



기술 보고서

How to enable StorageGRID in your environment

NetApp
April 26, 2024

목차

기술 보고서	1
NetApp StorageGRID 및 빅데이터 분석	1
Hadoop S3A 튜닝	4

기술 보고서

NetApp StorageGRID 및 빅데이터 분석

NetApp StorageGRID 사용 사례

NetApp StorageGRID 오브젝트 스토리지 솔루션은 확장성, 데이터 가용성, 보안 및 고성능을 제공합니다. 모든 규모와 다양한 산업 분야의 조직이 광범위한 사용 사례에 StorageGRID S3를 사용합니다. 몇 가지 일반적인 시나리오를 살펴보겠습니다.

- 빅 데이터 분석: * StorageGRID S3는 기업에서 Apache Spark, Splunk Smartstore 및 Dremio와 같은 도구를 사용하여 분석을 위해 대량의 정형 및 비정형 데이터를 저장하는 데이터 레이크로 자주 사용됩니다.
- 데이터 계층화: * NetApp 고객은 ONTAP의 FabricPool 기능을 사용하여 고성능 로컬 계층 간에 데이터를 자동으로 StorageGRID로 이동합니다. 계층화하면 콜드 데이터를 저렴한 오브젝트 스토리지에서 즉시 사용 가능한 상태로 유지하면서 고가의 플래시 스토리지를 핫 데이터용으로 확보할 수 있습니다. 따라서 성능과 비용 절감 효과가 극대화됩니다.
- 데이터 백업 및 재해 복구: * 기업에서는 StorageGRID S3를 안정적이고 비용 효율적인 솔루션으로 사용하여 중요한 데이터를 백업하고 재해 발생 시 복구할 수 있습니다.
- 응용 프로그램용 데이터 저장: * StorageGRID S3는 응용 프로그램용 스토리지 백엔드로 사용할 수 있으므로 개발자가 파일, 이미지, 비디오 및 기타 유형의 데이터를 쉽게 저장하고 검색할 수 있습니다.
- 콘텐츠 전송: * StorageGRID S3를 사용하여 정적 웹 사이트 콘텐츠, 미디어 파일 및 소프트웨어 다운로드를 전 세계 사용자에게 저장하고 제공할 수 있으며, StorageGRID의 지리적 분산 및 글로벌 네임스페이스를 활용하여 빠르고 안정적인 콘텐츠 전송을 수행할 수 있습니다.
- 데이터 계층화: * NetApp 고객은 ONTAP FabricPool 기능을 사용하여 고성능 로컬 계층 간에 데이터를 자동으로 StorageGRID로 이동합니다. 계층화하면 콜드 데이터를 저렴한 오브젝트 스토리지에서 즉시 사용 가능한 상태로 유지하면서 고가의 플래시 스토리지를 핫 데이터용으로 확보할 수 있습니다. 따라서 성능과 비용 절감 효과가 극대화됩니다.
- 데이터 아카이브: * StorageGRID는 다양한 스토리지 유형을 제공하고 공용 장기 저비용 스토리지 옵션에 대한 계층화를 지원하며, 규정 준수 또는 기간별 목적으로 보존해야 하는 데이터의 보관 및 장기 보존에 이상적인 솔루션입니다.
- 오브젝트 스토리지 사용 사례 *

[StorageGRID 사용 사례 다이어그램, 폭 = 396, 높이 = 394]

위 중 빅 데이터 분석은 최상위 사용 사례 중 하나이며 사용량이 증가하고 있습니다.

데이터 레이크에 **StorageGRID**를 사용해야 하는 이유

- 협업 증가 - 업계 표준 API 액세스를 지원하는 대규모 공유 멀티 사이트 멀티 테넌시
- 운영 비용 절감 - 자동 복구, 자동화된 단일 스케일아웃 아키텍처를 통해 운영 간소화
- 확장성 - 기존 Hadoop 및 데이터 웨어하우스 솔루션과 달리 StorageGRID S3 오브젝트 스토리지는 스토리지를 컴퓨팅과 데이터와 분리하므로 기업이 필요에 따라 스토리지 요구사항을 확장할 수 있습니다.
- 내구성 및 안정성 - StorageGRID는 99.9999999%의 내구성을 제공하여 저장된 데이터가 데이터 손실에 대한 저항성이 높습니다. 또한 고가용성을 제공하여 데이터에 항상 액세스할 수 있도록 보장합니다.
- 보안 - StorageGRID는 암호화, 액세스 제어 정책, 데이터 라이프사이클 관리, 오브젝트 잠금 및 버전 관리와 같은

