



FlexCache 볼륨 관리

ONTAP 9

NetApp
February 12, 2026

목차

FlexCache 볼륨 관리	1
ONTAP FlexCache 볼륨에 대해 알아보십시오	1
비디오	1
ONTAP FlexCache 볼륨에 대해 지원되거나 지원되지 않는 기능	2
FlexCache 볼륨과 원래 볼륨 간 ONTAP 버전 지원	3
지원되는 프로토콜	3
지원되는 기능	3
ONTAP FlexCache 볼륨 사이징 지침을 참조하십시오	8
ONTAP FlexCache 볼륨을 생성합니다	8
FlexCache 다시 쓰기	14
ONTAP FlexCache Write-Back에 대해 자세히 알아보십시오	14
ONTAP FlexCache write-back 지침	15
ONTAP FlexCache 후기입 아키텍처	16
ONTAP FlexCache 후기입 사용 사례	20
ONTAP FlexCache write-back 사전 요구 사항	22
ONTAP FlexCache write-back 상호 운용성	23
ONTAP FlexCache write-back 설정 및 관리	24
ONTAP FlexCache 후기입(Write-Back)에 대한 질문과 대답(FAQ)	28
FlexCache 이중성	29
FlexCache 이중성에 대한 FAQ	29
NAS FlexCache 볼륨에 대한 S3 액세스를 활성화합니다	30
FlexCache 볼륨 관리	36
ONTAP FlexCache 볼륨 감사에 대해 자세히 알아보십시오	36
원본 볼륨의 ONTAP FlexCache 볼륨 속성을 동기화합니다	37
ONTAP FlexCache 관계 구성을 업데이트합니다	38
ONTAP FlexCache 볼륨에서 파일 액세스 시간 업데이트를 활성화합니다	38
ONTAP FlexCache 볼륨에 대한 글로벌 파일 잠금을 사용합니다	39
ONTAP FlexCache 볼륨을 미리 채웁니다	41
ONTAP FlexCache 관계 삭제	42
핫스팟 교정을 위한 FlexCache	43
ONTAP FlexCache 볼륨으로 고성능 컴퓨팅 워크로드의 핫 스팟 해결	43
ONTAP FlexCache 핫스팟 개선 솔루션 설계	44
ONTAP FlexCache 밀도 결정	47
ONTAP SVM 간 또는 SVM 내 HDFA 옵션을 결정합니다	49
HDFA 및 ONTAP 데이터 LIF를 구성합니다	51
ONTAP NAS 연결을 분산하도록 클라이언트를 구성합니다	54

FlexCache 볼륨 관리

ONTAP FlexCache 볼륨에 대해 알아보십시오

NetApp FlexCache 기술은 데이터 액세스를 가속하고 WAN 대기 시간을 줄이며, 클라이언트가 동일한 데이터에 반복적으로 액세스해야 하는 읽기 집약적인 워크로드의 WAN 대역폭 비용을 줄여줍니다. FlexCache 볼륨을 생성할 때 원래 볼륨의 활성 데이터(핫 데이터)만 포함된 기존(오리진) 볼륨의 원격 캐시를 생성합니다.

FlexCache 볼륨이 포함된 핫 데이터의 읽기 요청을 받으면 데이터가 클라이언트에 도달하기 위해 멀리 이동할 필요가 없기 때문에 원본 볼륨보다 빠르게 응답할 수 있습니다. FlexCache 볼륨이 자주 읽지 않는 데이터(콜드 데이터)에 대한 읽기 요청을 수신하는 경우, 원래 볼륨에서 필요한 데이터를 검색한 다음 클라이언트 요청을 처리하기 전에 데이터를 저장합니다. 그런 다음 해당 데이터에 대한 후속 읽기 요청이 FlexCache 볼륨에서 직접 처리됩니다. 첫 번째 요청을 받은 이후에는 더 이상 네트워크를 통해 데이터를 이동하거나 과부하 상태인 시스템에서 데이터를 처리할 필요가 없습니다. 예를 들어, 자주 요청하는 데이터에 대한 단일 액세스 포인트에서 클러스터 내에 병목 현상이 발생한다고 가정합니다. 클러스터 내에서 FlexCache 볼륨을 사용하여 핫 데이터에 여러 마운트 지점을 제공할 수 있으므로 병목 현상이 줄어들고 성능이 향상됩니다. 다른 예로, 여러 클러스터에서 액세스하는 볼륨으로 네트워크 트래픽을 줄여야 한다고 가정합니다. FlexCache 볼륨을 사용하여 원본 볼륨의 핫 데이터를 네트워크 내의 클러스터를 통해 배포할 수 있습니다. 따라서 사용자에게 보다 가까운 액세스 지점을 제공하여 WAN 트래픽이 감소합니다.

또한 FlexCache 기술을 사용하여 클라우드 및 하이브리드 클라우드 환경의 성능을 향상할 수 있습니다. FlexCache 볼륨은 온프레미스 데이터 센터의 데이터를 클라우드로 캐싱하여 워크로드를 하이브리드 클라우드로 전환하는 데 도움이 될 수 있습니다. 또한 FlexCache 볼륨을 사용하여 한 클라우드 공급자에서 다른 클라우드 공급자로 또는 동일한 클라우드 공급자의 두 지역 간에 데이터를 캐싱하여 클라우드 사일로를 제거할 수 있습니다.

ONTAP 9.10.1부터 가능합니다 "[글로벌 파일 잠금 사용](#)" 모든 FlexCache 볼륨에 걸쳐 있습니다. 전역 파일 잠금은 사용자가 이미 다른 사용자가 연 파일에 액세스하지 못하도록 합니다. 그런 다음, 원래 볼륨의 업데이트가 모든 FlexCache 볼륨에 동시에 배포됩니다.

ONTAP 9.9.1부터 FlexCache 볼륨은 찾을 수 없는 파일 목록을 유지합니다. 이렇게 하면 클라이언트가 존재하지 않는 파일을 검색할 때 오리진으로 여러 통화를 전송할 필요가 없도록 하여 네트워크 트래픽을 줄일 수 있습니다.

추가 목록 "[FlexCache 볼륨 및 원래 볼륨에 대해 지원되는 기능](#)" ONTAP 버전에서 지원되는 프로토콜 목록을 포함하여 사용할 수도 있습니다.

ONTAP FlexCache 기술의 아키텍처에 대한 자세한 내용은 에서 확인할 수 있습니다 "[TR-4743: ONTAP의 FlexCache](#)".

비디오

FlexCache를 통해 글로벌 데이터의 WAN 지연 시간 및 읽기 시간을 줄이는 방법

ONTAP FlexCache

Data Access Where You Need It

Use Case

© 2020 NetApp, Inc. All rights reserved.

 NetApp



ONTAP FlexCache의 성능 이점에 대해 알아보십시오!

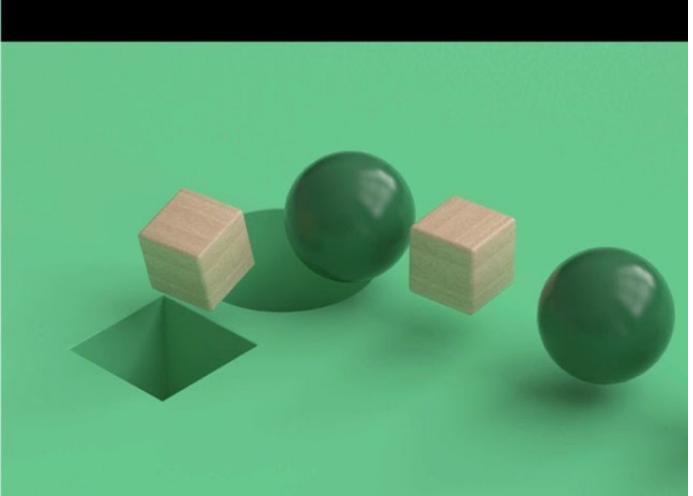
ONTAP FlexCache

Data Access Where You Need It

Tech Clip

© 2020 NetApp, Inc. All rights reserved.

 NetApp



ONTAP FlexCache 볼륨에 대해 지원되거나 지원되지 않는 기능

ONTAP 9.5부터 FlexCache 볼륨을 구성할 수 있습니다. FlexVol 볼륨은 원본 볼륨으로 지원되며 FlexGroup 볼륨은 FlexCache 볼륨으로 지원됩니다. ONTAP 9.7부터 FlexVol 볼륨과

FlexGroup 볼륨이 모두 원본 볼륨으로 지원됩니다. 원래 볼륨과 FlexCache 볼륨에 대해 지원되는 기능 및 프로토콜은 다양합니다.



캐시 볼륨과 원본 볼륨이 모두 지원되는 ONTAP 버전에서 실행되는 한 상호 운용할 수 있습니다. 기능은 지원 기능이 도입된 ONTAP 버전 또는 이후 ONTAP 버전을 실행하는 캐시와 오리진 둘 다 실행 중인 경우에만 지원됩니다.

FlexCache 볼륨과 원래 볼륨 간 ONTAP 버전 지원

오리진 볼륨과 캐시 볼륨 간에 지원되는 권장 ONTAP 버전은 이전 버전 4개 또는 이후 버전 4개를 넘지 않습니다. 예를 들어 캐시가 ONTAP 9.14.1을 사용하는 경우 오리진에서 사용할 수 있는 가장 초기 버전은 ONTAP 9.10.1입니다.

지원되는 프로토콜

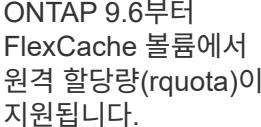
프로토콜	원본 볼륨에서 지원됩니까?	FlexCache 볼륨에서 지원됩니까?
NFSv3	예	예
NFSv4	예 NFSv4.x 프로토콜을 사용하여 캐시 볼륨을 액세스하려면 오리진 클러스터와 캐시 클러스터 모두 ONTAP 9.10.1 이상을 사용해야 합니다. 오리진 클러스터와 FlexCache 클러스터는 서로 다른 ONTAP 버전을 가질 수 있지만 둘 다 ONTAP 9.10.1 이상 버전이어야 합니다. 예를 들어 오리진에는 ONTAP 9.10.1이 있고 캐시에는 ONTAP 9.11.1이 포함될 수 있습니다.	ONTAP 9.10.1부터 지원됩니다. NFSv4.x 프로토콜을 사용하여 캐시 볼륨을 액세스하려면 오리진 클러스터와 캐시 클러스터 모두 ONTAP 9.10.1 이상을 사용해야 합니다. 오리진 클러스터와 FlexCache 클러스터는 서로 다른 ONTAP 버전을 가질 수 있지만 둘 다 ONTAP 9.10.1 이상 버전이어야 합니다. 예를 들어 오리진에는 ONTAP 9.10.1이 있고 캐시에는 ONTAP 9.11.1이 포함될 수 있습니다.
NFSv4.2	예	아니요
중소기업	예	예 ONTAP 9.8부터 지원됩니다.

지원되는 기능

피처	원본 볼륨에서 지원됩니까?	FlexCache 볼륨에서 지원됩니까?
----	----------------	-----------------------

자율적 랜섬웨어 방어	예 ONTAP 9.10.1부터 FlexVol 원본 볼륨에 대해 지원되며 ONTAP 9.13.1부터 FlexGroup 원본 볼륨에 대해 지원됩니다. 을 참조하십시오 "자율 랜섬웨어 보호 사용 사례 및 고려사항" .	아니요
안티바이러스	예 ONTAP 9.7부터 지원됩니다.	해당 없음 오리진에서 바이러스 백신 검사를 구성하는 경우 캐시에 필요하지 않습니다. 원본 바이러스 백신 검사는 쓰기 소스에 관계없이 쓰기가 커밋되기 전에 바이러스에 감염된 파일을 감지합니다. FlexCache에서 바이러스 백신 검사를 사용하는 방법에 대한 자세한 내용은 를 참조하십시오 "FlexCache with ONTAP 기술 보고서" .
감사	예 ONTAP 9.7부터 지원됩니다. 기본 ONTAP 감사를 사용하여 FlexCache 관계에서 NFS 파일 액세스 이벤트를 감사할 수 있습니다. 자세한 내용은 을 참조하십시오 FlexCache 볼륨을 감사할 때의 고려 사항	예 ONTAP 9.7부터 지원됩니다. 기본 ONTAP 감사를 사용하여 FlexCache 관계에서 NFS 파일 액세스 이벤트를 감사할 수 있습니다. 자세한 내용은 을 참조하십시오 FlexCache 볼륨을 감사할 때의 고려 사항
Cloud Volumes ONTAP	예 ONTAP 9.6부터 지원됩니다	예 ONTAP 9.6부터 지원됩니다
컴팩션	예 ONTAP 9.6부터 지원됩니다	예 ONTAP 9.7부터 지원됩니다
압축	예 ONTAP 9.6부터 지원됩니다	예 ONTAP 9.6부터 지원됩니다
중복 제거	예	예 인라인 중복제거는 ONTAP 9.6부터 FlexCache 볼륨에서 지원됩니다. ONTAP 9.7부터 시작되는 FlexCache 볼륨에서는 다중 볼륨 중복제거가 지원됩니다.

FabricPool	예	예 ONTAP 9.7부터 지원됩니다	 FabricPool 계층화가 활성화된 원본 볼륨에 대한 캐시로 FlexCache 볼륨을 생성할 수 있지만 FlexCache 볼륨 자체는 계층화할 수 없습니다.
FlexCache 재해 복구	예	예 ONTAP 9.9.1부터 지원됨, NFSv3 프로토콜 포함, 전용. FlexCache 볼륨은 개별 SVM 또는 개별 클러스터에 있어야 합니다.	
FlexGroup 볼륨	예 ONTAP 9.7부터 지원됩니다	예	
FlexVol 볼륨	예	아니요	
FPolicy를 참조하십시오	예 ONTAP 9.7부터 지원됩니다	예 ONTAP 9.7부터 NFS에 대해 지원됩니다. ONTAP 9.14.1부터 SMB에 대해 지원됩니다.	
MetroCluster 구성	예 ONTAP 9.7부터 지원됩니다	예 ONTAP 9.7부터 지원됩니다	
Microsoft ODX(Offloaded Data Transfer)	예	아니요	
NetApp 애그리게이트 암호화(NAE)	예 ONTAP 9.6부터 지원됩니다	예 ONTAP 9.6부터 지원됩니다	
NetApp 볼륨 암호화(NVE)	예 ONTAP 9.6부터 지원됩니다	예 ONTAP 9.6부터 지원됩니다	

ONTAP S3 NAS 버킷	예 ONTAP 9.12.1부터 지원됩니다.	예 ONTAP 9.18.1부터 지원됩니다.
QoS를 참조하십시오	예	예  FlexCache 볼륨에 대해 파일 레벨 QoS가 지원되지 않습니다.
Qtree	예 ONTAP 9.6부터 qtree를 작성하고 수정할 수 있습니다. 소스에서 생성된 qtree는 캐시에서 액세스할 수 있습니다.	아니요
할당량	예 ONTAP 9.6부터 FlexCache 원본 볼륨에 대한 할당량 적용은 사용자, 그룹, qtree가 지원됩니다.	아니요  FlexCache 쓰기 사운드 모드(기본 모드)를 사용하면 캐시의 쓰기가 원본 볼륨으로 전달됩니다. 할당량은 오리진에서 적용됩니다.  ONTAP 9.6부터 FlexCache 볼륨에서 원격 할당량(rquota)이 지원됩니다.
SMB 변경 통지	예	예 ONTAP 9.14.1부터 SMB 변경 알림이 캐시에서 지원됩니다.
SnapLock 볼륨	아니요	아니요
SnapMirror 비동기식 관계 *	예	아니요

	<ul style="list-style-type: none"> FlexCache 원점: 원본 FlexVol의 FlexCache 볼륨을 가질 수 있습니다 원본 FlexGroup의 FlexCache 볼륨을 가질 수 있습니다 SnapMirror 관계의 원본 기본 볼륨에서 FlexCache 볼륨을 가질 수 있습니다. ONTAP 9.8부터 SnapMirror 보조 볼륨은 FlexCache 원본 볼륨이 될 수 있습니다. SnapMirror 2차 볼륨은 활성 SnapMirror 업데이트 없이 유지 상태여야 하며, 그렇지 않으면 FlexCache 생성이 실패합니다. 	SnapMirror 동기식 관계
아니요	아니요	SnapRestore
예	아니요	스냅샷 수
예	아니요	SVM DR 구성
예	<p>ONTAP 9.5 버전부터 지원됩니다. SVM DR 관계의 기본 SVM은 원본 볼륨을 사용할 수 있습니다. 그러나 ONTAP ONTAP 이전 버전을 실행 중인 경우 SVM DR 관계가 끊어지면 새 원본 볼륨을 사용하여 FlexCache 관계를 다시 생성해야 합니다.</p> <p>ONTAP 9.18.1 버전부터 원본 SVM이 장애 조치를 수행하면 캐시가 자동으로 재해 복구 사이트의 원본으로 전환됩니다. 수동 복구 단계가 필요 없어집니다.</p> <p>FlexCache 볼륨 생성 방법에 대해 알아보세요.</p>	스토리지 레벨 액세스 가드(슬래그)
아니요	아니요	씬 프로비저닝
예	<p>예</p> <p>ONTAP 9.7부터 지원됩니다</p>	볼륨 클론 복제

예	아니요	볼륨 이동
예	예(볼륨 구성요소에만 해당) FlexCache 볼륨의 볼륨 구성 요소 이동은 ONTAP 9.6 이상에서 지원됩니다.	볼륨 재호스팅
아니요	아니요	어레이 통합을 위한 vStorage API(VAAI)



FlexVol 9 릴리즈 9.5 이전 버전에서는 원본 FlexCache 볼륨이 7-Mode에서 작동하는 Data ONTAP 8.2.x를 실행하는 시스템에서 생성된 ONTAP 볼륨에만 데이터를 제공할 수 있습니다. FlexVol 9.5부터 오리진 ONTAP 볼륨은 ONTAP 9 시스템의 FlexCache 볼륨에 데이터를 제공할 수도 있습니다. 7-Mode FlexCache에서 ONTAP 9 FlexCache로 마이그레이션하는 방법에 대한 자세한 내용은 ["NetApp 기술 보고서 4743: FlexCache in ONTAP"](#)를 참조하십시오.

ONTAP FlexCache 볼륨 사이징 지침을 참조하십시오

볼륨 프로비저닝을 시작하기 전에 FlexCache 볼륨의 제한을 알고 있어야 합니다.

FlexVol 볼륨의 크기 제한은 원본 볼륨에 적용됩니다. FlexCache 볼륨의 크기는 원본 볼륨보다 작거나 같을 수 있습니다. FlexCache 볼륨 크기에 대한 모범 사례는 원본 볼륨 크기의 10% 이상이어야 합니다.

또한 FlexCache 볼륨에 대한 다음과 같은 추가 제한도 알고 있어야 합니다.

제한	ONTAP 9.8 이상	ONTAP 9.7	ONTAP 9.6 - 9.5
원본 볼륨에서 생성할 수 있는 최대 FlexCache 볼륨 수입니다	100	10	10
노드당 권장되는 최대 오리진 볼륨 수입니다	100	100	10
노드당 권장되는 최대 FlexCache 볼륨 수입니다	100	100	10
노드당 FlexCache 볼륨에 권장되는 최대 FlexGroup 구성 수	800	800	40
노드당 FlexCache 볼륨당 최대 구성요소 수	32	32	32

관련 정보

- ["NetApp 상호 운용성"](#)

ONTAP FlexCache 볼륨을 생성합니다

자주 사용되는 객체에 액세스할 때 성능을 향상시키기 위해 동일한 ONTAP 클러스터 내에 FlexCache 볼륨을 생성할 수 있습니다. 데이터 센터가 여러 위치에 있는 경우 원격 ONTAP

클러스터에 FlexCache 볼륨을 생성하여 데이터 액세스 속도를 높일 수 있습니다.

