처럼 MetroCluster 시스템이 있다고 해서 데이터베이스 운영에 대한 모범 사례가 반드시 추가되지 않거나 변경되는 것은 아닙니다. 고객 MetroCluster 시스템에서 현재 실행 중인 데이터베이스의 대부분은 단일 인스턴스이며 Oracle on ONTAP 설명서의 권장 사항을 따릅니다.

사전 구성된 OS로 페일오버

SyncMirror은 재해 복구 사이트에서 데이터의 동기식 복사본을 제공하지만, 데이터를 사용하려면 운영 체제와 관련 애플리케이션이 필요합니다. 기본 자동화를 통해 전체 환경의 장애 조치 시간을 크게 개선할 수 있습니다. VCS(Veritas Cluster Server)와 같은 클러스터웨어 제품은 사이트 전체에 클러스터를 생성하는 데 자주 사용되며, 대부분의 경우 간단한 스크립트로 페일오버 프로세스를 구동할 수 있습니다.

운영 노드가 손실되면 대체 사이트에서 데이터베이스를 온라인으로 전환하도록 클러스터웨어(또는 스크립트)가 구성됩니다. 한 가지 옵션은 데이터베이스를 구성하는 NFS 또는 SAN 리소스에 대해 사전 구성된 대기 서버를 생성하는 것입니다. 운영 사이트에 장애가 발생하면 클러스터웨어 또는 스크립트된 대체 시스템이 다음과 유사한 일련의 작업을

수행합니다.

1. MetroCluster 강제 전환
2. FC LUN 검색 수행(SAN만 해당)
3. 파일 시스템 마운트 및/또는 ASM 디스크 그룹 마운트
4. 데이터베이스를 시작하는 중입니다

이 방법의 주요 요구 사항은 원격 사이트에서 실행 중인 OS입니다. Oracle 바이너리로 사전 구성되어야 합니다. 즉, Oracle 패치 적용과 같은 작업이 운영 및 대기 사이트에서 수행되어야 합니다. 또는 재해가 선언된 경우 Oracle 바이너리를 원격 사이트로 미러링하고 마운트할 수 있습니다.

실제 활성화 절차는 간단합니다. LUN 검색과 같은 명령은 FC 포트당 몇 개의 명령만 사용하면 됩니다. 파일 시스템 마운트는 예 불과합니다 mount CLI에서 단일 명령으로 명령 및 데이터베이스와 ASM을 모두 시작하고 중지할 수 있습니다. 볼륨 및 파일 시스템이 전환 전 재해 복구 사이트에서 사용되지 않는 경우에는 설정할 필요가 없습니다 `dr-force-nvfail On` 볼륨.

가상화된 OS로 페일오버

데이터베이스 환경의 페일오버는 운영 체제 자체를 포함하도록 확장할 수 있습니다. 이론적으로 이 페일오버는 부팅 LUN에서 수행할 수 있지만 대부분의 경우 가상화된 OS에서 수행됩니다. 절차는 다음 단계와 유사합니다.

1. MetroCluster 강제 전환
2. 데이터베이스 서버 가상 머신을 호스팅하는 데이터 저장소를 마운트합니다
3. 가상 머신 시작
4. 데이터베이스를 수동으로 시작하거나 데이터베이스를 자동으로 시작하도록 가상 시스템을 구성하면 ESX 클러스터가 사이트에 걸쳐 있을 수 있습니다. 재해 발생 시 전환 후 재해 복구 사이트에서 가상 시스템을 온라인으로 전환할 수 있습니다. 재해 발생 시 가상 데이터베이스 서버를 호스팅하는 데이터 저장소를 사용하지 않는 한 설정할 필요가 없습니다 `dr-force-nvfail` 연결된 볼륨에서.

MetroCluster에서 Oracle RAC 확장

많은 고객이 사이트 간에 Oracle RAC 클러스터를 확장하여 완벽한 Active-Active 구성을 실현함으로써 RTO를 최적화합니다. Oracle RAC의 쿼럼 관리를 포함해야 하기 때문에 전체 설계가 더 복잡해집니다. 또한, 두 사이트에서 데이터에 액세스할 수 있으므로 강제 전환으로 인해 최신 데이터 복사본이 사용될 수 있습니다.

두 사이트 모두에 데이터 복사본이 있지만 현재 애그리게이트를 소유하고 있는 컨트롤러만 데이터를 제공할 수 있습니다. 따라서 확장된 RAC 클러스터의 경우 원격 노드가 사이트 간 연결에서 I/O를 수행해야 합니다. 결과적으로 I/O 지연 시간이 추가되지만 이 지연 시간은 일반적으로 문제가 되지 않습니다. RAC 상호 연결 네트워크도 사이트 간에 확장해야 하므로 지연 시간이 짧은 고속 네트워크가 필요합니다. 추가된 지연 시간으로 인해 문제가 발생할 경우 클러스터를 액티브-패시브 방식으로 작동할 수 있습니다. 그런 다음 I/O 집약적인 작업을 애그리게이트가 속한 컨트롤러에 로컬인 RAC 노드로 보내야 함. 그런 다음 원격 노드가 가벼운 I/O 작업을 수행하거나 온전한 대기 서버로만 사용됩니다.

액티브-액티브 확장 RAC가 필요한 경우 MetroCluster 대신 ASM 미러링을 고려해야 합니다. ASM 미러링을 사용하면 데이터의 특정 복제본을 선호할 수 있습니다. 따라서 모든 읽기가 로컬에서 실행되는 확장 RAC 클러스터를 구축할 수 있습니다. 읽기 I/O가 사이트를 통과하지 않으므로 지연 시간이 가장 짧습니다. 모든 쓰기 작업은 사이트 간 연결을 전송해야 하지만 동기식 미러링 솔루션에서 이러한 트래픽은 피할 수 없습니다.



가상화된 부팅 디스크를 비롯한 부팅 LUN을 Oracle RAC와 함께 사용하는 경우, 이 명령을 사용합니다. `misscount` 매개 변수를 변경해야 할 수 있습니다. RAC 시간 초과 매개변수에 대한 자세한 내용은 을 참조하십시오 ["ONTAP 지원 Oracle RAC"](#).

2개 사이트 구성

2개 사이트의 확장 RAC 구성은 운영 중단 없이 많은 재해 시나리오에서도 가동 중단 없이 지속되는 액티브-액티브 데이터베이스 서비스를 제공할 수 있습니다.

RAC 보팅 파일

MetroCluster에서 확장 RAC를 구축할 때 가장 먼저 고려해야 할 사항은 쿼럼 관리입니다. Oracle RAC에는 디스크 하트비트와 네트워크 하트비트를 관리하는 두 가지 메커니즘이 있습니다. 디스크 하트비트는 보팅 파일을 사용하여 스토리지 액세스를 모니터링합니다. 단일 사이트 RAC 구성의 경우 기본 스토리지 시스템이 HA 기능을 제공하는 한 단일 보팅 리소스로 충분합니다.

이전 버전의 Oracle에서는 보팅 파일이 물리적 스토리지 장치에 배치되었지만 현재 버전의 Oracle에서는 보팅 파일이 ASM 디스크 그룹에 저장됩니다.



Oracle RAC는 NFS에서 지원됩니다. 그리드 설치 프로세스 중에 그리드 파일에 사용되는 NFS 위치를 ASM 디스크 그룹으로 제공하기 위한 일련의 ASM 프로세스가 생성됩니다. 이 프로세스는 최종 사용자에게 거의 투명하며 설치가 완료된 후 지속적인 ASM 관리가 필요하지 않습니다.

2개 사이트 구성의 첫 번째 요구 사항은 무중단 재해 복구 프로세스를 보장하는 방식으로 각 사이트에서 투표 파일의 절반 이상을 항상 액세스할 수 있도록 하는 것입니다. 이 작업은 투표 파일이 ASM 디스크 그룹에 저장되기 전에는 간단했지만, 오늘날 관리자는 ASM 중복의 기본 원칙을 이해해야 합니다.

ASM 디스크 그룹에는 이중화를 위한 세 가지 옵션이 있습니다 `external`, `normal`, 및 `high`. 즉, 미러링되지 않은, 미러링된, 3웨이 미러링이 있습니다. 라는 새로운 옵션입니다 `Flex` 사용 가능하지만 거의 사용되지 않습니다. 이중화 수준 및 중복 장치의 배치가 장애 시나리오에서 수행되는 작업을 제어합니다. 예를 들면 다음과 같습니다.

- 에 투표 파일 배치 `diskgroup` 와 함께 `external` 이중화 리소스는 사이트 간 연결이 끊긴 경우 하나의 사이트를 제거할 수 있도록 보장합니다.
- 에 투표 파일 배치 `diskgroup` 와 함께 `normal` 사이트당 하나의 ASM 디스크만 있는 중복으로 인해 두 사이트 간 연결이 끊어지면 두 사이트 모두에서 노드 제거가 보장됩니다.
- 에 투표 파일 배치 `diskgroup` 와 함께 `high` 한 사이트에 두 개의 디스크가 있고 다른 사이트에 한 개의 디스크가 있는 중복성을 통해 두 사이트가 모두 작동 중이고 상호 연결할 수 있는 경우 활성-활성 작업이 가능합니다. 그러나 단일 디스크 사이트가 네트워크에서 분리되어 있으면 해당 사이트가 제거됩니다.

RAC 네트워크 하트비트

Oracle RAC 네트워크 하트비트는 클러스터 상호 연결에서 노드 가용성을 모니터링합니다. 클러스터에 남아 있으려면 노드가 다른 노드의 절반 이상에 연결할 수 있어야 합니다. 2개 사이트 아키텍처에서는 이 요구 사항으로 인해 RAC 노드 수를 다음과 같이 선택할 수 있습니다.

- 사이트당 동일한 수의 노드를 배치하면 네트워크 연결이 끊어질 경우 한 사이트에서 제거됩니다.
- 한 사이트에 N 노드를 배치하고 반대쪽 사이트에 N+1 노드를 배치하면 사이트 간 연결이 끊어지면 사이트가 네트워크 쿼럼에 더 많은 노드를 남기고 사이트를 더 적은 수의 노드로 제거할 수 있습니다.

Oracle 12cR2 이전에는 사이트 손실 중에 퇴거가 발생하는 측을 제어할 수 없었습니다. 각 사이트에 동일한 수의

노드가 있는 경우 일반적으로 부팅되는 첫 번째 RAC 노드가 마스터 노드에 의해 제거됩니다.

Oracle 12cR2에는 노드 가중치 기능이 도입되었습니다. 이 기능을 통해 관리자는 Oracle이 브레인 분할 조건을 해결하는 방법을 보다 효과적으로 제어할 수 있습니다. 간단한 예로, 다음 명령을 실행하면 RAC의 특정 노드에 대한 기본 설정이 설정됩니다.

```
[root@host-a ~]# /grid/bin/crsctl set server css_critical yes
CRS-4416: Server attribute 'CSS_CRITICAL' successfully changed. Restart
Oracle High Availability Services for new value to take effect.
```

Oracle High-Availability Services를 다시 시작한 후 구성은 다음과 같습니다.

```
[root@host-a lib]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
```

노드 host-a 이(가) 중요 서버로 지정되었습니다. 2개의 RAC 노드가 격리된 경우 host-a 존속, 그리고 host-b 퇴거시킵니다.



자세한 내용은 Oracle 백서 "Oracle Clusterware 12c Release 2 기술 개요"를 참조하십시오.

12cR2 이전 버전의 Oracle RAC의 경우 다음과 같이 CRS 로그를 확인하여 마스터 노드를 식별할 수 있습니다.

```
[root@host-a ~]# /grid/bin/crsctl status server -f | egrep
'^NAME|CSS_CRITICAL='
NAME=host-a
CSS_CRITICAL=yes
NAME=host-b
CSS_CRITICAL=no
[root@host-a ~]# grep -i 'master node' /grid/diag/crs/host-
a/crs/trace/crsd.trc
2017-05-04 04:46:12.261525 : CRSSE:2130671360: {1:16377:2} Master Change
Event; New Master Node ID:1 This Node's ID:1
2017-05-04 05:01:24.979716 : CRSSE:2031576832: {1:13237:2} Master Change
Event; New Master Node ID:2 This Node's ID:1
2017-05-04 05:11:22.995707 : CRSSE:2031576832: {1:13237:221} Master
Change Event; New Master Node ID:1 This Node's ID:1
2017-05-04 05:28:25.797860 : CRSSE:3336529664: {1:8557:2} Master Change
Event; New Master Node ID:2 This Node's ID:1
```

이 로그는 마스터 노드가 임을 나타냅니다 2 및 노드입니다 host-a 의 ID가 있습니다 1. 이는 실제로 그 점을

의미합니다 `host-a` 은(는) 마스터 노드가 아닙니다. 마스터 노드의 ID는 명령을 사용하여 확인할 수 있습니다
`olsnodes -n`.

```
[root@host-a ~]# /grid/bin/olsnodes -n
host-a 1
host-b 2
```

ID가 인 노드입니다 2 있습니다 `host-b`, 마스터 노드입니다. 각 사이트의 노드 수가 동일한 구성에서 사이트는 을(를) 사용합니다 `host-b` 어떤 이유로든 두 세트의 네트워크 연결이 끊길 경우 존속되는 사이트입니다.

마스터 노드를 식별하는 로그 항목이 시스템에서 제외될 수 있습니다. 이 경우 OCR(Oracle Cluster Registry) 백업의 타임스탬프를 사용할 수 있습니다.

```
[root@host-a ~]# /grid/bin/ocrconfig -showbackup
host-b      2017/05/05 05:39:53      /grid/cdata/host-cluster/backup00.ocr
0
host-b      2017/05/05 01:39:53      /grid/cdata/host-cluster/backup01.ocr
0
host-b      2017/05/04 21:39:52      /grid/cdata/host-cluster/backup02.ocr
0
host-a      2017/05/04 02:05:36      /grid/cdata/host-cluster/day.ocr      0
host-a      2017/04/22 02:05:17      /grid/cdata/host-cluster/week.ocr     0
```

이 예는 마스터 노드가 임을 보여 줍니다 `host-b`. 또한 에서 마스터 노드가 변경되었음을 나타냅니다 `host-a` 를 선택합니다 `host-b` 5월 4일 2시 5분에서 21시 39분 사이. 이 마스터 노드를 식별하는 방법은 이전 OCR 백업 이후 마스터 노드가 변경될 수 있기 때문에 CRS 로그도 확인한 경우에만 사용하는 것이 안전합니다. 이 변경 사항이 발생한 경우 OCR 로그에 표시됩니다.

대부분의 고객은 전체 환경과 각 사이트에서 동일한 수의 RAC 노드를 서비스하는 단일 보팅 디스크 그룹을 선택합니다. 디스크 그룹은 데이터베이스가 포함된 사이트에 배치해야 합니다. 그 결과, 연결이 끊어지면 원격 사이트에서 제거됩니다. 원격 사이트에는 더 이상 쿼럼이 없고 데이터베이스 파일에 액세스할 수 없지만 로컬 사이트는 평소와 같이 계속 실행됩니다. 연결이 복원되면 원격 인스턴스를 다시 온라인 상태로 만들 수 있습니다.

재해가 발생할 경우 데이터베이스 파일과 보팅 디스크 그룹을 정상 사이트에서 온라인으로 전환하기 위해 전환을 수행해야 합니다. AUSO가 재해에 의해 전환을 트리거할 경우 클러스터가 동기화하고 스토리지 리소스가 정상적으로 온라인 상태가 되기 때문에 NVFAIL이 트리거되지 않습니다. AUSO는 매우 빠른 작동이며, 이전에 완료되어야 합니다 `disktimeout` 기간이 만료됩니다.

사이트는 두 곳밖에 없기 때문에 자동화된 외부 티브레이킹 소프트웨어를 사용할 수 없으며, 이는 강제 전환이 수동 작업이어야 한다는 것을 의미합니다.

3개 사이트 구성

확장된 RAC 클러스터는 3개의 사이트로 훨씬 더 쉽게 설계할 수 있습니다. MetroCluster 시스템의 절반을 호스팅하는 두 사이트도 데이터베이스 워크로드를 지원하고, 세 번째 사이트는 데이터베이스와 MetroCluster 시스템을 위한 Tiebreaker 역할을 합니다. Oracle Tiebreaker 구성은 세 번째 사이트에 투표하는 데 사용되는 ASM 디스크 그룹의 구성원을 배치하는 것만큼 간단할 수 있으며, RAC 클러스터에 홀수 노드 수가 있는지 확인하기 위해 세 번째 사이트에 운영 인스턴스를 포함할 수도 있습니다.



확장 RAC 구성에서 NFS를 사용하는 방법에 대한 중요한 정보는 "취급 장애 그룹"에 관한 Oracle 설명서를 참조하십시오. 요약하면, 취급 리소스를 호스팅하는 세 번째 사이트에 대한 연결이 끊겨 기본 Oracle 서버 또는 Oracle RAC 프로세스가 중단되지 않도록 소프트 옵션을 포함하도록 NFS 마운트 옵션을 수정해야 할 수 있습니다.

SnapMirror 활성 동기화

SnapMirror 활성 동기화가 포함된 Oracle 데이터베이스

SnapMirror 액티브 동기화는 개별 Oracle 데이터베이스 및 애플리케이션 환경에 선택적 RPO = 0 동기식 미러링을 지원합니다.

SnapMirror 액티브 동기화는 기본적으로 SAN에 대한 향상된 SnapMirror 기능으로, 호스트가 LUN을 호스팅하는 시스템과 해당 복제본을 호스팅하는 시스템 모두에서 LUN에 액세스할 수 있습니다.

SnapMirror 활성 동기화 및 SnapMirror Sync는 복제 엔진을 공유하지만 SnapMirror 액티브 동기화에는 엔터프라이즈 애플리케이션에 대한 투명한 애플리케이션 페일오버 및 페일백과 같은 추가 기능이 포함되어 있습니다.

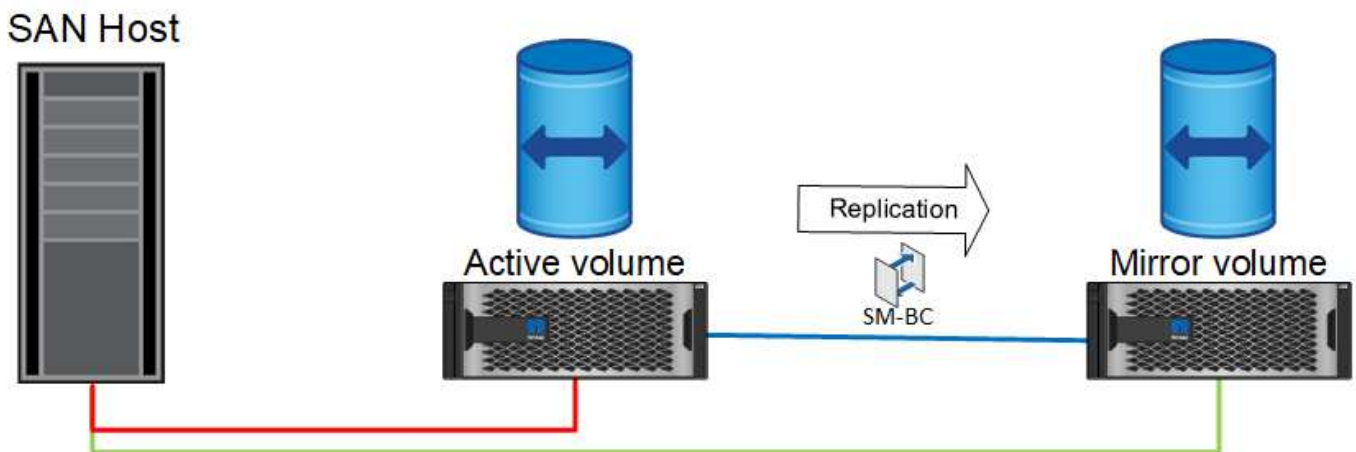
실제로 이 솔루션은 개별 워크로드에 대해 선택적 세부 RPO=0 동기식 복제를 지원하여 MetroCluster의 세부 버전과 비슷하게 작동합니다. 하위 레벨의 경로 동작은 MetroCluster와 매우 다르지만 호스트 관점에서 최종 결과는 비슷합니다.

경로 액세스

SnapMirror Active Sync를 사용하면 운영 스토리지 및 원격 스토리지 시스템 모두에서 호스트 운영 체제에 스토리지 장치를 표시할 수 있습니다. 경로는 스토리지 시스템과 호스트 간에 최적화된 경로를 식별하기 위한 업계 표준 프로토콜인 ALUA(Asymmetric Logical Unit Access)를 통해 관리됩니다.

입출력에 가장 액세스하기 위한 디바이스 경로는 활성/최적화 경로로 간주되고 나머지 경로는 활성/최적화되지 않은 경로로 간주됩니다.

SnapMirror 활성 동기화 관계는 서로 다른 클러스터에 있는 SVM 쌍 사이입니다. 두 SVM 모두 데이터를 제공할 수 있지만 ALUA는 LUN이 상주하는 드라이브를 현재 소유하는 SVM을 우선적으로 사용합니다. 원격 SVM에 대한 IO는 SnapMirror 활성 동기화 상호 연결을 통해 에서 프록시를 수행합니다.



동기식 복제

정상 작동 시 원격 복제본은 항상 RPO=0 동기식 복제본이며 한 가지 예외가 있습니다. 데이터를 복제할 수 없는 경우 SnapMirror 활성 동기화를 사용하면 데이터를 복제하고 IO 서비스를 재개해야 합니다. 이 옵션은 복제 링크의 손실을 거의 재해에 가까운 문제로 간주하거나 데이터를 복제할 수 없을 때 비즈니스 작업이 중단되는 것을 원하지 않는 고객이 선호합니다.

스토리지 하드웨어

다른 스토리지 재해 복구 솔루션과 달리 SnapMirror Active Sync는 비대칭적 플랫폼 유연성을 제공합니다. 각 사이트의 하드웨어는 동일할 필요가 없습니다. 이 기능을 사용하면 SnapMirror 액티브 동기화를 지원하는 데 사용되는 하드웨어를 적절한 크기로 조정할 수 있습니다. 전체 운영 워크로드를 지원해야 하는 경우 원격 스토리지 시스템이 기본 사이트와 동일할 수 있지만 재해로 인해 I/O가 감소할 경우 원격 사이트의 소규모 시스템보다 비용 효율적입니다.

ONTAP 중재자

ONTAP 중재자는 NetApp 지원에서 다운로드되는 소프트웨어 응용 프로그램입니다. 중재자는 운영 및 원격 사이트 스토리지 클러스터 모두에 대한 페일오버 작업을 자동화합니다. 온프레미스 또는 클라우드에서 호스팅되는 소규모 가상 머신(VM)에 구축할 수 있습니다. 구성이 완료되면 두 사이트의 장애 조치 시나리오를 모니터링하는 세 번째 사이트 역할을 합니다.

SnapMirror 활성 동기화를 통한 Oracle 데이터베이스 페일오버

SnapMirror 액티브 동기화에서 Oracle 데이터베이스를 호스팅하는 주된 이유는 계획된 스토리지 이벤트 및 계획되지 않은 스토리지 이벤트 중에 투명한 페일오버를 제공하기 위해서입니다.

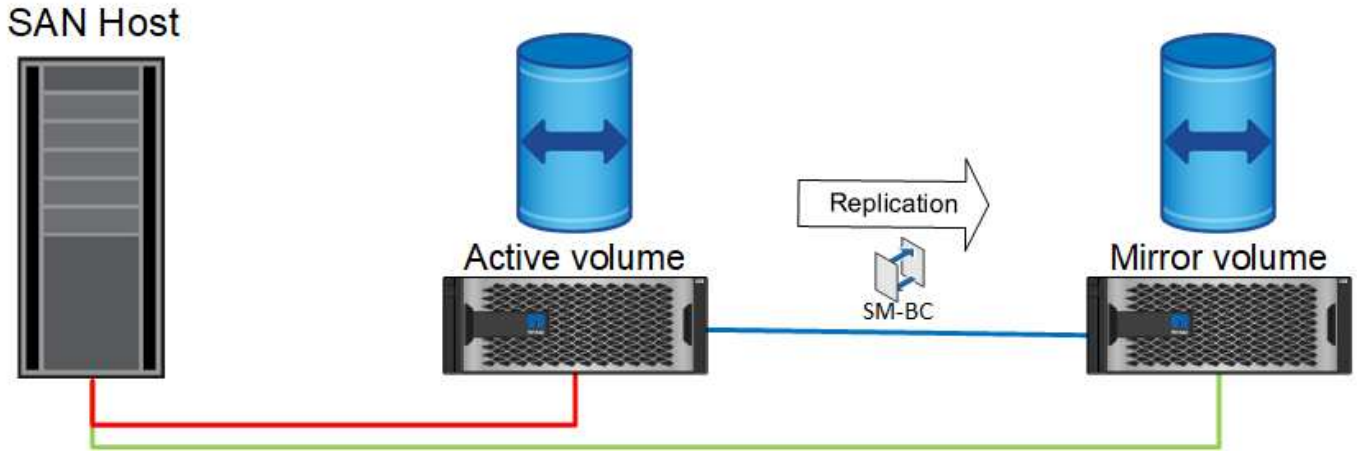
SnapMirror 액티브 동기화는 계획된 스토리지 페일오버 작업과 계획되지 않은 두 가지 유형의 스토리지 페일오버 작업을 지원하며, 두 가지 방식으로 작동합니다. 계획된 페일오버는 관리자가 원격 사이트로 빠르게 전환하기 위해 수동으로 시작하는 반면, 계획되지 않은 페일오버는 3차 사이트의 중재자가 자동으로 시작합니다. 계획된 페일오버의 주된 목적은 증분 패치 및 업그레이드를 수행하고, 재해 복구 테스트를 수행하거나, 전체 활성 동기화 기능을 입증하기 위해 1년 내내 사이트 간 작업을 전환하는 공식적인 정책을 채택하는 것입니다.

이 다이어그램은 정상, 장애 조치 및 장애 복구 작업 중에 발생하는 상황을 보여 줍니다. 쉽게 이해할 수 있도록 복제된 LUN을 설명합니다. 실제 SnapMirror 활성 동기화 구성에서는 각 볼륨에 하나 이상의 LUN이 포함되어 있는 볼륨을 기반으로 복제가 수행되지만 그림을 더 단순하게 만들기 위해 볼륨 계층이 제거되었습니다.