다양한 보안 기능을 제공하여 S3 버킷에 저장된 데이터를 보호합니다

- StorageGRID S3 데이터 레이크 *

[StorageGRID 데이터 레이크 예, width=614, height=345]

S3 오브젝트 스토리지에서 가장 잘 작동하는 데이터 웨어하우스 또는 데이터 레이크가 있습니다

NetApp는 Hive, Delta Lake 및 Dremio 등 3개의 데이터 웨어하우스/호수 하우스 에코시스템을 통해 StorageGRID의 벤치마크를 수행했습니다. ["아파치 아이스버그: 확실한 가이드"](#) 데이터 웨어하우스 및 데이터 레이크 하우스에 대한 간략한 소개와 이 두 아키텍처의 장점 포함

- 벤치마크 도구 - TPC-DS - <https://www.tpc.org/tpcds/>
- 빅 데이터 에코시스템
 - VM 5개로 구성된 클러스터, 각각 128G RAM 및 24개의 vCPU, 시스템 디스크용 SSD 스토리지
 - Hive 3.1.3 포함 Hadoop 3.3.5(이름 노드 1개 + 데이터 노드 4개)
 - Spark 3.0.0(마스터 1명 + 작업자 4명) 및 Hadoop 3.3.5 지원 델타 레이크
 - Dremio v23(마스터 1개 + 실행자 4개)
- 오브젝트 스토리지
 - NetApp ® StorageGRID ® 11.6(SG6060 + SG1000 로드 밸런서 3개 포함)
 - 오브젝트 보호 - 복사본 2개
- 데이터베이스 크기 1000GB
- 각 쿼리 테스트에 대해 일관된 결과를 얻기 위해 3개 에코시스템에서 캐시를 모두 사용할 수 없습니다.

TPC-DS에는 쿼리 벤치마킹을 위한 99개의 복잡한 SQL 쿼리가 포함되어 있습니다. 전체 시간(분)을 측정하여 99개의 쿼리를 모두 완료했으며 결과를 분석하기 위한 S3 요청의 유형과 수를 세분화하여 더 깊이 있게 조사했습니다. 아래의 첫 번째 표는 99개의 모든 쿼리의 총 시간을 보여 주며 두 번째 표는 각 에코시스템에서 StorageGRID로 전송된 S3 요청의 수와 유형을 요약한 것입니다.

- TPC-DS 쿼리 결과 *

에코시스템	하이프	델타 레이크	드리미오
지원합니다	NetApp (영어) StorageGRID (영어)	NetApp (영어) StorageGRID (영어)	NetApp (영어) StorageGRID (영어)
드라이브 유형입니다	HDD	HDD	HDD
표 형식	마루	마루	마루 1
데이터베이스 크기	1000g	1000g	1000g
TPCDS 99 쿼리 총 시간(분)	1084 - 2도	55	47

파케(Parquet)와 아이스버그(Iceberg) 테이블 형식을 모두 테스트한 결과도 이와 유사합니다.

1. 조회 번호 72를 완료할 수 없습니다.
 - TPC-DS 쿼리 - S3 요청 분석 *

S3 요청	하이브	델타 레이크	드리미오
가져오기	1,117,184	2,074,610	4,414,227
관찰: 모든 제품군 GET	80% 범위 32MB 객체에서 2KB ~ 2MB, 초당 50 ~ 100회의 요청	73% 범위는 32MB 개체에서 100KB 미만으로, 초당 1000 - 1400개의 요청 수입니다	90% 1M 바이트 범위는 256MB 객체, 초당 2000- 2300개 요청에서 가져옵니다
개체 나열	312,053입니다	24,158입니다	240
머리 (존재하지 않는 객체)	156,027	12,103	192
머리 (존재하는 객체)	982,126	922,732	1,845
총 요청 수입니다	2,567,390입니다	3,033,603입니다	4,416,504입니다