이 작업에 대해

- ONTAP 9.18.1부터 볼륨 생성 시 `-is-s3-enabled` 옵션을 `true`로 설정하여 FlexCache 볼륨에서 NAS S3 버킷 액세스를 활성화할 수 있습니다. 이 옵션은 기본적으로 비활성화되어 있습니다.
- ONTAP 9.18.1부터 FlexCache SVM-DR 관계에 속하는 SVM이 있는 원본 볼륨에 대한 캐시 볼륨을 생성하는 것을 지원합니다.

ONTAP 9.18.1 이상을 실행하는 경우 스토리지 관리자는 SVM-DR 관계에 속한 원본 볼륨의 캐시 볼륨을 생성하기 전에 SVM-DR 관계에 속한 기본 및 보조 원본 SVM 모두와 캐시 SVM을 피어링해야 합니다.

- ONTAP 9.14.0부터 암호화된 소스에서 암호화되지 않은 FlexCache 볼륨을 만들 수 있습니다.
- ONTAP 9.7부터 FlexVol 볼륨과 FlexGroup 볼륨 모두 원본 볼륨으로 지원됩니다.
- ONTAP 9.5부터 FlexCache는 FlexVol 볼륨을 원본 볼륨으로, FlexGroup 볼륨을 FlexCache 볼륨으로 지원합니다.

시작하기 전에

- ONTAP 9.5 이상을 실행 중이어야 합니다.
- ONTAP 9.6 이하를 실행 중인 경우를 실행해야 ["FlexCache 라이센스를 추가합니다"](#)합니다.

ONTAP 9.7 이상에는 FlexCache 라이센스가 필요하지 않습니다. ONTAP 9.7부터는 FlexCache 기능이 ONTAP에 포함되어 더 이상 라이센스나 활성화가 필요하지 않습니다.



HA 쌍이 사용 중인 경우 ["SAS 또는 NVMe 드라이브\(SED, NSE, FIPS\) 암호화"](#), 항목의 지침을 따라야 합니다. ["FIPS 드라이브 또는 SED를 보호되지 않는 모드로 되돌리는 중입니다"](#) 시스템을 초기화하기 전에 HA 쌍 내의 모든 드라이브(부팅 옵션 4 또는 9) 이렇게 하지 않을 경우 드라이브를 용도 변경할 경우 향후의 데이터 손실이 발생할 수 있습니다.

예 1. 단계

시스템 관리자

1. FlexCache 볼륨이 원본 볼륨과 다른ONTAP 클러스터에 있는 경우 클러스터 피어 관계를 생성합니다.
 - a. 로컬 클러스터에서 * 보호 > 개요 * 를 클릭합니다.
 - b. Intercluster Settings * 를 확장하고 * Add Network Interfaces * 를 클릭한 후 클러스터에 대한 클러스터 간 네트워크 인터페이스를 추가합니다.

원격 클러스터에서 이 단계를 반복합니다.

 - c. 원격 클러스터에서 * 보호 > 개요 * 를 클릭합니다. Cluster Peers 섹션을 클릭하고 * Generate Passphrase * 를 클릭합니다.
 - d. 생성된 암호를 복사하여 로컬 클러스터에 붙여넣습니다.
 - e. 로컬 클러스터의 클러스터 피어에서 * 피어 클러스터 * 를 클릭하고 로컬 및 원격 클러스터를 피어로 지정합니다.
2. SVM 피어 관계 생성:
스토리지 VM 피어 아래에서  및 * 피어 스토리지 VM * 을 클릭하여 스토리지 VM을 피어링합니다.

3. 스토리지 > 볼륨 * 를 선택합니다.
4. 추가 * 를 선택합니다.
5. More Options * 를 선택한 다음 * Add as cache for a remote volume * 를 선택합니다.



ONTAP 9.8 이상을 실행 중인 경우 QoS를 비활성화하거나 사용자 지정 QoS 정책을 선택하려면 * 추가 옵션 * 를 클릭한 다음 * 스토리지 및 최적화 * 에서 * 성능 서비스 수준 * 을 선택합니다.

CLI를 참조하십시오

1. 생성할 FlexCache 볼륨이 다른 클러스터에 있는 경우 클러스터 피어 관계를 생성합니다.
 - a. 대상 클러스터에서 데이터 보호 소스 클러스터와 피어 관계를 생성합니다.

```
cluster peer create -generate-passphrase -offer-expiration
MM/DD/YYYY HH:MM:SS|1...7days|1...168hours -peer-addrs
<peer_LIF_IPs> -initial-allowed-vserver-peers <svm_name>,..|*
-ipspace <ipspace_name>
```

ONTAP 9.6부터는 클러스터 피어 관계를 만들 때 TLS 암호화가 기본적으로 활성화됩니다. TLS 암호화는 오리진과 FlexCache 볼륨 간의 인터클러스터 통신에 지원됩니다. 필요한 경우 클러스터 피어 관계에 대해 TLS 암호화를 비활성화할 수도 있습니다.

```

cluster02::> cluster peer create -generate-passphrase -offer
-expiration 2days -initial-allowed-vserver-peers *

        Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
        Expiration Time: 6/7/2017 08:16:10 EST
        Initial Allowed Vserver Peers: *
        Intercluster LIF IP: 192.140.112.101
        Peer Cluster Name: Clus_7ShR (temporary generated)

Warning: make a note of the passphrase - it cannot be displayed
again.

```

- a. 소스 클러스터에서 소스 클러스터를 대상 클러스터에 인증합니다.

```
cluster peer create -peer-addrs <peer_LIF_IPs> -ipspace <ipspace>
```

```
cluster01::> cluster peer create -peer-addrs
192.140.112.101,192.140.112.102
```

Notice: Use a generated passphrase or choose a passphrase of 8 or more characters.

To ensure the authenticity of the peering relationship, use a phrase or sequence of characters that would be hard to guess.

Enter the passphrase:

Confirm the passphrase:

Clusters cluster02 and cluster01 are peered.

2. FlexCache 볼륨이 원본 볼륨과 다른 SVM에 있는 경우 애플리케이션으로서 "FlexCache"와 SVM 피어 관계를 생성합니다.

- a. SVM이 다른 클러스터에 있는 경우 피어링 SVM에 대한 SVM 권한을 생성합니다.

```
vserver peer permission create -peer-cluster <cluster_name>
-vserver <svm-name> -applications flexcache
```

다음 예에서는 모든 로컬 SVM에 적용되는 SVM 피어 권한을 생성하는 방법을 보여줍니다.

```
cluster1::> vserver peer permission create -peer-cluster cluster2  
-vserver "*" -applications flexcache  
  
Warning: This Vserver peer permission applies to all local Vservers.  
After that no explicit  
"vserver peer accept" command required for Vserver peer relationship  
creation request  
from peer cluster "cluster2" with any of the local Vservers. Do you  
want to continue? {y|n}: y
```

a. SVM 피어 관계 생성:

```
vserver peer create -vserver <local_SVM> -peer-vserver  
<remote_SVM> -peer-cluster <cluster_name> -applications flexcache
```

3. FlexCache 볼륨 생성:

```
volume flexcache create -vserver <cache_svm> -volume  
<cache_vol_name> -auto-provision-as flexgroup -size <vol_size>  
-origin-vserver <origin_svm> -origin-volume <origin_vol_name> -is-s3  
-enabled true|false
```

다음 예에서는 FlexCache 볼륨을 생성하고 프로비저닝할 기준 애큰리게이트를 자동으로 선택합니다.

```
cluster1::> volume flexcache create -vserver vs_1 -volume fc1 -auto  
-provision-as flexgroup -origin-volume vol_1 -size 160MB -origin  
-vserver vs_1  
[Job 443] Job succeeded: Successful
```

다음 예에서는 FlexCache 볼륨을 생성하고 접합 경로를 설정합니다.

```
cluster1::> volume flexcache create -vserver vs34 -volume fc4 -aggr  
-list aggr34,aggr43 -origin-volume origin1 -size 400m -junction-path  
/fc4  
[Job 903] Job succeeded: Successful
```

다음 예제는 FlexCache 볼륨에서 S3 액세스를 활성화합니다.

```
cluster1::> volume flexcache create -vserver vs3 -volume  
cache_vs3_vol33 -origin-volume vol33 -origin-vserver vs3 -junction  
-path /cache_vs3_vol33 -is-s3-enabled true
```

4. FlexCache 볼륨과 원본 볼륨에서 FlexCache 관계를 확인합니다.

a. 클러스터에서 FlexCache 관계 보기:

```
volume flexcache show
```

```
cluster1::> volume flexcache show  
Vserver Volume      Size      Origin-Vserver Origin-Volume  
Origin-Cluster  
-----  
-----  
vs_1    fc1          160MB     vs_1        vol_1  
cluster1
```

b. 오리진 클러스터의 모든 FlexCache 관계 보기: + 'volume FlexCache origin show-caches'

```
cluster::> volume flexcache origin show-caches  
Origin-Vserver Origin-Volume   Cache-Vserver   Cache-Volume  
Cache-Cluster  
-----  
-----  
vs0          ovoll1         vs1           cfg1  
clusA  
vs0          ovoll1         vs2           cfg2  
clusB  
vs_1         vol_1          vs_1          fc1  
cluster1
```

결과

FlexCache 볼륨이 성공적으로 생성되었습니다. 클라이언트는 FlexCache 볼륨의 연결 경로를 사용하여 볼륨을 마운트할 수 있습니다.

관련 정보

["클러스터 및 SVM 피어링"](#)

FlexCache 다시 쓰기

ONTAP FlexCache Write-Back에 대해 자세히 알아보십시오

ONTAP 9.15.1에 도입된 FlexCache Write-Back은 캐시에서 쓰는 대체 작업 모드입니다. Write-back을 사용하면 데이터가 오리진으로 전달될 때까지 기다리지 않고 캐시의 안정적인 스토리지에 쓰기를 커밋하고 클라이언트에서 이를 확인할 수 있습니다. 데이터는 비동기적으로 오리진으로 다시 플러시됩니다. 그 결과, 특정 워크로드 및 환경에서 거의 로컬에 가까운 속도로 쓰기를 수행할 수 있는 전 세계적으로 분산된 파일 시스템이 탄생하여 탁월한 성능 이점을 제공합니다.



ONTAP 9.12.1에서는 공개 미리 보기로 쓰기 저장 기능을 도입했습니다. 이 기능을 다시 쓰기 버전 1(wbv1)이라고 하며, ONTAP 9.15.1에서 다시 쓰기 버전 2(wbv2)라고 하는 Write-back과 동일하다고 간주해서는 안 됩니다.

Write-back 대 write-around

FlexCache는 ONTAP 9.5에 도입된 이후 읽기 쓰기 가능한 캐시였지만 write-around 모드로 작동했습니다. 캐시의 쓰기가 안정적인 스토리지로 커밋되도록 오리진으로 전송되었습니다. 오리진에서 쓰기 작업을 안정적인 스토리지에 커밋한 후 캐시에 대한 쓰기를 확인했습니다. 그러면 캐시는 클라이언트에 대한 쓰기를 승인합니다. 이로 인해 모든 쓰기는 캐시와 오리진 사이에서 네트워크를 통과해야 하는 페널티가 발생합니다. FlexCache WRITE-Back(후기입)



ONTAP 9.15.1로 업그레이드한 후 기존의 write-around 캐시를 write-back 캐시로 변환하고 필요한 경우 다시 write-around로 변환할 수 있습니다. 그러나 이로 인해 문제가 발생할 경우 진단 로그를 읽기가 더 어려워질 수 있습니다.

	쓰기	다시 쓰기
ONTAP 버전	9.6 이상	9.15.1 이상
사용 사례	읽기 집약적 워크로드	쓰기 집약적 워크로드
데이터가 에 커밋되었습니다	원점	캐시
고객 경험	WAN과 유사합니다	LAN과 유사합니다
제한	오리진당 100개	오리진당 10개
"CAP 정리"	파티션에 사용할 수 있으며 허용 가능합니다	사용 가능하고 일관적입니다

FlexCache Write-Back 용어

FlexCache Write-Back과 관련된 주요 개념 및 용어 이해

기간	정의
* 더티 데이터 *	캐시의 안정적인 스토리지에 커밋되었지만 오리진으로 플러시되지 않은 데이터
* 독점 잠금 위임(XLD) *	캐시에 파일 단위로 부여된 프로토콜 수준 잠금 권한입니다. 이 권한을 통해 캐시는 오리진에 접속하지 않고 클라이언트에 전용 쓰기 잠금을 제공할 수 있습니다.

기간	정의
* 공유 잠금 위임(SLD) *	캐시에 파일 단위로 부여된 프로토콜 수준 잠금 권한입니다. 이 권한을 통해 캐시는 오리진에 접속하지 않고 공유 읽기 잠금을 클라이언트에 배포할 수 있습니다.
* 다시 쓰기 *	캐시에 대한 쓰기가 해당 캐시의 안정적인 스토리지에 커밋되어 클라이언트에 즉시 인식되는 FlexCache 작업 모드입니다. 데이터는 원본에 비동기적으로 다시 기록됩니다.
* 기입 *	캐시에 대한 쓰기가 오리진으로 전달되어 안정적인 스토리지에 커밋되는 FlexCache 작업 모드입니다. 커밋되면 오리진에서 캐시에 대한 쓰기를 승인하고 캐시가 클라이언트에 대한 쓰기를 확인합니다.
* DDR(Dirty Data Record System) *	파일별로 다시 쓰기 가능 캐시에서 더티 데이터를 추적하는 독점 메커니즘입니다.
* 원점 *	모든 FlexCache 캐시 볼륨의 소스 데이터를 포함하는 FlexGroup 또는 FlexVol 단일 데이터 소스로서 잠금을 오픈스트레이션하고 100% 데이터 일관성, 통화 및 일관성을 보장합니다.
* 캐시 *	FlexCache 오리진의 스파스 캐시 볼륨인 FlexGroup입니다.

정합성 보장, 전류 및 일관성

FlexCache는 언제 어디서나 원하는 데이터를 사용할 수 있는 NetApp의 솔루션입니다. FlexCache는 100% 일관된, 전류 및 일관적인 시간입니다.

- * 정합성: * 데이터는 액세스 위치에 관계없이 동일합니다.
- * 현재: * 데이터는 항상 최신 상태입니다.
- * Coherent : * 데이터가 올바르거나 손상되지 않았습니다.

ONTAP FlexCache write-back 지침

FlexCache 쓰기 저장에는 원본과 캐시 간의 많은 복잡한 상호작용이 포함됩니다. 최적의 성능을 얻으려면 환경이 다음 지침을 따라야 합니다. 이 가이드라인은 콘텐츠 생성 당시 사용 가능한 최신 주요 ONTAP 버전(ONTAP 9.17.1)을 기반으로 합니다.

비운영 환경에서 운영 워크로드를 테스트하는 것이 가장 좋습니다. 이 지침을 벗어나는 FlexCache Write-Back을 구현할 경우 더욱 중요합니다.

다음 지침은 NetApp에서 내부적으로 잘 테스트되었습니다. 이 호텔 안에 머무시는 것이 좋습니다. 그렇지 않으면 예기치 않은 동작이 발생할 수 있습니다.

- ONTAP 9.17.1P1에서는 FlexCache 쓰기 저장 기능에 대한 중요한 개선 사항이 도입되었습니다. 9.17.1P1 이후에는 원본 클러스터와 캐시 클러스터 모두에서 현재 권장되는 릴리스를 실행하는 것이 강력히 권장됩니다. 9.17.1 코드라인을 실행할 수 없는 경우, 다음으로 제안되는 릴리스는 최신 P 릴리스인 9.16.1입니다. ONTAP 9.15.1에는 FlexCache 쓰기 저장에 필요한 모든 수정 사항과 개선 사항이 포함되어 있지 않으므로 프로덕션 워크로드에는 권장되지 않습니다.
- 현재 반복에서 FlexCache write-back 캐시는 전체 FlexCache 볼륨에 대해 단일 구성요소로 구성해야 합니다. 다중 구성 FlexCache는 캐시에서 원치 않는 데이터 제거를 초래할 수 있습니다.
- 100GB 미만의 파일에 대해 테스트를 실행했으며, 캐시와 원본 간의 WAN 왕복 시간이 200ms를 초과하지 않도록 했습니다. 이러한 한도를 벗어나는 작업 부하로 인해 예상치 못한 성능 특성이 나타날 수 있습니다.
- SMB 대체 데이터 스트림에 쓰면 주 파일이 캐시에서 제거됩니다. 기본 파일의 모든 더티 데이터를 오리진으로

플러시해야 해당 파일에 대해 다른 작업을 수행할 수 있습니다. 대체 데이터 스트림도 오리진으로 전달됩니다.

- 파일 이름을 바꾸면 파일이 캐시에서 제거됩니다. 파일의 모든 더티 데이터를 오리진으로 플러시해야 해당 파일에 대해 다른 작업을 수행할 수 있습니다.
- 이때 write-back 지원 FlexCache 볼륨의 파일에 대해 변경하거나 설정할 수 있는 유일한 속성은 다음과 같습니다.
 - 타임스탬프
 - 모드 비트
 - NT ACL을 선택합니다
 - 소유자
 - 그룹
 - 크기

변경되거나 설정된 다른 특성은 오리진으로 전달되어 캐시에서 파일을 제거할 수 있습니다. 캐시에서 다른 속성을 변경하거나 설정해야 할 경우, 어카운트 팀에 PVR을 개설하도록 요청하십시오.

- 오리진에서 생성된 스냅샷으로 인해 해당 오리진 볼륨과 연결된 모든 write-back 지원 캐시에서 미처리 더티 데이터를 모두 불러옵니다. 이렇게 하려면 많은 양의 쓰기 저장 작업이 진행 중인 경우 이러한 더티 파일을 제거하는 데 시간이 걸릴 수 있으므로 여러 번 작업을 다시 시도해야 할 수 있습니다.
- 쓰기를 위한 SMB 기회 잠금(Oplock)은 쓰기 백이 활성화된 FlexCache 볼륨에서 지원되지 않습니다.
- 원점은 80% 미만으로 남아 있어야 합니다. 오리진 볼륨에 20% 이상의 공간이 남아 있지 않으면 캐시 볼륨에 전용 잠금 위임이 부여되지 않습니다. 이 경우 다시 쓰기 가능 캐시에 대한 호출은 오리진으로 전달됩니다. 이렇게 하면 오리진에서 공간이 부족해지는 것을 방지할 수 있으며, 이로 인해 다시 쓰기 가능 캐시에 더티 데이터가 분리된 상태로 남게 됩니다.
- 낮은 대역폭 및/또는 손실이 많은 클러스터 간 네트워크는 FlexCache 쓰기 성능에 상당한 부정적인 영향을 미칠 수 있습니다. 작업 부하에 따라 크게 달라지므로 특정 대역폭 요구 사항은 없지만 캐시와 원본 간의 클러스터 간 링크 상태를 확인하는 것이 강력히 권장됩니다.

ONTAP FlexCache 후기입 아키텍처

FlexCache는 쓰기 작업 모드인 Write-back 및 Write-around를 포함하여 강력한 일관성을 염두에 두고 설계되었습니다. 기존의 Write-Around 작동 모드와 ONTAP 9.15.1에 도입된 새로운 Write-back 작동 모드 모두 액세스된 데이터가 항상 100% 일관적이고, 최신 상태이며, 정합성이 보장됩니다.

다음 개념은 FlexCache Write-Back 작동 방식을 자세히 설명합니다.

위임

잠금 위임 및 데이터 위임을 통해 FlexCache는 Write-back 및 Write-around 캐시를 모두 유지할 수 있으며, 데이터의 정합성이 보장되고 최신 상태를 유지할 수 있습니다. 오리진은 두 대표단을 조율합니다.

잠금 위임

잠금 위임은 프로토콜 수준 잠금 권한입니다. 오리진에서 필요에 따라 클라이언트에 프로토콜 잠금을 발급하기 위해 파일별로 캐시에 권한을 부여합니다. 여기에는 [배타적 잠금 위임\(XLD\)](#) 및 [SLD\(공유 잠금 위임\)](#)가 포함됩니다.

XLD 및 write-back

ONTAP가 총돌하는 쓰기를 조정할 필요가 없도록 하기 위해 클라이언트가 파일에 쓰기를 요청하는 캐시에 XLD가 부여됩니다. 중요한 것은 모든 파일에 대해 XLD가 한 번에 하나만 존재할 수 있다는 것입니다. 즉, 한 파일에 대해 한 번에 여러 writer가 있을 수 없습니다.