정상 작동

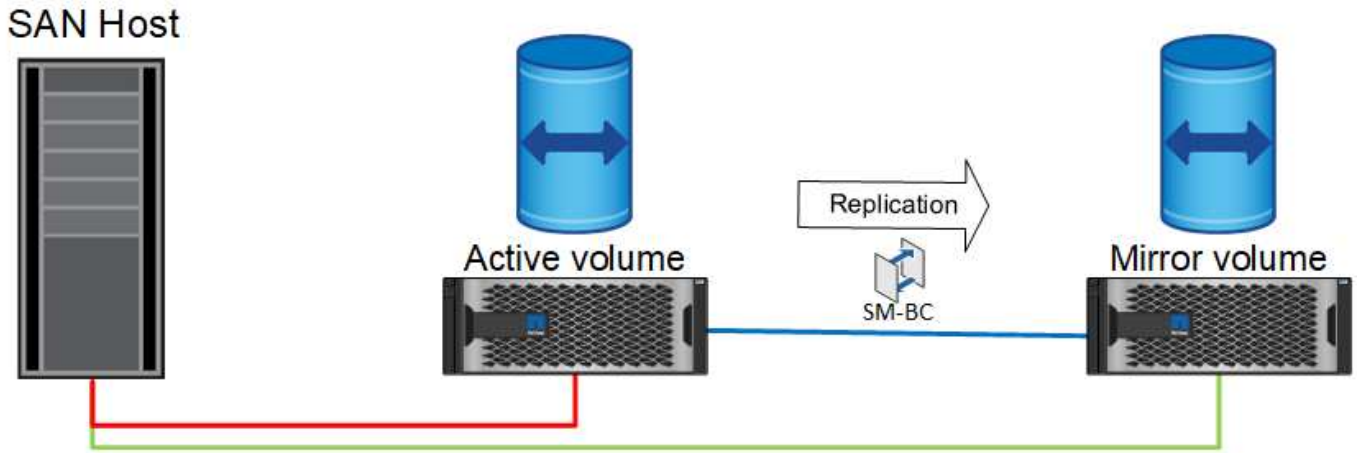
정상 작동 시 로컬 또는 원격 복제본에서 LUN에 액세스할 수 있습니다. 빨간색 선은 ALUA에서 광고한 최적화된 경로를 나타내며, 그 결과 입출력이 이 경로를 통해 우선적으로 전송되어야 합니다.

녹색 선은 활성 경로이지만 해당 경로의 IO가 SnapMirror 활성 동기화 경로를 통해 전달되어야 하기 때문에 지연 시간이 더 많이 발생합니다. 추가 지연 시간은 SnapMirror 활성 동기화에 사용되는 사이트 간 상호 연결의 속도에 따라 달라집니다.



실패

계획되거나 계획되지 않은 페일오버 때문에 액티브 미러 복제본을 사용할 수 없게 되면 더 이상 사용할 수 없게 됩니다. 그러나 원격 시스템에는 동기식 복제본이 있고 원격 사이트에 대한 SAN 경로가 이미 존재합니다. 원격 시스템에서 해당 LUN에 대한 IO를 처리할 수 있습니다.



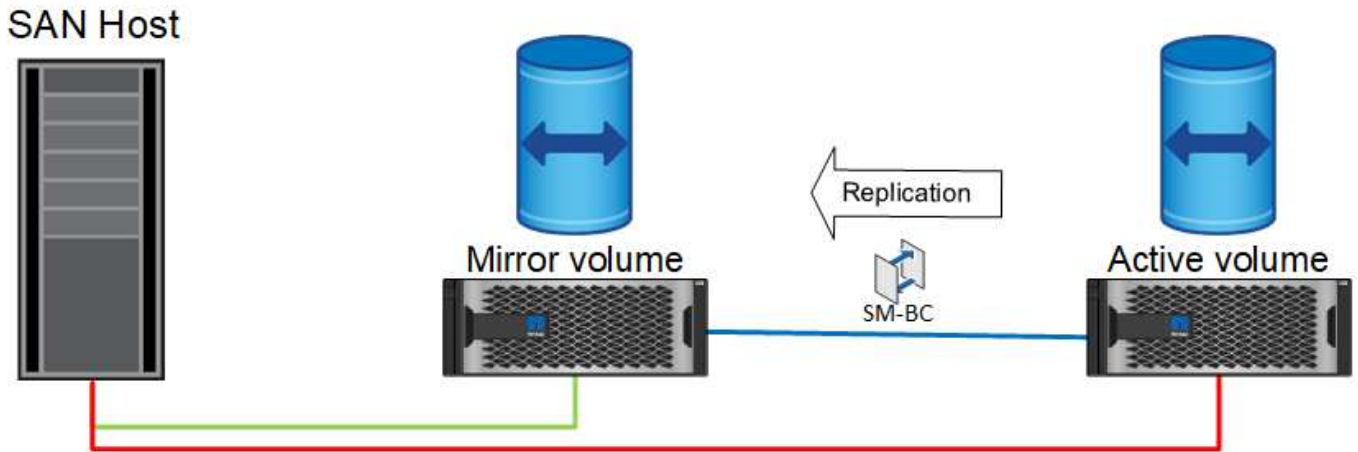
페일오버

페일오버하면 원격 복제본이 활성 복제본이 됩니다. 경로가 Active에서 Active/Optimized로 변경되고 IO는 데이터 손실 없이 계속 처리됩니다.



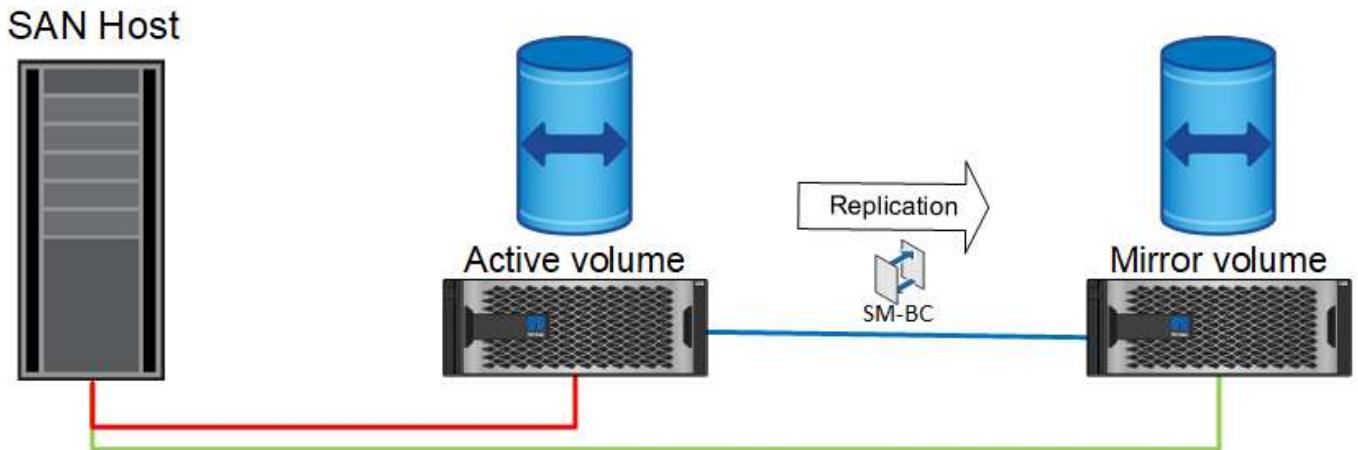
복구

소스 시스템이 서비스로 반환되면 SnapMirror 활성 동기화는 복제를 다시 동기화하지만 다른 방향을 실행할 수 있습니다. 이제 이 구성은 기본적으로 시작점과 동일하지만 활성 미러 사이트가 대칭 이동된 경우는 예외입니다.



장애 복구

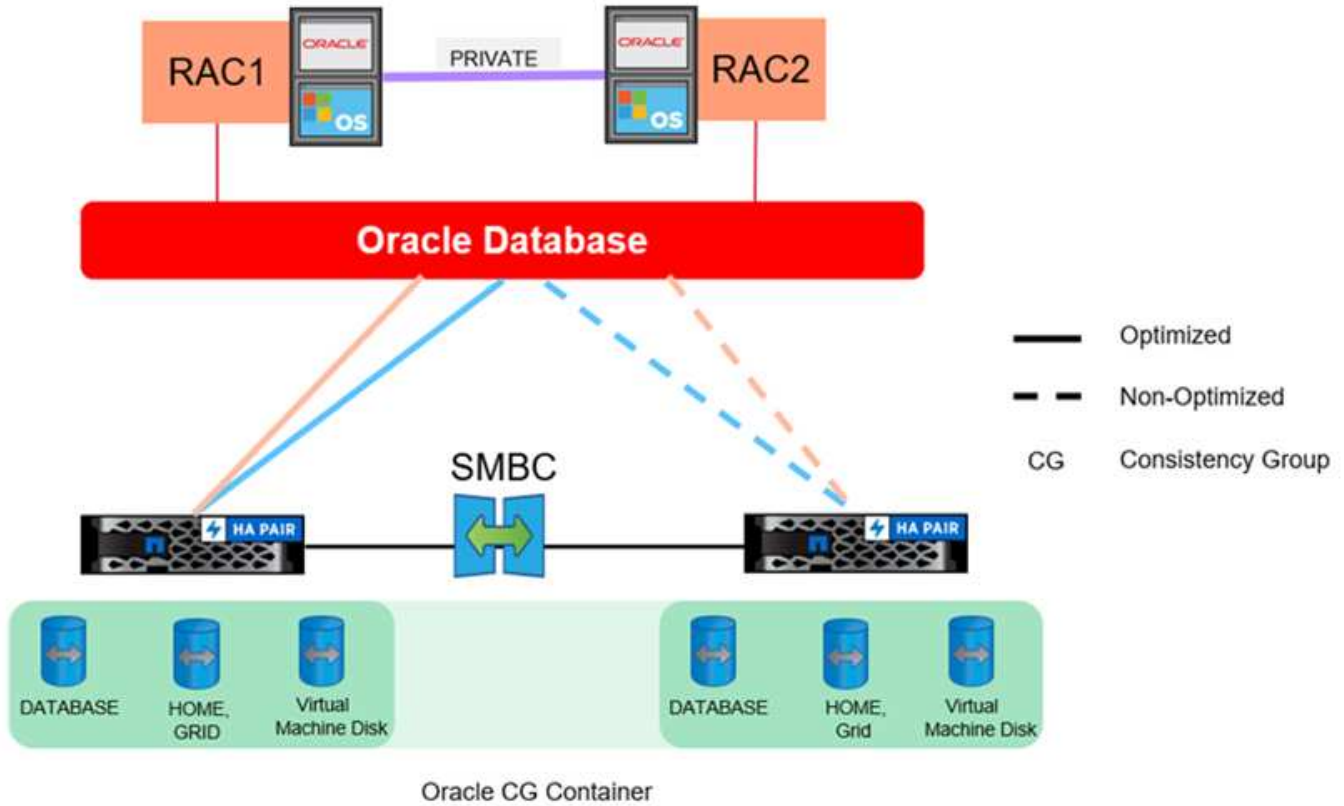
필요한 경우 관리자는 페일백을 수행한 후 LUN의 활성 복사본을 원래 컨트롤러로 다시 이동할 수 있습니다.



SnapMirror 활성 동기화가 포함된 단일 인스턴스 Oracle 데이터베이스

아래 다이어그램은 Oracle 데이터베이스의 기본 및 원격 스토리지 클러스터 모두에서 스토리지 디바이스를 조닝 또는 연결하는 간단한 구축 모델을 보여 줍니다.

Oracle은 운영 환경에만 구성되어 있습니다. 이 모델은 스토리지 측 재해 발생 시 애플리케이션 다운타임 없이 데이터 손실 없이 원활한 스토리지 페일오버를 해결합니다. 그러나 이 모델은 사이트 장애 시 데이터베이스 환경의고가용성을 제공하지 않습니다. 이 유형의 아키텍처는 스토리지 서비스의고가용성을 갖춘 데이터 손실 없는 솔루션을 찾고 있지만 데이터베이스 클러스터의 전체 손실에는 수동 작업이 필요하다는 점을 수용하려는 고객에게 유용합니다.



또한 이 방법을 사용하면 Oracle 라이선스 비용을 절감할 수 있습니다. 원격 사이트에서 Oracle 데이터베이스 노드를 미리 구성하려면 대부분의 Oracle 라이선스 계약에 따라 모든 코어에 라이선스를 부여해야 합니다. Oracle 데이터베이스 서버를 설치하고 나머지 데이터 복사본을 마운트하는 데 소요되는 시간으로 인한 지연이 허용되는 경우 이 설계는 비용 효율성이 매우 높습니다.

SnapMirror 활성 동기화가 포함된 Oracle RAC

SnapMirror 액티브 동기화는 로드 밸런싱 또는 개별 애플리케이션 페일오버와 같은 데이터 세트 복제를 세부적으로 제어할 수 있습니다. 전체 아키텍처는 확장된 RAC 클러스터처럼 보이지만 일부 데이터베이스는 특정 사이트 전용이며 전체 부하가 분산됩니다.

예를 들어 6개의 개별 데이터베이스를 호스팅하는 Oracle RAC 클러스터를 구축할 수 있습니다. 세 데이터베이스의 스토리지는 주로 사이트 A에서 호스팅되고, 다른 세 데이터베이스의 스토리지는 사이트 B에서 호스팅됩니다. 이 구성은 사이트 간 트래픽을 최소화하여 최상의 성능을 보장합니다. 또한 활성 경로를 사용하여 스토리지 시스템에 로컬인 데이터베이스 인스턴스를 사용하도록 애플리케이션을 구성합니다. 이렇게 하면 RAC 상호 연결 트래픽이 최소화됩니다. 마지막으로, 이 전반적인 설계를 통해 모든 컴퓨팅 리소스가 균일하게 사용되도록 합니다. 워크로드가 변경되면 사이트 간에 데이터베이스를 선택적으로 장애 조치하여 로딩이 고르게 이루어질 수 있습니다.

세분화를 제외하고 SnapMirror 액티브 시너지를 사용하는 Oracle RAC의 기본 원칙 및 옵션은 와 동일합니다
["MetroCluster 기반 Oracle RAC"](#)

Oracle 데이터베이스 및 SnapMirror 활성 동기화 장애 시나리오

SnapMirror 활성 동기화(SM-AS) 장애 시나리오가 각각 다른 결과를 가지고 있습니다.

시나리오	결과
복제 링크 오류입니다	중재자는 이 브레인 분할 시나리오를 인식하고 마스터 복제본이 있는 노드에서 입출력을 재개합니다. 사이트 간 연결이 다시 온라인 상태가 되면 대체 사이트가 자동 재동기화를 수행합니다.
1차 사이트 스토리지 장애	자동 비계획 페일오버는 중재자에 의해 시작됩니다. I/O 중단 없음.
원격 사이트 스토리지 장애	I/O 중단은 없습니다. 네트워크가 동기화 복제를 중단하도록 하고 마스터가 입출력을 계속 제공할 수 있는 올바른 소유자임을 설정하여 일시 정지가 발생합니다. 따라서 입출력이 몇 초 동안 일시 중지되면 입출력이 재개됩니다. 사이트가 온라인 상태일 때 자동으로 다시 동기화됩니다.
중재자와 스토리지 어레이 간의 중재자 또는 링크 상실	입출력이 계속되고 원격 클러스터와 동기화된 상태로 유지되지만 중재자가 없는 경우 예상치 못한 자동 페일오버 및 페일백은 불가능합니다.
HA 클러스터의 스토리지 컨트롤러 중 하나가 손실되었습니다	HA 클러스터의 파트너 노드가 테이크오버(NDO)를 시도합니다. 테이크오버가 실패하면 중재자는 스토리지의 두 노드가 모두 다운되었음을 알리고 원격 클러스터에 대한 비계획 페일오버를 자동으로 수행합니다.
디스크 손실	입출력은 최대 3개의 연속 디스크 장애에 대해 계속됩니다. RAID-TEC의 일부입니다.
일반적인 구축 환경에서 전체 사이트 손실	오류가 발생한 사이트의 서버는 더 이상 사용할 수 없습니다. 클러스터링을 지원하는 응용 프로그램은 두 사이트에서 모두 실행되고 대체 사이트에서 계속 작동하도록 구성할 수 있습니다. 하지만 대부분의 응용 프로그램은 SM-AS에서 중재자가 필요한 방식과 유사한 3차 사이트 Tiebreaker가 필요합니다. 애플리케이션 레벨 클러스터가 없으면 정상 사이트에서 애플리케이션을 시작해야 합니다. 이 경우 가용성에 영향을 미치지만 RPO=0은 유지됩니다. 데이터는 손실되지 않습니다.

Oracle 데이터베이스 마이그레이션

Oracle 데이터베이스를 ONTAP 스토리지 시스템으로 마이그레이션

새로운 스토리지 플랫폼의 기능 활용에는 한 가지 필연적인 요구사항이 있으므로 데이터를 새로운 스토리지 시스템에 배치해야 합니다. ONTAP를 사용하면 ONTAP에서 ONTAP로의 마이그레이션 및 업그레이드, 외부 LUN 가져오기, 호스트 운영 체제 또는 Oracle 데이터베이스 소프트웨어를 직접 사용하는 절차 등 마이그레이션 프로세스를 간소화할 수 있습니다.



이 문서는 이전에 게시된 기술 보고서 _TR-4534: Oracle 데이터베이스를 NetApp 스토리지 시스템으로 마이그레이션 _을(를) 대체합니다

새 데이터베이스 프로젝트의 경우 데이터베이스 및 애플리케이션 환경이 제대로 구축되기 때문에 이 문제가 발생하지 않습니다. 그러나 마이그레이션은 비즈니스 중단, 마이그레이션 완료에 필요한 시간, 필요한 기술 세트 및 위험 최소화와 관련하여 특별한 과제를 안고 있습니다.

스크립트

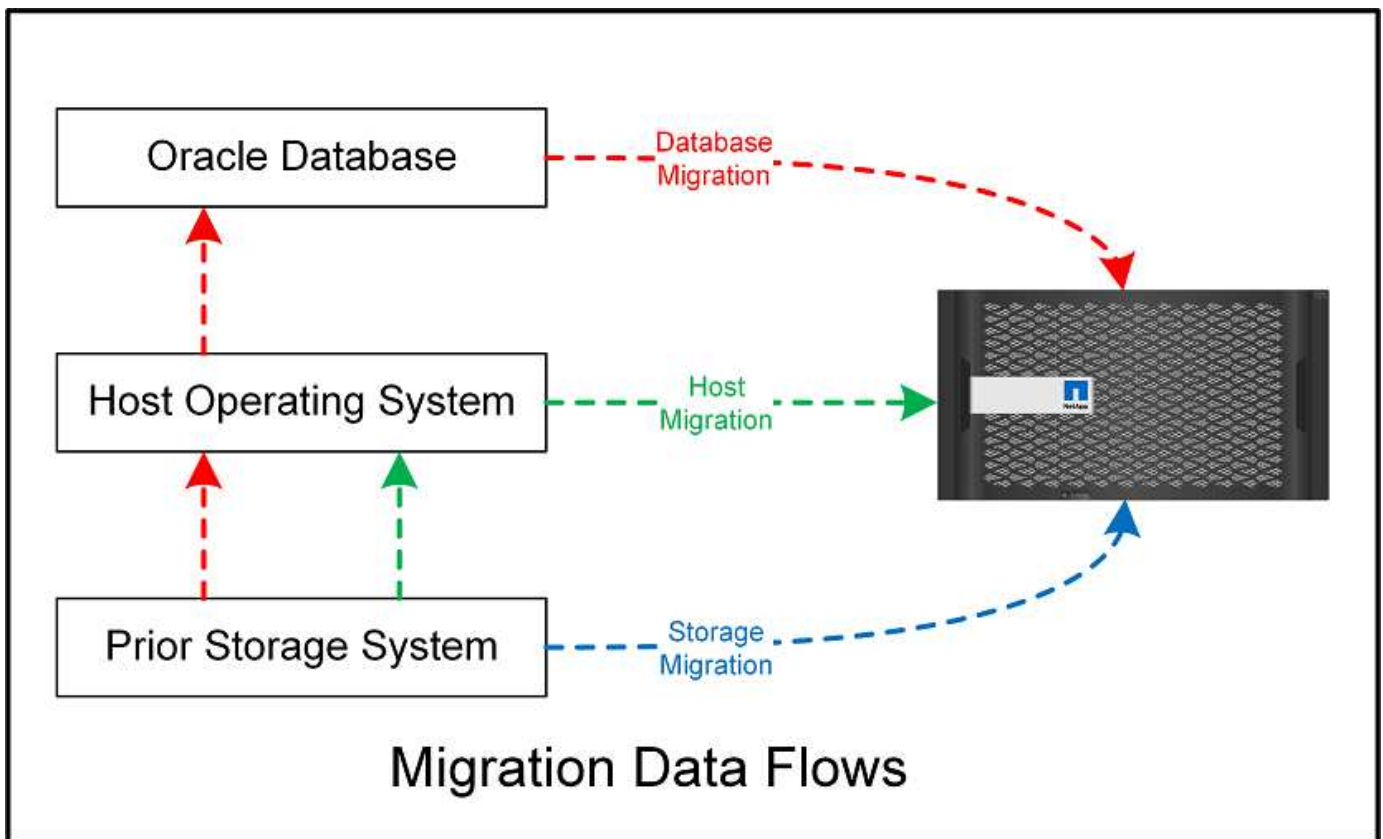
샘플 스크립트는 이 설명서에 나와 있습니다. 이러한 스크립트는 사용자 오류 가능성을 줄이기 위해 마이그레이션의 다양한 측면을 자동화하는 샘플 방법을 제공합니다. 이 스크립트는 마이그레이션을 담당하는 IT 직원의 전반적인 요구사항을 줄이고 전반적인 프로세스를 가속화할 수 있습니다. 이 스크립트는 NetApp 프로페셔널 서비스 및 NetApp 파트너가 수행하는 실제 마이그레이션 프로젝트로부터 작성되었습니다. 이 설명서 전반에 걸쳐 사용 예제가 나와 있습니다.

Oracle 데이터베이스 마이그레이션 계획

Oracle 데이터 마이그레이션은 데이터베이스, 호스트 또는 스토리지 배열의 세 가지 레벨 중 하나에서 수행할 수 있습니다.

데이터베이스, 호스트 운영 체제 또는 스토리지 시스템 등 전체 솔루션의 어떤 구성 요소가 데이터 이동을 담당하는지에 따라 차이가 있습니다.

아래 그림에서는 마이그레이션 수준과 데이터 흐름의 예를 보여 줍니다. 데이터베이스 레벨 마이그레이션의 경우 데이터가 원래 스토리지 시스템에서 호스트 및 데이터베이스 계층을 통해 새로운 환경으로 이동됩니다. 호스트 레벨 마이그레이션은 비슷하지만 데이터가 애플리케이션 계층을 거치지 않고 호스트 프로세스를 사용하여 새 위치에 기록됩니다. 마지막으로, 스토리지 레벨 마이그레이션을 통해 NetApp FAS 시스템과 같은 어레이가 데이터 이동을 담당합니다.



데이터베이스 수준 마이그레이션은 일반적으로 Oracle 계층에서 마이그레이션을 완료하기 위해 대기 데이터베이스를

통한 Oracle 로그 전달을 사용하는 것을 의미합니다. 호스트 레벨 마이그레이션은 호스트 운영 체제 구성의 기본 기능을 사용하여 수행됩니다. 이 구성에는 cp, tar 및 Oracle RMAN(Recovery Manager)과 같은 명령을 사용하거나 LVM(Logical Volume Manager)을 사용하여 파일 시스템의 기본 바이트를 재배치하는 파일 복사 작업이 포함됩니다. Oracle ASM(Automatic Storage Management)은 데이터베이스 애플리케이션 수준 이하로 실행되기 때문에 호스트 수준 기능으로 분류됩니다. ASM은 호스트에서 일반적인 논리적 볼륨 관리자를 대체합니다. 마지막으로, 데이터는 스토리지 어레이 레벨에서 마이그레이션될 수 있으며, 이는 운영 체제 레벨 아래에서 마이그레이션됩니다.

계획 고려 사항

마이그레이션을 위한 최상의 옵션은 마이그레이션할 환경의 규모, 다운타임을 방지해야 하는 필요성, 마이그레이션을 수행하는 데 필요한 전반적인 노력 등 여러 요소의 조합에 따라 달라집니다. 대규모 데이터베이스에는 마이그레이션을 위해 더 많은 시간과 노력이 필요하지만, 이와 같은 마이그레이션의 복잡성은 최소화됩니다. 작은 데이터베이스는 신속하게 마이그레이션할 수 있지만, 수천 개의 마이그레이션을 해야 하는 경우 그 규모가 커지면 복잡한 문제가 발생할 수 있습니다. 마지막으로, 데이터베이스가 클수록 비즈니스 크리티컬에 해당할 가능성이 커지므로 백엔드 경로를 유지하면서 다운타임을 최소화해야 합니다.

여기에서는 마이그레이션 전략 계획을 위한 몇 가지 고려 사항에 대해 설명합니다.

데이터 크기입니다

크기가 커토버 시간에 영향을 미칠 필요는 없지만 마이그레이션할 데이터베이스의 크기는 마이그레이션 계획에 분명히 영향을 미칩니다. 대량의 데이터를 마이그레이션해야 하는 경우 주요 고려 사항은 대역폭입니다. 복제 작업은 일반적으로 효율적인 순차적 I/O를 사용하여 수행됩니다 보수적인 추정치로, 복제 작업에 사용 가능한 네트워크 대역폭의 50%를 사용한다고 가정합니다. 예를 들어, 8GB FC 포트는 이론적으로 약 800Mbps를 전송할 수 있습니다. 50%의 활용률을 가정하면 약 400Mbps의 속도로 데이터베이스를 복사할 수 있습니다. 따라서 10TB 데이터베이스를 이 속도로 약 7시간 내에 복사할 수 있습니다.

장거리 마이그레이션을 위해서는 일반적으로 에 설명된 로그 전달 프로세스와 같이 보다 창의적인 접근 방식이 필요합니다 "**온라인 데이터 파일 이동**". 장거리 IP 네트워크는 LAN 또는 SAN 속도와 가까운 곳에서는 거의 대역폭을 가지지 않습니다. 한 예로, NetApp는 아카이브 로그 생성 속도가 매우 높은 220TB 데이터베이스의 장거리 마이그레이션을 지원했습니다. 이 방법은 가능한 최대 대역폭을 제공하기 때문에 데이터 전송을 위해 선택한 방식은 테이프를 매일 배송하는 방식이었습니다.

데이터베이스 수입니다

많은 경우 대량의 데이터를 이동할 때 발생하는 문제는 데이터 크기가 아니라 데이터베이스를 지원하는 구성의 복잡성입니다. 단순히 50TB의 데이터베이스를 마이그레이션해야 한다는 사실을 아는 것만으로는 충분하지 않습니다. 단일 50TB 미션 크리티컬 데이터베이스, 4천 개의 기존 데이터베이스 컬렉션 또는 운영 데이터와 비운영 데이터의 조합이 될 수 있습니다. 경우에 따라 대부분의 데이터는 소스 데이터베이스의 클론으로 구성됩니다. 특히, 새 아키텍처에서 NetApp FlexClone 볼륨을 활용하도록 설계된 경우 이러한 클론을 쉽게 다시 생성할 수 있으므로 이러한 클론을 마이그레이션할 필요가 없습니다.

