



## 추가 참고 사항

### Enterprise applications

NetApp  
February 10, 2026

# 목차

추가 참고 사항 .....	1
성능 최적화 및 벤치마킹 .....	1
Oracle Automatic Workload Repository 및 벤치마킹 .....	1
Oracle AWR과 문제 해결 .....	2
CALIBRATE_IO 를 선택합니다 .....	2
SLOB2 를 참조하십시오 .....	3
스윙벤치 .....	3
HammerDB .....	3
오리온 .....	3
부실 NFSv3 잠금 .....	3
WAFL 정렬 확인 .....	4
정렬 .....	5
잘못 정렬됨 .....	7
로깅 재실행 .....	8

# 추가 참고 사항

## 성능 최적화 및 벤치마킹

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 $\mu$ 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 명료하지 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 명료하지 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

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 `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
$uname=`uname -n`;
chomp($uname);
($name, $aliases, $addrtype, $length, @addrs) = gethostbyname $uname;
print "uname -n yields: $uname\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.

## WAFL 정렬 확인

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  $32 \times 512 = 16,384$ 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 않았습니다.

## 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 © 2026 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.