파일 쓰기 요청이 다시 쓰기 가능 캐시에 들어오면 다음 단계가 수행됩니다.

1. 캐시는 요청된 파일에 대한 XLD가 이미 있는지 확인합니다. 이 경우 다른 클라이언트가 캐시의 파일에 쓰지 않는 한 클라이언트에 쓰기 잠금을 부여합니다. 요청된 파일에 대한 XLD가 캐시에 없는 경우 오리진에서 해당 파일을 요청합니다. 이것은 클러스터 간 네트워크를 통과하는 독점 호출입니다.
2. 캐쉬로부터 XLD 요청을 수신하면 오리진에서 다른 캐쉬에 해당 파일에 대해 미처리 XLD가 있는지 확인합니다. 이 경우 해당 파일의 XLD를 호출하면 해당 캐시에서 오리진으로 플러시를 트리거합니다. [더티 데이터](#)
3. 해당 캐시의 더티 데이터가 다시 플러시되고 오리진의 안정적인 스토리지로 커밋되면 오리진에서 해당 파일에 대한 XLD를 요청 캐시에 부여합니다.
4. 파일의 XLD가 수신되면 캐시는 클라이언트에 잠금을 부여하고 쓰기가 시작됩니다.

이러한 단계 중 일부를 다루는 상위 수준 시퀀스 다이어그램은 시퀀스 [\[write-back-sequence-diagram\]](#) 다이어그램에서 다룹니다.

클라이언트 관점에서 볼 때 모든 잠금은 쓰기 잠금이 요청될 때 약간의 지연이 발생할 수 있는 표준 FlexVol 또는 FlexGroup에 쓰는 것처럼 작동합니다.

현재 반복에서 쓰기 저장 가능 캐시에 파일의 XLD가 있는 경우 ONTAP는 작업을 포함하여 다른 캐시에 있는 해당 파일에 대한 * 모든 * 액세스를 차단합니다. READ



원점 구성요소당 170개의 XLD 제한이 있습니다.

데이터 위임

데이터 위임은 해당 파일에 대해 캐시된 데이터가 최신 상태임을 오리진에서 캐시에 제공하는 파일별 보증입니다. 캐시에 파일에 대한 데이터 위임이 있는 한, 오리진에 연결할 필요 없이 해당 파일에 대한 캐시된 데이터를 클라이언트에 제공할 수 있습니다. 캐시에 파일에 대한 데이터 위임이 없는 경우 클라이언트에서 요청한 데이터를 받으려면 원본에 연결해야 합니다.

다시 쓰기 모드에서는 다른 캐시나 오리진에서 해당 파일에 대해 XLD를 사용하는 경우 파일의 데이터 위임이 취소됩니다. 이렇게 하면 다른 모든 캐시 및 오리진에 있는 클라이언트에서 파일을 효과적으로 차단하고 읽기 작업을 수행할 수도 있습니다. 이는 오래된 데이터에 절대 액세스하지 못하도록 하기 위해 반드시 해결해야 하는 절충입니다.

Write-back이 설정된 캐시에서 읽기는 일반적으로 Write-around 캐시의 읽기처럼 작동합니다. Write-Around 및 Write-back이 설정된 캐시 모두에서 요청된 파일이 읽기가 실행되는 위치와 다른 Write-back이 설정된 캐시에 단독 쓰기 잠금이 있는 경우 초기 성능 저하가 발생할 수 READ 있습니다. XLD를 해지해야 하며, 다른 캐시의 읽기를 서비스하기 전에 더티 데이터를 오리진으로 커밋해야 합니다.

더티 데이터 추적

캐시에서 오리진으로 다시 쓰기는 비동기적으로 수행됩니다. 즉, 더티 데이터는 오리진에 즉시 다시 기록되지 않습니다. ONTAP는 파일당 더티 데이터를 추적하기 위해 더티 데이터 레코드 시스템을 사용합니다. 각 DDR(더티 데이터 레코드)은 특정 파일에 대해 약 20MB의 더티 데이터를 나타냅니다. 파일이 활발하게 작성되면 ONTAP는 두 개의 DDR이 채워지고 세 번째 DDR이 쓰여진 후에 더티 데이터를 다시 플러시하기 시작합니다. 따라서 쓰기 중에 캐시에 약 40MB의 더티 데이터가 남아 있습니다. 상태 저장 프로토콜(NFSv4.x, SMB)의 경우 파일을 닫을 때 나머지 40MB의 데이터가 오리진으로 다시 플러시됩니다. 상태 비저장 프로토콜(NFSv3)의 경우, 다른 캐시에서 파일에 대한 액세스를 요청하거나 파일이 2분 이상 유휴 상태가 되면 최대 5분 동안 40MB의 데이터가 다시 플러시됩니다. 타이머가 트리거되거나 공간이

트리거되는 더티 데이터 플러싱에 대한 자세한 내용은 [을 참조하십시오. 캐시 스크러버](#)

DDR 및 스크러버 외에도 일부 프런트엔드 NAS 작업은 파일에 대한 모든 더티 데이터를 플러시합니다.

- SETATTR
 - mtime, atime 및/또는 ctime만 수정하는 SETATTR은 캐시에서 처리할 수 있으므로 WAN의 페널티를 피할 수 있습니다.
- CLOSE
- OPEN 확인할 수 있습니다
- READ 확인할 수 있습니다
- REaddir 확인할 수 있습니다
- REaddirplus 확인할 수 있습니다
- WRITE 확인할 수 있습니다

연결 해제 모드

파일의 XLD가 write-around 캐시에 보관되고 해당 캐시가 오리진에서 연결이 끊기면 해당 파일에 대한 읽기는 다른 캐시와 오리진에서 계속 허용됩니다. 이 동작은 다시 쓰기 가능 캐시에 의해 XLD가 유지되는 경우에 다릅니다. 이 경우 캐시의 연결이 끊긴 경우 파일 읽기가 모든 곳에서 중단됩니다. 이를 통해 100% 일관성, 통화 및 일관성을 유지할 수 있습니다. 오리진에서 클라이언트에 쓰기 승인되어 있는 모든 데이터를 사용할 수 있으므로 읽기는 write-around 모드로 허용됩니다. 연결 해제 중 Write-back 모드에서는 오리진에서 다시 쓰기 가능 캐시에 기록되고 인식되는 모든 데이터가 연결 해제되기 전에 오리진으로 만들어지는 것을 보장할 수 없습니다.

파일에 대한 XLD가 있는 캐시의 연결이 장시간 끊어진 경우 시스템 관리자는 원본에서 XLD를 수동으로 취소할 수 있습니다. 이렇게 하면 정상적인 캐시 및 오리진에서 파일에 대한 입출력이 재개됩니다.



XLD를 수동으로 해지하면 연결이 끊어진 캐시에서 파일에 대한 더티 데이터가 손실됩니다. XLD를 수동으로 취소하는 작업은 캐시와 원본 간에 심각한 장애가 발생한 경우에만 수행해야 합니다.

캐시 스크러버

ONTAP에는 타이머 만료 또는 공간 임계값이 위반되는 것과 같은 특정 이벤트에 대한 응답으로 실행되는 스크러버가 있습니다. 스크러버는 스크러브되는 파일에 대해 배타적인 잠금을 가지고 스크립트가 완료될 때까지 해당 파일에 대한 입출력을 효과적으로 동결합니다.

스크러버에는 다음이 포함됩니다.

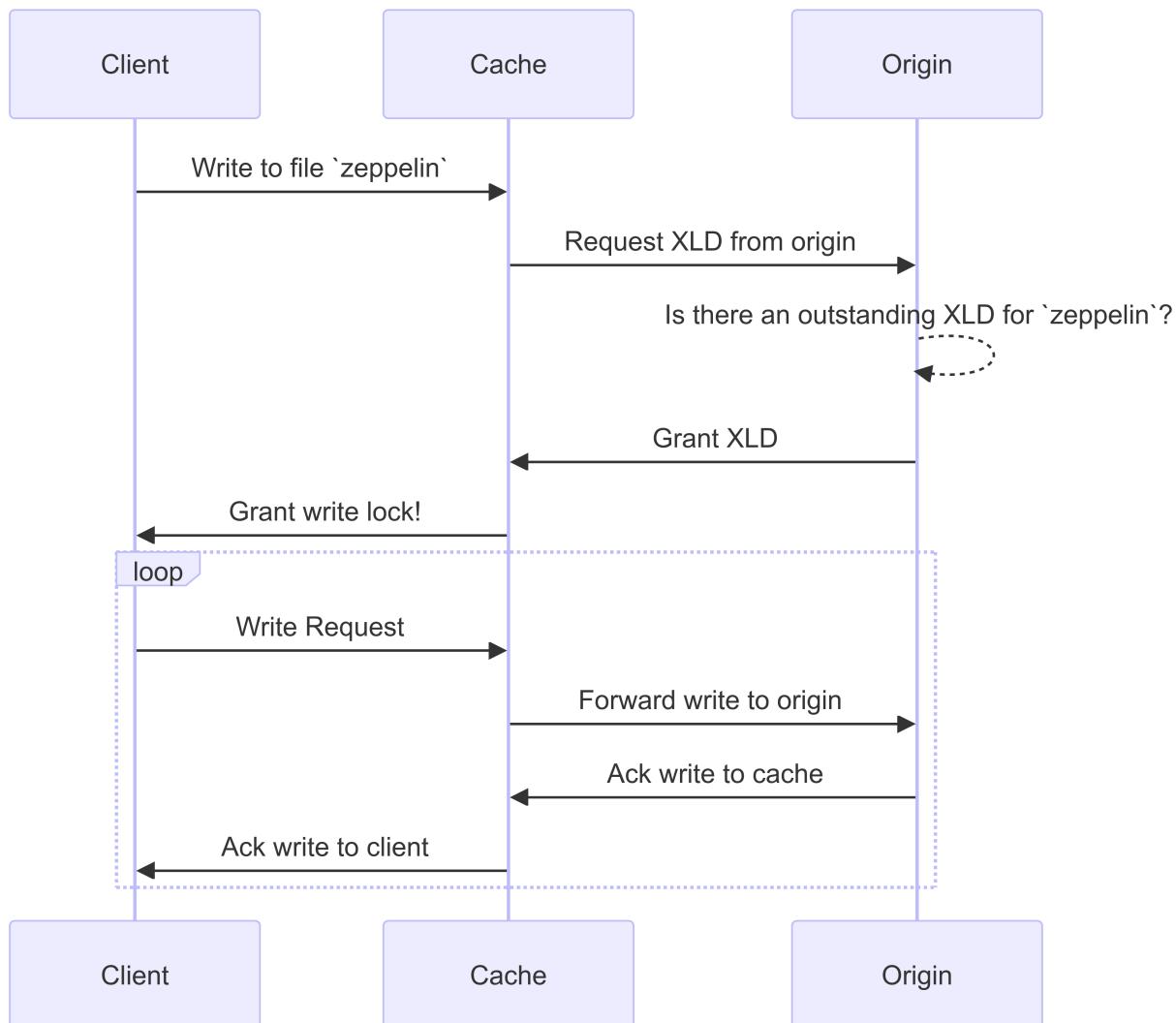
- *mtime-based scrubber on the cache: * 이 스크러버는 5분마다 시작되며 2분 동안 수정되지 않은 모든 파일을 스크러브합니다. 파일에 대한 더티 데이터가 여전히 캐시에 있는 경우 해당 파일에 대한 입출력이 정지되고 다시 쓰기가 트리거됩니다. 다시 쓰기가 완료된 후 IO가 재개됩니다.
- * mtime-based scrubber on origin: * 캐시의 mtime 기반 scrubber와 마찬가지로 5분마다 실행됩니다. 하지만 수정되지 않은 파일을 15분 동안 스크립하여 inode의 위임을 불러옵니다. 이 스크러버는 다시 쓰기를 시작하지 않습니다.
- * RW limit-based scrubber on origin: * ONTAP는 오리진 구성요소당 얼마나 많은 RW 잠금 위임이 처리되는지 모니터링합니다. 이 숫자가 170을 초과하면 ONTAP는 LRU(Least-Recently-Used)를 기준으로 쓰기 잠금 위임을 스크립하기 시작합니다.

- * 캐시 상의 공간 기반 스크러버: * FlexCache 볼륨이 90%에 도달하면 캐시가 스크러빙되어 LRU 기준으로 제거됩니다.
- * 오리진에서 공간 기반 스크러버: * FlexCache 오리진 볼륨이 90%에 도달하면 캐시가 스크러빙되어 LRU 기준으로 제거됩니다.

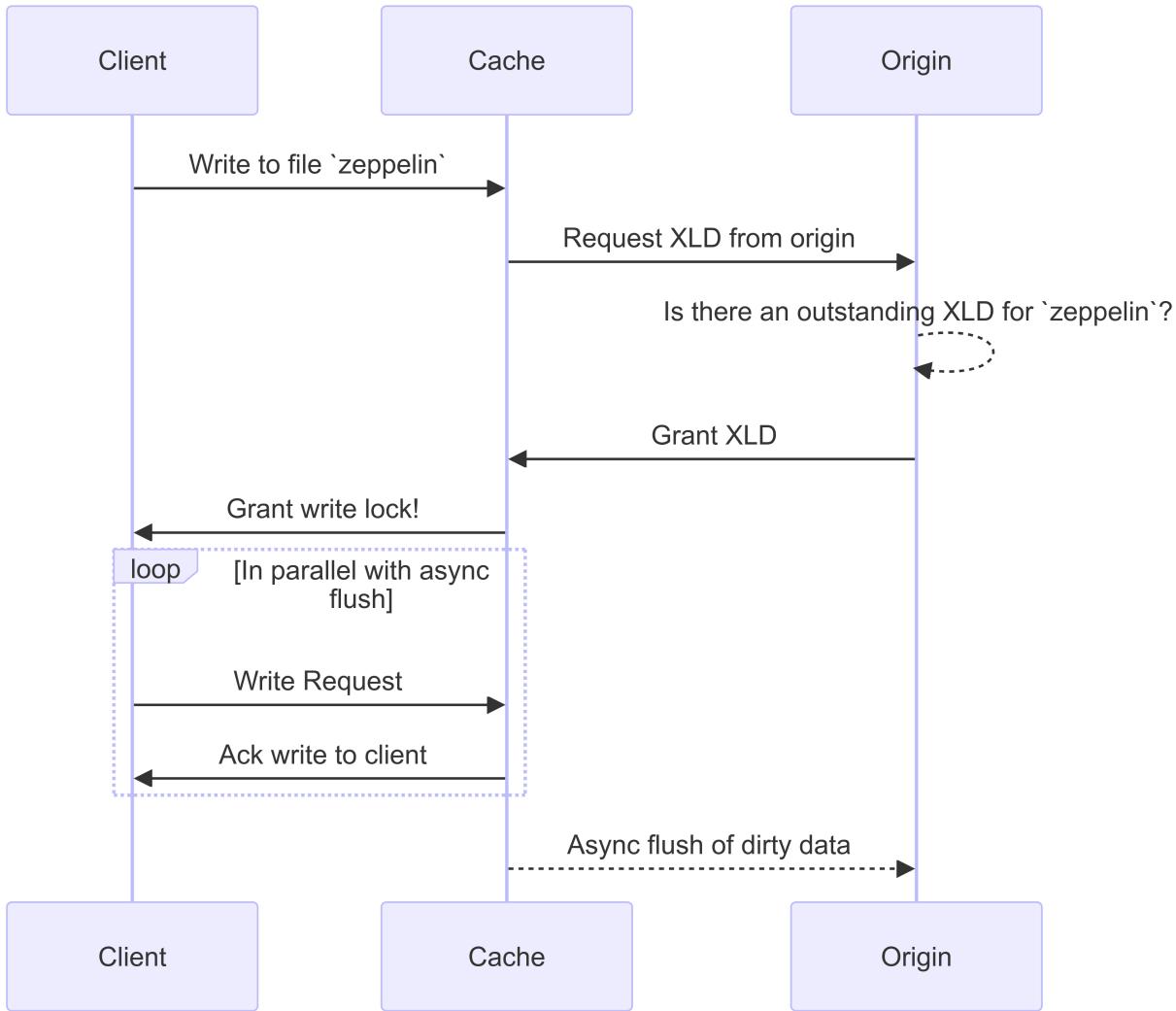
시퀀스 다이어그램

이러한 시퀀스 다이어그램은 write-around 모드와 write-back 모드 간의 쓰기 확인의 차이를 나타냅니다.

쓰기



다시 쓰기



ONTAP FlexCache 후기입 사용 사례

쓰기 프로파일은 다시 쓰기 가능 FlexCache에 가장 적합합니다. 작업을 테스트하여 Write-back 또는 write-around가 최상의 성능을 제공하는지 확인해야 합니다.



Write-back은 write-around를 대체하는 것이 아닙니다. 쓰기 작업이 많은 워크로드를 처리하기 위해 쓰기팅을 설계했지만 여전히 많은 워크로드에서 쓰기 작업이 더 적합합니다.

타겟 워크로드

파일 크기

파일 크기는 및 파일 호출 사이에 실행된 쓰기 수보다 덜 OPEN CLOSE 중요합니다. 작은 파일은 기본적으로 호출 수가 적기 때문에 WRITE 다시 쓰기에 적합하지 않습니다. 큰 파일은 및 호출 사이에 더 많은 쓰기를 가질 수 OPEN CLOSE 있지만, 이는 보장되지 않습니다.

["FlexCache write-back 지침"](#)최대 파일 크기에 대한 최신 권장 사항은 페이지를 참조하십시오.

쓰기 크기

클라이언트에서 쓸 때 쓰기 호출이 아닌 다른 수정 NAS 호출이 관련됩니다. 여기에는 다음이 포함되지만 이에 국한되지

않습니다.

- CREATE
- OPEN
- CLOSE
- SETATTR
- SET_INFO

SETATTR SET_INFO atime, ctime, owner group 또는 를 size 설정한 호출이 mtime 캐시에서 처리됩니다. 이러한 나머지 호출은 오리진에서 처리되어야 하며, 작업 중인 파일에 대해 Write-back이 활성화된 캐시에 누적된 더티 데이터의 Write-Back을 트리거해야 합니다. 파일 입출력은 쓰기가 완료될 때까지 정지됩니다.

이러한 호출이 WAN을 통과해야 한다는 것을 알면 다시 쓰기에 적합한 워크로드를 식별하는 데 도움이 됩니다. 일반적으로 OPEN CLOSE 위에 나열된 다른 호출 중 하나를 실행하지 않고 및 호출 간에 수행할 수 있는 쓰기가 많을수록 성능 향상 Write-Back이 더 좋습니다.

쓰기 후 읽기

지금까지 FlexCache에서 쓰기 후 읽기 워크로드의 성능은 좋지 않았습니다. 이는 9.15.1 이전의 write-around 동작 방식 때문이다. WRITE `파일에 대한 호출은 오리진에서 커밋되어야 하며, 후속 `READ 호출은 데이터를 다시 캐시로 가져와야 합니다. 이로 인해 두 작업 모두 WAN 페널티가 발생합니다. 따라서 FlexCache의 경우 쓰기 후 읽기 워크로드는 쓰기 방지 모드로 사용하지 않습니다. 9.15.1에 쓰기 저장 기능이 도입됨에 따라 데이터가 캐시에서 커밋되고 캐시에서 즉시 읽을 수 있으므로 WAN 페널티가 없어집니다. 워크로드에 FlexCache 볼륨에 쓰기 후 읽기가 포함된 경우에는 write-back 모드로 작동하도록 캐시를 구성해야 합니다.



쓰기 후 읽기가 작업 부하에서 중요한 부분인 경우 다시 쓰기 모드로 작동하도록 캐시를 구성해야 합니다.