마이그레이션 계획을 수립하려면 범위에 포함되는 데이터베이스의 수와 우선 순위를 어떻게 지정해야 하는지 파악해야 합니다. 데이터베이스 수가 증가함에 따라 기본 마이그레이션 옵션은 스택에서 더 낮거나 낮은 경향이 있습니다. 예를 들어, RMAN과 짧은 운영 중단으로 단일 데이터베이스를 쉽게 복사할 수 있습니다. 이것이 호스트 수준 복제입니다.

데이터베이스가 50개인 경우 RMAN 복제본을 수신하도록 새 파일 시스템 구조를 설정하는 대신 데이터를 제자리로 이동하는 것이 더 쉬울 수 있습니다. 이 프로세스는 호스트 기반 LVM 마이그레이션을 통해 데이터를 이전 LUN에서 새 LUN으로 재배치하는 방법으로 수행할 수 있습니다. 이렇게 하면 DBA(데이터베이스 관리자) 팀에서 OS 팀으로 역할이 이전되고, 결과적으로 데이터가 데이터베이스와 관련하여 투명하게 마이그레이션됩니다. 파일 시스템 구성이 변경되지 않았습니다.

끝으로, 서버 200대에서 500개의 데이터베이스를 마이그레이션해야 하는 경우 ONTAP FLI(Foreign LUN Import)

기능과 같은 스토리지 기반 옵션을 사용하여 LUN을 직접 마이그레이션할 수 있습니다.

재건축 요구 사항

일반적으로 새 스토리지 어레이의 기능을 활용하기 위해 데이터베이스 파일 레이아웃을 변경해야 하지만 항상 그렇지 않습니다. 예를 들어, EF-Series All-Flash 어레이의 기능은 주로 SAN 성능과 SAN 안정성을 최우선으로 합니다. 대부분의 경우 데이터 레이아웃과 관련하여 특별한 고려 사항 없이 데이터베이스를 EF-Series 어레이로 마이그레이션할 수 있습니다. 높은 IOPS, 짧은 지연 시간 및 강력한 안정성만 필요합니다. RAID 구성 또는 Dynamic Disk Pool과 같은 요소와 관련된 모범 사례가 있지만 EF-Series 프로젝트에서 이러한 기능을 활용하기 위해 전체 스토리지 아키텍처를 크게 변경할 필요는 없습니다.

반면, ONTAP로 마이그레이션하려면 일반적으로 최종 구성이 최대한의 가치를 제공할 수 있도록 데이터베이스 레이아웃을 좀 더 고려해야 합니다. ONTAP는 그 자체로 특별한 아키텍처 노력 없이 데이터베이스 환경에 많은 기능을 제공합니다. 가장 중요한 것은 현재 하드웨어의 수명이 다할 때 새 하드웨어로 중단 없이 마이그레이션할 수 있는 기능을 제공한다는 것입니다. 일반적으로 ONTAP로의 마이그레이션은 수행해야 하는 마지막 마이그레이션입니다. 후속 하드웨어가 업그레이드되고 데이터가 중단 없이 새로운 미디어로 마이그레이션됩니다.

어떤 계획을 세우면 훨씬 더 많은 혜택을 누릴 수 있습니다. 가장 중요한 고려 사항은 스냅샷 사용과 관련됩니다. 스냅샷은 거의 즉각적인 백업, 복원 및 클론 복제 작업을 수행하기 위한 기반입니다. 스냅샷 성능의 예로, 알려진 가장 큰 용도는 6개의 컨트롤러에서 약 250개의 LUN에서 실행되는 996TB의 단일 데이터베이스를 사용하는 것입니다. 이 데이터베이스는 2분 내에 백업되고 2분 내에 복원되며 15분 내에 복제될 수 있습니다. 추가 이점으로는 워크로드의 변화에 대응하여 클러스터 주변으로 데이터를 이동하는 기능과 QoS(서비스 품질) 제어 애플리케이션을 통해 다중 데이터베이스 환경에서 양호하고 일관된 성능을 제공하는 기능이 있습니다.

QoS 제어, 데이터 재배치, 스냅샷, 클론 복제 등의 기술은 거의 모든 구성에서 작동합니다. 그러나 일반적으로 이점을 극대화하기 위해서는 몇 가지 생각이 필요합니다. 새로운 스토리지 어레이에 대한 투자를 최대화하기 위해 데이터베이스 스토리지 레이아웃을 변경해야 하는 경우도 있습니다. 호스트 기반 또는 스토리지 기반 마이그레이션은 원본 데이터 레이아웃을 복제하므로 이러한 설계 변경은 마이그레이션 전략에 영향을 미칠 수 있습니다. 마이그레이션을 완료하고 ONTAP에 최적화된 데이터 레이아웃을 제공하기 위해 추가 단계가 필요할 수 있습니다. 예 나와 있는 절차 "[Oracle 마이그레이션 절차 개요](#)" 그리고 나중에 데이터베이스를 마이그레이션하는 것뿐만 아니라 최소한의 노력으로 최적의 최종 레이아웃으로 마이그레이션하는 몇 가지 방법을 보여 줍니다.

컷오버 시간

컷오버 중에 허용되는 최대 서비스 중단 시간을 결정해야 합니다. 전체 마이그레이션 프로세스로 인해 운영이 중단된다고 생각하는 일반적인 실수입니다. 서비스 중단이 시작되기 전에 다양한 작업을 완료할 수 있으며, 다양한 옵션을 통해 운영 중단 또는 운영 중단 없이 마이그레이션을 완료할 수 있습니다. 불가피한 운영 중단이 불가피한 경우에도 컷오버 시간의 기간이 절차마다 다르므로 허용되는 최대 서비스 중단 시간을 정의해야 합니다.

예를 들어 10TB 데이터베이스를 복사하려면 일반적으로 약 7시간이 소요됩니다. 비즈니스 요구 사항이 7시간 동안 중단되는 경우 파일 복사는 쉽고 안전한 마이그레이션 옵션입니다. 5시간이 허용되지 않는 경우 간단한 로그 전달 프로세스를 수행합니다(참조 "[Oracle 로그 전달](#)")를 최소한의 노력으로 설정하여 컷오버 시간을 약 15분으로 단축할 수 있습니다. 이 시간 동안 데이터베이스 관리자가 이 프로세스를 완료할 수 있습니다. 15분이 허용되는 경우 스크리핑을 통해 최종 컷오버 프로세스를 자동화하여 컷오버 시간을 단 몇 분으로 단축할 수 있습니다. 언제든지 마이그레이션의 속도를 높일 수 있지만 시간과 노력을 들여야 합니다. 컷오버 시간 목표는 비즈니스에 허용되는 성과를 기준으로 해야 합니다.

뒤로 이동 경로

마이그레이션 시 위험을 완전히 없앨 수는 없습니다. 기술이 완벽하게 작동하더라도 사용자 오류가 발생할 가능성이 항상 있습니다. 선택한 마이그레이션 경로와 관련된 위험은 실패한 마이그레이션의 결과와 함께 고려해야 합니다. 예를 들어, Oracle ASM의 투명한 온라인 스토리지 마이그레이션 기능은 주요 기능 중 하나이며 이 방법은 가장 신뢰할 수 있는 기능 중 하나입니다. 그러나 이 메서드를 사용하여 데이터를 복구할 수 없는 방식으로 복사하고 있습니다. 드문

경우지만 ASM에서 문제가 발생하는 경우에는 쉬운 백아웃 경로가 없습니다. 유일한 옵션은 원래 환경을 복원하거나 ASM을 사용하여 마이그레이션을 원래 LUN으로 되돌리는 것입니다. 시스템에서 이러한 작업을 수행할 수 있다고 가정하면 원래 스토리지 시스템에서 스냅샷 유형 백업을 수행하면 위험이 최소화될 수 있지만 제거되지는 않습니다.

예행 연습

일부 마이그레이션 절차는 실행 전에 완전히 검증되어야 합니다. 마이그레이션 및 전환 프로세스의 예행 연습은 마이그레이션을 성공적으로 수행하고 다운타임을 최소화해야 하는 미션 크리티컬 데이터베이스에 대한 일반적인 요청입니다. 또한 사용자 수용 테스트는 마이그레이션 후 작업의 일부로 포함되는 경우가 많으며 이러한 테스트가 완료된 후에만 전체 시스템을 운영 환경으로 되돌릴 수 있습니다.

예행 연습이 필요한 경우 몇 가지 ONTAP 기능을 통해 프로세스를 훨씬 쉽게 수행할 수 있습니다. 특히 스냅샷은 테스트 환경을 재설정하고 데이터베이스 환경의 공간 효율적인 여러 복제본을 신속하게 생성할 수 있습니다.

절차를 참조하십시오

Oracle 마이그레이션 절차 개요

Oracle 마이그레이션 데이터베이스에 다양한 절차를 사용할 수 있습니다. 올바른 운영 체제는 비즈니스 요구 사항에 달려 있습니다.

대부분의 경우 시스템 관리자와 DBA는 물리적 볼륨 데이터를 재배포하거나, 미러링 및 방해 요소를 제거하거나, Oracle RMAN을 활용하여 데이터를 복제하는 방법을 선호하고 있습니다.

이러한 절차는 사용 가능한 옵션 중 일부에 익숙하지 않은 IT 직원을 위한 지침으로 주로 제공됩니다. 또한 절차를 통해 각 마이그레이션 방법에 대한 작업, 시간 요구사항 및 기술 집합 요구를 파악할 수 있습니다. 이를 통해 NetApp 및 파트너 프로페셔널 서비스 또는 IT 관리 부서와 같은 타사는 각 절차의 요구사항을 보다 완벽하게 이해할 수 있습니다.

마이그레이션 전략을 만드는 단일 모범 사례는 없습니다. 계획을 작성하려면 먼저 가용성 옵션을 파악한 다음 비즈니스 요구에 가장 적합한 방법을 선택해야 합니다. 아래 그림은 기본적인 고려 사항과 고객이 내린 일반적인 결론을 보여 주지만 모든 상황에 보편적으로 적용되는 것은 아닙니다.

예를 들어 한 단계만 거치면 전체 데이터베이스 크기에 대한 문제가 발생합니다. 다음 단계는 데이터베이스가 1TB 이하인지 여부에 따라 달라집니다. 권장 단계는 일반적인 고객 사례를 기반으로 한 권장 사항입니다. 대부분의 고객은 DataGuard를 사용하여 작은 데이터베이스를 복사하지 않을 수도 있지만 일부 고객은 그렇지 않을 수도 있습니다. 대부분의 고객은 시간이 필요하기 때문에 50TB 데이터베이스를 복사하지 않으려고 하지만 어떤 고객은 이 작업을 허용할 만큼 충분한 유지 관리 기간이 있을 수 있습니다.

가장 적합한 마이그레이션 경로에 대한 고려 사항 유형의 순서도를 찾을 수 있습니다 ["여기"](#).

온라인 데이터 파일 이동

Oracle 12cR1 이상에는 데이터베이스가 온라인 상태일 때 데이터 파일을 이동하는 기능이 포함되어 있습니다. 또한 서로 다른 파일 시스템 유형 간에 작동합니다. 예를 들어 데이터 파일을 xfs 파일 시스템에서 ASM으로 재배포할 수 있습니다. 이 방법은 필요할 수 있는 개별 데이터 파일 이동 작업의 수로 인해 일반적으로 대규모로 사용되지 않지만 데이터 파일이 적은 작은 데이터베이스에서는 고려할 만한 옵션입니다.

또한 데이터 파일을 단순히 이동하는 것이 기존 데이터베이스의 부분을 마이그레이션하는 좋은 방법입니다. 예를 들어, 유틸 블록을 오브젝트 저장소에 저장할 수 있는 FabricPool 볼륨과 같이 사용량이 적은 데이터 파일을 더 비용 효율적인 스토리지로 재배포할 수 있습니다.

데이터베이스 수준에서 마이그레이션하면 데이터베이스가 데이터를 재배포할 수 있습니다. 특히 로그 전송을 의미합니다. RMAN 및 ASM 같은 기술은 Oracle 제품이지만, 마이그레이션을 위해 파일을 복제하고 볼륨을 관리하는 호스트 레벨에서 작동합니다.

로그 전달

데이터베이스 수준 마이그레이션의 기반은 Oracle 아카이브 로그이며, 여기에는 데이터베이스의 변경 사항 로그가 포함됩니다. 대부분의 경우 아카이브 로그는 백업 및 복구 전략의 일부입니다. 복구 프로세스는 데이터베이스 복원으로 시작한 다음 하나 이상의 아카이브 로그를 재생하여 데이터베이스를 원하는 상태로 만듭니다. 이와 동일한 기본 기술을 사용하여 운영 중단이 거의 또는 전혀 없는 마이그레이션을 수행할 수 있습니다. 더욱 중요한 점은 이 기술을 통해 원래 데이터베이스를 그대로 유지하면서 마이그레이션을 수행할 수 있으므로 백업 경로가 유지됩니다.

마이그레이션 프로세스는 데이터베이스 백업을 보조 서버로 복원하는 것부터 시작됩니다. 다양한 방법으로 그렇게 할 수 있지만 대부분의 고객은 일반 백업 애플리케이션을 사용하여 데이터 파일을 복원합니다. 데이터 파일이 복원되면 사용자가 로그 전달 방법을 설정합니다. 기본 데이터베이스에서 생성된 아카이브 로그의 지속적인 피드를 만들고 복원된 데이터베이스에서 다시 재생하여 두 로그 모두 동일한 상태에 가깝게 유지하는 것이 목표입니다. 전환 시간이 되면 소스 데이터베이스가 완전히 종료되고 최종 아카이브 로그가 복사되고 경우에 따라 재실행 로그가 재생됩니다. 리두 로그에는 커밋된 최종 트랜잭션이 포함될 수 있으므로 리두 로그도 고려해야 합니다.

이러한 로그를 전송하고 재생한 후에는 두 데이터베이스가 서로 일관됩니다. 이 시점에서 대부분의 고객은 몇 가지 기본 테스트를 수행합니다. 마이그레이션 프로세스 중에 오류가 발생하면 로그 재생에서 오류를 보고하고 실패합니다. 알려진 쿼리 또는 응용 프로그램 기반 작업을 기반으로 몇 가지 빠른 테스트를 수행하여 구성이 최적화되었는지 확인하는 것이 좋습니다. 또한 원래 데이터베이스를 종료하기 전에 최종 테스트 테이블을 하나 만들어 마이그레이션된 데이터베이스에 있는지 확인하는 것이 일반적입니다. 이 단계를 수행하면 최종 로그 동기화 중에 오류가 발생하지 않습니다.

단순한 로그 전달 마이그레이션은 원본 데이터베이스와 관련하여 대역 외 방식으로 구성할 수 있으므로 업무상 중요한 데이터베이스에 특히 유용합니다. 소스 데이터베이스에 대한 구성 변경이 필요하지 않으며 마이그레이션 환경의 복원 및 초기 구성은 운영 작업에 영향을 미치지 않습니다. 로그 전달이 구성된 후 운영 서버에 일부 입출력 요구 사항이 배치됩니다. 그러나 로그 전달은 아카이브 로그의 단순 순차 읽기로 구성되므로 운영 데이터베이스 성능에 영향을 미칠 가능성은 낮습니다.

로그 전달은 장거리, 높은 변경률 마이그레이션 프로젝트에 특히 유용한 것으로 입증되었습니다. 한 예로, 하나의 220TB 데이터베이스가 약 500마일 떨어진 새로운 위치로 마이그레이션되었습니다. 변경률이 매우 높았고 보안 제한으로 인해 네트워크 연결이 사용되지 않았습니다. 로그 배송은 테이프 및 택배사를 사용하여 수행되었습니다. 소스 데이터베이스의 복제본은 아래에 설명된 절차를 사용하여 초기에 복원되었습니다. 그런 다음 최종 테이프 세트가 제공되고 로그가 복제 데이터베이스에 적용된 전환 시간까지 택배사에 의해 로그가 매주 배송되었습니다.

Oracle DataGuard

경우에 따라 전체 DataGuard 환경이 보장됩니다. DataGuard라는 용어를 사용하여 로그 전달 또는 대기 데이터베이스 구성을 참조하는 것은 올바르지 않습니다. Oracle DataGuard는 데이터베이스 복제 관리를 위한 포괄적인 프레임워크이지만 복제 기술은 아닙니다. 마이그레이션 과정에서 완전한 DataGuard 환경의 주된 이점은 한 데이터베이스에서 다른 데이터베이스로 투명하게 전환하는 것입니다. 또한 새로운 환경의 성능 또는 네트워크 연결 문제와 같은 문제가 발견될 경우 Dataguard는 원래 데이터베이스로 투명하게 전환할 수 있습니다. 완전히 구성된 DataGuard 환경에서는 애플리케이션이 기본 데이터베이스 위치의 변경을 감지할 수 있도록 데이터베이스 계층뿐만 아니라 응용 프로그램도 구성해야 합니다. 일반적으로 DataGuard를 사용하여 마이그레이션을 완료할 필요는 없지만 일부 고객은 내부에서 광범위한 DataGuard 전문 지식을 보유하고 있으며 마이그레이션 작업에 이미 의존하고 있습니다.

재건축

앞서 설명한 것처럼 스토리지 어레이의 고급 기능을 활용하려면 데이터베이스 레이아웃을 변경해야 하는 경우가 있습니다. 또한 ASM에서 NFS 파일 시스템으로 이동하는 것과 같은 스토리지 프로토콜이 변경될 경우 파일 시스템 레이아웃이 변경될 수도 있습니다.

DataGuard를 비롯한 로그 전달 방법의 주요 이점 중 하나는 복제 대상이 소스와 일치하지 않아도 된다는 것입니다. 로그 전달 방식을 사용하여 ASM에서 일반 파일 시스템으로 마이그레이션하거나 그 반대로 마이그레이션하는 데 문제가 없습니다. 대상 위치에서 데이터 파일의 정확한 레이아웃을 변경하여 플러그형 데이터베이스(PDB) 기술의 사용을 최적화하거나 특정 파일에 대해 선택적으로 QoS 제어를 설정할 수 있습니다. 즉, 로그 전달을 기반으로 하는 마이그레이션 프로세스를 통해 데이터베이스 스토리지 레이아웃을 쉽고 안전하게 최적화할 수 있습니다.

서버 리소스

데이터베이스 수준 마이그레이션의 한 가지 제한 사항은 보조 서버의 필요성입니다. 이 두 번째 서버를 사용하는 방법에는 두 가지가 있습니다.

1. 두 번째 서버를 데이터베이스의 영구적인 새 홈으로 사용할 수 있습니다.
2. 두 번째 서버를 임시 스테이징 서버로 사용할 수 있습니다. 새 스토리지로의 데이터 마이그레이션이 완료되고 테스트된 후 LUN 또는 NFS 파일 시스템이 스테이징 서버에서 분리되어 원래 서버에 다시 연결됩니다.

첫 번째 옵션은 가장 쉽지만 매우 강력한 서버가 필요한 대규모 환경에서는 이 옵션을 사용하는 것이 불가능할 수 있습니다. 두 번째 옵션은 파일 시스템을 원래 위치로 다시 재배치하기 위해 추가 작업이 필요합니다. 이는 파일 시스템을 스테이징 서버에서 마운트 해제하고 원래 서버에 다시 마운트할 수 있기 때문에 NFS를 스토리지 프로토콜로 사용하는 간단한 작업입니다.

블록 기반 파일 시스템은 FC 조닝 또는 iSCSI 이니시에이터를 업데이트하기 위해 추가 작업이 필요합니다. 대부분의 논리적 볼륨 관리자(ASM 포함)에서는 원래 서버에서 LUN을 사용할 수 있게 되면 LUN이 자동으로 감지되어 온라인 상태로 전환됩니다. 그러나 일부 파일 시스템 및 LVM 구현에서는 데이터를 내보내고 가져오는 데 더 많은 작업이 필요할 수 있습니다. 정확한 절차는 다양할 수 있지만 일반적으로 마이그레이션을 완료하고 원래 서버에서 데이터를 다시 저장하는 간단하고 반복 가능한 절차를 설정하는 것이 쉽습니다.

단일 서버 환경 내에서 로그 전달을 설정하고 데이터베이스를 복제할 수 있지만 새 인스턴스에는 로그를 재생하기 위한 다른 프로세스 SID가 있어야 합니다. SID가 다른 프로세스 ID의 다른 집합에서 데이터베이스를 임시로 가져온 후 나중에 변경할 수 있습니다. 하지만 이렇게 하면 복잡한 관리 작업이 많이 발생할 수 있으며 데이터베이스 환경에 사용자 오류가 발생할 위험이 있습니다.

호스트 레벨 마이그레이션

호스트 레벨에서 데이터를 마이그레이션한다는 것은 호스트 운영 체제와 관련 유틸리티를 사용하여 마이그레이션을 완료하는 것을 의미합니다. 이 프로세스에는 Oracle RMAN 및 Oracle ASM을 비롯하여 데이터를 복사하는 모든 유틸리티가 포함됩니다.

데이터 복사

단순 복사 작업의 값은 과소 평가되지 않아야 합니다. 오늘날의 네트워크 인프라는 초당 기가바이트 단위의 속도로 데이터를 이동할 수 있으며 파일 복사 작업은 효율적인 순차적 읽기 및 쓰기 I/O를 기반으로 합니다. 로그 전달과 비교할 때 호스트 복제 작업에서 더 많은 중단이 불가피하지만 마이그레이션은 단순한 데이터 이동 그 이상입니다. 여기에는 일반적으로 네트워킹, 데이터베이스 재시작 시간 및 마이그레이션 후 테스트 변경 사항이 포함됩니다.

데이터를 복사하는 데 필요한 실제 시간은 중요하지 않을 수 있습니다. 또한 원본 데이터를 그대로 유지하므로 복제 작업은 보장된 백아웃 경로를 유지합니다. 마이그레이션 프로세스 중에 문제가 발생하면 원본 데이터가 있는 원본 파일 시스템을 다시 활성화할 수 있습니다.

플랫폼 변경

플랫폼 변경이란 CPU 유형의 변경을 의미합니다. 데이터베이스를 기존 Solaris, AIX 또는 HP-UX 플랫폼에서 x86 Linux로 마이그레이션할 경우 CPU 아키텍처의 변경으로 인해 데이터를 다시 포맷해야 합니다. SPARC, IA64 및 전원 CPU는 빅 엔디언 프로세서라고 하는 반면 x86 및 x86_64 아키텍처는 리틀 엔디언이라고 합니다. 따라서 Oracle 데이터 파일 내의 일부 데이터는 사용 중인 프로세서에 따라 순서가 다르게 지정됩니다.

기존에는 DataPump를 사용하여 플랫폼 간에 데이터를 복제해 왔습니다. 데이터 덤프는 대상 데이터베이스에서 보다 빠르게 가져올 수 있는 특수한 유형의 논리적 데이터 내보내기를 만드는 유틸리티입니다. DataPump 는 데이터의 논리적 복사본을 만들기 때문에 프로세서 엔디언의 종속성을 남깁니다. 데이터덤프는 여전히 일부 고객이 플랫폼 재구축을 위해 사용하고 있지만 Oracle 11g에서는 더욱 빠른 옵션인 교차 플랫폼 전송 테이블스페이스를 사용할 수 있게 되었습니다. 이렇게 하면 테이블스페이스를 다른 엔디언 형식으로 변환할 수 있습니다. 이것은 물리적 바이트를 논리적 데이터로 변환한 다음 다시 물리적 바이트로 변환해야 하는 DataPump 내보내기보다 더 나은 성능을 제공하는 물리적 변환입니다.

DataPump 및 이식 가능한 테이블스페이스에 대한 자세한 내용은 NetApp 설명서를 참조하십시오. 하지만 NetApp는 새로운 CPU 아키텍처를 사용하여 새 스토리지 시스템 로드로 마이그레이션할 때 고객을 지원하는 경험을 바탕으로 몇 가지 권장 사항을 제시합니다.

- DataPump를 사용 중인 경우 마이그레이션을 완료하는 데 필요한 시간을 테스트 환경에서 측정해야 합니다. 고객은 마이그레이션을 완료하는 데 필요한 시간에 놀라기도 합니다. 이와 같이 예기치 않은 추가 다운타임은 운영 중단을 일으킬 수 있습니다.
- 많은 고객들이 교차 플랫폼 전송 가능 테이블스페이스는 데이터 변환이 필요하지 않다고 잘못 생각합니다. 엔디언이 다른 CPU를 사용하는 경우 RMAN이 사용됩니다 convert 데이터 파일에 대한 작업은 미리 수행해야 합니다. 이것은 즉각적인 작업이 아닙니다. 경우에 따라 서로 다른 데이터 파일에서 여러 스레드가 작동하므로 변환 프로세스가 빨라질 수 있지만 변환 프로세스를 피할 수는 없습니다.

논리적 볼륨 관리자 기반 마이그레이션

LVM은 하나 이상의 LUN 그룹을 만들어 일반적으로 익스텐트라고 하는 작은 단위로 분할하는 방식으로 작동합니다. 그런 다음 익스텐트 풀이 기본적으로 가상화된 논리적 볼륨을 생성하기 위한 소스로 사용됩니다. 이 가상화 계층은 다음과 같은 다양한 방식으로 가치를 제공합니다.

- 논리적 볼륨은 여러 LUN에서 그린 익스텐트를 사용할 수 있습니다. 논리적 볼륨에 파일 시스템을 생성할 때 모든 LUN의 전체 성능을 사용할 수 있습니다. 또한 볼륨 그룹에 모든 LUN의 로드가 짝수일 뿐이므로 성능이 더욱 예측 가능합니다.
- 논리적 볼륨의 크기는 익스텐트를 추가하거나 경우에 따라 제거할 수 있습니다. 논리적 볼륨에서 파일 시스템의 크기를 조정하는 작업은 일반적으로 중단되지 않습니다.
- 기본 익스텐트를 이동하여 논리적 볼륨을 운영 중단 없이 마이그레이션할 수 있습니다.

LVM을 사용한 마이그레이션은 익스텐트 이동 또는 익스텐트 미러링/디머러링의 두 가지 방법 중 하나로 작동합니다. LVM 마이그레이션은 효율적인 대규모 블록 순차적 I/O를 사용하며 성능 문제는 거의 발생하지 않습니다. 이 문제가 발생할 경우 일반적으로 I/O 속도를 제한하는 옵션이 있습니다. 이렇게 하면 마이그레이션을 완료하는 데 필요한 시간이 길어지고 호스트 및 스토리지 시스템의 I/O 부담이 줄어듭니다.

미러 및 미러

AIX LVM과 같은 일부 볼륨 관리자는 사용자가 각 익스텐트의 복제본 수를 지정하고 각 복제본을 호스팅하는 디바이스를 제어할 수 있도록 합니다. 마이그레이션은 기존의 논리적 볼륨을 만들고 기본 익스텐트를 새 볼륨에 미러링하고 복사본이 동기화될 때까지 기다린 다음 이전 복사본을 삭제하여 수행됩니다. 백업 경로가 필요한 경우 미러 복사본이 삭제되기 전에 원본 데이터의 스냅샷을 생성할 수 있습니다. 또는 포함된 미러 복제본을 강제로 삭제하기 전에 서버를 잠시 종료하여 원래 LUN을 마스킹할 수 있습니다. 이렇게 하면 복구 가능한 데이터 복사본이 원래 위치에

보존됩니다.