첫 번째 테이블에서, 우리는 델타 호수와 Dremio가 Hive보다 훨씬 더 빠르다는 것을 볼 수 있습니다. 두 번째 표에서 Hive는 많은 S3 목록 오브젝트 요청을 전송했습니다. 이 요청은 모든 오브젝트 스토리지 플랫폼에서 일반적으로 느리며, 특히 많은 오브젝트가 포함된 버킷을 다룰 경우 매우 느립니다. 따라서 전체 쿼리 시간이 크게 증가합니다. 또 다른 관찰은 Dremio가 Hive에서 초당 50-100개의 요청을 처리하는 데 비해 초당 2,000-2,300개의 요청을 동시에 보낼 수 있다는 것입니다. Hive 및 Hadoop S3A는 표준 파일 시스템을 모방하여 S3 오브젝트 스토리지에 Hive 느림 효과를 제공합니다.

Hive 또는 Spark와 함께 Hadoop(HDFS 또는 S3 오브젝트 스토리지)을 사용하려면 Hadoop 및 Hive/Spark와 각 서비스의 설정이 상호 작용하는 방법에 대한 폭넓은 지식이 필요합니다. 이러한 두 서비스의 설정이 1,000개 이상인 경우입니다. 설정은 서로 관련이 있는 경우가 매우 많으며 단독으로 변경할 수 없습니다. 사용할 설정과 값의 최적 조합을 찾기 위해서는 엄청난 시간과 노력이 필요합니다.

Dremio는 완벽한 Apache Arrow를 사용하여 쿼리 성능을 획기적으로 향상하는 데이터 레이크 엔진입니다. Apache Arrow는 효율적인 데이터 공유와 빠른 분석을 위해 표준화된 원주 메모리 형식을 제공합니다. Arrow는 데이터 serialization 및 deserialization의 필요성을 제거하여 복잡한 데이터 프로세스와 시스템 간의 성능 및 상호 운용성을 향상시키도록 설계된 언어 독립적 접근 방식을 사용합니다.

Dremio의 성능은 주로 Dremio 클러스터의 컴퓨팅 성능에 의해 좌우됩니다. Dremio는 S3 오브젝트 스토리지 연결에 Hadoop의 S3A 커넥터를 사용하지만 Hadoop은 필요하지 않으며 대부분의 Hadoop의 fs.s3a 설정은 Dremio에서 사용되지 않습니다. 따라서 다양한 Hadoop s3a 설정을 배우고 테스트하는 데 시간을 들이지 않고도 Dremio 성능을 손쉽게 튜닝할 수 있습니다.

이러한 벤치마크 결과에서 알 수 있듯이 S3 기반 워크로드에 최적화된 빅데이터 분석 시스템이 주요 성능 요인이라는 결론을 내릴 수 있습니다. Dremio는 쿼리 실행을 최적화하고, 메타데이터를 효율적으로 사용하며, S3 데이터에 대한 원활한 액세스를 제공하므로 S3 스토리지로 작업할 때 Hive에 비해 성능이 향상됩니다. 이를 참조하십시오 ["페이지" StorageGRID를 사용하여 Dremio S3 데이터 소스를 구성합니다.](#)

아래 링크를 방문하여 StorageGRID와 Dremio가 함께 작동하여 현대적이고 효율적인 데이터 레이크 인프라를 제공하는 방법과 NetApp가 Hive+ HDFS에서 Dremio+ StorageGRID로 마이그레이션하여 빅데이터 분석 효율성을 획기적으로 개선한 방법에 대해 자세히 알아보십시오.

- ["NetApp StorageGRID로 빅데이터의 성능을 향상하십시오"](#)
- ["StorageGRID 및 Dremio를 사용하는 현대적이고 강력하고 효율적인 데이터 레이크 인프라"](#)
- ["NetApp이 제품 분석을 통해 고객 경험을 재정의하는 방법"](#)

Hadoop S3A 튜닝

Hadoop S3A 커넥터는 Hadoop 기반 애플리케이션과 S3 오브젝트 스토리지 간의 원활한 상호 작용을 지원합니다. S3 오브젝트 스토리지로 작업할 때 성능을 최적화하려면 Hadoop S3A Connector를 튜닝해야 합니다. 세부 조정을 시작하기에 앞서, Hadoop과 그 구성 요소에 대한 기본적인 이해를 갖겠습니다.