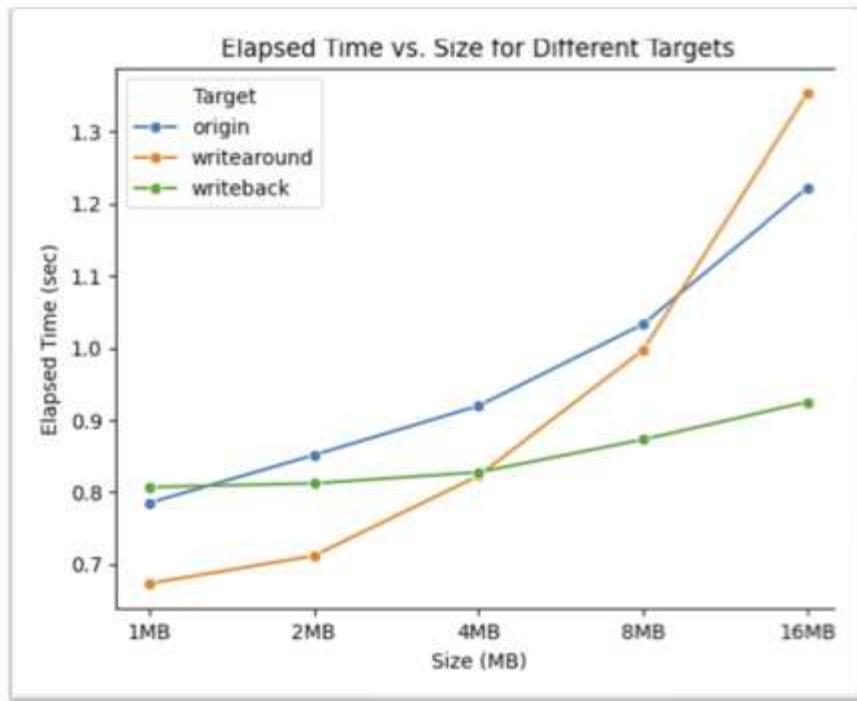
쓰기 후 쓰기

파일이 캐시에 더티 데이터를 축적하면 캐시는 데이터를 원본에 비동기적으로 씁니다. 이렇게 되면 원래 위치로 다시 플러시될 때까지 계속 대기 중인 더티 데이터로 파일이 닫히는 경우가 발생합니다. 방금 닫은 파일에 대해 다른 열기 또는 쓰기가 들어오면 더티 데이터가 모두 오리진으로 플러시될 때까지 쓰기가 일시 중단됩니다.

지연 시간 고려

FlexCache가 Write-back 모드로 작동하는 경우 지연 시간이 증가함에 따라 NAS 클라이언트에 더 많은 이점을 제공합니다. 하지만 지연 시간이 짧은 환경에서 얻을 수 있는 이점보다 쓰기 작업의 오버헤드가 훨씬 더 중요합니다. 일부 NetApp 테스트에서는 캐시와 원본 간 지연 시간이 8ms를 최소한으로 초과했을 때 쓰기 저장 이점이 시작되었습니다. 이 지연 시간은 워크로드에 따라 달라지므로 워크로드의 반환 시점을 테스트해야 합니다.

다음 그림은 NetApp 랩 테스트에서 Write-back에 대한 반환 지점을 보여 줍니다. x 축은 파일 크기이고 y 축은 경과 시간입니다. 이 테스트에서는 NFSv3, 256KB 및 64ms의 WAN 지연 시간으로 마운트했습니다. rsize wsize 이 테스트는 캐시와 오리진에 대해 작은 ONTAP Select 인스턴스와 단일 스레드 쓰기 작업을 사용하여 수행되었습니다. 결과는 다를 수 있습니다.



클러스터 내 캐싱에는 Write-back을 사용해서는 안 됩니다. 클러스터 내 캐싱은 오리진과 캐시가 같은 클러스터에 있을 때 발생합니다.

ONTAP FlexCache write-back 사전 요구 사항

쓰기 저장 모드로 FlexCache를 배포하기 전에 이러한 성능, 소프트웨어, 라이센스 및 시스템 구성 요구 사항을 충족하는지 확인하십시오.

CPU 및 메모리

쓰기 저장이 활성화된 캐시에서 시작된 쓰기 저장 메시지를 수용하기 위해 각 원본 클러스터 노드에 최소 128GB의 RAM과 20개의 CPU가 있는 것이 강력히 권장됩니다. A400 이상과 동일합니다. 오리진 클러스터가 복수 쓰기 가능 FlexCaches의 오리진으로 작동하는 경우 더 많은 CPU와 RAM이 필요합니다.



작업 부하에 크기가 작은 원본을 사용하면 다시 쓰기 가능 캐시 또는 오리진에서 성능에 큰 영향을 줄 수 있습니다.

ONTAP 버전입니다

- 오리진 *MUST*에서 ONTAP 9.15.1 이상을 실행하고 있습니다.
- write-back 모드에서 작동해야 하는 캐싱 클러스터는 *must* ONTAP 9.15.1 이상을 실행하고 있습니다.
- write-back 모드로 작동할 필요가 없는 캐싱 클러스터는 일반적으로 지원되는 모든 ONTAP 버전을 실행할 수 있습니다.

라이센싱

ONTAP 구매에 쓰기 작업 모드를 포함한 FlexCache가 포함됩니다. 추가 라이센스가 필요하지 않습니다.

피어링

- 오리진 및 캐시 클러스터는 이어야 합니다 "[클러스터 피어링된](#)"
- 원본 및 캐시 클러스터의 SVM(서버 가상 머신)은 FlexCache 옵션을 사용해야 합니다 "[SVM 피어링이 발생했습니다](#)".



캐시 클러스터를 다른 캐시 클러스터로 피어링할 필요가 없습니다. 또한 캐시 SVM을 다른 캐시 SVM에 피어링할 필요가 없습니다.

ONTAP FlexCache write-back 상호 운용성

write-back 모드로 FlexCache를 구축할 때 고려해야 할 상호 운용성 사항을 이해합니다.

ONTAP 버전입니다

쓰기 저장 모드를 사용하려면 캐시와 오리진 * 둘 다에서 ONTAP 9.15.1 이상을 실행하고 있어야 합니다.



Write-back 사용 캐시가 필요하지 않은 클러스터는 이전 버전의 ONTAP를 실행할 수 있지만 해당 클러스터는 write-around 모드에서만 작동할 수 있습니다.

사용자 환경에서 ONTAP 버전을 혼합하여 사용할 수 있습니다.

클러스터	ONTAP 버전입니다	후기입 지원 여부
* 원점 *	ONTAP 9.15.1	해당 없음 †
* 클러스터 1 *	ONTAP 9.15.1	예
* 클러스터 2 *	ONTAP 9.14.1	아니요

클러스터	ONTAP 버전입니다	후기입 지원 여부
* 원점 *	ONTAP 9.14.1	해당 없음 †
* 클러스터 1 *	ONTAP 9.15.1	아니요
* 클러스터 2 *	ONTAP 9.15.1	아니요

†_Origins는 캐시가 아니므로 write-back 또는 write-around 지원은 적용할 수 없습니다. _



에서는 [\[example2-table\]](#)오리진이 엄격한 요구 사항인 ONTAP 9.15.1 이상을 실행하고 있지 않기 때문에 두 클러스터 모두 다시 쓰기 모드를 활성화할 수 없습니다.

클라이언트 상호 운용성

ONTAP에서 일반적으로 지원하는 모든 클라이언트는 FlexCache 볼륨이 write-around 모드로 작동하든 write-back 모드로 작동하든 상관없이 볼륨에 액세스할 수 있습니다. 지원되는 클라이언트의 최신 목록은 NetApp을 참조하십시오. "[상호 운용성 매트릭스](#)"

클라이언트 버전은 특별히 중요하지 않지만 NFSv3, NFSv4.0, NFSv4.1, SMB2.x 또는 SMB3.x를 지원할 수 있을 만큼 새로운 클라이언트여야 합니다 SMB1 및 NFSv2는 더 이상 사용되지 않는 프로토콜이며 지원되지 않습니다.

다시 쓰기 및 덮어쓰기를 수행합니다

에서 볼 수 있듯이 [example1-table]write-back 모드에서 작동하는 FlexCache는 write-around 모드로 작동하는 캐시와 함께 존재할 수 있습니다. 특정 작업 부하와 Write-Back을 비교하여 사용하는 것이 좋습니다.



write-back과 write-around 간에 워크로드 성능이 동일할 경우 write-around를 사용합니다.

ONTAP 기능 상호 운용성

FlexCache 기능 상호 운용성에 대한 최신 목록은 ["FlexCache 볼륨에 대해 지원되는 기능과 지원되지 않는 기능"](#)을 참조하십시오.

ONTAP FlexCache write-back 설정 및 관리

ONTAP 9.15.1부터 FlexCache 볼륨에서 FlexCache Write-back 모드를 활성화하여 예지 컴퓨팅 환경 및 쓰기 작업이 많은 워크로드가 있는 캐시에 더 나은 성능을 제공할 수 있습니다. 또한 필요한 경우 FlexCache 볼륨에 write-back이 설정되어 있는지 아니면 볼륨에 대한 write-back을 사용하지 않도록 설정할 수도 있습니다.

캐시 볼륨에 다시 쓰기가 설정되어 있으면 쓰기 요청이 원본 볼륨이 아닌 로컬 캐시로 전송됩니다.

시작하기 전에

고급 권한 모드여야 합니다.

Write-back이 설정된 새 FlexCache 볼륨을 생성합니다

단계

ONTAP System Manager 또는 ONTAP CLI를 사용하여 write-back을 사용하도록 설정한 새 FlexCache 볼륨을 생성할 수 있습니다.

시스템 관리자

1. FlexCache 볼륨이 원본 볼륨과 다른 클러스터에 있는 경우 클러스터 피어 관계를 생성합니다.
 - a. 로컬 클러스터에서 * 보호 > 개요 * 를 클릭합니다.
 - b. Intercluster Settings * 를 확장하고 * Add Network Interfaces * 를 클릭한 후 클러스터에 클러스터 간 인터페이스를 추가합니다.

원격 클러스터에서 이 작업을 반복합니다.

 - c. 원격 클러스터에서 * 보호 > 개요 * 를 클릭합니다. Cluster Peers 섹션을 클릭하고 * Generate Passphrase * 를 클릭합니다.
 - d. 생성된 암호를 복사하여 로컬 클러스터에 붙여넣습니다.
 - e. 로컬 클러스터의 클러스터 피어 아래에서 * 피어 클러스터 * 를 클릭하고 로컬 및 원격 클러스터를 피어링합니다.
2. FlexCache 볼륨이 원본 볼륨과 다른 클러스터에 있는 경우 SVM 피어 관계를 생성합니다.

스토리지 VM 피어 * 아래에서 * 피어 : 스토리지 VM * 을 클릭한 다음 * 피어 스토리지 VM * 을 클릭하여 스토리지 VM을 피어링합니다.

FlexCache 볼륨이 동일한 클러스터에 있는 경우 System Manager를 사용하여 SVM 피어 관계를 생성할 수 없습니다.

3. 스토리지 > 볼륨 * 을 선택합니다.
4. 추가 * 를 선택합니다.
5. More Options * 를 선택한 다음 * Add as cache for a remote volume * 를 선택합니다.
6. Enable FlexCache write-back * 을 선택합니다.

CLI를 참조하십시오

1. 생성할 FlexCache 볼륨이 다른 클러스터에 있는 경우 클러스터 피어 관계를 생성합니다.
 - a. 대상 클러스터에서 데이터 보호 소스 클러스터와 피어 관계를 생성합니다.

```
cluster peer create -generate-passphrase -offer-expiration
MM/DD/YYYY HH:MM:SS|1...7days|1...168hours -peer-addrs
<peer_LIF_IPs> -initial-allowed-vserver-peers <svm_name>,..|*
-ipspace <ipspace_name>
```

ONTAP 9.6부터는 클러스터 피어 관계를 만들 때 TLS 암호화가 기본적으로 활성화됩니다. TLS 암호화는 오리진과 FlexCache 볼륨 간의 인터클러스터 통신에 지원됩니다. 필요한 경우 클러스터 피어 관계에 대해 TLS 암호화를 비활성화할 수도 있습니다.

```

cluster02::> cluster peer create -generate-passphrase -offer
-expiration 2days -initial-allowed-vserver-peers *

        Passphrase: UCa+6lRVICXeL/gq1WrK7ShR
        Expiration Time: 6/7/2017 08:16:10 EST
        Initial Allowed Vserver Peers: *
        Intercluster LIF IP: 192.140.112.101
        Peer Cluster Name: Clus_7ShR (temporary generated)

Warning: make a note of the passphrase - it cannot be displayed
again.

```

- a. 소스 클러스터에서 소스 클러스터를 대상 클러스터에 인증합니다.

```
cluster peer create -peer-addrs <peer_LIF_IPs> -ipspace <ipspace>
```

```

cluster01::> cluster peer create -peer-addrs
192.140.112.101,192.140.112.102

```

Notice: Use a generated passphrase or choose a passphrase of 8 or more characters.

To ensure the authenticity of the peering relationship, use a phrase or sequence of characters that would be hard to guess.

Enter the passphrase:

Confirm the passphrase:

Clusters cluster02 and cluster01 are peered.

2. FlexCache 볼륨이 원본 볼륨과 다른 SVM에 있는 경우 애플리케이션으로서 "FlexCache"와 SVM 피어 관계를 생성합니다.

- a. SVM이 다른 클러스터에 있는 경우 피어링 SVM에 대한 SVM 권한을 생성합니다.

```
vserver peer permission create -peer-cluster <cluster_name>
-vserver <svm-name> -applications flexcache
```

다음 예에서는 모든 로컬 SVM에 적용되는 SVM 피어 권한을 생성하는 방법을 보여줍니다.

```

cluster1::> vserver peer permission create -peer-cluster cluster2
-vserver "*" -applications flexcache

Warning: This Vserver peer permission applies to all local Vservers.
After that no explicit
"vserver peer accept" command required for Vserver peer relationship
creation request
from peer cluster "cluster2" with any of the local Vservers. Do you
want to continue? {y|n}: y

```

a. SVM 피어 관계 생성:

```
vserver peer create -vserver <local_SVM> -peer-vserver
<remote_SVM> -peer-cluster <cluster_name> -applications flexcache
```

3. Write-Back(후기입)이 설정된 FlexCache 볼륨 생성:

```

volume flexcache create -vserver <cache_vserver_name> -volume
<cache_flexgroup_name> -aggr-list <list_of_aggregates> -origin
-volume <origin flexgroup> -origin-vserver <origin_vserver name>
-junction-path <junction_path> -is-writeback-enabled true

```

기존 FlexCache 볼륨에서 FlexCache write-back을 활성화합니다

ONTAP System Manager 또는 ONTAP CLI를 사용하여 기존 FlexCache 볼륨에서 FlexCache write-back을 활성화할 수 있습니다.

시스템 관리자

- 스토리지 > 볼륨 * 을 선택하고 기존 FlexCache 볼륨을 선택합니다.
- 볼륨의 개요 페이지에서 오른쪽 상단에 있는 * 편집 * 을 클릭합니다.
- 볼륨 편집 * 창에서 * FlexCache 다시 쓰기 활성화 * 를 선택합니다.

CLI를 참조하십시오

- 기존 FlexCache 볼륨에서 write-back 활성화:

```

volume flexcache config modify -volume <cache_flexgroup_name> -is
-writeback-enabled true

```

FlexCache write-back이 활성화되어 있는지 확인한다

단계

System Manager 또는 ONTAP CLI를 사용하여 FlexCache write-back이 활성화되어 있는지 확인할 수 있습니다.

시스템 관리자

- 스토리지 > 볼륨 * 을 선택하고 볼륨을 선택합니다.
- 볼륨 * 개요 * 에서 * FlexCache details * 를 찾아 FlexCache 볼륨에서 FlexCache write-back이 * Enabled * 로 설정되어 있는지 확인합니다.

CLI를 참조하십시오

- FlexCache write-back이 활성화되어 있는지 확인합니다.

```
volume flexcache config show -volume <cache_flexgroup_name> -fields  
is-writeback-enabled
```

FlexCache 볼륨에서 write-back을 비활성화합니다

FlexCache 볼륨을 삭제하려면 먼저 FlexCache Write-Back을 비활성화해야 합니다.

단계

System Manager 또는 ONTAP CLI를 사용하여 FlexCache write-back을 사용하지 않도록 설정할 수 있습니다.

시스템 관리자

- 스토리지 > 볼륨 * 을 선택하고 FlexCache write-back이 활성화된 기존 FlexCache 볼륨을 선택합니다.
- 볼륨의 개요 페이지에서 오른쪽 상단에 있는 * 편집 * 을 클릭합니다.
- 볼륨 편집 * 창에서 * FlexCache 다시 쓰기 활성화 * 를 선택 해제합니다.

CLI를 참조하십시오

- 후기입 해제:

```
volume flexcache config modify -volume <cache_vol_name> -is  
-writeback-enabled false
```

ONTAP FlexCache 후기입(Write-Back)에 대한 질문과 대답(FAQ)

이 FAQ는 질문에 대한 빠른 답변을 찾는 경우에 도움이 될 수 있습니다.

다시 쓰기를 사용하고 싶습니다. 어떤 버전의 ONTAP를 실행해야 합니까?

캐시와 오리진 모두 ONTAP 9.15.1 이상을 실행해야 합니다. 최신 P 릴리스를 실행하는 것이 좋습니다. 엔지니어링은

Write-back 사용 캐시의 성능과 기능을 지속적으로 개선하고 있습니다.

오리진에 액세스하는 클라이언트가 다시 쓰기 가능 캐시에 액세스하는 클라이언트에 영향을 줄 수 있습니까?

예. 오리진에는 캐시와 동일한 데이터 권한이 있습니다. 캐시에서 파일을 제거하거나 잠금/데이터 위임을 취소해야 하는 파일에 대해 작업이 실행되는 경우 캐시의 클라이언트에서 파일 액세스가 차단될 수 있습니다.

Write-back 지원 FlexCaches에 QoS를 적용할 수 있습니까?

예. 모든 캐시와 오리진에는 독립적인 QoS 정책이 적용될 수 있습니다. 이 기능은 write-back이 시작되는 인터클러스터 트래픽에 직접적인 영향을 주지 않습니다. 간접적으로 write-back 지원 캐시에서 프런트 엔드 트래픽을 제한하여 인터클러스터 write-back 트래픽을 줄일 수 있습니다.

멀티 프로토콜 **NAS**는 후기입 가능 **FlexCaches**에서 지원됩니까?

예. 멀티 프로토콜은 후기입 가능 FlexCaches에서 완벽하게 지원됩니다. 현재 NFSv4.2 및 S3는 write-around 또는 write-back 모드로 작동하는 FlexCache에서 지원되지 않습니다.

SMB 대체 데이터 스트림은 후기입 지원 FlexCaches에서 지원됩니까?

SMB ADS(대체 데이터 스트림)는 지원되지만 Write-Back을 통해 가속화되지는 않습니다. ADS에 대한 쓰기가 오리진으로 전달되어 WAN 지연 시간이 줄어듭니다. 또한 쓰기는 ADS가 캐시의 일부인 주 파일을 제거합니다.

캐시를 생성한 후 **write-around** 모드와 **write-back** 모드 간에 전환할 수 있습니까?

예. 당신은 할 일은 링크의 플래그를 토글하는 것입니다 `is-writeback-enabled:../FlexCache-writeback/FlexCache -writeback-enable-task.html[flexcache modify 명령]`.

캐시와 원본 간의 클러스터 간 링크에 대해 알아야 할 대역폭 고려 사항이 있습니까?

네. FlexCache 쓰기 저장은 캐시와 원본 간의 클러스터 간 링크에 크게 의존합니다. 대역폭이 낮거나 손실이 많은 네트워크는 성능에 상당한 부정적인 영향을 미칠 수 있습니다. 작업 부하에 따라 크게 달라지므로 특정 대역폭 요구 사항은 없습니다.

FlexCache 이중성

FlexCache 이중성에 대한 FAQ

이 FAQ는 ONTAP 9.18.1에서 도입된 FlexCache 이중화에 대한 일반적인 질문에 답변합니다.

자주 묻는 질문

"이원성"이란 무엇입니까?