익스텐트 마이그레이션

거의 모든 볼륨 관리자는 익스텐트의 마이그레이션을 허용하며 경우에 따라서는 여러 옵션이 존재하기도 합니다. 예를 들어 일부 볼륨 관리자에서는 관리자가 특정 논리적 볼륨의 개별 익스텐트를 이전 스토리지에서 새 스토리지로 재배치할 수 있습니다. Linux LVM2와 같은 볼륨 관리자는 `pvmove` 지정한 LUN 디바이스의 모든 익스텐트를 새 LUN으로 재배치하는 명령입니다. 이전 LUN을 이동한 후 제거할 수 있습니다.



운영 시 가장 큰 위험은 구성에서 사용되지 않은 오래된 LUN을 제거하는 것입니다. FC 조닝을 변경하고 오래된 LUN 디바이스를 제거할 때는 특히 주의해야 합니다.

Oracle 자동 스토리지 관리

Oracle ASM은 논리 볼륨 관리자와 파일 시스템이 결합된 시스템입니다. 상위 수준에서 Oracle ASM은 LUN 모음을 가져와 작은 할당 단위로 분할하고 ASM 디스크 그룹이라고 하는 단일 볼륨으로 제공합니다. ASM에는 이중화 수준을 설정하여 디스크 그룹을 미러링하는 기능도 포함되어 있습니다. 볼륨은 미러링되지 않은(외부 중복), 미러링(일반 중복) 또는 3웨이 미러링(높은 중복)일 수 있습니다. 이중화 수준은 생성 후 변경할 수 없기 때문에 설정 시 주의해야 합니다.

ASM은 파일 시스템 기능도 제공합니다. 파일 시스템이 호스트에서 직접 표시되지 않지만 Oracle 데이터베이스는 ASM 디스크 그룹에서 파일과 디렉토리를 생성, 이동 및 삭제할 수 있습니다. 또한 `asmcmd` 유틸리티를 사용하여 구조를 탐색할 수도 있습니다.

다른 LVM 구현과 마찬가지로 Oracle ASM은 사용 가능한 모든 LUN에서 각 파일의 I/O를 스트라이핑 및 로드 밸런싱을 통해 I/O 성능을 최적화합니다. 둘째, 기본 익스텐트를 재배치하여 ASM 디스크 그룹의 크기 조정과 마이그레이션을 모두 수행할 수 있습니다. Oracle ASM은 재조정 작업을 통해 프로세스를 자동화합니다. 새로운 LUN이 ASM 디스크 그룹에 추가되고 기존 LUN이 삭제되어 익스텐트 재배치와 디스크 그룹에서 제거된 LUN의 후속 드롭이 트리거됩니다. 이 프로세스는 가장 검증된 마이그레이션 방법 중 하나이며, 투명한 마이그레이션을 제공하는 ASM의 신뢰성이 가장 중요한 기능일 수 있습니다.



Oracle ASM의 미러링 수준은 고정되어 있으므로 미러 및 미러 마이그레이션 방법과 함께 사용할 수 없습니다.

스토리지 레벨 마이그레이션

스토리지 수준 마이그레이션은 애플리케이션 및 운영 체제 수준 모두에서 마이그레이션을 수행하는 것을 의미합니다. 과거에는 네트워크 수준에서 LUN을 복제할 특수 장치를 사용하기도 했지만 이제는 ONTAP에서 기본적으로 제공하는 이러한 기능을 사용할 수 있습니다.

SnapMirror를 참조하십시오

NetApp 시스템 간 데이터베이스 마이그레이션은 NetApp SnapMirror 데이터 복제 소프트웨어를 통해 거의 보편적으로 수행됩니다. 이 프로세스에는 마이그레이션할 볼륨의 미러 관계를 설정하고 볼륨이 동기화될 수 있도록 한 다음 컷오버 기간을 기다리는 작업이 포함됩니다. 소스 데이터베이스가 도착하면 소스 데이터베이스가 종료되고 최종 미러 업데이트가 한 번 수행되며 미러가 중단됩니다. 그러면 포함된 NFS 파일 시스템 디렉토리를 마운트하거나 포함된 LUN을 검색하고 데이터베이스를 시작하여 복제본 볼륨을 사용할 수 있습니다.

단일 ONTAP 클러스터 내에서 볼륨을 재배치하는 것은 마이그레이션으로 간주되는 것이 아니라 일상적인 마이그레이션으로 간주됩니다 `volume move` 작동. SnapMirror는 클러스터 내의 데이터 복제 엔진으로 사용됩니다. 이 프로세스는 완전히 자동화되어 있습니다. LUN 매핑이나 NFS 익스포트 권한과 같은 볼륨 특성을 볼륨 자체와 함께 이동할 때 수행해야 할 추가 마이그레이션 단계는 없습니다. 재할당은 호스트 작업의 중단 없이 수행됩니다. 경우에 따라 새로 재배치된 데이터에 가장 효율적인 방식으로 액세스할 수 있도록 네트워크 액세스를 업데이트해야 하지만, 이러한

작업은 중단되지 않습니다.

FLI(Foreign LUN Import)

FLI는 8.3 이상을 실행하는 Data ONTAP 시스템에서 다른 스토리지 어레이의 기존 LUN을 마이그레이션할 수 있는 기능입니다. 절차는 간단합니다. ONTAP 시스템은 다른 SAN 호스트처럼 기존 스토리지 시스템에 조닝됩니다. 그런 다음 Data ONTAP는 원하는 레거시 LUN을 제어하고 기본 데이터를 마이그레이션합니다. 또한 가져오기 프로세스에서는 데이터가 마이그레이션될 때 새 볼륨의 효율성 설정을 사용합니다. 즉, 마이그레이션 프로세스 중에 데이터를 인라인으로 압축 및 중복제거할 수 있습니다.

Data ONTAP 8.3에서 FLI를 처음 구현하면 오프라인 마이그레이션만 허용되었습니다. 이는 매우 빠른 전송이었지만 마이그레이션이 완료될 때까지 LUN 데이터를 사용할 수 없다는 것을 의미합니다. 온라인 마이그레이션은 Data ONTAP 8.3.1에서 도입되었습니다. 이러한 종류의 마이그레이션은 전송 프로세스 중에 ONTAP에서 LUN 데이터를 제공할 수 있으므로 작업 중단이 최소화됩니다. ONTAP를 통해 LUN을 사용하도록 호스트를 다시 조닝하는 동안 중단이 짧게 발생합니다. 그러나 이러한 변경이 이루어지면 데이터에 다시 액세스할 수 있고 마이그레이션 프로세스 내내 계속 액세스할 수 있습니다.

읽기 입출력은 복제 작업이 완료될 때까지 ONTAP를 통해 프록시되고 쓰기 입출력은 외부 및 ONTAP LUN 모두에 동기식으로 기록됩니다. 관리자가 전체 컷오버를 실행하여 외부 LUN을 해제하고 더 이상 쓰기를 복제하지 않는 한 두 LUN 복사본이 이 방식으로 동기화된 상태로 유지됩니다.

FLI는 FC와 함께 사용하도록 설계되었지만 iSCSI로 변경하려는 경우 마이그레이션이 완료된 후 마이그레이션된 LUN을 iSCSI LUN으로 쉽게 다시 매핑할 수 있습니다.

FLI의 기능 중 하나는 자동 정렬 감지 및 조정입니다. 여기서 정렬이란 LUN 장치의 파티션을 의미합니다. 최적의 성능을 얻으려면 I/O를 4K 블록에 맞춰 정렬해야 합니다. 파티션이 4K의 배수가 아닌 오프셋에 배치되면 성능이 저하됩니다.

정렬의 두 번째 측면은 파티션 오프셋을 조정하여 수정할 수 없는 파일 시스템 블록 크기입니다. 예를 들어, ZFS 파일 시스템의 기본 내부 블록 크기는 512바이트입니다. AIX를 사용하는 다른 고객은 512 또는 1,024바이트 블록 크기의 JFS2 파일 시스템을 생성하는 경우가 있습니다. 파일 시스템이 4K 경계에 맞춰 정렬될 수 있지만 해당 파일 시스템 내에서 생성된 파일은 그렇지 않고 성능이 저하됩니다.

FLI는 이러한 상황에서 사용해서는 안 됩니다. 마이그레이션 후에 데이터에 액세스할 수 있지만 이로 인해 파일 시스템의 성능이 심각하게 제한됩니다. 일반적으로 ONTAP에서 랜덤 덮어쓰기 워크로드를 지원하는 모든 파일 시스템은 4K 블록 크기를 사용해야 합니다. 이 워크로드는 데이터베이스 데이터 파일 및 VDI 구축과 같은 워크로드에 주로 적용됩니다. 블록 크기는 관련 호스트 운영 체제 명령을 사용하여 확인할 수 있습니다.

예를 들어, AIX에서는 블록 크기를 로 볼 수 있습니다 `lsfs -q`. Linux를 사용하면 `xfstool` 및 `tune2fs`에 사용할 수 있습니다 `xfstool` 및 `ext3/ext4` 있습니다. 와 함께 `zfs` 명령은 `zdb -C`.

블록 크기를 제어하는 매개 변수는 `ashift`입니다. 일반적으로 기본값은 9이며, 이는 2의 9 또는 512바이트를 의미합니다. 최적의 성능을 위해 `ashift` 값은 12(2-12=4K)여야 합니다. 이 값은 `zpool`이 생성될 때 설정되며 변경할 수 없습니다. 즉, 가 포함된 데이터 `zpool`이 됩니다 `ashift` 12가 아닌 경우 데이터를 새로 생성된 `zpool`으로 마이그레이션해야 합니다.

Oracle ASM은 기본 블록 크기를 가지고 있지 않습니다. 유일한 요구 사항은 ASM 디스크가 구축된 파티션이 올바르게 정렬되어야 한다는 것입니다.

7-Mode 전환 툴

7MTT(7-Mode 전환 툴)는 대규모 7-Mode 구성을 ONTAP로 마이그레이션하는 데 사용되는 자동화 유틸리티입니다. 대부분의 데이터베이스 고객은 전체 스토리지 공간을 재배치하지 않고 데이터베이스를 기준으로 환경을 마이그레이션하므로 다른 방법을 더욱 쉽게 찾을 수 있습니다. 또한 데이터베이스는 대규모 스토리지 환경에 포함되는

경우가 많습니다. 따라서 데이터베이스는 종종 개별적으로 마이그레이션되며, 7MTT를 사용하여 나머지 환경을 이동할 수 있습니다.

복잡한 데이터베이스 환경을 위한 스토리지 시스템을 보유한 고객 수는 소규모지만 상당수가 있습니다. 이러한 환경에는 많은 볼륨, 스냅샷 및 내보내기 권한, LUN 이니시에이터 그룹, 사용자 권한 및 Lightweight Directory Access Protocol 구성과 같은 수많은 구성 세부 정보가 포함될 수 있습니다. 이런 경우에는 7MTT의 자동화 기능을 사용하여 마이그레이션을 단순화할 수 있습니다.

7MTT는 다음 2가지 모드 중 하나로 작동할 수 있습니다.

- * CBT(Copy-Based Transition). * CBT를 사용하는 7MTT는 새로운 환경의 기존 7-Mode 시스템에서 SnapMirror 볼륨을 설정합니다. 데이터가 동기화되면 7MTT가 컷오버 프로세스를 오케스트레이션합니다.
- * CFT(Copy-Free Transition) * CFT를 지원하는 7MTT는 기존 7-Mode 디스크 쉘프의 데이터 이동 없이 변환을 기반으로 합니다. 데이터는 복사되지 않으며 기존 디스크 쉘프를 재사용할 수 있습니다. 기존 데이터 보호 및 스토리지 효율성 구성이 그대로 유지됩니다.

이 두 옵션 간의 주된 차이점은 복사가 필요 없는 전환은 원래의 7-Mode HA 쌍에 연결된 모든 디스크 쉘프를 새로운 환경으로 재배포해야 하는 큰 방식이라는 것입니다. 쉘프의 하위 집합을 이동할 수 있는 옵션은 없습니다. 복사 기반 접근 방식에서는 선택한 볼륨을 이동할 수 있습니다. 또한 디스크 쉘프를 재구성하고 메타데이터를 변환하는 데 연결된 연결이 필요하므로 무복사 전환으로 컷오버 기간도 길어질 수 있습니다. 현장 경험에 비추어 볼 때, NetApp는 디스크 쉘프를 재배포하고 재설정하는 데 1시간, 메타데이터 변환에 15분에서 2시간 동안 사용할 것을 권장합니다.

Oracle 데이터 파일 마이그레이션

단일 명령으로 개별 Oracle 데이터 파일을 이동할 수 있습니다.

예를 들어, 다음 명령은 파일 시스템에서 데이터 파일 IOPST.dbf를 이동합니다 /oradata2 파일 시스템으로 /oradata3.

```
SQL> alter database move datafile '/oradata2/NTAP/IOPS002.dbf' to  
'/oradata3/NTAP/IOPS002.dbf';  
Database altered.
```

이 방법으로 데이터 파일을 이동하는 것은 느낄 수 있지만 일반적으로는 일상적인 데이터베이스 워크로드에 지장을 줄 만큼 I/O가 충분히 생성하지 않아야 합니다. 반면 ASM 재조정을 통한 마이그레이션은 훨씬 빠르게 실행할 수 있지만 데이터가 이동되는 동안 전체 데이터베이스의 속도가 느려질 수 있습니다.

데이터 파일을 이동한 시간은 테스트 데이터 파일을 만든 다음 이동하는 방법으로 손쉽게 측정할 수 있습니다. 작업에 대해 경과된 시간은 v\$ 세션 데이터에 기록됩니다.

```

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

```

이 예에서는 데이터 파일 8로 이동했습니다. 데이터 파일 8은 21GB이며 마이그레이션하는 데 6분 정도 걸렸습니다. 필요한 시간은 스토리지 시스템의 기능, 스토리지 네트워크 및 마이그레이션 시 발생하는 전체 데이터베이스 활동에 따라 달라집니다.

로그 전달을 통한 **Oracle** 데이터베이스 마이그레이션

로그 전달을 사용하는 마이그레이션의 목표는 새 위치에 원본 데이터 파일의 복사본을 만든 다음 새 환경에 변경 사항을 전달하는 방법을 설정하는 것입니다.

설정된 후에는 로그 전송 및 재생을 자동화하여 복제 데이터베이스를 소스와 대부분 동기화된 상태로 유지할 수 있습니다. 예를 들어, Cron 작업은 (a) 가장 최근의 로그를 새 위치로 복사하고 (b) 15분마다 재생하도록 예약할 수 있습니다. 이렇게 하면 아카이브 로그를 15분 이상 재생해야 하므로 전환 시 작업 중단이 최소화됩니다.

아래에 나와 있는 절차는 기본적으로 데이터베이스 클론 작업입니다. 표시된 로직은 NetApp SnapManager for Oracle(SMO) 및 NetApp SnapCenter Oracle 플러그인 내의 엔진과 유사합니다. 일부 고객은 맞춤형 클론 복제 작업을 위해 스크립트 또는 WFA 워크플로우에 표시된 절차를 사용했습니다. 이 절차는 SMO 또는 SnapCenter를 사용하는 것보다 수동적이지만, ONTAP의 데이터 관리 API로 인해 프로세스가 더욱 간소화됩니다.

로그 전달 - 파일 시스템을 파일 시스템으로 전송합니다

이 예제에서는 Waffle이라는 데이터베이스를 일반 파일 시스템에서 다른 서버에 있는 다른 일반 파일 시스템으로 마이그레이션하는 방법을 보여 줍니다. 또한 SnapMirror를 사용하여 데이터 파일의 신속한 복사본을 만드는 방법을 보여 주지만 이것이 전체 절차의 필수 요소가 아닙니다.

데이터베이스 백업을 만듭니다

첫 번째 단계는 데이터베이스 백업을 만드는 것입니다. 특히 이 절차를 수행하려면 아카이브 로그 재생에 사용할 수 있는 데이터 파일 세트가 필요합니다.

방법입니다

이 예에서 소스 데이터베이스는 ONTAP 시스템에 있습니다. 데이터베이스 백업을 만드는 가장 간단한 방법은 스냅샷을 사용하는 것입니다. 데이터베이스는 가 있는 동안 몇 초 동안 핫 백업 모드로 전환됩니다 snapshot create 이 작업은 데이터 파일을 호스팅하는 볼륨에서 실행됩니다.

```
SQL> alter database begin backup;
Database altered.
```

```
Cluster01::*> snapshot create -vserver vserver1 -volume jfsc1_oradata
hotbackup
Cluster01::*>
```

```
SQL> alter database end backup;
Database altered.
```

그 결과 라는 디스크에 스냅샷이 생성됩니다 hotbackup 핫 백업 모드에 있는 동안 데이터 파일의 이미지가 포함됩니다. 적절한 아카이브 로그와 결합하여 데이터 파일의 일관성을 유지할 경우 이 스냅샷의 데이터를 복구 또는 클론의 기준으로 사용할 수 있습니다. 이 경우 새 서버에 복제됩니다.

새 환경으로 복원합니다

이제 새 환경에서 백업을 복원해야 합니다. 이는 Oracle RMAN, NetBackup 같은 백업 애플리케이션에서 복원 또는 핫 백업 모드에 있었던 데이터 파일의 간단한 복사 작업을 비롯하여 다양한 방법으로 수행할 수 있습니다.

이 예에서는 SnapMirror를 사용하여 스냅샷 핫 백업을 새 위치에 복제합니다.

1. 스냅샷 데이터를 수신할 새 볼륨을 생성합니다. 예서 미러링을 초기화합니다 jfsc1_oradata 를 선택합니다 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. 동기화가 완료되었음을 나타내는 SnapMirror가 상태를 설정한 후, 특히 원하는 스냅샷을 기반으로 미러를 업데이트합니다.


```
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. 동기화를 확인하려면 을 참조하십시오 newest-snapshot 미러 볼륨의 필드입니다.

```
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. 그런 다음 미러가 파손될 수 있습니다.

```
Cluster01:::> snapmirror break -destination-path vserver1:vol_oradata
Operation succeeded: snapmirror break for destination
"vserver1:vol_oradata".
Cluster01:::>
```

5. 새 파일 시스템을 마운트합니다. 블록 기반 파일 시스템을 사용하면 사용 중인 LVM에 따라 정확한 절차가 달라집니다. FC 조닝 또는 iSCSI 연결을 구성해야 합니다. LUN에 대한 연결이 설정된 후 Linux와 같은 명령이 표시됩니다 pvscan ASM이 검색할 수 있도록 올바르게 구성해야 하는 볼륨 그룹 또는 LUN을 찾는 데 필요한 수도 있습니다.

이 예에서는 단순 NFS 파일 시스템이 사용됩니다. 이 파일 시스템은 직접 마운트할 수 있습니다.

```
fas8060-nfs1:/vol_oradata          19922944   1639360   18283584   9%
/oradata
fas8060-nfs1:/vol_logs             9961472    128       9961344    1%
/logs
```

컨트롤 파일 만들기 템플릿을 만듭니다

그런 다음 컨트롤 파일 템플릿을 만들어야 합니다. 를 클릭합니다 backup controlfile to trace 명령은 텍스트

명령을 만들어 컨트롤 파일을 다시 만듭니다. 이 기능은 경우에 따라 백업에서 데이터베이스를 복원하는 데 유용할 수 있으며 데이터베이스 클론 생성과 같은 작업을 수행하는 스크립트와 함께 사용됩니다.

1. 다음 명령의 출력은 마이그레이션된 데이터베이스에 대한 컨트롤 파일을 다시 생성하는 데 사용됩니다.

```
SQL> alter database backup controlfile to trace as '/tmp/waffle.ctrl';
Database altered.
```

2. 컨트롤 파일을 만든 후 새 서버에 파일을 복사합니다.

```
[oracle@jpsc3 tmp]$ scp oracle@jpsc1:/tmp/waffle.ctrl /tmp/
oracle@jpsc1's password:
waffle.ctrl                                100% 5199
5.1KB/s  00:00
```

매개 변수 파일을 백업합니다

매개 변수 파일도 새 환경에 필요합니다. 가장 간단한 방법은 현재 spfile 또는 pfile 에서 pfile 을 만드는 것입니다. 이 예에서는 소스 데이터베이스가 spfile을 사용하고 있습니다.

```
SQL> create pfile='/tmp/waffle.tmp.pfile' from spfile;
File created.
```

ORATAB 항목을 만듭니다

오라타브 항목의 생성은 오라타브 같은 유틸리티의 적절한 기능을 위해 필요합니다. ORATAB 항목을 만들려면 다음 단계를 완료합니다.

```
WAFFLE:/orabin/product/12.1.0/dbhome_1:N
```

디렉토리 구조를 준비합니다

필요한 디렉터리가 없는 경우 해당 디렉터리를 만들어야 합니다. 그렇지 않으면 데이터베이스 시작 절차가 실패합니다. 디렉터리 구조를 준비하려면 다음과 같은 최소 요구 사항을 완료하십시오.

```
[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
```

매개 변수 파일 업데이트

1. 매개 변수 파일을 새 서버에 복사하려면 다음 명령을 실행합니다. 기본 위치는 입니다 \$ORACLE_HOME/dbs 디렉토리. 이 경우 pfile은 어디에나 배치할 수 있습니다. 마이그레이션 프로세스의 중간 단계로만 사용되고 있습니다.

```
[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. 필요에 따라 파일을 편집합니다. 예를 들어 아카이브 로그 위치가 변경된 경우 새 위치를 반영하도록 pfile을 변경해야 합니다. 이 예제에서는 제어 파일만 재배치되고 일부는 로그 및 데이터 파일 시스템 간에 배포됩니다.

```

[root@jfscl 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. 편집이 완료되면 이 pfile을 기반으로 spfile을 만듭니다.

```

SQL> create spfile from pfile='waffle.tmp.pfile';
File created.

```

컨트롤 파일을 다시 만듭니다

이전 단계에서 의 출력입니다 backup controlfile to trace 새 서버로 복사되었습니다. 필요한 출력의 특정 부분은 입니다 controlfile recreation 명령. 이 정보는 표시된 섹션 아래의 파일에서 찾을 수 있습니다 Set #1. NORESETLOGS. 라인부터 시작합니다 create controlfile reuse database 및 은 단어를 포함해야 합니다 noresetlogs. 세미콜론(;) 문자로 끝냅니다.

1. 이 예제 절차에서 파일은 다음과 같이 읽힙니다.

```
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. 다양한 파일의 새 위치를 반영하기 위해 이 스크립트를 편집합니다. 예를 들어, 높은 I/O를 지원하는 것으로 알려진 특정 데이터 파일은 고성능 스토리지 계층의 파일 시스템으로 리디렉션될 수 있습니다. 다른 경우에는 지정된 PDB의 데이터 파일을 전용 볼륨에 격리하는 것과 같은 관리자의 이유만으로 변경 내용이 변경될 수 있습니다.
3. 이 예에서 는 입니다 DATAFILE 스탠자는 변경되지 않은 상태로 유지되지만 다시 실행 로그는 의 새 위치로 이동됩니다 /redo 아카이브 로그와 공간을 공유하는 대신 /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>

```

파일이 잘못 배치되거나 매개 변수가 잘못 구성된 경우 수정해야 할 항목을 나타내는 오류가 생성됩니다. 데이터베이스가 마운트되었지만 아직 열려 있지 않으며 사용 중인 데이터 파일이 핫 백업 모드로 표시되어 있기 때문에 열 수 없습니다. 데이터베이스의 일관성을 유지하기 위해서는 먼저 아카이브 로그를 적용해야 합니다.

초기 로그 복제

데이터 파일의 일관성을 유지하려면 하나 이상의 로그 응답 작업이 필요합니다. 로그를 재생하는 데 사용할 수 있는 옵션은 다양합니다. 경우에 따라 원래 서버의 원래 아카이브 로그 위치를 NFS를 통해 공유할 수 있으며 로그 회신을 직접 수행할 수 있습니다. 다른 경우에는 아카이브 로그를 복사해야 합니다.

예를 들어, 단순 입니다 scp 작업은 소스 서버에서 마이그레이션 서버로 모든 현재 로그를 복사할 수 있습니다.

```

[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

```

초기 로그 재생

파일이 아카이브 로그 위치에 있으면 명령을 실행하여 재생할 수 있습니다 `recover database until cancel` 그 다음에 응답이 옵니다 `AUTO` 사용 가능한 모든 로그를 자동으로 재생합니다.


```

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

```

최종 아카이브 로그 응답에서 오류를 보고하지만 이는 정상입니다. 로그는 이를 나타냅니다 sqlplus 특정 로그 파일을 찾지만 찾지 못했습니다. 로그 파일이 아직 존재하지 않기 때문일 가능성이 높습니다.

아카이브 로그를 복사하기 전에 소스 데이터베이스를 종료할 수 있는 경우 이 단계는 한 번만 수행해야 합니다. 아카이브 로그가 복사되고 재생된 다음 프로세스를 계속 진행하여 중요한 재실행 로그를 복제하는 컷오버 프로세스로 이동할 수 있습니다.

증분 로그 복제 및 재생

대부분의 경우 마이그레이션은 즉시 수행되지 않습니다. 마이그레이션 프로세스가 완료되기까지 며칠이나 몇 주가 걸릴 수 있습니다. 즉, 로그가 계속해서 복제본 데이터베이스로 전송되고 재생되어야 합니다. 따라서 컷오버가 도착하면 최소한의 데이터를 전송하고 재생해야 합니다.