Hadoop이란?

- Hadoop * 은 대규모 데이터 처리 및 스토리지를 처리하도록 설계된 강력한 오픈 소스 프레임워크입니다. 이를 통해 컴퓨터 클러스터 간에 분산 스토리지 및 병렬 처리가 가능합니다.

Hadoop의 3가지 핵심 구성 요소는 다음과 같습니다.

- * Hadoop HDFS (Hadoop 분산 파일 시스템) * : 스토리지를 처리하고 데이터를 블록으로 분할하여 노드에 분산시킵니다.
- * Hadoop MapReduce *: 작업을 작은 청크로 분할하고 병렬로 실행하여 데이터를 처리합니다.
- * Hadoop YARN (또 다른 리소스 협상 담당자) : * **"리소스를 관리하고 작업을 효율적으로 예약합니다"**

Hadoop HDFS 및 S3A 커넥터

HDFS는 Hadoop 에코시스템의 핵심 구성 요소로, 효율적인 빅 데이터 처리에 중요한 역할을 합니다. HDFS는 안정적인 스토리지 및 관리를 지원합니다. 병렬 처리 및 최적화된 데이터 스토리지를 보장하여 데이터 액세스 및 분석 속도가 빨라집니다.

빅데이터 처리 시 HDFS는 대규모 데이터 세트를 위한 내결함성 스토리지를 제공하는 데 탁월합니다. 이 점은 데이터 복제를 통해 실현됩니다. 데이터 웨어하우스 환경에서 대량의 정형 데이터와 비정형 데이터를 저장하고 관리할 수 있습니다. 또한 Apache Spark, Hive, Pig 및 Flink와 같은 선도적인 빅 데이터 처리 프레임워크와 원활하게 통합되어 확장 가능하고 효율적인 데이터 처리가 가능합니다. Unix 기반(Linux) 운영 체제와 호환되기 때문에 빅 데이터 처리를 위해 Linux 기반 환경을 사용하는 것을 선호하는 기업에 이상적인 선택입니다.

시간이 지나면서 데이터 볼륨이 증가함에 따라 자체 컴퓨팅 및 스토리지를 사용하여 Hadoop 클러스터에 새 시스템을 추가하는 방식이 비효율적이 되었습니다. 선형적으로 확장하면 리소스를 효율적으로 사용하고 인프라를 관리하는 데 어려움이 발생합니다.

이러한 과제를 해결하기 위해 Hadoop S3A 커넥터는 S3 오브젝트 스토리지에 대한 고성능 I/O를 제공합니다. S3A를 사용하여 Hadoop 워크플로우를 구축하면 오브젝트 스토리지를 데이터 저장소로 활용할 수 있으며, 컴퓨팅과 스토리지를 독립적으로 확장할 수 있는 분리된 컴퓨팅 및 스토리지를 사용할 수 있습니다. 또한 컴퓨팅과 스토리지를 분리하면 컴퓨팅 작업에 적절한 양의 리소스를 투입하고 데이터 세트의 크기에 따라 용량을 제공할 수 있습니다. 따라서 Hadoop 워크플로우의 전체 TCO를 줄일 수 있습니다.

Hadoop S3A 커넥터 튜닝

S3는 HDFS와 다르게 동작하며 파일 시스템의 모양을 유지하려는 일부 시도는 공격적으로 최적화되지 않습니다. S3 리소스를 가장 효율적으로 활용하려면 신중한 튜닝/테스트/실험이 필요합니다.