듀얼리티(Duality)는 파일(NAS) 및 객체(S3) 프로토콜을 모두 사용하여 동일한 데이터에 통합적으로 접근할 수 있도록 합니다. FlexCache 지원 없이 ONTAP 9.12.1에서 처음 도입된 듀얼리티는 ONTAP 9.18.1에서 FlexCache 볼륨을 포함하도록 확장되어 FlexCache 볼륨에 캐시된 NAS 파일에 S3 프로토콜 액세스를 허용합니다.

FlexCache S3 버킷에서 지원되는 S3 작업에는 어떤 것들이 있습니까?

표준 S3 NAS 버킷에서 지원되는 S3 작업은 FlexCache S3 NAS 버킷에서도 지원되지만 COPY 작업은 예외입니다. 표준 S3 NAS 버킷에서 지원되지 않는 작업의 최신 목록은 "[상호 운용성 문서](#)"를 참조하십시오.

FlexCache 이중성 기능을 사용하여 FlexCache를 쓰기 후 저장 모드로 사용할 수 있습니까?

아니요. FlexCache 볼륨에 FlexCache S3 NAS 버킷을 생성하려면 FlexCache 볼륨이 반드시 쓰기 우회 모드여야 합니다. 쓰기 백 모드로 설정된 FlexCache 볼륨에 FlexCache S3 NAS 버킷을 생성하려고 하면 작업이 실패합니다.

하드웨어 제한 때문에 클러스터 중 하나를 **ONTAP 9.18.1**로 업그레이드할 수 없습니다. 캐시 클러스터만 **ONTAP 9.18.1**을 실행하는 경우 내 클러스터에서 **duality**가 계속 작동합니까?

아니요. 캐시 클러스터와 오리진 클러스터 모두 최소 유효 클러스터 버전이 9.18.1 이상이어야 합니다. ONTAP 버전 9.18.1보다 이전 버전인 오리진 클러스터와 피어링된 캐시 클러스터에 FlexCache S3 NAS 버킷을 생성하려고 하면 작업이 실패합니다.

MetroCluster 구성이 있습니다. **FlexCache** 이중화 기능을 사용할 수 있습니까?

아니요. FlexCache 이중성은 MetroCluster 구성에서 지원되지 않습니다.

FlexCache S3 NAS 버킷에 있는 파일에 대한 **S3** 액세스를 감사할 수 있습니까?

S3 감사는 FlexCache 볼륨에서 사용하는 NAS 감사 기능을 통해 제공됩니다. FlexCache 볼륨의 NAS 감사에 대한 자세한 내용은 "[FlexCache 감사에 대해 자세히 알아보세요](#)"를 참조하십시오.

캐시 클러스터가 원본 클러스터에서 연결이 끊어지면 어떻게 됩니까?

FlexCache S3 NAS 버킷에 대한 S3 요청은 캐시 클러스터가 원본 클러스터에서 연결이 끊어진 경우 503 Service Unavailable 오류와 함께 실패합니다.

FlexCache 이중성을 사용하여 **S3** 멀티파트 작업을 수행할 수 있습니까?

멀티파트 S3 작업이 제대로 작동하려면 기본 FlexCache 볼륨의 granular-data 필드가 'advanced'로 설정되어 있어야 합니다. 이 필드는 원본 볼륨에 설정된 값으로 설정됩니다.

FlexCache 이중화 기능은 **HTTP**와 **HTTPS** 액세스를 모두 지원합니까?

예. 기본적으로 HTTPS가 필요합니다. 필요한 경우 S3 서비스에서 HTTP 액세스를 허용하도록 구성할 수 있습니다.

NAS FlexCache 볼륨에 대한 **S3** 액세스를 활성화합니다

ONTAP 9.18.1부터 NAS FlexCache 볼륨에 대한 S3 액세스를 활성화할 수 있으며, 이를 "이중성"이라고도 합니다. 이를 통해 클라이언트는 NFS 및 SMB와 같은 기존 NAS 프로토콜 외에도 S3 프로토콜을 사용하여 FlexCache 볼륨에 저장된 데이터에 액세스할 수 있습니다. 다음 정보를 사용하여 FlexCache 이중성을 설정할 수 있습니다.

필수 구성 요소

시작하기 전에 다음 필수 조건을 모두 완료했는지 확인하십시오.

- S3 프로토콜과 원하는 NAS 프로토콜(NFS, SMB 또는 둘 다)에 대한 라이센스가 부여되고 SVM에 구성되어 있는지 확인하십시오.
- DNS 및 기타 필요한 서비스가 구성되어 있는지 확인하십시오.
- 클러스터 및 SVM 피어링됨
- FlexCache 볼륨 생성
- 데이터 LIF가 생성되었습니다



FlexCache 이중성에 대한 더 자세한 문서는 "[ONTAP S3 멀티프로토콜 지원](#)"을 참조하십시오.

1단계: 인증서 생성 및 서명

FlexCache 볼륨에 대한 S3 액세스를 활성화하려면 FlexCache 볼륨을 호스팅하는 SVM에 인증서를 설치해야 합니다. 이 예제에서는 자체 서명 인증서를 사용하지만 운영 환경에서는 신뢰할 수 있는 인증 기관(CA)에서 서명한 인증서를 사용해야 합니다.

1. SVM 루트 CA 생성:

```
security certificate create -vserver <svm> -type root-ca -common-name  
<arbitrary_name>
```

2. 인증서 서명 요청 생성:

```
security certificate generate-csr -common-name <dns_name_of_data_lif>  
-dns-name <dns_name_of_data_lif> -ipaddr <data_lif_ip>
```

출력 예:

```
-----BEGIN CERTIFICATE REQUEST-----  
MIICzjCCAbYCAQAwHzEdMBsGA1UEAxMUY2FjaGUxZy1kYXRhLm5hcY5sYWIwggEi  
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQcusJk075O8Uh329cHI6x+BaRS2  
w5wrqvzoYlidXtYmdCH3m1DDprBiAyfIwBC0/iU3Xd5NpB7nc1wK1CI2VEkrXGUG  
...  
vMIGN351+FgzLQ4X51KfoMXCV70NqIakxzEmkTIUDKv7n9EVZ4b5DTT1rL03X/nK  
+Bim2y2y180PaFB3NauZHTnIIzIc8zCp2IEqmFWyMDcdBjP9KS0+jNm4QhuXiM8F  
D7gm3g/O70qa5OxbAEa15o4NbO195U0T0rwqTaSzFG0XQnK2PmA1OIwS5ET35p3Z  
dLU=  
-----END CERTIFICATE REQUEST-----
```

개인 키 예:

```
-----BEGIN PRIVATE KEY-----  
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQcusJk075O8Uh32  
9cHI6x+BaRS2w5wrqvzoYlidXtYmdCH3m1DDprBiAyfIwBC0/iU3Xd5NpB7nc1wK  
1CI2VEkrXGUGwBtx1K4IlrCTB829Q1aLGAQXVYWnzhQc4tS5PW/DsQ8t7o1Z9zEI  
...  
rXGEDDaqp7jQGNXUGlbxO3zcBil1/A9Hc6oalNECgYBKwe3PeZamiwhIHLy9ph7w  
dJFFCshsPalMuAp2OuKIAAnNa916ft9y5kf9tIbskT+t5Dth8bmV9pwe8UZaK5eC4  
Svxm19jHT5Qql0DaZVUmMXFKyKoqPDdfvcDk2Eb5gMfIIb0a3TPC/jqqpDn9BzuH  
TO02fuRvRR/G/HUz2yRd+A==  
-----END PRIVATE KEY-----
```



나중에 참조할 수 있도록 인증서 요청서와 개인 키 사본을 보관하십시오.

3. 인증서에 서명합니다.

```
`root-ca`는 <<anchor1-step, SVM 루트 CA 생성>>에서 생성한 것입니다.
```

```
certificate sign -ca <svm_root_ca> -ca-serial <svm_root_ca_sn> -expire  
-days 364 -format PEM -vserver <svm>
```

4. [인증서 서명 요청 생성](#)에서 생성된 CSR(Certificate Signing Request)을 붙여넣습니다.

예:

```
-----BEGIN CERTIFICATE REQUEST-----  
MIICzjCCAbYCAQAwHzEdMBsGA1UEAxMUY2FjaGUxZy1kYXRhLm5hc5sYWIwggEi  
MA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCuJk07508Uh329cHI6x+BaRS2  
w5wrqvzoYlidXtYmdCH3m1DDprBiAyfIwBC0/iU3Xd5NpB7nc1wK1CI2VEkrXGUg  
...  
vMIGN351+FgzLQ4X51KfoMXCV70NqIakxzEmkTIUDKv7n9EVZ4b5DTT1rL03X/nK  
+Bim2y2y180PaFB3NauZHTnIIzIc8zCp2IEqmFWyMDcdBjP9KS0+jNm4QhuXiM8F  
D7gm3g/O70qa5OxbAEa15o4Nb0l95U0T0rwqTaSzFG0XQnK2PmA1OIwS5ET35p3Z  
dLU=  
-----END CERTIFICATE REQUEST-----
```

이 명령은 다음 예와 유사한 인증서를 콘솔에 출력합니다.

서명된 인증서 예:

```
-----BEGIN CERTIFICATE-----  
MIIDdzCCA1+gAwIBAgIIIGHolbgv5DPowDQYJKoZIhvcNAQELBQAwLjEfMB0GA1UE  
AxMWY2FjaGUtMTY0Zy1zdm0tcm9vdC1jYTELMakGA1UEBhMCVVMwHhcNMjUxMTIx  
MjIxNTU4WhcNMjYxMTIxMjIxNTU4WjAfMR0wGwYDVQQDExRjYWNoZTFnLWRhdGEu  
...  
qs7zhj3ikWE3Gp9s+QijKWxx/0HDd1UuGqy0QZNqNm/M0mqVnokJNk5F4fBFxMiR  
1o63BxL8xGIRdtTCjzb2Gq2Wj7Ec1Uw6CykEkxAcVk+XrRtArGkNtcYdtHfUsKVE  
wswwv0rNydrNnWhJLhS18TW5Tex+OMyTXgk9/3K8kB0mAMrtxxYjt8tm+gztkivf  
J0eo1uDJhaNxqwEZRzFyGaa4k1+56oFzRfTc  
-----END CERTIFICATE-----
```

5. 다음 단계를 위해 인증서를 복사합니다.

6. SVM에 서버 인증서를 설치합니다.

```
certificate install -type server -vserver <svm> -cert-name flexcache-duality
```

7. [인증서에 서명합니다](#)에서 서명된 인증서를 붙여넣습니다.

예:

```
Please enter Certificate: Press <Enter> [twice] when done
-----BEGIN CERTIFICATE-----
MIIDdzCCA1+gAwIBAgIIGHolbgv5DPowDQYJKoZIhvcNAQELBQAwLjEfMB0GA1UE
AxMWY2FjaGUtMTY0Zy1zdm0tcm9vdC1jYTELMAkGA1UEBhMCVVMwHhcNMjUxMTIx
MjIxNTU4WhcNMjYxMTIxMjIxNTU4WjAfMR0wGwYDVQQDEXRjYWNoZTFnLWRhdGEu
bmFzLmxhYjCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAK6wmTTvk7xS
...
qS7zhj3ikWE3Gp9s+QijKWxx/0HDd1UuGqy0QZNqNm/M0mqVnokJNk5F4fBFxMiR
1o63BxL8xGIRdtTCjjb2Gq2Wj7EC1Uw6CykEkxAcVk+XrRtArGkNtcYdtHfUsKVE
wswwv0rNydrNnWhJLhS18TW5Tex+OMyTXgk9/3K8kB0mAMrtxxYjt8tm+gztkivf
J0eo1uDJhaNxqwEZRzFyGaa4k1+56oFzRfTc
-----END CERTIFICATE-----
```

8. [인증서 서명 요청 생성](#)에서 생성된 개인 키를 붙여넣습니다.

예:

```
Please enter Private Key: Press <Enter> [twice] when done
-----BEGIN PRIVATE KEY-----
MIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAQcusJk07508Uh32
9cHI6x+BaRS2w5wrqvzoYlidXtYmdCH3m1DDprBiAyfIwBC0/iU3Xd5NpB7nc1wK
1CI2VEkrXGUGwBtx1K4I1rCTB829Q1aLG AQXV yWnzhQc4t5PW/DsQ8t7o1Z9zEI
W/gaEIajgpXIwGNWZ+weKQK+yoolxC+gy4IUE7WvnEUiezaIdoqzyPhYq5GC4XWF
0johpQugOPe0/w2nVFRWJoFQp3ZP3NZACx8H0qkRB6SjaM243XV2jnuEzx2joXvT
WHHH+IBAQ2JDs7s1TY0I20e49J2Fx2+HvUxDx4BHao7CCHA1+MnmEl+9E38wTaEk
NLsU724ZAgMBAAE CggEABHUy06wxci k5ho3S9Ik1FDZV3JWzsu5gGdLSQOHRd5W+
...
rXGEDDaqp7jQGNXUGlbxO3zcBil1/A9Hc6oalNECgYBKwe3PeZamiwhIHLY9ph7w
djFFCshsPalMuAp2OuKIA nNa916ft9y5kf9tIbskT+t5Dth8bmV9pwe8UZaK5eC4
Svxm19jHT5Qql oDaZVUmMXFKyKoqPDdfvcDk2Eb5gMfIIb0a3TPC/jqqpDn9BzuH
TOO2fuRvRR/G/HUz2yRd+A==
-----END PRIVATE KEY-----
```

9. 서버 인증서의 인증서 체인을 구성하는 인증 기관(CA)의 인증서를 입력하십시오.

이는 서버 인증서를 발급한 CA 인증서부터 시작하여 루트 CA 인증서까지 이어질 수 있습니다.

```
Do you want to continue entering root and/or intermediate certificates  
{y|n}: n
```

You should keep a copy of the private key and the CA-signed digital certificate for future reference.

The installed certificate's CA and serial number for reference:

```
CA: cache-164g-svm-root-ca  
serial: 187A256E0BF90CFA
```

10. SVM 루트 CA의 공개 키를 가져옵니다.

```
security certificate show -vserver <svm> -common-name <root_ca_cn> -ca  
<root_ca_cn> -type root-ca -instance  
  
-----BEGIN CERTIFICATE-----  
MIIDgTCCAmmgAwIBAgIIGHokTnbshKEwDQYJKoZIhvcNAQELBQAwLjEfMB0GA1UE  
AxMWY2FjaGUtMTY0Zy1zdm0tcm9vdC1jYTELMAkGA1UEBhMCVVMwHhcNMjUxMTIx  
MjE1NTIxWhcNMjYxMTIxMjE1NTIxWjAuMR8wHQYDVQQDEXZjYWNoZS0xNjRnLXN2  
bS1yb290LWNhMQswCQYDVQQGEwJVUzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC  
...  
DoOL7vZFFt44xd+rp0DwafhSnLH5HNhdIAfa2JvZW+eJ7rgevH9wmOzyclvaih13  
Ewtb6cz1a/mtESSYRNBMGkIGM/SFCy5v1ROZXcxF96XPbYQN4cW0AYI3AHYBZP0A  
H1NzDR8iml4k9IuKf6BHLFA+VwLTJJZKrdf5Jvjgh0trGAbQGI/Hp2Bjuiopkui+  
n4aa5Rz0JFQopqQddAYnMuvcq10CyNn7S0vF/XLd3fJaprH8kQ==  
-----END CERTIFICATE-----
```



이는 클라이언트가 SVM 루트 CA에서 서명한 인증서를 신뢰하도록 구성하는 데 필요합니다. 공개 키가 콘솔에 출력됩니다. 공개 키를 복사하여 저장하십시오. 이 명령의 값은 [SVM 루트 CA 생성](#)에 입력한 값과 동일합니다.

2단계: S3 서버 구성

1. S3 프로토콜 액세스 활성화:

```
vserver show -vserver <svm> -fields allowed-protocols
```



S3는 기본적으로 SVM 수준에서 허용됩니다.

2. 기존 정책 복제:

```
network interface service-policy clone -vserver <svm> -policy default-data-files -target-vserver <svm> -target-policy <any_name>
```

3. 복제된 정책에 S3를 추가합니다.

```
network interface service-policy add-service -vserver <svm> -policy <any_name> -service data-s3-server
```

4. 데이터 LIF에 새 정책을 추가합니다.

```
network interface modify -vserver <svm> -lif <data_lif> -service-policy duality
```



기존 LIF의 서비스 정책을 수정하면 중단이 발생할 수 있습니다. LIF를 중지했다가 새 서비스에 대한 리스너와 함께 다시 시작해야 합니다. TCP는 이 작업에서 빠르게 복구되어야 하지만 잠재적인 영향을 인지하고 있어야 합니다.

5. SVM에 S3 오브젝트 저장소 서버를 생성합니다.

```
vserver object-store-server create -vserver <svm> -object-store-server <dns_name_of_data_lif> -certificate-name flexcache-duality
```

6. FlexCache 볼륨에서 S3 기능을 활성화합니다.

`flexcache config` 옵션 `'-is-s3-enabled` 을 버킷을 생성하기 전에 `true`로 설정해야 합니다. 또한 `'-is-writeback-enabled` 옵션을 `false`로 설정해야 합니다.

다음 명령은 기존 FlexCache를 수정합니다.

```
flexcache config modify -vserver <svm> -volume <fcache_vol> -is-writeback-enabled false -is-s3-enabled true
```

7. S3 버킷 생성:

```
vserver object-store-server bucket create -vserver <svm> -bucket <bucket_name> -type nas -nas-path <flexcache_junction_path>
```

8. 버킷 정책 생성:

```
vserver object-store-server bucket policy add-statement -vserver <svm>  
-bucket <bucket_name> -effect allow
```

9. S3 사용자 생성:

```
vserver object-store-server user create -user <user> -comment ""
```

출력 예:

```
Vserver: <svm>>  
User: <user>>  
Access Key: WCOT7...Y7D6U  
Secret Key: 6143s...pd__P  
Warning: The secret key won't be displayed again. Save this key for  
future use.
```

10. 루트 사용자의 키 재생성:

```
vserver object-store-server user regenerate-keys -vserver <svm> -user  
root
```

출력 예:

```
Vserver: <svm>>  
User: root  
Access Key: US791...2F1RB  
Secret Key: tgYmn...8_3o2  
Warning: The secret key won't be displayed again. Save this key for  
future use.
```

3단계: 클라이언트 설정

다양한 S3 클라이언트를 사용할 수 있습니다. AWS CLI를 사용하는 것이 좋은 시작점입니다. 자세한 내용은 "[AWS CLI 설치](#)"을 참조하십시오.

FlexCache 볼륨 관리

ONTAP FlexCache 볼륨 감사에 대해 자세히 알아보십시오

ONTAP 9.7부터는 기본 ONTAP 감사 및 FPolicy를 사용한 파일 정책 관리를 통해 FlexCache

관계에서 NFS 파일 액세스 이벤트를 감사할 수 있습니다.

ONTAP 9.14.1부터 NFS 또는 SMB가 있는 FlexCache 볼륨에 대해 FPolicy가 지원됩니다. 이전에는 SMB가 있는 FlexCache 볼륨에 대해 FPolicy가 지원되지 않았습니다.

기본 감사 및 FPolicy는 FlexVol 볼륨에 사용되는 동일한 CLI 명령으로 구성 및 관리됩니다. 그러나 FlexCache 볼륨에는 몇 가지 다른 동작이 있습니다.

- * 기본 감사 *

- FlexCache 볼륨을 감사 로그의 대상으로 사용할 수 없습니다.
- FlexCache 볼륨의 읽기 및 쓰기를 감사하려면 캐시 SVM과 원본 SVM 모두에 대해 감사를 구성해야 합니다.
이는 파일 시스템 작업이 처리되는 곳에서 감사되기 때문입니다. 즉, 캐시 SVM에서 읽기를 감사하고 원본 SVM에서 쓰기를 감사합니다.
- 쓰기 작업의 출처를 추적하기 위해 SVM UUID 및 MSID가 감사 로그에 추가되어 쓰기 작업이 시작된 FlexCache 볼륨을 식별합니다.