이러한 작업은 여러 가지 방법으로 스크립팅할 수 있지만 일반적인 방법 중 하나는 일반적인 파일 복제 유틸리티인 rsync를 사용하는 것입니다. 이 유틸리티를 사용하는 가장 안전한 방법은 데몬으로 구성하는 것입니다. 예를 들면, rsyncd.conf 다음 파일은 라는 리소스를 만드는 방법을 보여 줍니다 waffle.arch Oracle 사용자 자격 증명으로 액세스되고 에 매핑됩니다 /logs/WAFFLE/arch. 가장 중요한 것은 리소스를 읽기 전용으로 설정하여 운영 데이터를 읽을 수는 있지만 변경할 수는 없다는 것입니다.

```
[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
```

다음 명령은 새 서버의 아카이브 로그 대상을 rsync 리소스와 동기화합니다 waffle.arch 원래 서버에 있습니다. 를 클릭합니다 t 의 인수입니다 rsync - potg 타임스탬프를 기준으로 파일 목록을 비교하고 새 파일만 복사하도록 합니다. 이 프로세스는 새 서버의 증분 업데이트를 제공합니다. 이 명령은 정기적으로 실행되도록 cron으로 예약할 수도 있습니다.

```

[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

```

로그를 수신한 후 재생해야 합니다. 이전 예에서는 sqlplus를 사용하여 수동으로 실행하는 방법을 보여 줍니다 `recover database until cancel` 쉽게 자동화할 수 있는 프로세스입니다. 여기에 표시된 예에서는 에 설명된 스크립트를 사용합니다 "데이터베이스에서 로그를 재생합니다". 스크립트에는 재생 작업이 필요한 데이터베이스를 지정하는 인수를 사용할 수 있습니다. 이렇게 하면 다중 데이터베이스 마이그레이션 작업에 동일한 스크립트를 사용할 수 있습니다.

```

[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

```

컷오버

새 환경으로 전환할 준비가 되면 아카이브 로그와 redo 로그를 모두 포함하는 하나의 최종 동기화를 수행해야 합니다. 원래 redo 로그 위치를 아직 모르는 경우 다음과 같이 식별할 수 있습니다.

```
SQL> select member from v$logfile;
MEMBER
-----
-----
/logs/WAFFLE/redo/redo01.log
/logs/WAFFLE/redo/redo02.log
/logs/WAFFLE/redo/redo03.log
```

1. 원본 데이터베이스를 종료합니다.
2. 원하는 방법으로 새 서버에서 아카이브 로그의 최종 동기화를 수행합니다.
3. 원본 redo 로그를 새 서버에 복사해야 합니다. 이 예에서는 redo 로그가 의 새 디렉토리로 재배치되었습니다
/redo.

```
[oracle@jfs3 logs]$ scp jfs1:/logs/WAFFLE/redo/* /redo/
oracle@jfs1'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. 이 단계에서 새 데이터베이스 환경에는 원본과 동일한 상태로 되돌리는 데 필요한 모든 파일이 포함됩니다. 아카이브 로그는 마지막으로 한 번 재생되어야 합니다.

```

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. 완료되면 재실행 로그를 재생해야 합니다. 메시지가 표시되는 경우 Media recovery complete 이 반환되고 프로세스가 성공하며 데이터베이스가 동기화되어 열 수 있습니다.

```

SQL> recover database;
Media recovery complete.
SQL> alter database open;
Database altered.

```

로그 전달 - 파일 시스템에 **ASM**을 전달합니다

이 예에서는 Oracle RMAN을 사용하여 데이터베이스를 마이그레이션하는 방법을 보여 줍니다. 이는 파일 시스템 로그 전달과 파일 시스템 로그 전달의 이전 예와 매우 유사하지만 ASM의 파일은 호스트에 표시되지 않습니다. ASM 디바이스에 있는 데이터를 마이그레이션하는 유일한 옵션은 ASM LUN을 재배치하거나 Oracle RMAN을 사용하여 복제 작업을 수행하는 것입니다.

RMAN은 Oracle ASM에서 파일을 복사하기 위한 요구 사항이지만 RMAN 사용은 ASM에 국한되지 않습니다. RMAN을 사용하여 모든 유형의 스토리지에서 다른 유형으로 마이그레이션할 수 있습니다.

이 예에서는 팬케이크라는 데이터베이스를 ASM 스토리지에서 경로의 다른 서버에 있는 일반 파일 시스템으로 재배치하는 방법을 보여 줍니다 /oradata 및 /logs.

데이터베이스 백업을 만듭니다

첫 번째 단계는 대체 서버로 마이그레이션할 데이터베이스의 백업을 만드는 것입니다. 소스가 Oracle ASM을 사용하므로 RMAN을 사용해야 합니다. 간단한 RMAN 백업은 다음과 같이 수행할 수 있습니다. 이 방법은 나중에 RMAN에서 쉽게 식별할 수 있는 태그가 지정된 백업을 생성합니다.

첫 번째 명령은 백업 대상 유형과 사용할 위치를 정의합니다. 두 번째는 데이터 파일의 백업만 시작합니다.

```

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

```

백업 제어 파일

백업 제어 파일은 이 절차의 뒷부분에서 필요합니다 duplicate database 작동.

```

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

```

매개 변수 파일을 백업합니다

매개 변수 파일도 새 환경에 필요합니다. 가장 간단한 방법은 현재 spfile 또는 pfile 에서 pfile 을 만드는 것입니다. 이 예제에서 원본 데이터베이스는 spfile을 사용합니다.

```

RMAN> create pfile='/rman/pancake/pfile' from spfile;
Statement processed

```

ASM 파일 이름 바꾸기 스크립트

현재 컨트롤 파일에 정의된 여러 파일 위치는 데이터베이스를 이동할 때 변경됩니다. 다음 스크립트는 프로세스를 쉽게 하기 위해 RMAN 스크립트를 생성합니다. 이 예에서는 데이터 파일 수가 매우 적은 데이터베이스를 보여 주지만 일반적으로 데이터베이스에는 수백 또는 수천 개의 데이터 파일이 포함되어 있습니다.

이 스크립트는 에서 찾을 수 있습니다 "[ASM에서 파일 시스템으로 이름 변환](#)" 그리고 이 두 가지를 수행합니다.

먼저 매개 변수를 만들어 라는 redo 로그 위치를 다시 정의합니다 log_file_name_convert. 기본적으로 대체 필드의 목록입니다. 첫 번째 필드는 현재 redo 로그의 위치이고 두 번째 필드는 새 서버의 위치입니다. 그런 다음 패턴이 반복됩니다.

두 번째 기능은 데이터 파일 이름 변경을 위한 템플릿을 제공하는 것입니다. 스크립트는 데이터 파일을 반복하고 이름 및 파일 번호 정보를 가져와서 RMAN 스크립트로 형식을 지정합니다. 그런 다음 임시 파일에서도 마찬가지입니다. 그 결과, 파일이 원하는 위치로 복구되도록 원하는 대로 편집할 수 있는 간단한 RMAN 스크립트가 생성됩니다.


```

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.

```

이 화면의 출력을 캡처합니다. 를 클릭합니다 log_file_name_convert 매개 변수는 아래 설명된 대로 pfile에 배치됩니다. 데이터 파일을 원하는 위치에 배치하려면 RMAN 데이터 파일 이름 바꾸기 및 중복 스크립트를 적절히 편집해야 합니다. 이 예제에서는 모두 에 배치됩니다 /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';
}

```

디렉토리 구조를 준비합니다

스크립트는 거의 실행할 준비가 되었지만 먼저 디렉토리 구조가 있어야 합니다. 필요한 디렉터리가 아직 없으면 해당 디렉터리를 만들어야 합니다. 그렇지 않으면 데이터베이스 시작 절차가 실패합니다. 아래의 예는 최소 요구 사항을 반영합니다.

```

[oracle@jfsc2 ~]$ mkdir /oradata/pancake
[oracle@jfsc2 ~]$ mkdir /logs/pancake
[oracle@jfsc2 ~]$ cd /orabin/admin
[oracle@jfsc2 admin]$ mkdir PANCAKE
[oracle@jfsc2 admin]$ cd PANCAKE
[oracle@jfsc2 PANCAKE]$ mkdir adump dpdump pfile scripts xdb_wallet

```

ORATAB 항목을 만듭니다

oraenv와 같은 유틸리티가 제대로 작동하려면 다음 명령이 필요합니다.

```
PANCAKE:/orabin/product/12.1.0/dbhome_1:N
```

매개 변수 업데이트

새 서버의 경로 변경 사항을 반영하도록 저장된 pfile을 업데이트해야 합니다. 데이터 파일 경로 변경은 RMAN 복제 스크립트에 의해 변경되며 거의 모든 데이터베이스를 변경해야 합니다 control_files 및 log_archive_dest 매개 변수. 또한 변경해야 하는 감사 파일 위치와 같은 매개 변수가 있을 수 있습니다 db_create_file_dest ASM 외부에서는 관련이 없을 수 있습니다. 숙련된 DBA는 계속하기 전에 제안된 변경 사항을 주의 깊게 검토해야 합니다.

이 예에서 주요 변경 사항은 제어 파일 위치, 로그 아카이브 대상 및 추가입니다 log_file_name_convert 매개 변수.

```

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'

```

새 매개 변수가 확인되면 매개 변수가 적용되어야 합니다. 여러 옵션이 있지만 대부분의 고객은 pfile 텍스트를 기반으로 spfile을 만듭니다.

```

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.

```

시동 nomount

데이터베이스를 복제하기 전의 마지막 단계는 데이터베이스 프로세스를 불러오지만 파일을 마운트하지 않는 것입니다. 이 단계에서는 spfile에 문제가 발생할 수 있습니다. 를 누릅니다 startup nomount 명령 실패 매개 변수 오류로 인해 실패합니다. pfile 템플릿을 종료하고 수정한 다음 spfile로 다시 로드한 후 다시 시도하십시오.

```

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

```

데이터베이스를 복제합니다

이전 RMAN 백업을 새 위치로 복원하는 데 이 프로세스의 다른 단계보다 시간이 더 오래 걸립니다. 데이터베이스 ID(DBID)를 변경하거나 로그를 재설정하지 않고 데이터베이스를 복제해야 합니다. 이렇게 하면 로그를 적용할 수 없습니다. 이는 복사본을 완전히 동기화하는 데 필요한 단계입니다.

RMAN을 aux로 데이터베이스에 연결하고 이전 단계에서 생성한 스크립트를 사용하여 중복 데이터베이스 명령을 실행합니다.

```

[oracle@jpsc2 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

```

초기 로그 복제

이제 원본 데이터베이스의 변경 내용을 새 위치로 전달해야 합니다. 이렇게 하려면 여러 단계를 조합해야 할 수 있습니다. 가장 간단한 방법은 소스 데이터베이스의 RMAN이 공유 네트워크 연결에 아카이브 로그를 기록하도록 하는 것입니다. 공유 위치를 사용할 수 없는 경우 RMAN을 사용하여 로컬 파일 시스템에 쓴 다음 RCP 또는 rsync를 사용하여 파일을 복사하는 방법이 있습니다.

이 예에서 는 입니다 /rman 디렉토리는 원래 데이터베이스와 마이그레이션된 데이터베이스 모두에서 사용할 수 있는 NFS 공유입니다.

여기서 한 가지 중요한 문제는 입니다 disk format 조항. 백업의 디스크 형식은 입니다 %h_%e_%a.dbf`즉, 데이터베이스에 대한 스레드 번호, 시퀀스 번호 및 활성화 ID 형식을 사용해야 합니다. 글자는 다르지만 이 문장은 과 일치합니다 `log_archive_format='%t_%s_%r.dbf pfile의 매개 변수입니다. 또한 이 매개 변수는 스레드 번호, 시퀀스 번호 및 활성화 ID 형식으로 아카이브 로그를 지정합니다. 결과적으로 소스의 로그 파일 백업이 데이터베이스에서 예상하는 명명 규칙을 사용하게 됩니다. 이렇게 하면 과 같은 작업이 수행됩니다 recover database sqlplus 는 재생될 아카이브 로그의 이름을 올바르게 예측하기 때문에 훨씬 더 간단합니다.

```

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

```

초기 로그 재생

파일이 아카이브 로그 위치에 있으면 명령을 실행하여 재생할 수 있습니다 `recover database until cancel` 그 다음에 응답이 옵니다 `AUTO` 사용 가능한 모든 로그를 자동으로 재생합니다. 매개 변수 파일이 현재 아카이브 로그를 로 리디렉션하고 있습니다 `/logs/archive` 하지만 RMAN이 로그를 저장하는 데 사용된 위치와 일치하지 않습니다. 데이터베이스를 복구하기 전에 다음과 같이 위치를 일시적으로 리디렉션할 수 있습니다.


```

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

```

최종 아카이브 로그 응답에서 오류를 보고하지만 이는 정상입니다. 이 오류는 sqlplus가 특정 로그 파일을 찾고 있지만 찾지 못했음을 나타냅니다. 로그 파일이 아직 존재하지 않기 때문일 수 있습니다.

아카이브 로그를 복사하기 전에 소스 데이터베이스를 종료할 수 있는 경우 이 단계는 한 번만 수행해야 합니다. 아카이브 로그가 복사되고 재생된 다음 프로세스를 계속 진행하여 중요한 재실행 로그를 복제하는 컷오버 프로세스로 이동할 수 있습니다.

증분 로그 복제 및 재생

대부분의 경우 마이그레이션은 즉시 수행되지 않습니다. 마이그레이션 프로세스가 완료되기까지 며칠이나 몇 주가 걸릴 수 있습니다. 즉, 로그가 계속해서 복제본 데이터베이스로 전송되고 재생되어야 합니다. 이렇게 하면 컷오버가 도착할 때 최소한의 데이터를 전송하고 재생해야 합니다.

이 프로세스는 쉽게 스크립팅할 수 있습니다. 예를 들어, 로그 전달에 사용되는 위치가 지속적으로 업데이트되도록 원본 데이터베이스에 다음 명령을 예약할 수 있습니다.

```
[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.
```

로그를 수신한 후 재생해야 합니다. 이전 예제에서는 sqlplus 를 사용하여 수동으로 실행하는 방법을 보여 주었습니다 `recover database until cancel` 쉽게 자동화할 수 있습니다. 여기에 표시된 예에서는 에 설명된 스크립트를 사용합니다 "대기 데이터베이스에서 로그를 재생합니다". 스크립트에는 재생 작업이 필요한 데이터베이스를 지정하는 인수를 사용할 수 있습니다. 이 프로세스에서는 다중 데이터베이스 마이그레이션 작업에 동일한 스크립트를 사용할 수 있습니다.

```
[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
```

컷오버

새 환경으로 컷오버할 준비가 되면 최종 동기화 하나를 수행해야 합니다. 일반 파일 시스템으로 작업할 때 원래 redo 로그가 복사되고 재생되므로 마이그레이션된 데이터베이스가 원본과 100% 동기화되도록 쉽게 할 수 있습니다. ASM과 함께 이 작업을 수행하는 좋은 방법은 없습니다. 보관 로그만 쉽게 다시 복사할 수 있습니다. 데이터가 손실되지 않도록 하려면 원본 데이터베이스의 최종 종료를 주의 깊게 수행해야 합니다.

1. 먼저 데이터베이스를 정지하여 변경 사항이 없는지 확인해야 합니다. 이 일시 중지에는 예약된 작업을 비활성화하거나, 수신키를 종료하거나, 응용 프로그램을 종료하는 작업이 포함될 수 있습니다.
2. 이 단계를 수행한 후 대부분의 DBA는 종료의 표시자 역할을 하는 더미 테이블을 생성합니다.
3. 로그 아카이빙을 강제 수행하여 더미 테이블 생성이 아카이브 로그 내에 기록되도록 합니다. 이렇게 하려면 다음 명령을 실행합니다.

```
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. 마지막 아카이브 로그를 복사하려면 다음 명령을 실행합니다. 데이터베이스를 사용할 수 있어야 하지만 열려 있지 않아야 합니다.

```
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. 아카이브 로그를 복사하려면 다음 명령을 실행합니다.

```

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. 마지막으로 새 서버에서 나머지 아카이브 로그를 재생합니다.

```

[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. 이 단계에서는 모든 데이터를 복제합니다. 데이터베이스를 대기 데이터베이스에서 활성 작업 데이터베이스로 변환할 준비가 된 다음 열 수 있습니다.

```

SQL> alter database activate standby database;
Database altered.
SQL> alter database open;
Database altered.

```

8. 더미 테이블이 있는지 확인한 다음 삭제합니다.

```

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.

```

무중단 재실행 로그 마이그레이션

재실행 로그를 제외하고 데이터베이스가 전체적으로 올바르게 구성된 경우가 있습니다. 이러한 현상은 여러 가지 이유로 발생할 수 있으며, 그 중 가장 일반적인 원인은 스냅샷과 관련이 있습니다. Oracle용 SnapManager, SnapCenter, NetApp Snap Creator 스토리지 관리 프레임워크와 같은 제품을 사용하면 데이터 파일 볼륨의 상태를 되돌리는 경우에만 거의 즉각적으로 데이터베이스 복구가 가능합니다. 재실행 로그가 데이터 파일과 공간을 공유하는 경우 재실행 로그가 삭제되어 데이터 손실이 발생할 수 있으므로 재버전을 안전하게 수행할 수 없습니다. 따라서 redo 로그를 재배치해야 합니다.

이 절차는 단순하며 중단 없이 수행할 수 있습니다.

현재 redo 로그 구성

1. 재실행 로그 그룹의 수와 해당 그룹 번호를 식별합니다.


```

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. redo 로그의 크기를 입력합니다.

```

SQL> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 524288000
2 524288000
3 524288000

```

새 로그를 만듭니다

1. 각 REDO 로그에 대해 일치하는 크기와 구성원 수가 있는 새 그룹을 만듭니다.

```

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. 새 구성을 확인합니다.

```

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.

```

오래된 로그를 삭제합니다

1. 이전 로그(그룹 1, 2, 3)를 삭제합니다.

```

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. 활성 로그를 삭제할 수 없는 오류가 발생하면 다음 로그로 스위치를 강제로 전환하여 잠금을 해제하고 글로벌 체크포인트를 강제로 설정합니다. 이 프로세스의 다음 예를 참조하십시오. 이 로그 파일에 활성 데이터가 있기 때문에 이전 위치에 있던 로그 파일 그룹 2를 삭제하려는 시도가 거부되었습니다.

```

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. 로그 보관 후 체크포인트를 수행하면 로그 파일을 삭제할 수 있습니다.

```
SQL> alter system archive log current;
System altered.
SQL> alter system checkpoint;
System altered.
SQL> alter database drop logfile group 2;
Database altered.
```

4. 그런 다음 파일 시스템에서 로그를 삭제합니다. 이 과정은 매우 세심한 주의를 기울여 수행해야 합니다.

Oracle 데이터베이스 호스트 데이터 복사본

데이터베이스 레벨 마이그레이션과 마찬가지로 호스트 계층에서 마이그레이션하면 스토리지 공급업체에 종속되지 않는 접근 방식이 제공됩니다.

다시 말해, 언젠가는 "그냥 파일 복사"가 가장 좋은 옵션입니다.

이러한 낮은 수준의 기술 접근 방식은 매우 기본적이라고 생각될 수 있지만, 특별한 소프트웨어가 필요하지 않고 원본 데이터가 프로세스 중에 안전하게 유지되기 때문에 상당한 이점을 제공합니다. 주된 제한 사항은 파일 복사 데이터 마이그레이션은 복사 작업을 시작하기 전에 데이터베이스를 종료해야 하기 때문에 중단 프로세스라는 점입니다. 파일 내의 변경 내용을 동기화하는 좋은 방법은 없으므로 복사하기 전에 파일을 완전히 정지해야 합니다.

복사 작업에 필요한 종료가 바람직하지 않은 경우 다음으로 가장 좋은 호스트 기반 옵션은 논리 볼륨 관리자(LVM)를 활용하는 것입니다. Oracle ASM을 비롯한 많은 LVM 옵션이 존재하며 모든 기능이 비슷하지만 몇 가지 제한도 고려해야 합니다. 대부분의 경우 다운타임 및 운영 중단 없이 마이그레이션을 수행할 수 있습니다.

파일 시스템 대 파일 시스템 복제

단순 복사 작업의 유용성을 과소 평가해서는 안 됩니다. 이 작업은 복사 프로세스 중에 다운타임이 발생하지만 매우 안정적인 프로세스이며 운영 체제, 데이터베이스 또는 스토리지 시스템에 대한 특별한 전문 지식이 필요하지 않습니다. 또한 원본 데이터에 영향을 주지 않기 때문에 매우 안전합니다. 일반적으로 시스템 관리자는 소스 파일 시스템을 읽기 전용으로 마운트하도록 변경한 다음 서버를 재부팅하여 현재 데이터가 손상되지 않도록 합니다. 복제 프로세스는 사용자 오류의 위험 없이 가능한 한 빨리 실행되도록 스크립트될 수 있습니다. I/O 유형은 데이터의 단순한 순차 전송이므로 대역폭 효율성이 매우 높습니다.

다음 예에서는 안전하고 신속한 마이그레이션을 위한 한 가지 옵션을 보여 줍니다.

방법입니다

마이그레이션할 환경은 다음과 같습니다.

- 현재 파일 시스템

```
ontap-nfs1:/host1_oradata          52428800  16196928  36231872  31%
/oradata
ontap-nfs1:/host1_logs             49807360   548032  49259328   2% /logs
```

- 새 파일 시스템

```

ontap-nfs1:/host1_logs_new      49807360      128  49807232    1%
/new/logs
ontap-nfs1:/host1_oradata_new  49807360      128  49807232    1%
/new/oradata

```

개요

데이터베이스를 종료하고 파일을 복사하기만 하면 DBA가 데이터베이스를 마이그레이션할 수 있지만, 많은 데이터베이스를 마이그레이션해야 하거나 가동 중지 시간을 최소화하는 것이 중요한 경우 프로세스가 쉽게 스크립팅됩니다. 또한 스크립트를 사용하면 사용자의 실수 가능성이 줄어듭니다.

표시된 예제 스크립트는 다음 작업을 자동화합니다.

- 데이터베이스를 종료합니다
- 기존 파일 시스템을 읽기 전용 상태로 변환합니다
- 소스의 모든 데이터를 타겟 파일 시스템으로 복제하여 모든 파일 사용 권한을 유지합니다
- 이전 파일 시스템과 새 파일 시스템의 마운트를 해제합니다
- 이전 파일 시스템과 동일한 경로에 새 파일 시스템을 다시 마운트합니다

절차를 참조하십시오

1. 데이터베이스를 종료합니다.

```

[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. 파일 시스템을 읽기 전용으로 변환합니다. 예서와 같이 스크립트를 사용하여 보다 빠르게 수행할 수 있습니다 "파일 시스템을 읽기 전용으로 변환합니다".

```
[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. 이제 파일 시스템이 읽기 전용인지 확인합니다.

```
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. 와 파일 시스템 콘텐츠를 동기화합니다 rsync 명령.

```
[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. 이전 파일 시스템을 마운트 해제하고 복제된 데이터를 재배치합니다. 예서와 같이 스크립트를 사용하여 보다 빠르게 수행할 수 있습니다 **"파일 시스템을 교체합니다"**.

```

[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. 새 파일 시스템이 올바른 위치에 있는지 확인합니다.

```
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. 데이터베이스를 시작합니다.

```
[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
```

완전 자동화된 컷오버

이 샘플 스크립트에서는 데이터베이스 SID의 인수 다음에 파일 시스템의 공용 구분 쌍이 오는 인수를 사용할 수 있습니다. 위의 예에서 명령은 다음과 같이 실행됩니다.

```
[root@host1 current]# ./migrate.oracle.fs.pl NTAP /logs,/new/logs
/oradata,/new/oradata
```

예제 스크립트가 실행되면 다음 순서를 수행하려고 시도합니다. 어떤 단계에서든 오류가 발생하면 종료됩니다.

1. 데이터베이스를 종료합니다.
2. 현재 파일 시스템을 읽기 전용 상태로 변환합니다.
3. 심표로 구분된 각 파일 시스템 인수 쌍을 사용하고 첫 번째 파일 시스템을 두 번째 파일 시스템과 동기화합니다.
4. 이전 파일 시스템을 분리합니다.
5. 를 업데이트합니다 /etc/fstab 다음과 같은 파일:

- a. 에서 백업을 생성합니다 /etc/fstab.bak.
 - b. 이전 및 새 파일 시스템에 대한 이전 항목을 주석 처리합니다.
 - c. 이전 마운트 지점을 사용하는 새 파일 시스템에 대한 새 항목을 생성합니다.
6. 파일 시스템을 마운트합니다.
7. 데이터베이스를 시작합니다.

다음 텍스트는 이 스크립트에 대한 실행 예제를 제공합니다.

```
[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 및 passwd 마이그레이션

ASM과 관련된 마이그레이션을 완료하는 데 있어 한 가지 어려움은 ASM 관련 spfile과 암호 파일입니다. 기본적으로 이러한 중요 메타데이터 파일은 정의된 첫 번째 ASM 디스크 그룹에 생성됩니다. 특정 ASM 디스크 그룹을 비우고 제거해야 하는 경우 해당 ASM 인스턴스를 제어하는 spfile 및 암호 파일을 재배치해야 합니다.

이러한 파일을 재배치해야 하는 또 다른 활용 사례는 SnapManager for Oracle 또는 SnapCenter Oracle 플러그인과 같은 데이터베이스 관리 소프트웨어를 구축하는 경우입니다. 이러한 제품의 기능 중 하나는 데이터 파일을 호스팅하는 ASM LUN의 상태를 되돌려 신속하게 데이터베이스를 복원하는 것입니다. 이렇게 하려면 복원을 수행하기 전에 ASM 디스크 그룹을 오프라인으로 전환해야 합니다. 특정 데이터베이스의 데이터 파일이 전용 ASM 디스크 그룹에서 격리되어 있는 한 이 문제는 문제가 되지 않습니다.

해당 디스크 그룹에 ASM spfile/passwd 파일도 포함되어 있는 경우 디스크 그룹을 오프라인으로 전환할 수 있는 유일한 방법은 전체 ASM 인스턴스를 종료하는 것입니다. 이는 중단 프로세스이므로 spfile/passwd 파일을 재배치해야 합니다.

방법입니다

1. 데이터베이스 SID = 토스트
2. 의 현재 데이터 파일 +DATA
3. 의 현재 로그 파일 및 제어 파일 +LOGS
4. 로 설정된 새 ASM 디스크 그룹 +NEWDATA 및 +NEWLOGS

ASM spfile/passwd 파일 위치

이러한 파일 재배치는 중단 없이 수행할 수 있습니다. 그러나 안전을 위해 NetApp에서는 파일이 재배치되고 구성이 올바르게 업데이트되었는지 확인할 수 있도록 데이터베이스 환경을 종료하는 것이 좋습니다. 서버에 여러 ASM 인스턴스가 있는 경우 이 절차를 반복해야 합니다.

ASM 인스턴스 식별

에 기록된 데이터를 기반으로 ASM 인스턴스를 식별합니다 oratab 파일. ASM 인스턴스는 + 기호로 표시됩니다.

```
-bash-4.1$ cat /etc/oratab | grep '^+'
+ASM:/orabin/grid:N          # line added by Agent
```

이 서버에는 +asm 이라는 ASM 인스턴스가 하나 있습니다.

모든 데이터베이스가 종료되었는지 확인합니다

사용 중인 ASM 인스턴스에 대한 smon 프로세스만 볼 수 있습니다. 다른 스몬 프로세스가 있으면 데이터베이스가 여전히 실행 중임을 나타냅니다.

```
-bash-4.1$ ps -ef | grep smon
oracle      857      1  0 18:26 ?          00:00:00 asm_smon_+ASM
```

유일한 smon 프로세스는 ASM 인스턴스 자체입니다. 즉, 다른 데이터베이스는 실행 중이지 않으므로 데이터베이스 작업을 중단하지 않고 계속 진행하는 것이 안전합니다.

