



호스트 구성

Enterprise applications

NetApp
May 09, 2024

목차

호스트 구성	1
IBM AIX를 사용하는 Oracle 데이터베이스	1
HP-UX를 사용하는 Oracle 데이터베이스	2
Oracle 데이터베이스 및 Linux	4
ASMLib/AFD를 사용하는 Oracle 데이터베이스(ASM 필터 드라이버)	8
Microsoft Windows를 사용하는 Oracle 데이터베이스	10
Solaris를 사용한 Oracle 데이터베이스	10

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 I/O를 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	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144
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	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,suid</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,forcedirectio, nointr,suid</code>
ORACLE_HOME	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,suid</code>

다음 표에는 RAC에 대한 HP-UX 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,noac,suid</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,suid</code>
CRS/투표	<code>rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144, nointr,noac, forcedirectio,suid</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 `noac` 및 `forcedirectio` 마운트 옵션으로 이동합니다. 이렇게 추가되면 호스트 운영 체제 캐싱이 비활성화되어 RAC 클러스터의 모든 인스턴스에서 데이터 상태에 관한 일관된 뷰를 얻게 됩니다. `noac`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 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 사용해 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 를 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.lf`(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 를 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 를 사용해야 합니다 actimeo=0 마운트 옵션을 사용하면 운영 체제 캐시에서 파일 데이터가 즉시 만료됩니다 그러면 이 옵션은 Linux NFS 클라이언트가 특정 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 orafstab 파일

다중 경로 옵션으로 Linux에서 DNFS를 사용할 때는 여러 서브넷을 사용해야 합니다. 그 외 운영 체제에서는 를 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 설치되나 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 페일오버가 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 않았었습니다. 여기서 반환되는 값은 4KB 블록 크기가 낮다는 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 않았었습니다. 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 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.

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 툴로 활용할 수 없게 됩니다. `RMAN`이 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 에 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	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,noac
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
CRS/투표	rw,bg,hard,[vers=3,vers=4.1],proto=tcp,timeo=600,rsize=262144,wsiz=262144,nointr,noac,forcedirectio
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 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_mvvector_max_size`.

다음 명령을 루트로 실행합니다.

```
[root@host1 ~]# echo "zfs_mvvector_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_mvvector_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` 12가 아닌 경우 데이터를 새로 생성된 `zpool`으로 마이그레이션해야 합니다.

zpool을 생성한 후의 값을 확인합니다. ashift 계속 진행하기 전에 값이 12가 아니라면 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.

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.