- FPolicy * 를 사용합니다

- FlexCache 볼륨에 대한 쓰기가 원본 볼륨에서 커밋되더라도 FPolicy 구성은 캐시 볼륨의 쓰기를 모니터링합니다. 이는 원본 볼륨에서 쓰기를 감사하는 네이티브 감사와는 다릅니다.
- ONTAP는 캐시 및 원본 SVM에 동일한 FPolicy 구성을 요구하지 않지만, 유사한 구성을 두 개 구축하는 것이 좋습니다. 원래 SVM과 같이 구성된 캐시에 대한 새로운 FPolicy 정책을 생성하지만, 새 정책 범위는 캐시 SVM으로 제한됩니다.
- FPolicy 구성의 확장 크기는 20KB(20480바이트)로 제한됩니다. FlexCache 볼륨에서 FPolicy 구성에 사용되는 확장의 크기가 20KB를 초과하면 EMS 메시지가 `nblade.fpolicy.extn.failed` 트리거됩니다.

원본 볼륨의 ONTAP FlexCache 볼륨 속성을 동기화합니다

FlexCache 볼륨의 일부 볼륨 속성은 항상 원본 볼륨의 볼륨 속성과 동기화되어야 합니다. 원본 볼륨에서 속성이 수정된 후 FlexCache 볼륨의 볼륨 속성이 자동으로 동기화되지 않으면 속성을 수동으로 동기화할 수 있습니다.

이 작업에 대해

FlexCache 볼륨의 다음 볼륨 속성은 항상 원본 볼륨의 볼륨 속성과 동기화되어야 합니다.

- 보안 스타일('-security-style')
- 볼륨 이름('-volume-name')
- 최대 디렉토리 크기("-maxdir-size")
- 최소 미리 읽기('-min-readahead')

단계

1. FlexCache 볼륨에서 볼륨 속성을 동기화합니다.

```
'volume FlexCache sync-properties-vserver svm_name-volume flexcache_volume'
```

```
cluster1::> volume flexcache sync-properties -vserver vs1 -volume fc1
```

ONTAP FlexCache 관계 구성을 업데이트합니다

볼륨 이동, 애그리게이트 재배치 또는 스토리지 페일오버와 같은 이벤트가 발생하면 원래 볼륨 및 FlexCache 볼륨의 볼륨 구성 정보가 자동으로 업데이트됩니다. 자동 업데이트가 실패하는 경우 EMS 메시지가 생성되어 FlexCache 관계에 대한 구성을 수동으로 업데이트해야 합니다.

원본 볼륨과 FlexCache 볼륨이 연결 해제 모드에 있는 경우 FlexCache 관계를 수동으로 업데이트하려면 몇 가지 추가 작업을 수행해야 할 수 있습니다.

이 작업에 대해

FlexCache 볼륨의 구성을 업데이트하려면 원본 볼륨에서 명령을 실행해야 합니다. 원본 볼륨의 구성을 업데이트하려면 FlexCache 볼륨에서 명령을 실행해야 합니다.

단계

- FlexCache 관계의 구성을 업데이트합니다.

```
'volume FlexCache config-refresh-peer-vserver peer_svm-peer-volume peer_volume_to_update-peer-endpoint-type [origin|cache]'
```

ONTAP FlexCache 볼륨에서 파일 액세스 시간 업데이트를 활성화합니다

ONTAP 9.11.1부터 FlexCache 볼륨의 '-atime-update' 필드를 활성화하여 파일 액세스 시간 업데이트를 허용할 수 있습니다. 또한 '-atime-update-period' 속성을 사용하여 액세스 시간 업데이트 기간을 설정할 수 있습니다. '-atime-update-period' 속성은 액세스 시간 업데이트가 발생할 수 있는 빈도와 오리진 볼륨으로 전파될 수 있는 시기를 제어합니다.

개요

ONTAP는 읽기, READLINK 및 readdir를 사용하여 읽은 파일 및 디렉터리에 대한 액세스 시간 업데이트를 관리할 수 있는 "-atime-update"라는 볼륨 수준 필드를 제공합니다. aTime은 자주 액세스하지 않는 파일과 디렉토리의 데이터 수명 주기 결정에 사용됩니다. 자주 액세스하지 않는 파일은 결국 아카이브 스토리지로 마이그레이션되며 나중에 테이프로 이동하는 경우가 많습니다.

atime-update 필드는 기존 볼륨과 새로 생성된 FlexCache 볼륨에서 기본적으로 비활성화됩니다. 9.11.1 이전 버전의 ONTAP 릴리스에서 FlexCache 볼륨을 사용하는 경우 원본 볼륨에서 읽기 작업을 수행할 때 캐시가 불필요하게 제거되지 않도록 atime-update 필드를 비활성화해야 합니다. 그러나 대규모 FlexCache 캐시를 사용하는 경우 관리자는 특수한 도구를 사용하여 데이터를 관리하고 핫 데이터가 캐시에 남아 있고 콜드 데이터가 제거되도록 합니다. atime-update가 비활성화된 경우에는 이 기능을 사용할 수 없습니다. 그러나 ONTAP 9.11.1부터 -atime-update-period, 캐시된 데이터를 관리하는 데 필요한 도구를 사용하고 사용할 수 -atime-update 있습니다.

시작하기 전에

- 모든 FlexCache 볼륨에서 ONTAP 9.11.1 이상이 실행되고 있어야 합니다.
- 권한 모드를 사용해야 advanced 합니다.

이 작업에 대해

'-atime-update-period'를 86400초로 설정하면 파일에 대해 수행된 읽기 유사 작업의 수에 관계없이 24시간 간격으로 한 번 이상의 액세스 시간을 업데이트할 수 없습니다.

'-atime-update-period'를 0으로 설정하면 각 읽기 액세스에 대한 메시지가 오리진으로 전송됩니다. 그런 다음 오리진에서 각 FlexCache 볼륨에 atime이 오래되어 성능에 영향을 미친다는 정보를 제공합니다.

단계

1. 권한 모드를 다음과 같이 설정합니다. advanced

세트 프리빌리지 고급

2. 파일 액세스 시간 업데이트를 활성화하고 업데이트 빈도를 설정합니다.

```
volume modify -volume vol_name -vserver <SVM name> -atime-update true -atime-update-period <seconds>
```

다음 예에서는 '-atime-update'를 활성화하고 '-atime-update-period'를 86400초 또는 24시간으로 설정합니다.

```
c1: volume modify -volume origin1 vs1_c1 -atime-update true -atime-update-period 86400
```

3. '-atime-update'가 활성화되어 있는지 확인합니다.

'volume show-volume.vol_name_-fields atime-update, atime-update-period'

```
c1::>*> volume show -volume cache1_origin1 -fields atime-update,atime-update-period
vserver volume          atime-update atime-update-period
-----
vs2_c1    cache1_origin1 true           86400
```

4. 이 설정된 후에는 -atime-update FlexCache 볼륨의 파일을 자동으로 스크러빙할 수 있는지 여부와 스크러빙 간격을 지정할 수 있습니다.

```
volume flexcache config modify -vserver <SVM name> -volume <volume_name> -is-atime-scrub-enabled <true|false> -atime-scrub-period <integer>
```

매개 변수에 대한 자세한 -is-atime-scrub-enabled 내용은 "[ONTAP 명령 참조입니다](#)"를 참조하십시오.

ONTAP FlexCache 볼륨에 대한 글로벌 파일 잠금을 사용합니다

ONTAP 9.10.1부터 전역 파일 잠금을 적용하여 모든 관련 캐시 파일에 대한 읽기를 방지할 수 있습니다.

글로벌 파일 잠금을 사용하면 모든 FlexCache 볼륨이 온라인 상태가 될 때까지 원본 볼륨의 수정 사항이 일시

중단됩니다. FlexCache 볼륨이 오프라인 상태일 때 일시 중지 및 수정 시간 초과로 인해 캐시와 원본 간 연결의 안정성을 제어할 수 있는 경우에만 전역 파일 잠금을 활성화해야 합니다.

시작하기 전에

- 글로벌 파일 잠금에서는 오리진이 포함된 클러스터와 모든 연결된 캐시에서 ONTAP 9.9.1 이상을 실행해야 합니다. 글로벌 파일 잠금은 새 FlexCache 볼륨 또는 기존 볼륨에 대해 활성화할 수 있습니다. 명령은 하나의 볼륨에서 실행할 수 있으며 모든 관련 FlexCache 볼륨에 적용됩니다.
- 전역 파일 잠금을 설정하려면 고급 권한 수준이어야 합니다.
- 9.9.1 이전 버전의 ONTAP로 되돌리는 경우 원본 및 연결된 캐시에서 글로벌 파일 잠금을 먼저 해제해야 합니다. 비활성화하려면 원본 볼륨에서 다음을 실행합니다. `volume flexcache prepare-to-downgrade -disable-feature-set 9.10.0`
- 글로벌 파일 잠금을 설정하는 프로세스는 오리진에 기존 캐시가 있는지 여부에 따라 달라집니다.
 - [enable-gfl-new]
 - [enable-gfl-existing]

새 FlexCache 볼륨에서 글로벌 파일 잠금을 설정합니다

단계

1. TRUE로 설정된 '-is-global-file-locking'으로 FlexCache 볼륨 생성:

```
'Volume FlexCache create volume_volume_name_-is-global-file-locking-enabled true'
```



기본값은 '-is-global-file-locking'입니다. 이후에 볼륨에 대해 볼륨 FlexCache 생성 명령을 실행할 때는 "true"로 설정된 '-is-global-file-locking enabled'로 전달해야 합니다.

기존 FlexCache 볼륨에서 글로벌 파일 잠금을 설정합니다

단계

1. 글로벌 파일 잠금은 원본 볼륨에서 설정해야 합니다.
2. 오리진에는 다른 기존 관계(예: SnapMirror)를 포함할 수 없습니다. 기존의 모든 관계는 분리되어야 합니다. 명령을 실행할 때 모든 캐시 및 볼륨이 연결되어 있어야 합니다. 연결 상태를 확인하려면 다음을 실행합니다.

볼륨 FlexCache 연결 상태 표시

나열된 모든 볼륨의 상태는 `connected`로 표시되어야 합니다. 자세한 내용은 ["FlexCache 관계의 상태를 봅니다"](#) 또는 ["원점에서 FlexCache 볼륨의 속성을 동기화합니다"](#).

3. 캐시에 글로벌 파일 잠금 설정:

```
'volume FlexCache origin config show/modify -volume_volume_name_-is-global-file-locking-enabled true'
```

관련 정보

- ["ONTAP 명령 참조입니다"](#)

ONTAP FlexCache 볼륨을 미리 채웁니다

FlexCache 볼륨을 미리 채워 캐시된 데이터에 액세스하는 데 걸리는 시간을 줄일 수 있습니다.

시작하기 전에

- 고급 권한 레벨의 클러스터 관리자여야 합니다
- Prepopulation에 대해 전달하는 경로가 존재해야 하거나 미리 채우기 작업이 실패합니다.

이 작업에 대해

- 미리 채우기는 파일만을 읽고 디렉토리를 통과합니다
- 를 클릭합니다 -isRecursion Flag 미리 채우기에 전달된 전체 디렉터리 목록에 적용됩니다

단계

1. FlexCache 볼륨을 미리 채웁니다.

```
volume flexcache prepopulate -cache-vserver vserver_name -cache-volume -path
-list path_list -isRecursion true|false
```

- 를 클릭합니다 -path-list 매개 변수는 원래 루트 디렉터리부터 시작하여 미리 채울 상대 디렉터리 경로를 나타냅니다. 예를 들어, 오리진 루트 디렉토리의 이름이 /origin이고 디렉토리 /origin/dir1 및 /origin/dir2가 포함된 경우 다음과 같이 경로 목록을 지정할 수 있습니다. -path-list dir1, dir2 또는 -path-list /dir1, /dir2.
- 의 기본값입니다 -isRecursion 매개 변수는 True입니다.

이 예에서는 단일 디렉토리 경로를 미리 채웁니다.

```
cluster1::*: flexcache prepopulate start -cache-vserver vs2 -cache
-volume fg_cachevol_1 -path-list /dir1
(volume flexcache prepopulate start)
[JobId 207]: FlexCache prepopulate job queued.
```

다음 예제에서는 여러 디렉터리의 파일을 미리 채웁니다.

```
cluster1::*: flexcache prepopulate start -cache-vserver vs2 -cache
-volume fg_cachevol_1 -path-list /dir1,/dir2,/dir3,/dir4
(volume flexcache prepopulate start)
[JobId 208]: FlexCache prepopulate job queued.
```

다음 예제에서는 단일 파일을 미리 채웁니다.

```
cluster1::*> flexcache prepopulate start -cache-vserver vs2 -cache
-volume fg_cachevol_1 -path-list /dir1/file1.txt
(volume flexcache prepopulate start)
[JobId 209]: FlexCache prepopulate job queued.
```

이 예제에서는 오리진에서 가져온 모든 파일을 미리 채웁니다.

```
cluster1::*> flexcache prepopulate start -cache-vserver vs2 -cache
-volume fg_cachevol_1 -path-list / -isRecursion true
(volume flexcache prepopulate start)
[JobId 210]: FlexCache prepopulate job queued.
```

이 예제에는 선행 모집단의 잘못된 경로가 포함되어 있습니다.

```
cluster1::*> flexcache prepopulate start -cache-volume
vol_cache2_vs3_c2_vol_origin1_vs1_c1 -cache-vserver vs3_c2 -path-list
/dirl, dir5, dir6
(volume flexcache prepopulate start)

Error: command failed: Path(s) "dir5, dir6" does not exist in origin
volume
"vol_origin1_vs1_c1" in Vserver "vs1_c1".
```

2. 읽은 파일 수 표시:

job show-id job_ID-ins를 선택합니다

관련 정보

- ["작업 표시"](#)

ONTAP FlexCache 관계 삭제

더 이상 FlexCache 볼륨이 필요하지 않은 경우 FlexCache 관계 및 FlexCache 볼륨을 삭제할 수 있습니다.

시작하기 전에

FlexCache 쓰기 되돌림 기능이 활성화된 경우 FlexCache 볼륨을 삭제하기 전에 먼저 해당 기능을 비활성화해야 합니다. ["FlexCache 볼륨에서 write-back을 비활성화합니다"](#)을 참조하십시오.

단계

1. FlexCache 볼륨이 있는 클러스터에서 FlexCache 볼륨을 오프라인으로 설정합니다.

```
'volume offline-vserver svm_name-volume volume_name'
```

2. FlexCache 볼륨을 삭제합니다.

```
'볼륨 FlexCache 삭제 - vserver svm_name - volume volume volume_name'
```

FlexCache 관계 세부 정보가 원본 볼륨 및 FlexCache 볼륨에서 제거됩니다.

핫스팟 교정을 위한 FlexCache

ONTAP FlexCache 볼륨으로 고성능 컴퓨팅 워크로드의 핫 스팟 해결

애니메이션 렌더링 또는 EDA와 같은 여러 고성능 컴퓨팅 워크로드에서 흔히 발생하는 문제는 주목을 받고 있습니다. 핫스팟팅은 클러스터 또는 네트워크의 특정 부분이 다른 영역에 비해 상당한 부하를 경험하여 해당 위치에 집중된 과도한 데이터 트래픽으로 인해 성능 병목 현상이 발생하고 전반적인 효율성이 감소되는 상황을 말합니다. 예를 들어, 파일 또는 여러 파일이 실행 중인 작업에 대한 수요가 많아 해당 파일에 대한 요청을 처리하는 데 사용되는 CPU에 병목 현상이 발생합니다(볼륨 선호도를 통해). FlexCache는 이 병목 현상을 완화할 수 있지만 올바르게 설정해야 합니다.

이 문서에서는 핫 스팟팅을 개선하기 위해 FlexCache를 설정하는 방법에 대해 설명합니다.



2024년 7월부터 이전에 PDF로 게시된 기술 보고서의 콘텐츠가 ONTAP 제품 문서와 통합되었습니다. 이 ONTAP 핫스팟 개선 기술 보고서 내용은 발행일 현재 완전히 새로 작성되었으며 이전 형식은 아직 작성되지 않았습니다.

주요 개념

핫스팟 개선 계획을 세울 때는 이러한 필수 개념을 이해하는 것이 중요합니다.

- * 고밀도 FlexCache(HDF) *: 캐시 용량 요구 사항이 허용하는 한 몇 개의 노드로 확장되도록 압축된 FlexCache
- * HDF Array(HDFA) *: 클러스터 전체에 분산된 동일한 오리진의 캐시인 HDFS 그룹입니다
- * SVM 간 HDFA *: 서버 가상 머신(SVM)당 HDFA의 HDF 1개
- * 내부 SVM HDFA *: HDFA의 모든 HDFS가 하나의 SVM에 있음
- * 동서 트래픽 *: 클러스터 백엔드 트래픽은 간접 데이터 액세스로부터 생성됩니다

다음 단계

- "고밀도 FlexCache를 사용하여 핫 스팟을 개선하는 방법을 이해합니다"
- "FlexCache 어레이 밀도를 결정합니다"
- "HDFS의 밀도를 결정하고 SVM 간 HDFA 및 SVM 내 HDFA와 함께 NFS를 사용하여 HDFS에 액세스할지 여부를 결정합니다"
- "HDFA 및 데이터 LIF를 구성하여 ONTAP 구성에서 클러스터 내 캐싱을 사용할 경우의 이점을 실현하십시오"
- "클라이언트 구성을 사용하여 ONTAP NAS 연결을 분산하도록 클라이언트를 구성하는 방법에 대해 알아봅니다"

ONTAP FlexCache 핫스팟 개선 솔루션 설계

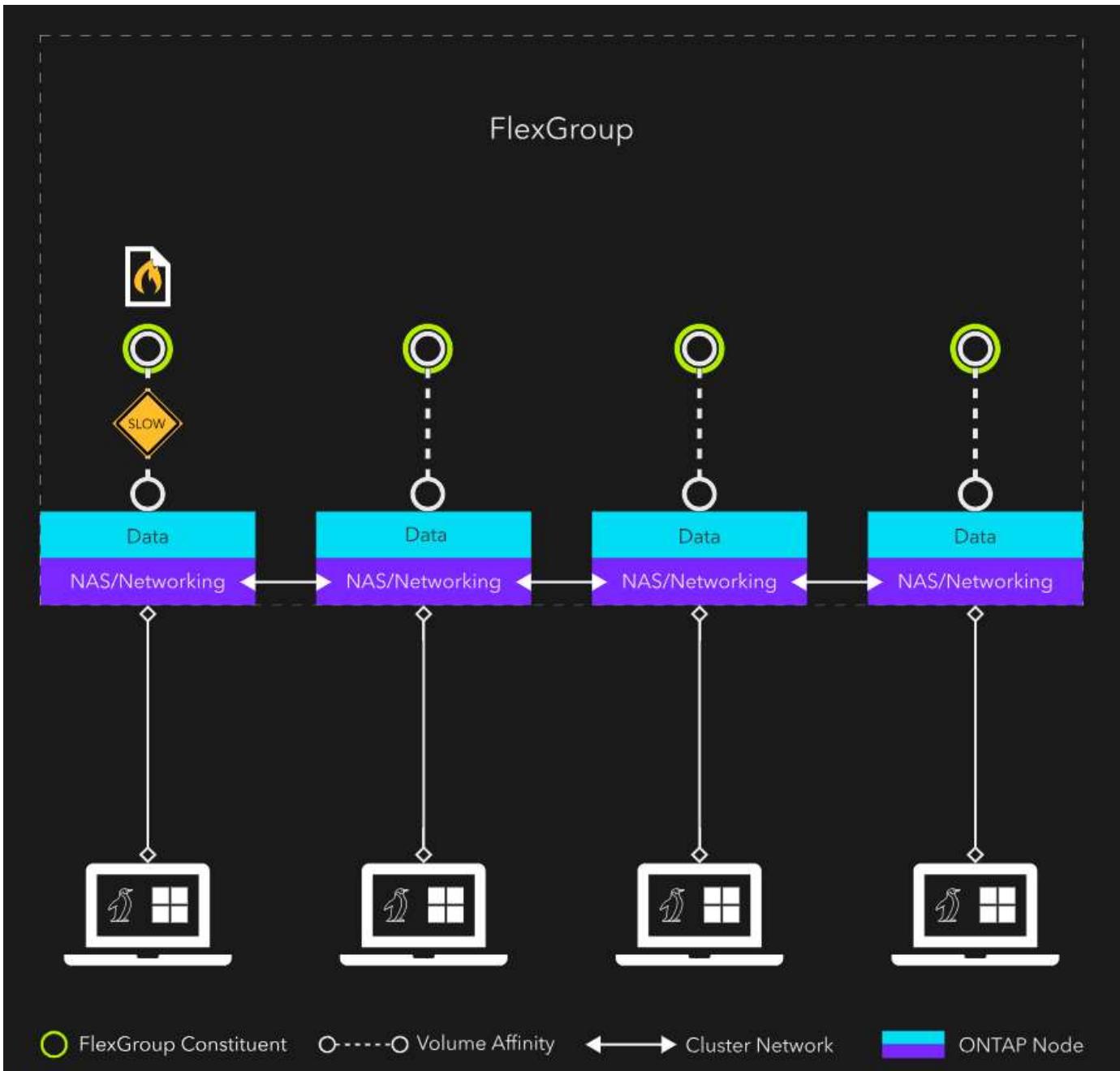
핫스팟을 해결하려면 병목 현상의 근본 원인, 자동 프로비저닝 FlexCache가 충분하지 않은 이유, FlexCache 솔루션을 효과적으로 설계하는 데 필요한 기술 세부 사항을 살펴봅니다. 고밀도 FlexCache 스토리지(HDFA)를 이해하고 구현하면 성능을 최적화하고 수요가 많은 작업 부하에서 병목 현상을 제거할 수 있습니다.