파일을 찾습니다

를 사용하여 ASM spfile 및 암호 파일의 현재 위치를 식별합니다 spget 및 pwget 명령.

```
bash-4.1$ asmcmd
ASMCMD> spget
+DATA/spfile.ora
```

```
ASMCMD> pwget --asm
+DATA/orapwasm
```

두 파일은 모두 의 하단에 있습니다 +DATA 디스크 그룹입니다.

파일을 복사합니다

를 사용하여 파일을 새 ASM 디스크 그룹에 복사합니다 spcopy 및 pwcop 명령. 새 디스크 그룹이 최근에 생성되어 현재 비어 있는 경우 먼저 마운트해야 할 수 있습니다.

```
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
```

이제 파일이 에서 복사되었습니다 +DATA 를 선택합니다 +NEWDATA.

ASM 인스턴스를 업데이트합니다

이제 ASM 인스턴스를 업데이트하여 위치 변경을 반영해야 합니다. 를 클릭합니다 spset 및 pwset 명령은 ASM 디스크 그룹을 시작하는 데 필요한 ASM 메타데이터를 업데이트합니다.

```
ASMCMD> spset +NEWDATA/spfile.ora
ASMCMD> pwset --asm +NEWDATA/orapwasm
```

업데이트된 파일을 사용하여 **ASM**을 활성화합니다

이때 ASM 인스턴스는 이러한 파일의 이전 위치를 계속 사용합니다. 새 위치에서 파일을 다시 읽고 이전 파일의 잠금을 해제하려면 인스턴스를 다시 시작해야 합니다.

```
-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
```

이전 **spfile** 및 암호 파일을 제거합니다

절차가 성공적으로 수행되면 이전 파일이 더 이상 잠기지 않고 제거할 수 있습니다.

```
-bash-4.1$ asmcmd
ASMCMDB> rm +DATA/spfile.ora
ASMCMDB> rm +DATA/orapwasm
```

Oracle ASM에서 ASM으로의 복사

Oracle ASM은 기본적으로 경량의 복합 볼륨 관리자와 파일 시스템입니다. 파일 시스템이 바로 보이지 않으므로 RMAN을 사용하여 복사 작업을 수행해야 합니다. 복사 기반의 마이그레이션 프로세스는 안전하고 간단하지만 약간의 운영 중단이 발생합니다. 운영 중단을 최소화할 수 있지만 완전히 제거되지는 않습니다.

ASM 기반 데이터베이스의 무중단 마이그레이션을 원하는 경우 ASM의 기능을 활용하여 이전 LUN을 삭제하는 동시에 ASM 익스텐트를 새 LUN으로 재조정하는 것이 가장 좋습니다. 일반적으로 안전하면서 운영 중단이 일어나지 않지만 백아웃 경로는 제공되지 않습니다. 기능 또는 성능 문제가 발생할 경우 데이터를 소스로 다시 마이그레이션하는 방법만 사용할 수 있습니다.

데이터를 이동하지 않고 데이터베이스를 새 위치로 복사하여 원본 데이터를 그대로 유지하면 이러한 위험을 방지할 수 있습니다. 이 데이터베이스는 새 위치에서 완전히 테스트된 후 가동할 수 있으며, 문제가 발견될 경우 원래 데이터베이스를 폴백 옵션으로 사용할 수 있습니다.

이 절차는 RMAN과 관련된 여러 옵션 중 하나입니다. 초기 백업이 생성된 후 나중에 로그 재생을 통해 동기화되는 2단계 프로세스를 허용하도록 설계되었습니다. 이 프로세스는 초기 베이스라인 복사 중에 데이터베이스가 운영 상태를 유지하고 데이터를 제공할 수 있기 때문에 다운타임을 최소화하는 것이 좋습니다.

데이터베이스를 복사합니다

Oracle RMAN은 현재 ASM 디스크 그룹에 있는 소스 데이터베이스의 레벨 0(전체) 복제본을 생성합니다 +DATA 의 새 위치로 이동합니다 +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

```

아카이브 로그 스위치를 강제 적용합니다

아카이브 로그에 복사의 일관성을 완전히 유지하는 데 필요한 모든 데이터가 포함되도록 하려면 아카이브 로그 스위치를 강제로 사용해야 합니다. 이 명령을 사용하지 않으면 주요 데이터가 재실행 로그에 계속 존재할 수 있습니다.

```

RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current

```

원본 데이터베이스를 종료합니다

데이터베이스가 종료되고 제한된 읽기 전용 모드로 전환되기 때문에 이 단계에서 종단이 시작됩니다. 소스 데이터베이스를 종료하려면 다음 명령을 실행합니다.

```

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

```

제어 파일 백업

마이그레이션을 중단하고 원래 스토리지 위치로 되돌려야 하는 경우 controlfile을 백업해야 합니다. 백업 제어 파일 사본이 100% 필요한 것은 아니지만 데이터베이스 파일 위치를 원래 위치로 다시 설정하는 프로세스가 더 쉬워집니다.

```

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=8977760073
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:01
Finished backup at 06-DEC-15

```

매개 변수 업데이트

현재 spfile에는 이전 ASM 디스크 그룹 내의 현재 위치에 있는 컨트롤 파일에 대한 참조가 포함되어 있습니다. 중간 pfile 버전을 편집하여 쉽게 편집할 수 있도록 편집해야 합니다.

```

RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed

```

pfile을 업데이트합니다

새 ASM 디스크 그룹 이름을 반영하도록 이전 ASM 디스크 그룹을 참조하는 모든 매개 변수를 업데이트합니다. 그런 다음 업데이트된 pfile을 저장합니다. 를 확인합니다 db_create 매개 변수가 있습니다.

아래 예에서는 예에 대한 참조를 나타냅니다 +DATA 이(가) 로 변경되었습니다 +NEWDATA 노란색으로 강조 표시됩니다. 두 가지 주요 매개 변수는입니다 db_create 올바른 위치에 새 파일을 만드는 매개 변수입니다.

```
*.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'
```

init.ora 파일을 업데이트합니다

대부분의 ASM 기반 데이터베이스는 를 사용합니다 init.ora 에 있는 파일 \$ORACLE_HOME/dbs 디렉토리로, ASM 디스크 그룹의 spfile을 가리킵니다. 이 파일은 새 ASM 디스크 그룹의 위치로 리디렉션되어야 합니다.

```
-bash-4.1$ cd $ORACLE_HOME/dbs  
-bash-4.1$ cat initTOAST.ora  
SPFILE='+DATA/TOAST/spfileTOAST.ora'
```

이 파일을 다음과 같이 변경합니다.

```
SPFILE='+NEWLOGS/TOAST/spfileTOAST.ora
```

매개 변수 파일 재생성

이제 편집된 pfile의 데이터로 spfile을 채울 준비가 되었습니다.

```
RMAN> create spfile from pfile='/tmp/pfile';  
Statement processed
```

데이터베이스를 시작하여 새 **spfile** 사용을 시작합니다

데이터베이스를 시작하여 새로 생성된 spfile을 사용하고 시스템 매개변수에 대한 추가 변경 사항이 올바르게 기록되었는지 확인합니다.


```

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

```

제어파일을 복원합니다

RMAN에서 생성된 백업 제어 파일은 RMAN에서 새 spfile에 지정된 위치로 직접 복구할 수도 있습니다.

```

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

```

데이터베이스를 마운트하고 새 컨트롤 파일의 사용을 확인합니다.

```

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

```

로그 재생

데이터베이스는 현재 이전 위치에 있는 데이터 파일을 사용합니다. 복사본을 사용하려면 먼저 복사본을 동기화해야 합니다. 초기 복제 프로세스 중에 시간이 경과했으며 변경 사항이 주로 아카이브 로그에 기록되었습니다. 이러한 변경 사항은 다음과 같이 복제됩니다.

1. 아카이브 로그가 포함된 RMAN 증분 백업을 수행합니다.

```
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. 로그를 재생합니다.

```

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

```

활성화

복원된 컨트롤 파일은 원래 위치에 있는 데이터 파일을 참조하며 복사된 데이터 파일의 경로 정보도 포함합니다.

1. 활성 데이터 파일을 변경하려면 `switch database to copy` 명령.

```

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"

```

활성 데이터 파일은 이제 복사된 데이터 파일이지만 최종 redo 로그에 변경 내용이 포함될 수 있습니다.

2. 나머지 로그를 모두 재생하려면 `recover database` 명령. 메시지가 표시되는 경우 `media recovery complete` 프로세스가 성공했다는 메시지가 나타납니다.

```

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

```

이 프로세스는 일반 데이터 파일의 위치만 변경했습니다. 임시 데이터 파일은 이름을 바꿔야 하지만 임시 파일이므로 복사할 필요가 없습니다. 데이터베이스가 현재 다운되어 임시 데이터 파일에 활성 데이터가 없습니다.

3. 임시 데이터 파일을 재배치하려면 먼저 해당 위치를 확인합니다.

```

RMAN> select file#||' '||name from v$tempfile;
FILE#||' '||NAME
-----
-----
1 +DATA/TOAST/TEMPFILE/temp.263.897683145

```

4. 각 데이터 파일의 새 이름을 설정하는 RMAN 명령을 사용하여 임시 데이터 파일을 재배치합니다. OMF(Oracle Managed Files)에서는 전체 이름이 필요하지 않으며 ASM 디스크 그룹이면 충분합니다. 데이터베이스가 열리면 OMF는 ASM 디스크 그룹의 적절한 위치에 연결됩니다. 파일을 재배치하려면 다음 명령을 실행합니다.

```

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

```

로그 마이그레이션을 다시 실행합니다

마이그레이션 프로세스는 거의 완료되었지만 재실행 로그는 여전히 원본 ASM 디스크 그룹에 있습니다. REDO 로그는 직접 재배치할 수 없습니다. 대신 새 redo 로그 세트가 생성되어 구성에 추가된 다음 이전 로그가 삭제됩니다.

1. 재실행 로그 그룹의 수와 해당 그룹 번호를 식별합니다.

```

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. redo 로그의 크기를 입력합니다.

```

RMAN> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1  52428800
2  52428800
3  52428800

```

3. 각 redo 로그에 대해 일치하는 구성을 가진 새 그룹을 생성합니다. OMF를 사용하지 않는 경우 전체 경로를 지정해야 합니다. 이는 를 사용하는 예이기도 합니다 db_create_online_log 매개 변수. 앞에서 설명한 것처럼 이 매개 변수는 +NEWLOGS 로 설정되었습니다. 이 구성을 사용하면 파일 위치나 특정 ASM 디스크 그룹을 지정할 필요 없이 다음 명령을 사용하여 새로운 온라인 로그를 생성할 수 있습니다.

```

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. 데이터베이스를 엽니다.

```

SQL> alter database open;
Database altered.

```

5. 이전 로그를 삭제합니다.

```

RMAN> alter database drop logfile group 1;
Statement processed

```

6. 활성 로그를 삭제할 수 없는 오류가 발생하면 다음 로그로 스위치를 강제로 전환하여 잠금을 해제하고 글로벌

체크포인트를 강제로 설정합니다. 예를 들면 다음과 같습니다. 이 로그 파일에 활성 데이터가 있기 때문에 이전 위치에 있던 로그 파일 그룹 3을 삭제하려는 시도가 거부되었습니다. 체크포인트 다음에 로그 아카이빙을 수행하면 로그 파일을 삭제할 수 있습니다.

```
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. 환경을 검토하여 모든 위치 기반 매개 변수가 업데이트되었는지 확인합니다.

```
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. 다음 스크립트는 이 프로세스를 단순화하는 방법을 보여 줍니다.

```

[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. ASM 디스크 그룹이 완전히 비워진 경우 에서 디스크 그룹을 마운트 해제할 수 있습니다 `asmcmd`. 그러나 대부분의 경우 다른 데이터베이스 또는 ASM `spfile/passwd` 파일에 속하는 파일이 여전히 존재할 수 있습니다.

```

-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>

```

Oracle ASM에서 파일 시스템 복사입니다

Oracle ASM-파일 시스템 복사 절차는 ASM과 ASM/ASM 복제 절차와 매우 유사하며, 이점과 제한 사항이 유사합니다. 기본적인 차이점은 ASM 디스크 그룹과 달리 가시적인 파일 시스템을 사용할 때 다양한 명령 및 구성 매개 변수의 구문입니다.

데이터베이스를 복사합니다

Oracle RMAN은 현재 ASM 디스크 그룹에 위치한 소스 데이터베이스의 레벨 0(전체) 복제본을 생성하는 데 사용됩니다 +DATA 의 새 위치로 이동합니다 /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

```

아카이브 로그 스위치를 강제 적용합니다

아카이브 로그 스위치를 강제로 사용하면 아카이브 로그에 복제본의 일관성을 완전히 유지하는 데 필요한 모든 데이터가 포함되도록 할 수 있습니다. 이 명령을 사용하지 않으면 주요 데이터가 재실행 로그에 계속 존재할 수 있습니다. 아카이브 로그 스위치를 강제로 전환하려면 다음 명령을 실행합니다.


```
RMAN> sql 'alter system archive log current';
sql statement: alter system archive log current
```

원본 데이터베이스를 종료합니다

데이터베이스가 종료되고 제한된 액세스 읽기 전용 모드로 전환되기 때문에 이 단계에서 중단이 시작됩니다. 소스 데이터베이스를 종료하려면 다음 명령을 실행합니다.

```
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
```

제어 파일 백업

마이그레이션을 중단하고 원래 스토리지 위치로 되돌려야 하는 경우 제어 파일을 백업합니다. 백업 제어 파일 사본이 100% 필요한 것은 아니지만 데이터베이스 파일 위치를 원래 위치로 다시 설정하는 프로세스가 더 쉬워집니다.

```
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
```

매개 변수 업데이트

```
RMAN> create pfile='/tmp/pfile' from spfile;
Statement processed
```

pfile을 업데이트합니다

이전 ASM 디스크 그룹을 참조하는 모든 매개 변수는 업데이트되어야 하며, 경우에 따라 더 이상 관련이 없을 때 삭제해야 합니다. 새 파일 시스템 경로를 반영하도록 이 경로를 업데이트하고 업데이트된 pfile을 저장합니다. 전체 대상 경로가 나열되어 있는지 확인합니다. 이러한 매개 변수를 업데이트하려면 다음 명령을 실행합니다.

```
*.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'
```

원본 init.ora 파일을 비활성화합니다

이 파일은 에 있습니다 \$ORACLE_HOME/dbs 디렉토리이며 일반적으로 ASM 디스크 그룹의 spfile에 대한 포인터로 사용되는 pfile에 있습니다. 원본 spfile이 더 이상 사용되지 않도록 하려면 이름을 바꿉니다. 그러나 마이그레이션을 중단해야 하는 경우 이 파일이 필요하므로 삭제하지 마십시오.

```
[oracle@jfsc1 ~]$ cd $ORACLE_HOME/dbs  
[oracle@jfsc1 dbs]$ cat initTOAST.ora  
SPFILE='+ASM0/TOAST/spfileTOAST.ora'  
[oracle@jfsc1 dbs]$ mv initTOAST.ora initTOAST.ora.prev  
[oracle@jfsc1 dbs]$
```

매개 변수 파일 재생성

이 단계는 spfile 재배포의 마지막 단계입니다. 원본 spfile은 더 이상 사용되지 않으며 현재 중간 파일을 사용하여 데이터베이스가 시작(마운트되지 않음)됩니다. 이 파일의 내용은 다음과 같이 새 spfile 위치에 쓸 수 있습니다.

```
RMAN> create spfile from pfile='/tmp/pfile';  
Statement processed
```

데이터베이스를 시작하여 새 **spfile** 사용을 시작합니다

중간 파일의 잠금을 해제하고 새 spfile 파일만 사용하여 데이터베이스를 시작하려면 데이터베이스를 시작해야 합니다. 데이터베이스를 시작하면 새 spfile 위치가 올바르고 데이터가 유효하다는 것도 증명됩니다.

```

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

```

제어파일을 복원합니다

경로에 백업 제어 파일이 생성되었습니다 /tmp/TOAST.ctrl 절차의 앞부분에 있습니다. 새 spfile은 제어 파일 위치를 로 정의합니다 /logfs/TOAST/ctrl/ctrlfile1.ctrl 및 /logfs/TOAST/redo/ctrlfile2.ctrl. 그러나 해당 파일은 아직 존재하지 않습니다.

1. 이 명령은 컨트롤 파일 데이터를 spfile에 정의된 경로로 복원합니다.

```

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. mount 명령을 실행하여 제어 파일이 올바르게 검색되고 유효한 데이터가 포함되도록 합니다.

```

RMAN> alter database mount;
Statement processed
released channel: ORA_DISK_1

```

를 확인합니다 control_files 매개 변수에서 다음 명령을 실행합니다.

```

SQL> show parameter control_files;
NAME                                TYPE                                VALUE
-----                                -
control_files                       string
/logs/TOAST/arch/control01.ctl
,
/logs/TOAST/redo/control02.c
tl

```

로그 재생

데이터베이스가 현재 이전 위치의 데이터 파일을 사용하고 있습니다. 복사본을 사용하려면 먼저 데이터 파일을 동기화해야 합니다. 초기 복제 프로세스 중에 시간이 경과했으며 변경 사항은 주로 아카이브 로그에 기록되었습니다. 이러한 변경 사항은 다음 두 단계에 복제됩니다.

1. 아카이브 로그가 포함된 RMAN 증분 백업을 수행합니다.

```

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. 로그를 재생합니다.

```

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

```

활성화

복원된 컨트롤 파일은 원래 위치에 있는 데이터 파일을 참조하며 복사된 데이터 파일의 경로 정보도 포함합니다.

1. 활성 데이터 파일을 변경하려면 `switch database to copy` 명령:

```

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. 데이터 파일은 완전히 일관되어야 하지만 온라인 재실행 로그에 기록된 나머지 변경 내용을 재생하려면 마지막 단계가 필요합니다. `switch` 명령을 사용하여 이러한 변경 사항을 재생하고 복사본을 원본과 100% 동일하게 만듭니다. 하지만 복사본이 아직 열려 있지 않습니다.

```

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

```

임시 데이터 파일 재배치

1. 원본 디스크 그룹에서 여전히 사용 중인 임시 데이터 파일의 위치를 식별합니다.

```

RMAN> select file#||' '||name from v$tempfile;
FILE#||' '||NAME
-----
-----
1 +ASM0/TOAST/temp01.dbf

```

2. 데이터 파일을 재배치하려면 다음 명령을 실행합니다. tempfiles가 많은 경우 텍스트 편집기를 사용하여 RMAN 명령을 생성한 다음 잘라내어 붙여 넣습니다.

```

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

```

로그 마이그레이션을 다시 실행합니다

마이그레이션 프로세스는 거의 완료되었지만 재실행 로그는 여전히 원본 ASM 디스크 그룹에 있습니다. REDO 로그는 직접 재배치할 수 없습니다. 대신 새 redo 로그 세트가 생성되고 구성에 추가되며, 그 다음에 이전 로그가 삭제됩니다.

1. 재실행 로그 그룹의 수와 해당 그룹 번호를 식별합니다.

```

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. redo 로그의 크기를 입력합니다.

```

RMAN> select group#||' '||bytes from v$log;
GROUP#||' '||BYTES
-----
-----
1 52428800
2 52428800
3 52428800

```

3. 각 redo 로그에 대해 새 파일 시스템 위치를 사용하여 현재 redo 로그 그룹과 동일한 크기를 사용하여 새 그룹을 생성합니다.

```

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. 이전 스토리지에 있는 이전 로그 파일 그룹을 제거합니다.

```

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. 활성 로그를 삭제하는 블록에 오류가 발생하는 경우 다음 로그로 스위치를 강제 전환하여 잠금을 해제하고 글로벌 체크포인트를 강제 적용합니다. 예를 들면 다음과 같습니다. 이 로그 파일에 활성 데이터가 있기 때문에 이전 위치에

있던 로그 파일 그룹 3을 삭제하려는 시도가 거부되었습니다. 로그 아카이빙과 체크포인트가 지나면 로그 파일을 삭제할 수 있습니다.

```

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. 환경을 검토하여 모든 위치 기반 매개 변수가 업데이트되었는지 확인합니다.

```

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. 다음 스크립트는 이 프로세스를 보다 쉽게 만드는 방법을 보여 줍니다.


```

[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. ASM 디스크 그룹이 완전히 비워진 경우 에서 디스크 그룹을 마운트 해제할 수 있습니다 `asmcmd`. 대부분의 경우 다른 데이터베이스 또는 ASM `spfile/passwd` 파일에 속하는 파일이 계속 존재할 수 있습니다.

```

-bash-4.1$ . oraenv
ORACLE_SID = [TOAST] ? +ASM
The Oracle base remains unchanged with value /orabin
-bash-4.1$ asmcmd
ASMCMD> umount DATA
ASMCMD>

```

데이터 파일 정리 절차

마이그레이션 프로세스로 인해 Oracle RMAN의 사용 방식에 따라 긴 구문 또는 암호화된 데이터 파일이 생성될 수 있습니다. 여기에 표시된 예에서는 의 파일 형식으로 백업이 수행되었습니다 `/oradata/TOAST/%U.%U RMAN`이 각 데이터 파일에 대해 기본 고유 이름을 생성해야 함을 나타냅니다. 결과는 다음 텍스트에 표시된 것과 유사합니다. 데이터 파일의 기존 이름은 이름 안에 포함됩니다. 이 작업은 에 나와 있는 스크립트된 접근 방식을 사용하여 정리할 수 있습니다 "[ASM 마이그레이션 정리](#)".

```

[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 재조정

앞서 설명한 대로 재조정 프로세스를 사용하여 Oracle ASM 디스크 그룹을 새 스토리지 시스템으로 투명하게 마이그레이션할 수 있습니다. 요약하면 재조정 프로세스에서는 크기가 동일한 LUN을 기존 LUN 그룹에 추가한 다음 이전 LUN의 삭제 작업을 수행해야 합니다. Oracle ASM은 기본 데이터를 최적의 레이아웃으로 새 스토리지로 자동으로 재이동한 다음 완료되면 이전 LUN을 해제합니다.

마이그레이션 프로세스는 효율적인 순차적 I/O를 사용하며 일반적으로 성능 중단을 일으키지 않지만 필요할 때 마이그레이션 속도를 조절할 수 있습니다.

마이그레이션할 데이터를 식별합니다

```

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

```

새 LUN을 생성합니다

동일한 크기의 새 LUN을 생성하고 필요에 따라 사용자 및 그룹 멤버십을 설정합니다. LUN은 로 표시되어야 합니다 CANDIDATE 디스크.

```
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
```

새 LUN을 추가합니다

추가 및 삭제 작업은 함께 수행할 수 있지만 일반적으로 두 단계로 새 LUN을 추가하는 것이 더 쉽습니다. 먼저 새 LUN을 디스크 그룹에 추가합니다. 이 단계를 수행하면 익스텐트의 절반이 현재 ASM LUN에서 새 LUN으로 마이그레이션됩니다.

재조정 성능은 데이터가 전송되는 속도를 나타냅니다. 숫자가 클수록 데이터 전송의 병렬 처리 수가 높아집니다. 마이그레이션은 성능 문제를 일으킬 소지가 없는 효율적인 순차적 I/O 작업을 통해 수행됩니다. 그러나 필요한 경우 진행 중인 마이그레이션의 균형 조정 성능을 로 조정할 수 있습니다 alter diskgroup [name] rebalance power [level] 명령. 일반적인 마이그레이션은 5의 값을 사용합니다.

```
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.
```

작동을 모니터링합니다

재조정 작업을 여러 방법으로 모니터링하고 관리할 수 있습니다. 이 예에서는 다음 명령을 사용했습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
-----
1 REBAL RUN
1 REBAL WAIT
```

마이그레이션이 완료되면 재조정 작업이 보고되지 않습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

기존 **LUN**을 삭제합니다

이제 마이그레이션이 절반 정도 완료되었습니다. 몇 가지 기본 성능 테스트를 수행하여 환경이 양호한지 확인하는 것이 좋습니다. 확인 후 이전 LUN을 삭제하여 나머지 데이터를 재배포할 수 있습니다. 그러나 LUN이 즉시 해제되지는 않습니다. 삭제 작업은 Oracle ASM에 먼저 익스텐트를 재배포한 다음 LUN을 해제하라는 신호를 보냅니다.

```
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.
```

작동을 모니터링합니다

재조정 작업은 여러 가지 방법으로 모니터링 및 관리할 수 있습니다. 이 예에서는 다음 명령을 사용했습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
GROUP_NUMBER OPERA STAT
-----
1 REBAL RUN
1 REBAL WAIT
```

마이그레이션이 완료되면 재조정 작업이 보고되지 않습니다.

```
SQL> select group_number,operation,state from v$asm_operation;
no rows selected
```

이전 LUN을 제거합니다

디스크 그룹에서 기존 LUN을 제거하기 전에 헤더 상태에 대한 최종 확인 작업을 수행해야 합니다. ASM에서 LUN을 릴리즈하면 더 이상 이름이 나열되지 않고 헤더 상태가 로 표시됩니다 FORMER. 이는 이러한 LUN을 시스템에서 안전하게 제거할 수 있음을 나타냅니다.

```
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 마이그레이션

여기에 제시된 절차는 라는 볼륨 그룹의 LVM 기반 마이그레이션 원칙을 보여줍니다 datavg. 그 예가 Linux LVM에서 도출되었지만 원칙은 AIX, HP-UX 및 VxVM에도 동일하게 적용됩니다. 정확한 명령은 다를 수 있습니다.

1. 현재 에 있는 LUN을 식별합니다 datavg 볼륨 그룹:

```
[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. 물리적 크기가 같거나 약간 더 큰 새 LUN을 생성하고 물리적 볼륨으로 정의합니다.

```
[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. 새 볼륨을 볼륨 그룹에 추가합니다.

```
[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. 를 발행합니다 pvmove 명령을 사용하여 각 현재 LUN의 익스텐트를 새 LUN으로 재배포합니다. 를 클릭합니다 - i [seconds] argument 는 작업의 진행률을 모니터링합니다.

```

[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. 이 프로세스가 완료되면 `lv` 사용하여 볼륨 그룹에서 이전 LUN을 삭제합니다 `vgreduce` 명령. 성공하면 이제 시스템에서 LUN을 안전하게 제거할 수 있습니다.