이 문서의 Hadoop 옵션은 Hadoop 3.3.5를 기반으로 합니다. 을 참조하십시오 ["Hadoop 3.3.5 core-site.xml"](#) 사용 가능한 모든 옵션

참고 – 일부 Hadoop fs.s3a 설정의 기본값은 Hadoop 버전마다 다릅니다. 현재 Hadoop 버전과 관련된 기본값을 확인하십시오. 이러한 설정이 Hadoop core-site.xml에 지정되지 않은 경우 기본값이 사용됩니다. Spark 또는 Hive 구성 옵션을 사용하여 런타임에 값을 재정의할 수 있습니다.

이 페이지로 이동해야 합니다 "[아파치 하둡 페이지](#)" 각 fs.s3a 옵션을 이해합니다. 가능한 경우 비운영 Hadoop 클러스터에서 테스트하여 최적의 값을 찾습니다.

읽어야 합니다 "[S3A 커넥터로 작업할 때 성능을 극대화합니다](#)" 기타 튜닝 권장 사항

몇 가지 주요 고려 사항을 살펴보겠습니다.

- 1. 데이터 압축 *

StorageGRID 압축을 활성화하지 마십시오. 대부분의 빅 데이터 시스템은 전체 객체를 검색하는 대신 바이트 범위 GET를 사용합니다. 압축된 객체와 함께 GET 바이트 범위를 사용하면 GET 성능이 크게 저하됩니다.

- 2. S3A 커밋 *

일반적으로 magic s3a committer가 권장됩니다. 이를 참조하십시오 "[일반 S3A 커밋 옵션 페이지](#)" 마법 커밋 및 관련 s3a 설정에 대한 더 나은 이해를 얻기 위해.

매직 커미터:

Magic Committer는 특히 S3Guard에 의존하여 S3 오브젝트 저장소에서 일관된 디렉토리 목록을 제공합니다.

이제 일관된 S3(이 경우)를 통해 Magic Committer를 모든 S3 버킷과 함께 안전하게 사용할 수 있습니다.

선택 및 실험:

사용 사례에 따라 클러스터 HDFS 파일 시스템에 의존하는 스테이징 커밋자와 Magic committer 중에서 선택할 수 있습니다.

두 가지를 모두 실험하여 귀사의 워크로드 및 요구사항에 가장 적합한 솔루션을 결정하십시오.

요약하면, S3A committers는 S3에 대한 일관적이고, 고성능의 안정적인 출력 약속이라는 근본적인 과제에 대한 솔루션을 제공합니다. 내부 설계로 데이터 무결성을 유지하면서 효율적인 데이터 전송을 보장합니다.

[S3A 옵션 표]

- 3. 스레드, 연결 풀 크기 및 블록 크기 *
- 단일 버킷과 상호 작용하는 각 * S3A * 클라이언트에는 업로드 및 복사 작업을 위한 개방형 HTTP 1.1 연결 전용 풀과 스레드가 있습니다.
- "[이러한 풀 크기를 조정하여 성능과 메모리/스레드 사용량 간의 균형을 맞출 수 있습니다](#)".
- S3에 데이터를 업로드하면 블록으로 나뉩니다. 기본 블록 크기는 32MB입니다. fs.s3a.block.size 속성을 설정하여 이 값을 사용자 지정할 수 있습니다.
- 블록 크기가 클수록 업로드 시 다중 파트 관리 오버헤드를 줄여 대용량 데이터 업로드의 성능을 향상시킬 수 있습니다. 대용량 데이터 세트의 권장 값은 256MB 이상입니다.

[S3A 옵션 표]

- 4. 멀티 파트 업로드 *

s3a committers * 항상 * MPU(멀티 파트 업로드)를 사용하여 데이터를 S3 버킷에 업로드합니다. 작업 실패, 추측에 의한 작업 실행, 커밋 전 작업 중단 등을 허용하기 위해 필요합니다. 다음은 다중 파트 업로드와 관련된 몇 가지 주요 사양입니다.

- 최대 개체 크기: 5TiB(테라바이트)
- 업로드당 최대 부품 수: 10,000.
- 부품 번호: 1 ~ 10,000 범위(포함).
- 부품 크기: 5MiB에서 5GiB 사이. 멀티 파트 업로드의 마지막 부분에 대한 최소 크기 제한은 없습니다.

S3 멀티 파트 업로드에 더 작은 파트 크기를 사용하면 장점과 단점이 모두 있습니다.