병목 현상 이해

다음은 [이미지](#)일반적인 단일 파일 핫 스팟팅 시나리오를 보여 줍니다. 이 볼륨은 노드당 단일 구성요소가 있는 FlexGroup이며 파일은 노드 1에 상주합니다.

모든 NAS 클라이언트의 네트워크 연결을 클러스터의 서로 다른 노드에 분산시키는 경우 핫 파일이 상주하는 볼륨 선호도를 지원하는 CPU의 병목 현상이 발생합니다. 또한 파일이 상주하는 위치가 아닌 노드에 연결된 클라이언트에서 수신되는 통화에 클러스터 네트워크 트래픽(동부-서부 트래픽)을 도입합니다. 동서부의 트래픽 오버헤드는 일반적으로 작지만 고성능 컴퓨팅 워크로드의 경우에는 비트 수가 거의 없습니다.

그림 1: **FlexGroup** 단일 파일 핫스팟 시나리오

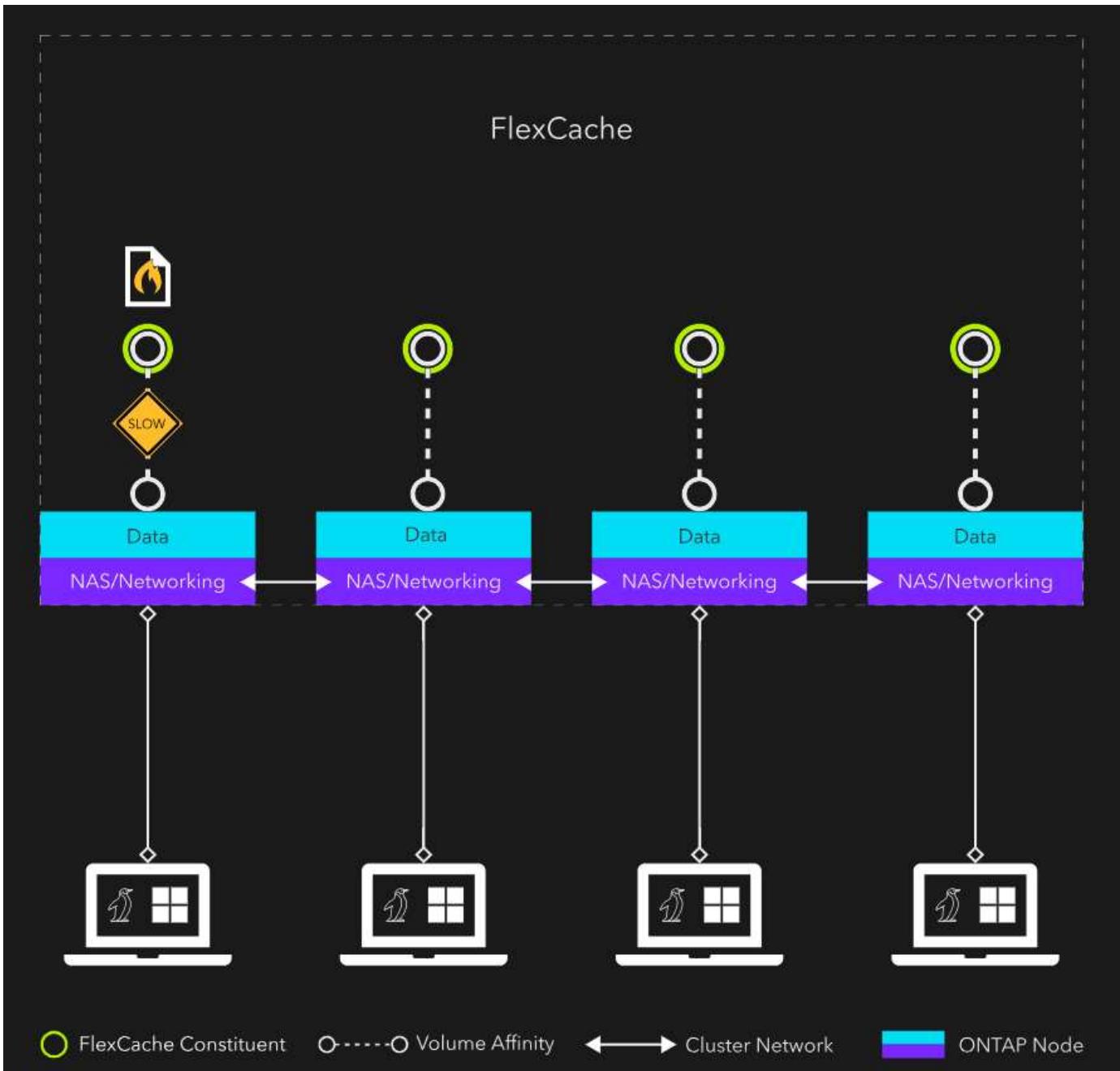


자동 프로비저닝된 **FlexCache**가 해답이 아닌 이유

핫스팟을 해결하려면 CPU 병목 현상을 제거하고 동서 트래픽도 제거합니다. FlexCache가 제대로 설정되어 있으면 도움이 될 수 있습니다.

다음 예에서 FlexCache System Manager, NetApp Console 또는 기본 CLI 인수를 통해 자동으로 프로비저닝됩니다. 그림 1 그리고 그림 2 처음에는 똑같아 보입니다. 둘 다 4노드, 단일 구성 NAS 컨테이너입니다. 유일한 차이점은 그림 1의 NAS 컨테이너가 FlexGroup이고, 그림 2의 NAS 컨테이너가 FlexCache라는 것입니다. 각 그림은 동일한 병목 현상을 보여줍니다. 즉, 핫 파일에 대한 액세스를 제공하는 볼륨 친화성을 위한 노드 1의 CPU와 지연 시간에 영향을 미치는 동서 트래픽입니다. 자동 프로비저닝된 FlexCache 병목 현상을 제거하지 못했습니다.

그림 2: 자동 프로비저닝 **FlexCache** 시나리오



FlexCache의 해부 구조

핫스팟 교정을 위한 FlexCache를 효과적으로 설계하려면 FlexCache에 대한 몇 가지 기술 세부 사항을 이해해야 합니다.

FlexCache는 항상 스파스 FlexGroup입니다. FlexGroup은 여러 개의 FlexVol로 구성됩니다. 이러한 FlexVol을 FlexGroup 구성요소라고 합니다. 기본 FlexGroup 레이아웃의 경우, 클러스터의 노드당 구성요소가 하나 이상 있습니다. 구성 요소는 추상화 계층 아래에 "함께 결합"되어 단일 대형 NAS 컨테이너로 클라이언트에 제공됩니다. 파일이 FlexGroup에 기록되면 수집 휴리스틱에서 파일이 저장될 구성요소를 결정합니다. 클라이언트의 NAS 연결을 포함하는 구성요소이거나 다른 노드일 수 있습니다. 모든 것이 추상화 계층 아래에서 작동하고 클라이언트에게 보이지 않기 때문에 위치는 관련이 없습니다.

FlexGroup에 대한 이해도를 FlexCache에 적용해 보겠습니다. FlexCache는 FlexGroup을 기반으로 구축되기 때문에에 설명된 대로 클러스터의 모든 노드에 대해 단일 FlexCache가 기본적으로 제공됩니다.[그림 1](#) 대부분의 경우 이것은 매우 유용합니다. 클러스터의 모든 리소스를 활용하고 있습니다.

하지만 핫 파일을 수정하는 경우에는 단일 파일 및 동서 트래픽의 CPU와 같은 두 가지 병목 현상 때문에 이 방법이 적합하지 않습니다. 핫 파일의 모든 노드에 구성 요소가 있는 FlexCache를 생성하는 경우 해당 파일은 여전히 구성요소 중 하나에 상주합니다. 즉, 핫 파일에 대한 모든 액세스를 서비스하는 CPU가 하나 있습니다. 또한 핫 파일에 도달하는 데 필요한 동서 트래픽의 양을 제한해야 합니다.

이 솔루션은 고밀도 FlexCaches의 어레이입니다.

고밀도 FlexCache의 해부 구조

고집적 HDF(FlexCache)는 캐시된 데이터의 용량 요구 사항이 허용하는 만큼 소수의 노드에 구성 요소가 있습니다. 목표는 단일 노드에서 캐시를 활성화하는 것입니다. 용량 요구사항에 따라 그렇게 할 수 없는 경우 몇 개의 노드에만 구성요소를 사용할 수 있습니다.

예를 들어, 24노드 클러스터에는 3개의 고밀도 FlexCh가 있을 수 있습니다.

- 노드 1에서 8까지 확장되는 경로입니다
- 노드 9에서 16까지 확장되는 초입니다
- 노드 17-24에 걸쳐 3분의 1을 제공합니다

이 세 개의 HDFS는 하나의 HDFA(High-Density FlexCache Array)를 구성합니다. 각 HDF 내에 파일이 균등하게 배포되면 클라이언트가 요청한 파일이 프런트엔드 NAS 접속에 로컬로 상주할 확률은 8%입니다. 각각 2개의 노드만 사용하는 12개의 HDFS를 사용하는 경우 파일이 로컬일 가능성이 50% 높습니다. HDF를 단일 노드로 축소하여 24개 노드로 만들 수 있다면 파일이 로컬임을 보증할 수 있습니다.

이 구성은 모든 동서 트래픽을 제거하며 가장 중요한 것은 핫 파일에 액세스하기 위한 24개의 CPU/볼륨 선호도를 제공합니다.

다음 단계

["FlexCache 어레이 밀도를 결정합니다"](#)

관련 정보

["FlexGroup 및 TR에 대한 설명서"](#)

ONTAP FlexCache 밀도 결정

첫 번째 핫스팟 개선 설계 결정은 FlexCache 밀도를 파악하는 것입니다. 다음 예는 4노드 클러스터입니다. 파일 수가 각 HDF의 모든 구성 요소에 균등하게 분포되어 있다고 가정합니다. 또한 모든 노드에 걸쳐 프런트엔드 NAS 연결이 고르게 분포되어 있다고 가정합니다.

이러한 예만 사용할 수 있는 구성은 아니지만 공간 요구 사항과 사용 가능한 리소스가 허용하는 한 많은 HDFS를 만드는 기본 설계 원칙을 이해해야 합니다.



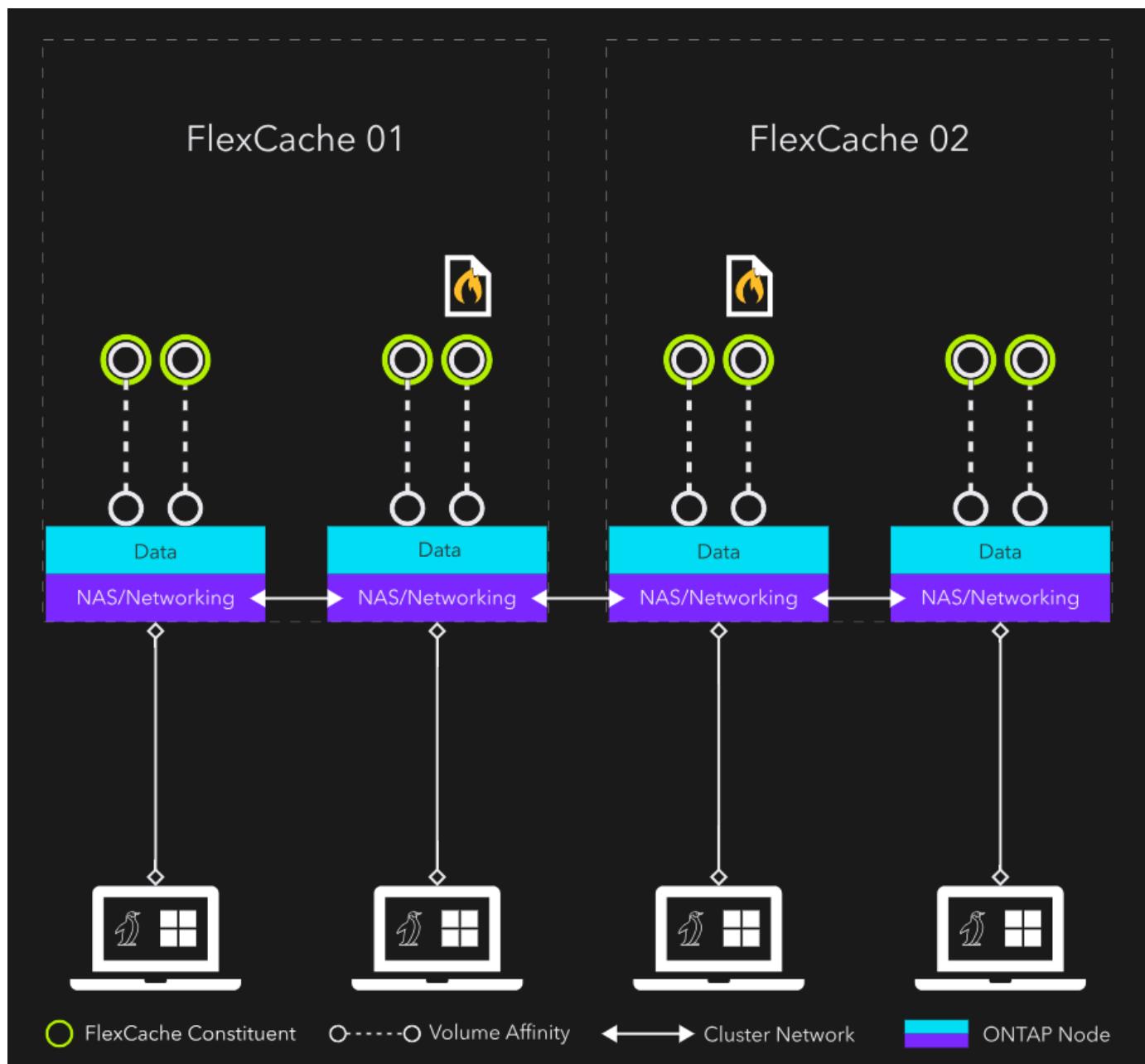
HDFA는 다음 구문을 사용하여 표현됩니다. $HDFs \times HDFA \times nodes \times HDF \times constituents \times node \times HDF$

2x2x2 HDFA 구성

[그림 1](#) 은 2x2x2x2 HDFA 구성의 예입니다. 두 개의 HDFS, 각각 두 개의 노드를 스파닝하고 각 노드는 두 개의 구성 볼륨을 포함합니다. 이 예에서 각 클라이언트는 핫 파일에 직접 액세스할 확률이 50%입니다. 네 고객 중 두 명은 동서

트래픽을 가지고 있습니다. 중요한 것은 이제 두 개의 HDFS가 존재한다는 것입니다. 즉, 핫 파일의 두 가지 캐시를 의미합니다. 현재 핫 파일에 대한 액세스를 처리하는 CPU/볼륨 선호도가 2개 있습니다.

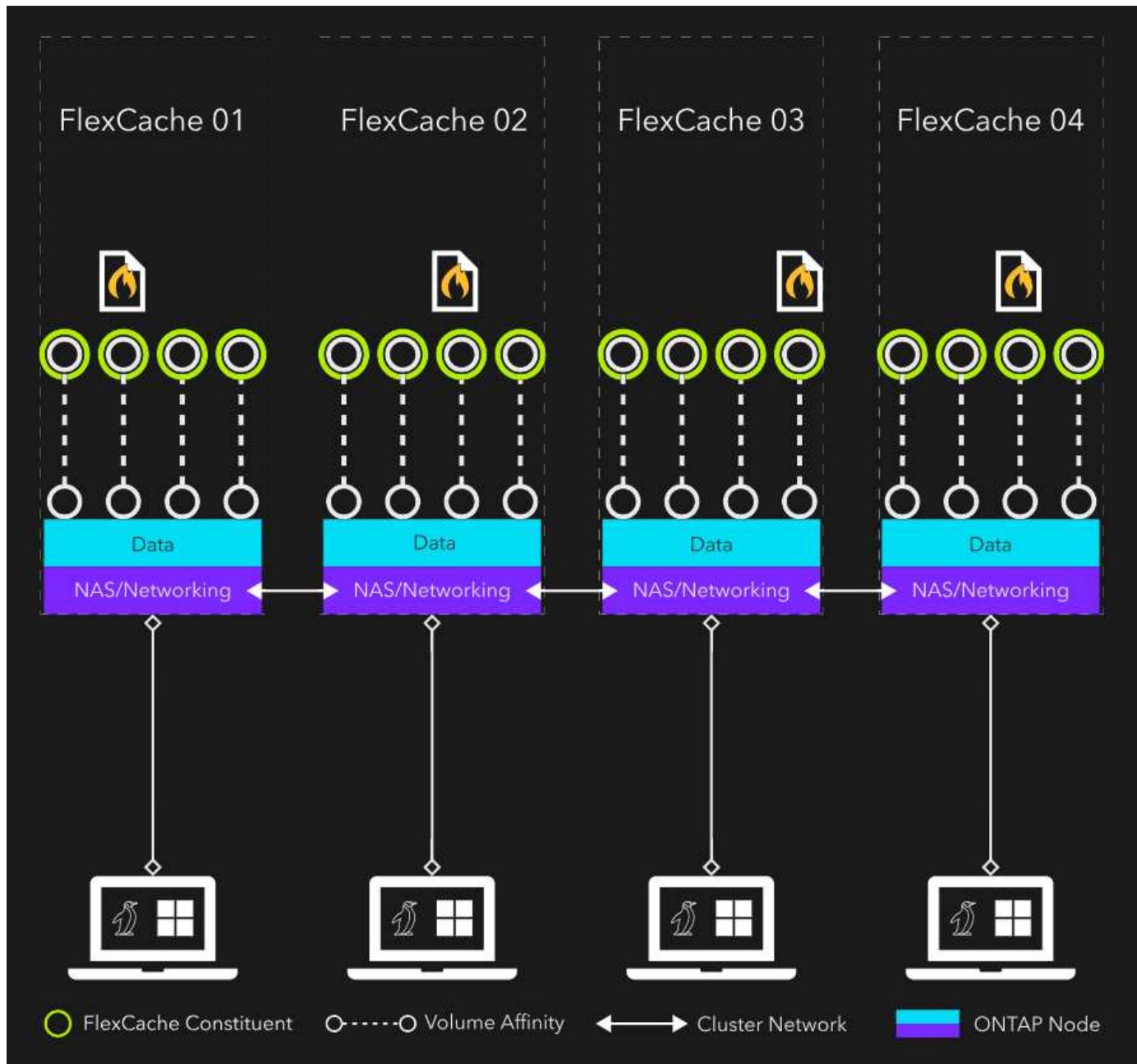
그림 1: 2x2x2 HDFA 구성



4x1x4 HDFA 구성

그림 2 최적의 구성을 나타냅니다. 4x1x4 HDFA 구성의 한 예로 4개의 HDFS가 단일 노드에 포함되고 각 노드는 4개의 구성 요소를 포함합니다. 이 예에서는 각 클라이언트가 핫 파일의 캐시에 직접 액세스할 수 있도록 보장합니다. 4개의 다른 노드에 4개의 캐시된 파일이 있으므로 4개의 서로 다른 CPU/볼륨 선호도가 핫 파일에 대한 서비스 액세스를 지원합니다. 또한 동서 트래픽이 생성되지 않습니다.