```
[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"
```

외부 LUN импорт

FLI-Planning을 사용한 Oracle 마이그레이션

FLI를 사용하여 SAN 리소스를 마이그레이션하는 절차는 NetApp에 설명되어 있습니다 "[TR-4380: 외부 LUN 가져오기를 사용한 SAN 마이그레이션](#)".

데이터베이스 및 호스트 관점에서 볼 때 특별한 단계가 필요하지 않습니다. FC 존이 업데이트되고 LUN을 ONTAP에서 사용할 수 있게 되면 LVM이 LUN에서 LVM 메타데이터를 읽을 수 있어야 합니다. 또한 볼륨 그룹을 추가 구성 단계 없이 사용할 수 있습니다. 드문 경우지만 환경에 이전 스토리지 어레이를 참조하는 하드 코딩된 구성 파일이 포함될 수 있습니다. 예를 들어, 을 포함하는 Linux 시스템이 해당됩니다 `/etc/multipath.conf` 주어진 디바이스의 WWN을 참조하는 규칙은 FLI에 의해 도입된 변경 사항을 반영하도록 업데이트되어야 합니다.



지원되는 구성에 대한 자세한 내용은 NetApp 호환성 매트릭스를 참조하십시오. 사용 환경이 포함되지 않은 경우 NetApp 담당자에게 문의하십시오.

이 예제는 Linux 서버에서 호스팅되는 ASM 및 LVM LUN의 마이그레이션을 보여줍니다. FLI는 다른 운영 체제에서 지원되며 호스트측 명령이 다를 수 있지만 원칙이 동일하고 ONTAP 절차는 동일합니다.

LVM LUN을 식별합니다

준비의 첫 번째 단계는 마이그레이션할 LUN을 식별하는 것입니다. 여기에 나와 있는 예에서는 SAN 기반 파일 시스템 두 개가 에 마운트되어 있습니다 `/orabin` 및 `/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
```

볼륨 그룹의 이름은 (볼륨 그룹 이름) - (논리적 볼륨 이름) 형식을 사용하는 디바이스 이름에서 추출할 수 있습니다. 이 경우 볼륨 그룹이 호출됩니다 sanvg.

를 클릭합니다 pvdisplay 명령을 사용하여 다음 명령을 사용하여 이 볼륨 그룹을 지원하는 LUN을 식별할 수 있습니다. 이 경우 를 구성하는 LUN이 10개 있습니다 sanvg 볼륨 그룹:

```
[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
```

ASM LUN 식별

ASM LUN도 마이그레이션해야 합니다. sysasm 사용자로 sqlplus에서 LUN 및 LUN 경로 수를 가져오려면 다음 명령을 실행합니다.

```
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 네트워크 변경 사항

현재 환경에는 마이그레이션할 LUN 20개가 포함되어 있습니다. ONTAP가 현재 LUN에 액세스할 수 있도록 현재 SAN을 업데이트합니다. 데이터가 아직 마이그레이션되지 않았지만 ONTAP는 현재 LUN에서 구성 정보를 읽어 해당 데이터의 새 홈을 생성해야 합니다.

AFF/FAS 시스템에서 하나 이상의 HBA 포트를 이니시에이터 포트에 구성해야 합니다. 또한 ONTAP이 외부 스토리지 어레이의 LUN에 액세스할 수 있도록 FC 존을 업데이트해야 합니다. 일부 스토리지에는 LUN 마스킹이 구성되어 있으며, 이 경우 특정 LUN을 액세스할 수 있는 WWN이 제한됩니다. 이 경우 LUN 마스킹도 업데이트하여 ONTAP WWN에 대한 액세스 권한을 부여해야 합니다.

이 단계가 완료된 후 ONTAP는 를 사용하여 외부 스토리지 어레이를 볼 수 있어야 합니다 `storage array show` 명령. 반환되는 키 필드는 시스템에서 외부 LUN을 식별하는 데 사용되는 접두사입니다. 아래 예에서는 외부 스토리지의 LUN을 보여 줍니다 FOREIGN_1 의 접두사를 사용하여 ONTAP 내에 나타납니다 FOR-1.

외부 어레이를 식별합니다

```
Cluster01::> storage array show -fields name,prefix
name          prefix
-----
FOREIGN_1     FOR-1
Cluster01::>
```

외부 LUN 식별

LUN은 를 전달하여 나열할 수 있습니다 `array-name` 를 누릅니다 `storage disk show` 명령. 반환되는 데이터는 마이그레이션 절차 중에 여러 번 참조됩니다.

```

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$HuVWBo
FOR-1.20  800DT$HuVWBn
20 entries were displayed.
Cluster01::>

```

외부 스토리지 **LUN**을 가져오기 후보로 등록합니다

외부 LUN은 처음에 특정 LUN 유형으로 분류됩니다. 데이터를 가져오려면 먼저 LUN에 외부 태그가 지정되어야 하므로 가져오기 프로세스의 대상이 되어야 합니다. 이 단계는 에 일련 번호를 전달하여 완료합니다 storage disk modify 명령을 사용합니다. 이 프로세스에서는 ONTAP 내의 LUN에만 Foreign으로 태그를 지정합니다. 외부 LUN 자체에 데이터가 기록되지 않습니다.

```

Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBW} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBX} -is
-foreign true
...
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBn} -is
-foreign true
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWBo} -is
-foreign true
Cluster01::*>

```

마이그레이션된 LUN을 호스팅할 볼륨을 생성합니다

마이그레이션된 LUN을 호스팅하려면 볼륨이 필요합니다. 정확한 볼륨 구성은 ONTAP 기능을 활용하는 전체 계획에 따라 다릅니다. 이 예에서는 ASM LUN이 한 볼륨에 배치되고 LVM LUN은 두 번째 볼륨에 배치됩니다. 따라서 계층화, 스냅샷 생성 또는 QoS 제어 설정과 같은 목적으로 LUN을 독립 그룹으로 관리할 수 있습니다.

를 설정합니다 snapshot-policy `to` none. 마이그레이션 프로세스에는 많은 데이터 회전율이 포함될 수 있습니다. 따라서 원하지 않는 데이터가 스냅샷에 캡처되기 때문에 실수로 스냅샷을 생성하는 경우 공간 소비가 크게 증가할 수 있습니다.

```
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::>
```

ONTAP LUN을 생성합니다

볼륨을 생성한 후에는 새 LUN을 생성해야 합니다. 일반적으로 LUN을 생성할 때 사용자가 LUN 크기와 같은 정보를 지정해야 하지만 이 경우에는 외부 디스크 인수가 명령에 전달됩니다. 따라서 ONTAP는 지정된 일련 번호에서 현재 LUN 구성 데이터를 복제합니다. 또한 LUN 형태 및 파티션 테이블 데이터를 사용하여 LUN 정렬을 조정하고 최적의 성능을 설정합니다.

이 단계에서는 올바른 외부 LUN이 올바른 새 LUN과 일치하는지 확인하기 위해 외부 스토리지와 일련 번호를 상호 참조해야 합니다.

```
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)
```

가져오기 관계를 만듭니다

이제 LUN이 생성되었지만 복제 대상으로 구성되지 않았습니다. 이 단계를 수행하려면 먼저 LUN을 오프라인 상태로 전환해야 합니다. 이 추가 단계는 사용자 오류로부터 데이터를 보호하도록 설계되었습니다. ONTAP에서 온라인 LUN에

대해 마이그레이션을 수행할 수 있었다면 인쇄 오류로 인해 활성 데이터를 덮어쓸 위험이 발생할 수 있습니다. 사용자가 먼저 LUN을 오프라인으로 전환하도록 하는 추가 단계는 올바른 타겟 LUN이 마이그레이션 대상으로 사용되는지 확인하는 데 도움이 됩니다.

```
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
```

LUN이 오프라인 상태가 된 후 외부 LUN 일련 번호를 에 전달하여 임포트 관계를 설정할 수 있습니다 `lun import create` 명령.

```
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::*>
```

모든 임포트 관계가 설정되면 LUN을 다시 온라인 상태로 전환할 수 있습니다.

```
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::*>
```

이니시에이터 그룹을 생성합니다

igroup(이니시에이터 그룹)은 ONTAP LUN 마스킹 아키텍처의 일부입니다. 호스트에 처음으로 액세스 권한이 부여되지 않으면 새로 생성된 LUN에 액세스할 수 없습니다. 이 작업은 액세스 권한을 부여해야 할 FC WWN 또는 iSCSI 이니시에이터 이름을 나열하는 igroup을 생성하여 수행합니다. 이 보고서가 작성된 시점을 기준으로 FLI는 FC LUN에 대해서만 지원됩니다. 그러나 예 나와 있는 것처럼 iSCSI 사후 마이그레이션으로 변환하는 작업은 간단합니다 "[프로토콜 변환](#)".

이 예에서는 호스트의 HBA에서 사용 가능한 두 포트에 해당하는 두 개의 WWN이 포함된 igroup을 생성합니다.

```
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
```

호스트에 새 LUN 매핑

igroup 작성 후에 LUN이 정의된 igroup에 매핑됩니다. 이 LUN은 이 igroup에 포함된 WWN에만 사용할 수 있습니다. NetApp는 마이그레이션 프로세스에서 이 단계에서 호스트가 ONTAP에 조닝되지 않은 것으로 가정합니다. 이는 호스트가 외부 스토리지와 새 ONTAP 시스템에 동시에 조닝되는 경우 동일한 일련 번호를 가진 LUN이 각 어레이에서 검색될 위험이 있기 때문에 중요합니다. 이 상황은 다중 경로 오작동이나 데이터 손상으로 이어질 수 있습니다.

```
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::*>
```

FLI-컷오버를 사용한 Oracle 마이그레이션

FC 네트워크 구성을 변경해야 하기 때문에 외부 LUN 임포트 중에는 중단이 불가피합니다. 그러나 운영 중단은 데이터베이스 환경을 재시작하고 FC 조닝을 업데이트하여 호스트 FC 연결을 외부 LUN에서 ONTAP로 전환하는 데 필요한 시간보다 훨씬 오래 지속되지 않습니다.

이 프로세스는 다음과 같이 요약할 수 있습니다.

1. 외부 LUN에서 모든 LUN 작업을 중지합니다.
2. 호스트 FC 연결을 새 ONTAP 시스템으로 리디렉션합니다.
3. 가져오기 프로세스를 트리거합니다.
4. LUN을 다시 검색합니다.
5. 데이터베이스를 다시 시작합니다.

마이그레이션 프로세스가 완료될 때까지 기다리지 않아도 됩니다. 특정 LUN의 마이그레이션이 시작되는 즉시 ONTAP에서 사용할 수 있으며 데이터 복사 프로세스가 진행되는 동안 데이터를 제공할 수 있습니다. 모든 읽기는 외부 LUN으로 전달되고 모든 쓰기는 두 스토리지에 동기식으로 기록됩니다. 복사 작업은 매우 빠르고 FC 트래픽 리디렉션의 오버헤드가 최소화되므로 성능에 미치는 영향은 일시적이고 최소화해야 합니다. 문제가 있는 경우 마이그레이션 프로세스가 완료되고 가져오기 관계가 삭제될 때까지 환경 다시 시작을 지연시킬 수 있습니다.

데이터베이스를 종료합니다

이 예에서 환경을 정지하는 첫 번째 단계는 데이터베이스를 종료하는 것입니다.

```
[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>
```

그리드 서비스를 종료합니다

마이그레이션되는 SAN 기반 파일 시스템 중 하나에 Oracle ASM 서비스도 포함됩니다. 기본 LUN을 정지하려면 파일 시스템을 마운트 해제해야 합니다. 즉, 이 파일 시스템에서 열려 있는 파일이 있는 프로세스를 모두 중지해야 합니다.

```
[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]$
```

파일 시스템을 마운트 해제합니다

모든 프로세스가 종료되면 마운트 해제 작업이 성공합니다. 사용 권한이 거부되면 파일 시스템에 잠금이 설정된 프로세스가 있어야 합니다. `l` 를 클릭합니다 `fuser` 명령은 이러한 프로세스를 식별하는 데 도움이 될 수 있습니다.

```
[root@host1 ~]# umount /orabin
[root@host1 ~]# umount /backups
```

볼륨 그룹을 비활성화합니다

지정된 볼륨 그룹의 모든 파일 시스템이 마운트 해제된 후 볼륨 그룹을 비활성화할 수 있습니다.

```
[root@host1 ~]# vgchange --activate n sanvg
  0 logical volume(s) in volume group "sanvg" now active
[root@host1 ~]#
```

FC 네트워크 변경 사항

이제 FC 존을 업데이트하여 호스트에서 외부 스토리지에 대한 모든 액세스를 제거하고 ONTAP에 대한 액세스를 설정할 수 있습니다.

가져오기 프로세스를 시작합니다

LUN 가져오기 프로세스를 시작하려면 `l` 를 실행합니다 `lun import start` 명령.


```

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*>

```

가져오기 진행 상황을 모니터링합니다

를 사용하여 가져오기 작업을 모니터링할 수 있습니다 `lun import show` 명령. 아래와 같이 20개의 LUN을 모두 가져오는 작업이 진행 중입니다. 따라서 이제 데이터 복사 작업이 계속 진행되더라도 ONTAP를 통해 데이터에 액세스할 수 있습니다.

```

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$HuVWBa /vol/new_asm/LUN5 4
vserver1   800DT$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$HuVWBj /vol/new_lvm/LUN3 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN4 3
vserver1   800DT$HuVWBk /vol/new_lvm/LUN5 3
vserver1   800DT$HuVWB1 /vol/new_lvm/LUN6 4
vserver1   800DT$HuVWBm /vol/new_lvm/LUN7 3
vserver1   800DT$HuVWBn /vol/new_lvm/LUN8 2
vserver1   800DT$HuVWBo /vol/new_lvm/LUN9 2
20 entries were displayed.

```

오프라인 프로세스가 필요한 경우 까지 서비스 재검색 또는 재시작을 연기합니다 `lun import show` 명령은 모든 마이그레이션이 성공적이고 완료되었음을 나타냅니다. 그런 다음 에 설명된 대로 마이그레이션 프로세스를 완료할 수 있습니다 ["외부 LUN импорт - 완료"](#).

온라인 마이그레이션이 필요한 경우 새 집에서 LUN을 다시 검색하고 서비스를 시작합니다.

SCSI 장치 변경 사항을 검색합니다

대부분의 경우 새 LUN을 다시 검색하는 가장 간단한 옵션은 호스트를 재시작하는 것입니다. 이렇게 하면 오래된 오래된 장치가 자동으로 제거되고 모든 새 LUN이 올바르게 검색되며 다중 경로 장치와 같은 관련 장치가 구축됩니다. 이 예제에서는 데모를 위한 완전한 온라인 프로세스를 보여 줍니다.

주의: 호스트를 다시 시작하기 전에 의 모든 항목이 있는지 확인하십시오 `/etc/fstab` 마이그레이션된 참조 SAN 리소스가 주석 처리되었습니다. 이렇게 하지 않고 LUN 액세스에 문제가 있으면 운영 체제가 부팅되지 않을 수 있습니다. 이 상황은 데이터를 손상시키지 않습니다. 그러나 구조 모드 또는 유사한 모드로 부팅하고 를 수정하는 것은 매우 불편할 수 있습니다 `/etc/fstab` 문제 해결을 위해 운영 체제를 부팅할 수 있습니다.

이 예에 사용된 Linux 버전의 LUN을 로 다시 검색할 수 있습니다 `rescan-scsi-bus.sh` 명령. 명령이 성공하면 각 LUN 경로가 출력에 표시되어야 합니다. 출력에서 해석하기가 어려울 수 있지만, 조닝 및 `igroup` 구성이 올바르게 을 포함하는 많은 LUN이 표시되어야 합니다 NETAPP 공급업체 문자열

```

[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.

```

다중 경로 장치를 확인합니다

LUN 검색 프로세스는 다중 경로 장치의 재구성을 트리거하지만 Linux 다중 경로 드라이버에는 간헐적인 문제가 있는 것으로 알려져 있습니다. 의 출력입니다 `multipath - ll` 출력이 예상한 대로 나타나는지 확인해야 합니다. 예를 들어, 아래 출력에는 와 연결된 다중 경로 장치가 나와 있습니다 NETAPP 공급업체 문자열 각 디바이스에는 4개의 경로가 있으며, 우선 순위가 50이고 우선 순위가 10인 2개의 경로가 있습니다. 정확한 출력은 Linux 버전에 따라 다를 수 있지만 이 출력은 예상한 대로 표시됩니다.



사용하는 Linux 버전에 대한 호스트 유틸리티 설명서를 참조하여 를 확인하십시오
/etc/multipath.conf 설정이 올바릅니다.

```
[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 sdbn 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
```

LVM 볼륨 그룹을 다시 활성화합니다

LVM LUN이 제대로 검색되면 가 나타납니다 `vgchange --activate y` 명령이 성공해야 합니다. 이것은 논리적

볼륨 관리자의 값에 대한 좋은 예입니다. 볼륨 그룹 메타데이터가 LUN 자체에 기록되므로 LUN의 WWN이나 일련 번호도 변경하는 것은 중요하지 않습니다.

OS는 LUN을 검색하여 LUN에 기록된 소량의 데이터를 발견했습니다. 이 데이터는 LUN에 속한 물리적 볼륨으로 식별됩니다 `sanvg volumegroup`. 그런 다음 필요한 모든 장치를 구축했습니다. 볼륨 그룹을 다시 활성화하기만 하면 됩니다.

```
[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
```

파일 시스템을 다시 마운트합니다

볼륨 그룹이 다시 활성화된 후 모든 원본 데이터가 손상되지 않은 상태로 파일 시스템을 마운트할 수 있습니다. 앞서 설명했듯이 백 그룹에서 데이터 복제가 아직 활성 상태인 경우에도 파일 시스템이 완전히 작동합니다.

```
[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
```

ASM 장치를 다시 검색합니다

SCSI 장치를 다시 검색할 때 ASMLib 장치를 다시 검색해야 합니다. ASMLib를 다시 시작한 다음 디스크를 검사하여 온라인으로 재검색을 확인할 수 있습니다.



이 단계는 ASMLib가 사용되는 ASM 구성에만 관련이 있습니다.

주의: ASMLib를 사용하지 않는 경우 /dev/mapper 디바이스가 자동으로 다시 생성되어야 합니다. 그러나 사용 권한이 올바르지 않을 수 있습니다. ASMLib가 없는 경우 ASM에 대한 기본 장치에 특수 권한을 설정해야 합니다. 이러한 작업은 일반적으로 둘 중 하나의 특수 항목을 통해 수행됩니다 /etc/multipath.conf 또는 udev 두 규칙 집합에서 사용할 수 있습니다. 이러한 파일은 WWN 또는 일련 번호 측면에서 환경의 변경 사항을 반영하도록 업데이트하여 ASM 장치에 올바른 권한이 있는지 확인해야 할 수 있습니다.

이 예에서는 ASMLib를 다시 시작하고 디스크 검색을 통해 원래 환경과 동일한 10개의 ASM LUN을 표시합니다.

```
[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"
```

그리드 서비스를 다시 시작합니다

이제 LVM 및 ASM 장치가 온라인 상태이고 사용 가능해졌으므로 그리드 서비스를 다시 시작할 수 있습니다.

```
[root@host1 ~]# cd /orabin/product/12.1.0/grid/bin
[root@host1 bin]# ./crsctl start has
```

데이터베이스를 다시 시작합니다

그리드 서비스가 다시 시작된 후 데이터베이스를 불러올 수 있습니다. 데이터베이스를 시작하기 전에 ASM 서비스를 완전히 사용할 수 있도록 몇 분 정도 기다려야 할 수 있습니다.

```
[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>
```

FLI를 사용한 **Oracle** 마이그레이션 완료

호스트 관점에서 보면 마이그레이션이 완료되지만 가져오기 관계가 삭제될 때까지 외부 스토리지에서 입출력이 계속 제공됩니다.

관계를 삭제하기 전에 모든 LUN에 대해 마이그레이션 프로세스가 완료되었는지 확인해야 합니다.

```

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$HuVWB e /vol/new_asm/LUN9 completed
vserver1 800DT$HuVWBf /vol/new_lvm/LUN0 completed
vserver1 800DT$HuVWBg /vol/new_lvm/LUN1 completed
vserver1 800DT$HuVWBh /vol/new_lvm/LUN2 completed
vserver1 800DT$HuVWB i /vol/new_lvm/LUN3 completed
vserver1 800DT$HuVWBj /vol/new_lvm/LUN4 completed
vserver1 800DT$HuVWBk /vol/new_lvm/LUN5 completed
vserver1 800DT$HuVWB l /vol/new_lvm/LUN6 completed
vserver1 800DT$HuVWBm /vol/new_lvm/LUN7 completed
vserver1 800DT$HuVWBn /vol/new_lvm/LUN8 completed
vserver1 800DT$HuVWB o /vol/new_lvm/LUN9 completed
20 entries were displayed.

```

관계 가져오기를 삭제합니다

마이그레이션 프로세스가 완료되면 마이그레이션 관계를 삭제합니다. I/O를 완료한 후에는 ONTAP의 드라이브에서만 I/O를 처리합니다.

```

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

```

외부 LUN 등록을 취소합니다

마지막으로 디스크를 수정하여 를 제거합니다 is-foreign 지정.


```

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$HuVVBn} -is
-foreign false
Cluster01::*> storage disk modify {-serial-number 800DT$HuVWB0} -is
-foreign false
Cluster01::*>

```

FLI-프로토콜 변환을 통한 Oracle 마이그레이션

LUN에 액세스하는 데 사용되는 프로토콜을 변경하는 것은 일반적인 요구사항입니다.

데이터를 클라우드로 마이그레이션하는 전체 전략의 일부이기도 한 경우도 있습니다. TCP/IP는 클라우드의 프로토콜이며 FC에서 iSCSI로 변경하면 다양한 클라우드 환경으로 쉽게 마이그레이션할 수 있습니다. 그렇지 않으면 iSCSI가 IP SAN의 감소된 비용을 활용하는 것이 바람직할 수도 있습니다. 경우에 따라 마이그레이션이 임시 조치로 다른 프로토콜을 사용할 수 있습니다. 예를 들어, 외부 스토리지 시스템과 ONTAP 기반 LUN이 동일한 HBA에 공존할 수 없는 경우 기존 스토리지의 데이터를 복제할 수 있을 정도로 긴 iSCSI LUN을 사용할 수 있습니다. 그런 다음 이전 LUN을 시스템에서 제거한 후 FC로 다시 변환할 수 있습니다.

다음 절차는 FC에서 iSCSI로 변환하는 방법을 보여 주지만 전반적인 원칙은 역방향 iSCSI에서 FC로 변환하는 방법에 적용됩니다.

iSCSI 이니시에이터를 설치합니다

대부분의 운영 체제에는 기본적으로 소프트웨어 iSCSI 초기자가 포함되어 있지만 포함되어 있지 않은 경우 쉽게 설치할 수 있습니다.

```

[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.e17 will be
updated
--> Processing Dependency: iscsi-initiator-utils = 6.2.0.873-32.e17 for
package: iscsi-initiator-utils-iscsiuio-6.2.0.873-32.e17.x86_64
---> Package iscsi-initiator-utils.x86_64 0:6.2.0.873-32.0.2.e17 will be
an update
--> Running transaction check
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.e17 will
be updated
---> Package iscsi-initiator-utils-iscsiuio.x86_64 0:6.2.0.873-32.0.2.e17

```

will be an update

--> Finished Dependency Resolution

Dependencies Resolved

=====
===

Package	Arch	Version	Repository
---------	------	---------	------------

=====
===

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.e17.x86_64
3/4
Verifying   : iscsi-initiator-utils-6.2.0.873-32.e17.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 /]#
```

iSCSI 이니시에이터 이름을 식별합니다

설치 프로세스 중에 고유한 iSCSI 이니시에이터 이름이 생성됩니다. Linux에서는 에 있습니다
/etc/iscsi/initiatorname.iscsi 파일. 이 이름은 IP SAN에서 호스트를 식별하는 데 사용됩니다.

```
[root@host1 /]# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1992-05.com.redhat:497bd66ca0
```

새 이니시에이터 그룹을 생성합니다

igroup(이니시에이터 그룹)은 ONTAP LUN 마스킹 아키텍처의 일부입니다. 호스트에 처음으로 액세스 권한이 부여되지 않으면 새로 생성된 LUN에 액세스할 수 없습니다. 이 단계는 액세스가 필요한 FC WWN 또는 iSCSI 이니시에이터 이름을 나열하는 igroup을 생성하여 수행합니다.

이 예에서는 Linux 호스트의 iSCSI 이니시에이터가 포함된 igroup이 생성됩니다.

```
Cluster01::*> igroup create -igroup linuxiscsi -protocol iscsi -ostype
linux -initiator iqn.1994-05.com.redhat:497bd66ca0
```

환경을 종료합니다

LUN 프로토콜을 변경하기 전에 LUN을 완전히 정지해야 합니다. 변환 중인 LUN 중 하나의 데이터베이스를 종료하고 파일 시스템을 마운트 해제해야 하며 볼륨 그룹을 비활성화해야 합니다. ASM이 사용되는 경우 ASM 디스크 그룹이 분리되고 모든 그리드 서비스가 종료되는지 확인합니다.

FC 네트워크에서 LUN 매핑을 해제합니다

LUN이 완전히 정지된 후 원본 FC igroup에서 매핑을 제거합니다.

```
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
```

LUN을 IP 네트워크에 다시 매핑합니다

새 iSCSI 기반 이니시에이터 그룹에 각 LUN에 대한 액세스 권한을 부여합니다.

```
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::*>
```

iSCSI 대상을 검색합니다

iSCSI 검색에는 두 단계가 있습니다. 첫 번째는 LUN 검색과 다른 타겟을 검색하는 것입니다. 를 클릭합니다 `iscsiadm` 아래 표시된 명령은 에서 지정한 포털 그룹을 검색합니다 `-p argument` 및 는 iSCSI 서비스를 제공하는 모든 IP 주소 및 포트의 목록을 저장합니다. 이 경우 기본 포트 3260에 iSCSI 서비스가 있는 네 개의 IP 주소가 있습니다.



대상 IP 주소에 연결할 수 없는 경우 이 명령을 완료하는 데 몇 분 정도 걸릴 수 있습니다.

```
[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
```

iSCSI LUN을 검색합니다

iSCSI 대상이 검색된 후 iSCSI 서비스를 다시 시작하여 사용 가능한 iSCSI LUN을 검색하고 다중 경로 또는 ASMLib 디바이스와 같은 관련 디바이스를 구축합니다.

```
[root@host1 ~]# service iscsi restart
Redirecting to /bin/systemctl restart iscsi.service
```

환경을 다시 시작합니다

볼륨 그룹을 다시 활성화하고, 파일 시스템을 다시 마운트하고, RAC 서비스를 다시 시작하는 등의 방법으로 환경을 다시 시작합니다. 예방 조치로, NetApp 변환 프로세스가 완료된 후 서버를 재부팅하여 모든 구성 파일이 올바르게 오래된 모든 디바이스가 제거되도록 하는 것이 좋습니다.

주의: 호스트를 다시 시작하기 전에 의 모든 항목이 있는지 확인하십시오 /etc/fstab 마이그레이션된 참조 SAN 리소스가 주석 처리되었습니다. 이 단계를 수행하지 않고 LUN 액세스에 문제가 있는 경우 운영 체제가 부팅되지 않을 수 있습니다. 이 문제는 데이터를 손상시키지 않습니다. 그러나 구조 모드 또는 유사한 모드로 부팅하고 수정하는 것은 매우 불편할 수 있습니다 /etc/fstab 문제 해결 노력을 시작할 수 있도록 운영 체제를 부팅할 수 있습니다.