- 장점 *:
 - 네트워크 문제에서 빠른 복구: 작은 부품을 업로드하면 네트워크 오류로 인해 실패한 업로드를 다시 시작할 때의 영향이 최소화됩니다. 부품에 오류가 발생하면 전체 오브젝트가 아닌 특정 부분만 다시 업로드하면 됩니다.
 - 향상된 병렬 처리: 다중 스레딩 또는 동시 연결을 활용하여 더 많은 파트를 병렬로 업로드할 수 있습니다. 이 병렬화는 특히 큰 파일을 처리할 때 성능을 향상시킵니다.
- 단점 *:
 - 네트워크 오버헤드: 파트 크기가 작을수록 업로드할 파트가 더 많아지며 각 파트마다 자체 HTTP 요청이 필요합니다. HTTP 요청이 많을수록 개별 요청을 시작 및 완료하는 데 따르는 오버헤드가 증가합니다. 많은 수의 작은 파트를 관리하면 성능에 영향을 줄 수 있습니다.
 - 복잡성: 주문 관리, 부품 추적, 성공적인 업로드 보장은 번거로울 수 있습니다. 업로드를 중단해야 하는 경우 이미 업로드한 모든 부품을 추적하고 제거해야 합니다.

Hadoop의 경우 `fs.s3a.multipart.size`에 256MB 이상의 파트 크기가 권장됩니다. 항상 `fs.s3a.multipart.threshold` 값을 $2 \times \text{fs.s3a.multipart.size}$ 값으로 설정하십시오. 예를 들어 `fs.s3a.multipart.size=256M`, `fs.s3a.multipart.threshold`는 512M이어야 합니다.

대형 데이터 세트에 더 큰 파트 크기를 사용합니다. 특정 사용 사례와 네트워크 상태에 따라 이러한 요소의 균형을 맞추는 부품 크기를 선택하는 것이 중요합니다.

다중 부분 업로드는 입니다 "[3단계 프로세스](#)":

1. 업로드가 시작되면 StorageGRID에서 업로드 ID를 반환합니다.
2. 개체 부분은 `upload-id`를 사용하여 업로드됩니다.
3. 모든 객체 부분이 업로드되면 는 업로드 ID와 함께 완전한 멀티 파트 업로드 요청을 보냅니다. StorageGRID는 업로드된 부분에서 객체를 생성하며 클라이언트는 객체에 액세스할 수 있습니다.

전체 다중 파트 업로드 요청이 성공적으로 전송되지 않으면 부품은 StorageGRID에 남아 있고 객체를 생성하지 않습니다. 이 문제는 작업이 중단, 실패 또는 중단될 때 발생합니다. 업로드가 시작된 후 15일이 경과하면 멀티 파트 업로드가 완료되거나 중단되거나 StorageGRID가 이러한 부품을 제거할 때까지 파트가 그리드에 남아 있습니다. 버킷에 여러 개의(수억 ~ 수백만) 진행 중인 멀티 파트 업로드가 있는 경우 Hadoop이 'list-multipart-uploads'를 전송할 때(이 요청은 업로드 ID로 필터링되지 않음) 요청을 완료하는 데 시간이 오래 걸리거나 시간이 초과될 수 있습니다. 적절한 `fs.s3a.multipart.purge`를 `true`로 설정하여 적절한 `fs.s3a.multipart.purge.age` 값을 설정할 수 있습니다(예: 5-7일, 기본값 86400, 즉 1일을 사용하지 마십시오). 또는 NetApp 지원 팀에 문의하여 상황을 조사하십시오.

[S3A 옵션 표]

- 5. 메모리의 버퍼 쓰기 데이터 *

성능을 높이기 위해 쓰기 데이터를 S3에 업로드하기 전에 메모리에 버퍼링할 수 있습니다. 이렇게 하면 작은 쓰기 수를 줄이고 효율성을 높일 수 있습니다.

[S3A 옵션 표]

S3 및 HDFS는 서로 다른 방식으로 작동한다는 점을 기억하십시오. S3 리소스를 가장 효율적으로 활용하려면 신중한 튜닝/테스트/실험이 필요합니다.

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.