Oracle 마이그레이션 절차 샘플 스크립트

제공되는 스크립트는 다양한 OS 및 데이터베이스 작업을 스크립팅하는 방법의 예로 제공됩니다. 그들은 있는 그대로 제공됩니다. 특정 절차에 대한 지원이 필요한 경우 NetApp 또는 NetApp 리셀러에게 문의하십시오.

데이터베이스 종료

다음 Perl 스크립트는 Oracle SID의 단일 인수를 사용하고 데이터베이스를 종료합니다. Oracle 사용자 또는 루트로 실행할 수 있습니다.

```

#!/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;}

```

데이터베이스 시작

다음 Perl 스크립트는 Oracle SID의 단일 인수를 사용하고 데이터베이스를 종료합니다. Oracle 사용자 또는 루트로 실행할 수 있습니다.

```

#!/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;}

```

파일 시스템을 읽기 전용으로 변환합니다

다음 스크립트는 파일 시스템 인수를 사용하여 읽기 전용으로 마운트 해제 및 다시 마운트하려고 시도합니다. 이렇게 하면 데이터를 복제하기 위해 파일 시스템을 사용할 수 있어야 하지만 우발적인 손상으로부터 보호해야 하는 마이그레이션 프로세스 중에 유용합니다.

```

#!/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;

```

파일 시스템을 교체합니다

다음 스크립트 예제는 파일 시스템 하나를 다른 파일 시스템으로 바꾸는 데 사용됩니다. '/etc/fstab' 파일을 편집하므로 루트로 실행해야 합니다. 이전 파일 시스템과 새 파일 시스템의 심표로 구분된 단일 인수를 사용할 수 있습니다.

1. 파일 시스템을 교체하려면 다음 스크립트를 실행합니다.

```

#!/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;

```

이 스크립트 사용의 예로, 의 데이터를 가정합니다 /oradata 로 마이그레이션됩니다 /neworadata 및 /logs 로 마이그레이션됩니다 /newlogs. 이 작업을 수행하는 가장 간단한 방법 중 하나는 간단한 파일 복제 작업을 사용하여 새 디바이스를 원래 마운트 지점으로 재배치하는 것입니다.

2. 예 이전 파일 시스템과 새 파일 시스템이 있다고 가정합니다 /etc/fstab 다음과 같은 파일:

```

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. 이 스크립트를 실행하면 현재 파일 시스템을 마운트 해제하고 새 파일 시스템으로 대체합니다.

```

[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. 스크립트도 를 업데이트합니다 /etc/fstab 그에 따라 보관합니다. 여기에 표시된 예에서는 다음과 같은 변경 사항이 포함되어 있습니다.

```

#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

```

자동화된 데이터베이스 마이그레이션

이 예제에서는 마이그레이션을 완전히 자동화하기 위해 종료, 시작 및 파일 시스템 교체 스크립트를 사용하는 방법을 보여 줍니다.

```

#!/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;

```

파일 위치를 표시합니다

이 스크립트는 많은 중요한 데이터베이스 매개 변수를 수집하여 읽기 쉬운 형식으로 인쇄합니다. 이 스크립트는 데이터 레이아웃을 검토할 때 유용할 수 있습니다. 또한 스크립트를 다른 용도로 수정할 수도 있습니다.

```
#!/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 마이그레이션 정리

```

#!/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";

```

ASM에서 파일 시스템 이름으로 변환

```
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;
/
```

데이터베이스에서 로그를 재생합니다

이 스크립트는 마운트 모드에 있는 데이터베이스에 대해 Oracle SID의 단일 인수를 허용하고 현재 사용 가능한 모든 아카이브 로그를 재생하려고 시도합니다.

```
#!/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;
```

대기 데이터베이스에서 로그를 재생합니다

이 스크립트는 대기 데이터베이스용으로 설계되었다는 점을 제외하고 위의 스크립트와 동일합니다.


```

#!/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;

```

추가 참고 사항

Oracle 데이터베이스 성능 최적화 및 벤치마킹 절차

데이터베이스 스토리지 성능을 정확히 테스트한다는 것은 매우 복잡한 일입니다. 이 작업을 수행하려면 다음 문제에 대한 이해가 필요합니다.

- IOPS 및 처리량
- 포그라운드와 백그라운드 I/O 작업 간의 차이
- 지연 시간이 데이터베이스에 미치는 영향
- 스토리지 성능에도 영향을 주는 수많은 OS 및 네트워크 설정

비스토리지 데이터베이스도 고려해야 합니다. 스토리지 성능이 더 이상 성능을 제한하는 요인이 아니기 때문에 스토리지 성능 최적화가 유용한 이점을 전혀 제공하지 않는 경우도 있습니다.

이제 데이터베이스 고객의 대다수가 All-Flash 어레이를 선택하며 이 어레이에는 추가 고려사항이 수반됩니다. 예를

들어, 2노드 AFF A900 시스템에서 성능 테스트를 생각해 봅시다.

- 80/20 읽기/쓰기 비율로 2개의 A900 노드는 지연 시간이 150 μ s 표시를 넘어가기도 전에 1M 이상의 랜덤 데이터베이스 IOPS를 제공할 수 있습니다. 이는 대부분의 데이터베이스에서 현재 요구되는 성능 수준을 훨씬 뛰어넘는 것이라 어느 정도로 개선될지를 예측하기 어렵습니다. 스토리지는 병목 현상에서 제외될 것입니다
- 네트워크 대역폭은 성능 한계의 공통된 원인으로 부상하고 있습니다. 예를 들어, 회전식 디스크 솔루션은 I/O 지연 시간이 매우 길기 때문에 종종 데이터베이스 성능의 병목 현상을 발생시킵니다. All-Flash 어레이를 통해 지연 시간 한계를 제거하고 나면 이번에는 네트워크가 장애물이 되는 경우가 많습니다. 이는 진정한 네트워크 연결의 시각화가 어려운 가상화 환경과 블레이드 시스템에서 특히 두드러지며 대역폭 한계로 인해 스토리지 시스템 자체를 최대한 활용할 수 없는 경우 성능 테스트가 복잡해질 수 있습니다.
- All-Flash 어레이의 지연 시간이 대폭 개선되었기 때문에 All-Flash 어레이의 성능을 회전식 디스크가 포함된 어레이와 비교하는 것은 일반적으로 불가능합니다. 테스트 결과는 대체로 중요하지 않습니다.
- 데이터베이스는 스토리지 I/O에 의한 제약이 없기 때문에 최고의 IOPS 성능을 All-Flash 어레이와 비교하는 테스트는 대부분 유용하지 않습니다 한 어레이는 500K 랜덤 IOPS를 유지할 수 있고 다른 어레이는 300K를 유지할 수 있다고 가정해 봅시다. 이 차이는 데이터베이스가 실제로 99%의 시간을 CPU 처리에 사용하는 경우 아무런 상관이 없습니다. 워크로드는 스토리지 어레이의 전체 용량을 활용하지 않습니다. 이와 반대로, 최고 IOPS 용량은 스토리지 어레이가 최고 성능으로 로드될 것으로 예상되는 통합 플랫폼에서 매우 중요할 것입니다.
- 어떤 스토리지 테스트에서든 지연 시간과 IOPS를 항상 고려하십시오. 시중의 많은 스토리지 어레이가 IOPS 레벨이 매우 높다는 주장을 펼치지만 지연 시간 때문에 이들 IOPS는 그러한 레벨에서 쓸모가 없습니다. All-Flash 어레이의 일반적인 타겟은 1ms 표시입니다. 더 나은 테스트 방식은 가능한 최대의 IOPS를 측정하는 것이 아니라 평균 지연 시간이 1ms를 넘기 전에 스토리지 어레이가 유지할 수 있도록 할 IOPS를 결정하는 것입니다.

Oracle Automatic Workload Repository 및 벤치마킹

Oracle 성능 비교를 위한 이상적인 표준은 Oracle AWR(Automatic Workload Repository) 보고서입니다.

AWR 보고서에는 여러 유형이 있습니다. 스토리지 관점에서, 를 실행하여 생성된 보고서 `awrrpt.sql` 명령은 특정 데이터베이스 인스턴스를 대상으로 하고 지연 시간을 기반으로 스토리지 I/O 이벤트를 분석하는 상세한 히스토그램이 포함되어 있기 때문에 가장 종합적이고 유용합니다.

2개의 성능 어레이를 비교할 때 가장 이상적인 방식은 각 어레이에서 같은 워크로드를 실행하여 워크로드를 정확히 타겟으로 하는 AWR 보고서를 생성하는 것입니다. 장시간 실행되는 워크로드의 경우, 시작 시간과 종료 시간을 아우르는 경과 시간을 토대로 단일 AWR 보고서를 사용할 수 있으며 여러 보고서로 AWR 데이터를 분석하는 것이 더 좋습니다. 예를 들어, 일괄 작업이 자정부터 오전 6시까지 실행된 경우 자정부터 오전 1시, 오전 1시부터 오전 2시 등등 일련의 1시간 간격 AWR 보고서를 생성하십시오.

다른 경우에는 매우 짧은 쿼리를 최적화해야 합니다. 최고의 옵션은 쿼리가 시작될 때 생성된 AWR 스냅샷과 쿼리가 끝날 때 생성된 두 번째 AWR 스냅샷에 기반을 두고 AWR 보고서를 생성하는 것입니다. 그렇지 않으면, 분석 중인 쿼리의 활동을 종료하지 않게 만드는 백그라운드 활동을 최소화하기 위해 데이터베이스 서버가 정지되어야 합니다.



AWR 보고서를 생성할 수 없는 경우 Oracle Statspack 보고서가 좋은 대안이 될 수 있는데 여기에는 AWR 보고서와 같은 I/O 통계 대부분이 포함되어 있습니다.

Oracle AWR과 문제 해결

AWR 보고서는 성능 문제 분석을 위한 가장 중요한 툴이기도 합니다.

벤치마킹과 마찬가지로 성능 문제 해결을 위해서는 특정 워크로드를 정확하게 측정해야 합니다. NetApp 지원 센터에 성능 문제를 보고할 때 또는 NetApp이나 파트너 어카운트 팀과 협력하여 새로운 솔루션을 사용할 때는 가능하면 AWR 데이터를 제공하십시오.

AWR 데이터를 제공할 때는 다음 요구사항을 고려하십시오.

- 를 실행합니다 `awrrpt.sql` 명령을 사용하여 보고서를 생성합니다. 출력은 텍스트 또는 HTML 형식으로 가능합니다.
- Oracle RAC(Real Application Cluster)를 사용하는 경우 클러스터의 각 인스턴스에 대해 AWR 보고서를 생성합니다.
- 문제가 존재했던 특정 시간을 타겟으로 합니다. AWR 보고서에서 허용되는 최대 경과 시간은 일반적으로 1시간입니다. 수 시간 동안 문제가 지속되거나 일괄 작업 같이 시간이 많이 걸리는 작업인 경우, 분석할 기간 전체를 포괄하는 1시간 간격 AWR 보고서를 여러 개 제공합니다.
- 가능한 경우 AWR 스냅샷 간격을 15분으로 조정합니다. 이 설정을 통해 더 상세한 분석을 수행할 수 있으며 또한 의 추가 실행도 필요합니다 `awrrpt.sql` 각 15분 간격에 대한 보고서를 제공합니다.
- 매우 짧은 실행 쿼리가 문제인 경우, 작업이 시작될 때 생성된 AWR 스냅샷과 작업이 종료될 때 생성된 두 번째 AWR 스냅샷에 기반을 두고 AWR 보고서를 제공합니다. 그렇지 않으면, 분석 중인 작업의 활동을 종료하지 않게 만드는 백그라운드 활동을 최소화하기 위해 데이터베이스 서버가 정지되어야 합니다.
- 특정 시간에 성능 문제가 보고되었지만 그 외 시간에는 성능 문제가 없는 경우 비교를 위해 우수한 성능을 입증하는 추가 AWR 데이터를 제공합니다.

CALIBRATE_IO 를 선택합니다

를 클릭합니다 `calibrate_io` 명령은 스토리지 시스템의 테스트, 비교 또는 벤치마크에 사용할 수 없습니다. Oracle 설명서에 명시된 것처럼 이 절차는 스토리지의 I/O 기능을 보정합니다.

보정은 벤치마킹과 같지 않습니다. 이 명령의 목적은 데이터베이스 작업 보정을 위해 I/O를 발행하고 호스트에 발행된 I/O 레벨을 최적화하여 효율성을 개선하는 것입니다. 에 의해 수행되는 I/O 유형이기 때문입니다 `calibrate_io` 작동 방식은 실제 데이터베이스 사용자 I/O를 나타내지 않으며, 그 결과는 예측할 수 없으며, 재현이 불가능한 경우도 많습니다.

SLOB2 를 참조하십시오

Oracle 벤치마크인 SLOB2는 데이터베이스 성능을 평가하는 데 있어 선호되는 도구가 되었습니다. Kevin Closson이 개발했으며 에서 다운로드할 수 있습니다 "<https://kevinclosson.net/slob/>". 설치하고 구성하는 데에는 몇 분 정도가 걸리며 실제 Oracle 데이터베이스를 사용하여 사용자가 정의 가능한 테이블스페이스에 I/O 패턴을 생성합니다. 이 툴은 I/O를 통해 All-Flash 어레이를 포화할 수 있는 소수의 테스트 옵션 중 하나이며 IOPS가 낮지만 지연 시간에 민감한 스토리지 워크로드를 시뮬레이션하기 위해 훨씬 낮은 레벨의 I/O를 생성하는 데 유용합니다.

스윙벤치

Swingbench는 데이터베이스 성능을 테스트하는 데 유용할 수 있지만 스토리지에 중점을 두어 Swingbench를 사용하는 것은 매우 어렵습니다. NetApp이 관찰한 결과, AFF 어레이에 상당한 로드가 될 정도의 I/O가 생성된 Swingbench 테스트는 없었습니다. 드문 경우지만 지연 시간 관점에서 스토리지를 평가하는 데 OET(Order Entry Test)를 사용할 수 있습니다. 이는 데이터베이스에 특정 쿼리의 알려진 지연 시간 종속성이 있는 상황에서 유용할 수 있습니다. All-Flash 어레이의 잠재적 지연 시간을 실현하려면 호스트와 네트워크가 적절히 구성되었는지 주의를 기울여 확인해야 합니다.

HammerDB

HammerDB는 TPC-C 및 TPC-H 벤치마크를 시뮬레이션하는 데이터베이스 테스트 툴입니다. 테스트를 제대로 실행하기 위해 대규모 데이터 세트를 구성하는 데에는 상당한 시간이 걸릴 수 있지만 이는 OLTP와 데이터 웨어하우스 애플리케이션의 성능을 평가하는 데 효과적인 툴이 될 수 있습니다.

오리온

Oracle Orion 툴은 일반적으로 Oracle 9과 함께 사용되었지만 다양한 호스트 운영 체제의 변화에 따른 호환성이 보장되도록 유지보수되지 않았습니다. 이 툴은 운영 체제 및 스토리지 구성의 비호환성 때문에 Oracle 10 또는 Oracle 11에는 거의 사용되지 않습니다.

Oracle은 이 툴을 재작성했고 Oracle 12c에 기본으로 설치했습니다. 이 제품은 개선이 되었고 실제 Oracle 데이터베이스에서와 같은 호출을 다수 사용하지만 Oracle에서와 정확히 같은 코드 경로 또는 I/O 동작을 사용하지는 않습니다. 예를 들어, 대부분의 Oracle I/O는 동시에 수행되는데 이는 I/O 작업이 포그라운드에서 완료되는 경우 I/O가 완료될 때까지 데이터베이스가 멈춘다는 뜻입니다. 랜덤 I/O를 통한 스토리지 시스템의 단순한 서비스 장애는 실제 Oracle I/O의 재현이 아니며 스토리지 어레이를 비교하거나 구성 변화의 영향을 측정하는 직접적인 방법을 제공하지 않습니다.

그렇지만 특정 호스트-네트워크-스토리지 구성의 최대 성능에 관한 일반적인 측정이나 스토리지 시스템 상태를 알아내는 등 몇 가지 Orion 사용 사례가 있습니다. 매개 변수에 IOPS 고려사항, 처리량, 지연 시간을 포함하고 실제 워크로드를 충실히 복제하려 하는 한, 신중한 테스트를 통해 사용 가능한 Orion 테스트를 고안하여 스토리지 어레이를 비교하거나 구성 변경의 영향을 평가할 수 있습니다.

부실 NFSv3 잠금 및 Oracle 데이터베이스

Oracle 데이터베이스 서버가 충돌하는 경우 재시작했을 때 부실 NFS 잠금에서 문제가 발생할 수 있습니다. 서버에서 이름 확인을 구성할 때 특별히 주의를 기울이면 이 문제를 피할 수 있습니다.

이 문제가 발생하는 것은 잠금 생성과 잠금 해제가 두 가지의 약간 다른 호스트 이름 확인 방법을 사용하기 때문입니다. 여기에는 NLM(Network Lock Manager)와 NFS 클라이언트라는 2개의 프로세스가 관련되어 있습니다. NLM은 `l` 를 사용합니다 `uname -n` 를 사용하여 호스트 이름을 확인할 수 있습니다 `rpc.statd` 프로세스 사용 `gethostbyname()`. 운영 체제에서 이러한 부실 잠금을 제대로 삭제할 수 있으려면 이 호스트 이름이 일치해야 합니다. 예를 들어, 호스트가 찾고 있는 소유의 잠금이 필요할 수 있습니다 `dbserver5` 그러나 잠금이 호스트에 로 등록되었습니다 ``dbserver5.mydomain.org`. If(경우 `gethostbyname()` 과(와) 같은 값을 반환하지 않습니다 ``uname -a` 그런 다음 잠금 해제 프로세스가 성공하지 못했습니다.

다음 샘플 스크립트는 이름 확인이 완전히 일관되는지 확인합니다.

```
#!/usr/bin/perl
$username=`uname -n`;
chomp($username);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $username;
print "uname -n yields: $username\n";
print "gethostbyname yields: $name\n";
```

If(경우 `gethostbyname` 일치하지 않습니다 ``uname`` 부실 잠금일 가능성이 있습니다. 예를 들어, 이 결과는 다음과 같이 잠재적 문제를 드러냅니다.

```
uname -n yields: dbserver5
gethostbyname yields: dbserver5.mydomain.org
```

일반적으로 에서 호스트가 나타나는 순서를 변경하는 것이 해결책이 됩니다 `/etc/hosts`. 예를 들어, `hosts` 파일에 다음 항목이 포함되어 있다고 가정합니다.

```
10.156.110.201 dbserver5.mydomain.org dbserver5 loghost
```

이 문제를 해결하려면 정규화된 도메인 이름과 짧은 호스트 이름이 나타나는 순서를 변경합니다.

```
10.156.110.201 dbserver5 dbserver5.mydomain.org loghost
```

`gethostbyname()` 이제 쇼트 값을 반환합니다 `dbserver5`의 출력과 일치하는 호스트 이름입니다 `uname`. 이와 같이 서버 충돌이 일어나면 잠금이 자동으로 삭제됩니다.

Oracle 데이터베이스에 대한 WAFL 정렬 검증

올바른 WAFL 정렬은 우수한 성능을 얻는 데 매우 중요합니다. ONTAP이 4KB 유닛의 블록을 관리하기는 하나 ONTAP 4KB 유닛의 모든 작업을 수행하는 것은 아닙니다. 실제로 ONTAP은 다양한 크기의 블록 작업을 지원하지만 기본 계산은 4KB 유닛의 WAFL이 관리합니다.

'정렬'이라는 용어는 Oracle I/O가 이러한 4KB 유닛에 대응하는 방식을 의미합니다. 최적의 성능을 얻으려면 Oracle 8KB 블록이 드라이브의 4KB WAFL 물리적 블록 2개에 상주해야 합니다. 블록이 2KB로 오프셋되면 이 블록은 4KB 블록 1개의 절반에 상주하며, 별도의 4KB 블록 전체 그리고 4KB 블록 1/3의 절반에 상주합니다. 이 방식은 성능 저하를 초래합니다.

NAS 파일 시스템에서는 정렬이 문제가 되지 않습니다. Oracle 데이터 파일은 Oracle 블록 크기에 기반을 둔 파일의 시작에 맞춰 정렬됩니다. 따라서 블록 크기 8KB, 16KB, 32KB가 항상 정렬됩니다. 모든 블록 작업은 4킬로바이트 유닛에서 파일의 시작으로부터 오프셋됩니다.

이와 반대로, LUN의 경우 일반적으로 오프셋을 생성하는 일종의 드라이버 헤더 또는 파일 시스템 메타데이터가 시작 부분에 포함되어 있습니다. 최신 운영 체제는 기본 4KB 섹터를 사용할 수 있는 물리적 드라이브를 위해 설계되었기 때문에 정렬이 문제가 되는 경우는 드물며, 여기서도 최적의 성능을 위해서는 I/O를 4KB 경계에 맞춰 정렬해야 합니다.

그러나 몇 가지 예외가 있습니다. 데이터베이스가 4KB I/O에 최적화되지 않은 기존 운영 체제에서 마이그레이션되었을 수도 있고, 파티션이 생성될 때 사용자 오류에 의해 4KB 크기의 유닛이 아닌 오프셋이 발생했을 수도 있습니다.

다음 예는 Linux에만 해당되며 모든 운영 체제에 맞게 절차를 조정할 수 있습니다.

정렬

다음 예는 단일 파티션의 단일 LUN에서의 정렬 점검을 보여줍니다.

먼저 드라이브에서 사용 가능한 모든 파티션을 사용하는 파티션을 만듭니다.

```

[root@host0 iscsi]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xb97f94c1.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
The device presents a logical sector size that is smaller than
the physical sector size. Aligning to a physical sector (or optimal
I/O) size boundary is recommended, or performance may be impacted.
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-10240, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-10240, default 10240):
Using default value 10240
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
[root@host0 iscsi]#

```

다음 명령을 사용하여 정렬을 수학적으로 점검할 수 있습니다.

```

[root@host0 iscsi]# fdisk -u -l /dev/sdb
Disk /dev/sdb: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 65536 bytes
Disk identifier: 0xb97f94c1

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            32      20971519     10485744    83  Linux

```

출력은 유닛이 512바이트이고 파티션의 시작이 32유닛이라는 것을 보여줍니다. 전체 $32 \times 512 = 16,834$ 바이트이며 이는 모두 4KB WAFL 블록의 배수입니다. 이 파티션은 올바르게 정렬되었습니다.

올바른 정렬을 확인하려면 다음 단계를 완료하십시오.

1. LUN의 UUID(Universally Unique Identifier)를 식별합니다.

```
FAS8040SAP::> lun show -v /vol/jfs_luns/lun0
Vserver Name: jfs
LUN UUID: ed95d953-1560-4f74-9006-85b352f58fcd
Mapped: mapped`
```

2. ONTAP 컨트롤러의 노드 셸에 들어갑니다.

```
FAS8040SAP::> node run -node FAS8040SAP-02
Type 'exit' or 'Ctrl-D' to return to the CLI
FAS8040SAP-02> set advanced
set not found. Type '?' for a list of commands
FAS8040SAP-02> priv set advanced
Warning: These advanced commands are potentially dangerous; use
them only when directed to do so by NetApp
personnel.
```

3. 첫 번째 단계에서 식별된 타겟 UUID의 통계적 수집을 시작합니다.

```
FAS8040SAP-02*> stats start lun:ed95d953-1560-4f74-9006-85b352f58fcd
Stats identifier name is 'Ind0xffffffff08b9536188'
FAS8040SAP-02*>
```

4. 몇 가지 I/O를 수행합니다 를 사용하는 것이 중요합니다 iflag 동기식이고 버퍼링되지 않도록 하는 인수입니다.



이 명령을 사용할 때는 매우 주의해야 합니다. 의 후진 if 및 of 인수가 데이터를 삭제합니다.

```
[root@host0 iscsi]# dd if=/dev/sdb1 of=/dev/null iflag=dsync count=1000
bs=4096
1000+0 records in
1000+0 records out
4096000 bytes (4.1 MB) copied, 0.0186706 s, 219 MB/s
```

5. 통계를 정지하고 정렬 히스토그램을 확인합니다. 모든 I/O는 에 있어야 합니다 .0 버킷: I/O가 4KB 블록 경계에 맞춰 정렬되었음을 나타냅니다.

```

FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffffff08b9536188
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:186%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%

```

잘못 정렬됨

다음 예는 잘못 정렬된 I/O를 보여줍니다.

1. 4KB 경계에 맞춰 정렬되지 않은 파티션을 생성합니다. 최신 운영 체제에서 이는 기본 동작이 아닙니다.

```

[root@host0 iscsi]# fdisk -u /dev/sdb
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First sector (32-20971519, default 32): 33
Last sector, +sectors or +size{K,M,G} (33-20971519, default 20971519):
Using default value 20971519
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.

```

2. 파티션이 기본값 32섹터 대신 33섹터 오프셋으로 생성되었습니다. 예 설명된 절차를 반복합니다 "정렬". 히스토그램은 다음과 같이 나타납니다.


```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.0:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.1:136%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.3:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.4:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.5:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.6:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_align_histo.7:0%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:read_partial_blocks:31%
```

명확히 정렬 불량입니다. I/O는 대부분 * 에 속합니다 *.1 버킷 - 예상 오프셋과 일치합니다. 파티션이 생성되었을 때 최적화된 기본값보다 장치로 512바이트 더 멀리 이동했으며 이는 히스토그램이 512바이트로 오프셋된다는 뜻입니다.

또한 read_partial_blocks 통계는 0이 아닙니다. 즉, I/O가 수행되었지만 4KB 블록 전체를 채우지는 않았습니니다.

로깅 재실행

여기에 설명된 절차는 데이터 파일에 적용할 수 있습니다. Oracle 재실행 로그와 아카이브 로그는 I/O 패턴이 다릅니다. 예를 들어, 로깅 재실행은 단일 파일의 순환 덮어쓰기입니다. 기본값인 512바이트 블록 크기가 사용되는 경우 쓰기 통계는 다음과 같습니다.

```
FAS8040SAP-02*> stats stop
StatisticsID: Ind0xffffffff0468242e78
lun:ed95d953-1560-4f74-9006-85b352f58fcd:instance_uuid:ed95d953-1560-4f74-9006-85b352f58fcd
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.0:12%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.1:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.2:4%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.3:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.4:13%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.5:6%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.6:8%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_align_histo.7:10%
lun:ed95d953-1560-4f74-9006-85b352f58fcd:write_partial_blocks:85%
```

I/O가 모든 히스토그램 버킷 전체에 분산되나 이것으로 인해 성능이 저하되지는 않습니다. 하지만 로깅 재실행 속도가 매우 높다면 4KB 블록 크기를 사용하여 이점을 얻을 수 있습니다. 이 경우에는 로깅 재실행 LUN이 제대로 정렬되었는지 확인하는 것이 좋습니다. 그러나 데이터 파일 정렬에서처럼 우수한 성능을 얻는 데 있어 이 작업이 반드시 필요한 것은 아닙니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이선스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이선스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이선스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이선스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 <http://www.netapp.com/TM>에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.