그림 2: 4x1x4 HDFA 구성



다음 단계

HDFS의 집적도를 결정한 후 NFS를 사용하여 HDFS에 액세스할 경우 다른 설계 결정을 내려야 "SVM 간 HDFA 및 SVM 내 HDFA"합니다.

ONTAP SVM 간 또는 SVM 내 HDFA 옵션을 결정합니다

HDFS의 밀도를 확인한 후 NFS를 사용하여 HDFS에 액세스할지 여부를 결정하고 SVM 간 HDFA 및 SVM 내 HDFA 옵션에 대해 알아봅니다.



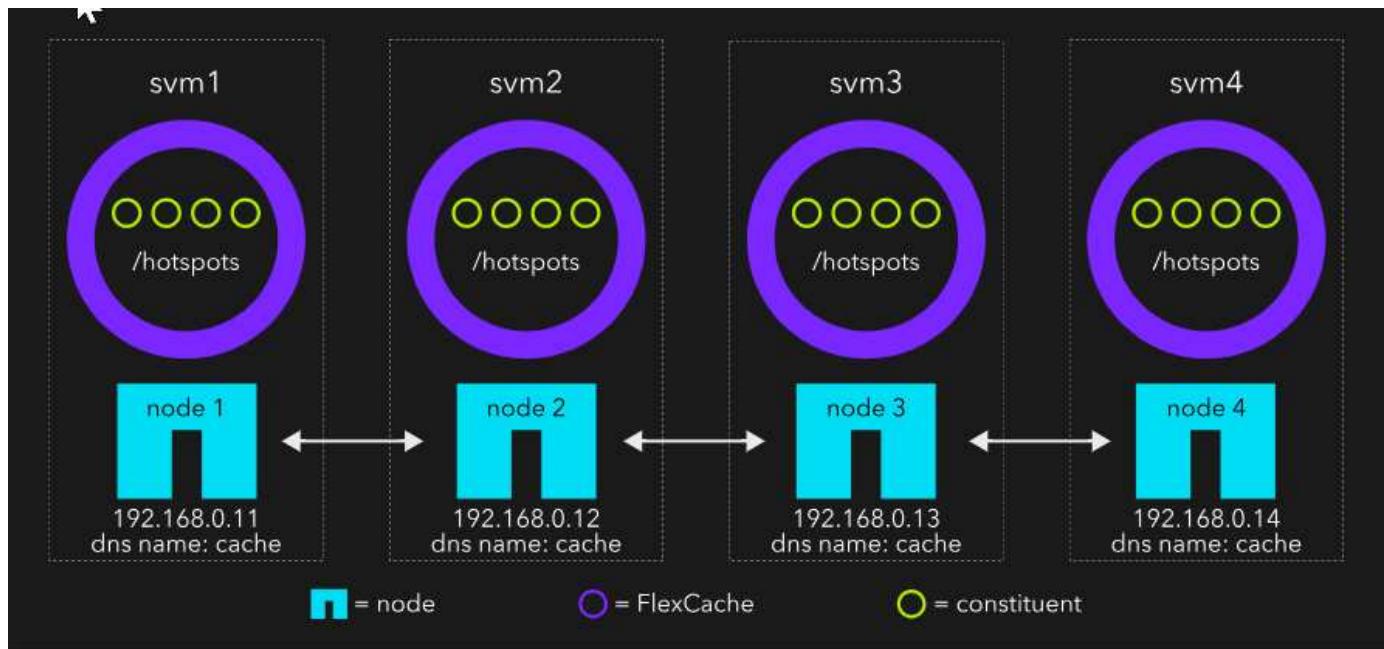
SMB 클라이언트만 HDFS에 액세스하는 경우 모든 HDFS를 단일 SVM에 생성해야 합니다. DFS 대상을 로드 밸런싱에 사용하는 방법은 Windows 클라이언트 구성과 함께 참조하십시오.

SVM 간 HDFA 구축

SVM 간 HDFA는 HDF에 대해 SVM을 생성해야 합니다. 따라서 HDFA 내의 모든 HDFS가 동일한 접합 경로를 갖게 되므로 클라이언트 측에서 보다 쉽게 구성할 수 있습니다.

이 [그림 1](#) 예에서 각 HDF는 자체 SVM에 있습니다. 이는 SVM 간 HDFA 구축입니다. 각 HDF에는 /핫스팟의 접합 경로가 있습니다. 또한 모든 IP에는 호스트 이름 캐시의 DNS A 레코드가 있습니다. 이 구성은 DNS 라운드 로빈을 활용하여 다양한 HDFS에 대한 로드 밸런싱 마운트를 수행합니다.

그림 1: 4x1x4 SVM 간 HDFA 구성

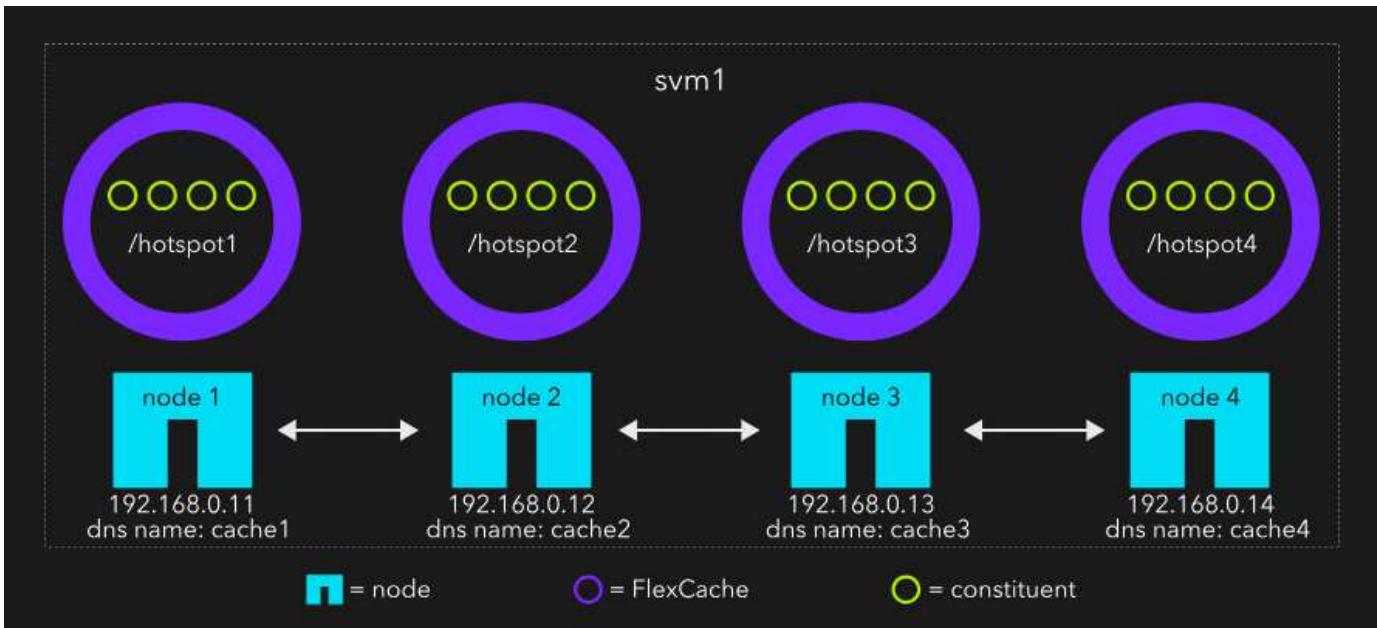


SVM HDFA 내부 구축

SVM 내부의 경우 각 HDF에 고유한 접합 경로가 필요하지만 모든 HDFS는 하나의 SVM에 있습니다. ONTAP에서는 하나의 SVM만 필요하기 때문에 이 설정은 더 쉬울 수 있지만 Linux 측면에서는 고급 구성이 필요하며 ONTAP에 데이터 LIF가 배치되어야 `autofs` 합니다.

이 [그림 2](#) 예에서 모든 HDF는 동일한 SVM에 있습니다. 이는 SVM 내 HDFA 구축이며 접합 경로가 고유해야 합니다. 로드 밸런싱이 제대로 작동하려면 각 IP에 대해 고유한 DNS 이름을 생성하고 호스트 이름이 확인되는 데이터 LIF를 HDF가 상주하는 노드에만 배치해야 합니다. 또한 에서 설명한 대로 여러 항목으로 를 구성해야 `'autofs' "Linux 클라이언트 구성"` 합니다.

그림 2: 4x1x4 내부 SVM HDFA 구성



다음 단계

이제 HDFA를 구축하는 방법에 대해 알아보았습니다. "["HDFA를 배포하고 클라이언트가 분산 방식으로 액세스할 수 있도록 구성합니다"](#)"

HDFA 및 ONTAP 데이터 LIF를 구성합니다

이 핫스팟 교정 솔루션의 이점을 실현하려면 HDFA 및 데이터 LIF를 적절히 구성해야 합니다. 이 솔루션은 동일한 클러스터에서 오리진 및 HDFA와 함께 클러스터 내 캐싱을 사용합니다.

다음은 두 가지 HDFA 샘플 구성입니다.

- 2x2x2 SVM 간 HDFA
- 4x1x4 내부 SVM HDFA

이 작업에 대해

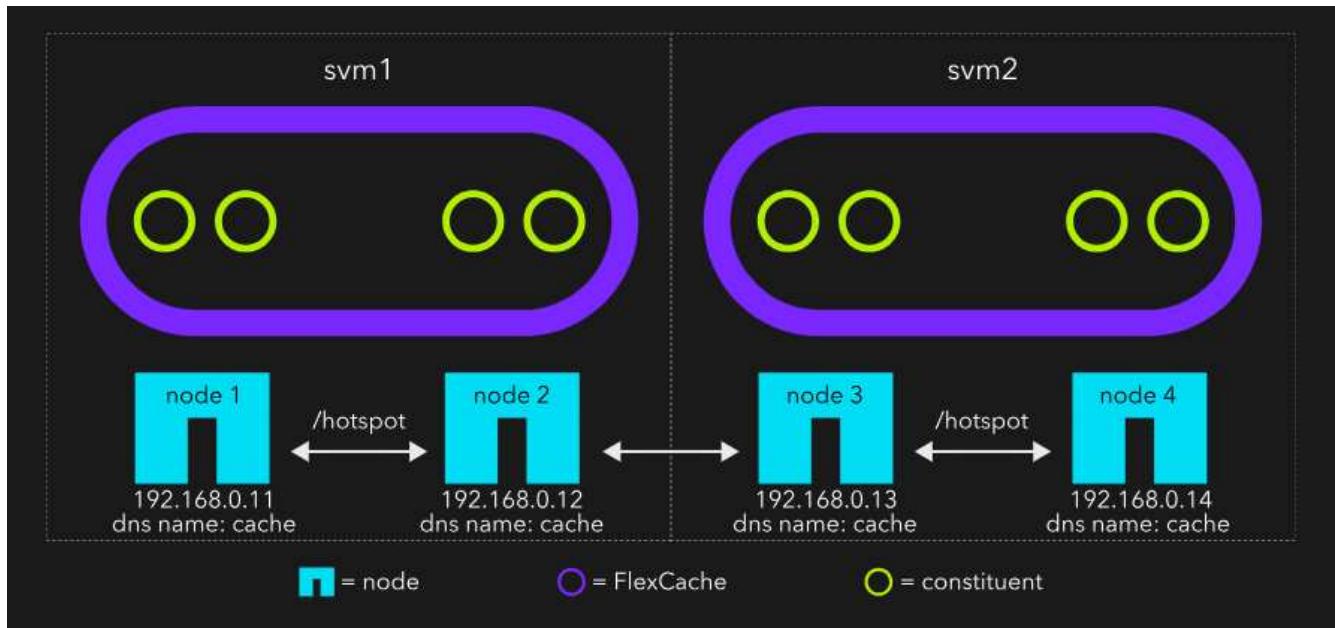
ONTAP CLI를 사용하여 이 고급 구성을 수행합니다. 명령에서 두 가지 구성을 사용해야 `flexcache create` 하며 한 가지 구성을 구성하지 않도록 해야 합니다.

- `-aggr-list`: HDFA를 제한하려는 노드의 노드 또는 하위 집합에 상주하는 애그리게이트 또는 애그리게이트 목록을 제공합니다.
- `-aggr-list-multiplier`: 옵션에 나열된 애그리게이트당 생성할 구성 요소의 수를 `aggr-list` 결정합니다. 두 개의 애그리게이트가 나열되고 이 값을 2로 설정하면 2개의 구성요소가 됩니다. NetApp에서는 애그리게이트당 최대 8개의 구성요소를 사용할 것을 권장하지만 16개로도 충분합니다.
- `-auto-provision-as`: 템 아웃 경우, CLI는 자동 채우기를 시도하고 값을 `'flexgroup'` 설정합니다. 이 설정이 구성되어 있지 않은지 확인합니다. 이 메시지가 나타나면 삭제합니다.

2x2x2 Inter-SVM HDFA 구성을 생성합니다

1. 그림 1과 같이 2x2x2x2 SVM 간 HDFA 구성을 지원하려면 준비 시트를 작성합니다.

그림 1: 2x2x2 Inter-SVM HDFA 레이아웃



SVM	HDFS당 노드 수	애그리게이트	노드당 구성 요소	접합 경로	데이터 LIF IP
svm1를 참조하십시오	node1, node2	aggr1, aggr2	2	/핫스팟	192.168.0.11, 192.168.0.12
svm2를 참조하십시오	node3, node4	aggr3, aggr4	2	/핫스팟	192.168.0.13, 192.168.0.14

2. HDFS를 생성합니다. 준비 시트의 각 행에 대해 다음 명령을 두 번 실행합니다. 두 번째 반복의 및 aggr-list 값을 조정해야 vserver 합니다.

```
cache::> flexcache create -vserver svm1 -volume hotspot -aggr-list
aggr1,aggr2 -aggr-list-multiplier 2 -origin-volume <origin_vol> -origin
-vserver <origin_svm> -size <size> -junction-path /hotspot
```

3. 데이터 LIF를 생성합니다. 명령을 4회 실행하여 준비 시트에 나열된 노드에서 SVM당 데이터 LIF 두 개를 생성합니다. 각 반복에 맞게 값을 적절히 조정해야 합니다.

```
cache::> net int create -vserver svm1 -home-port e0a -home-node node1
-address 192.168.0.11 -netmask-length 24
```

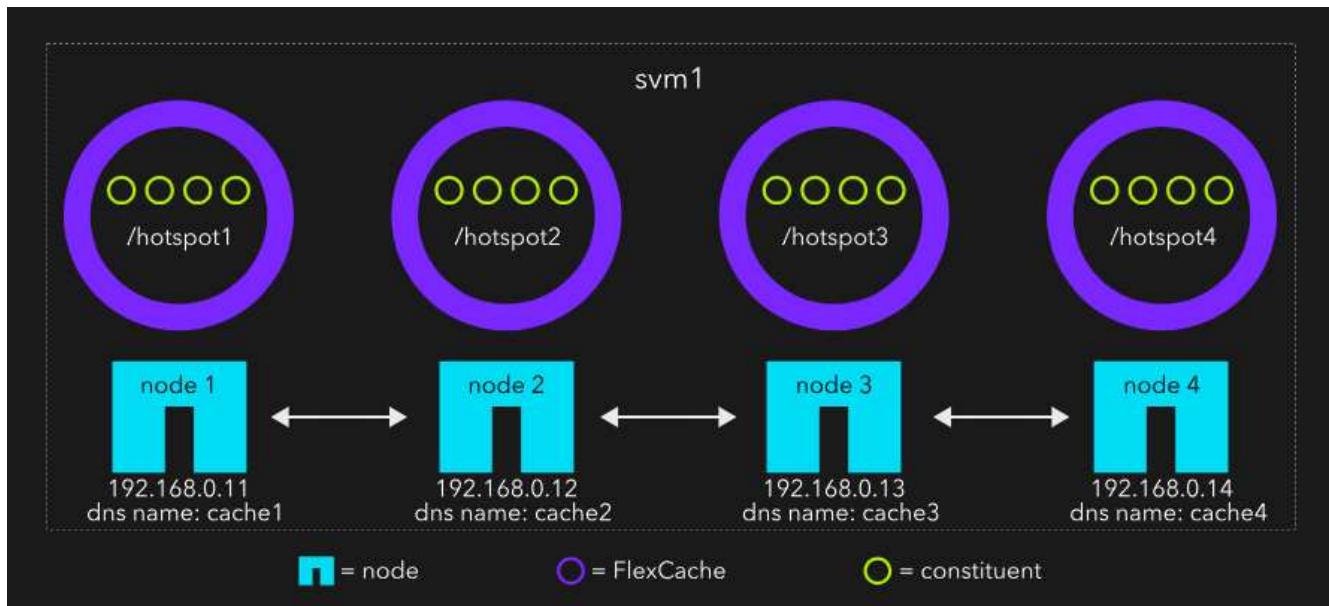
다음 단계

이제 HDFA를 적절히 활용할 수 있도록 클라이언트를 구성해야 합니다. 을 "[클라이언트 구성](#)"참조하십시오.

4x1x4 내부 SVM HDFA를 생성합니다

1. 그림 2와 같이 4x1x4 SVM 간 HDFA 구성을 지원하려면 준비 시트를 작성합니다.

그림 2: 4x1x4 내부 SVM HDFA 레이아웃



SVM	HDFS당 노드 수	애그리게이트	노드당 구성 요소	접합 경로	데이터 LIF IP
svm1을 참조하십시오	노드1	aggr1를 참조하십시오	4	/hotspot1 을 참조하십시오	192.168.0.11
svm1을 참조하십시오	노드2	aggr2를 참조하십시오	4	/hotspot2 을 참조하십시오	192.168.0.12
svm1을 참조하십시오	node3을 참조하십시오	aggr3를 참조하십시오	4	/hotspot3 을 참조하십시오	192.168.0.13
svm1을 참조하십시오	node4을 참조하십시오	aggr4를 참조하십시오	4	/hotspot4 을 참조하십시오	192.168.0.14

2. HDFS를 생성합니다. 준비 시트의 각 행에 대해 다음 명령을 네 번 실행합니다. 각 반복에 대한 및 junction-path 값을 조정해야 aggr-list 합니다.

```
cache::> flexcache create -vserver svm1 -volume hotspot1 -aggr-list
aggr1 -aggr-list-multiplier 4 -origin-volume <origin_vol> -origin
-vserver <origin_svm> -size <size> -junction-path /hotspot1
```

3. 데이터 LIF를 생성합니다. 명령을 4회 실행하여 SVM에 총 4개의 데이터 LIF를 생성합니다. 노드당 데이터 LIF는 하나만 있어야 합니다. 각 반복에 맞게 값을 적절히 조정해야 합니다.

```
cache::> net int create -vserver svm1 -home-port e0a -home-node node1
-address 192.168.0.11 -netmask-length 24
```

다음 단계

이제 HDFA를 적절히 활용할 수 있도록 클라이언트를 구성해야 합니다. 을 "클라이언트 구성"참조하십시오.

ONTAP NAS 연결을 분산하도록 클라이언트를 구성합니다

핫스팟을 해결하려면 CPU 병목 현상을 방지하기 위해 클라이언트를 적절히 구성합니다.

Linux 클라이언트 구성

SVM 내 또는 SVM 간 HDFA 구축을 선택하는 Linux에서 이를 사용하여 클라이언트가 서로 다른 HDFS 간에 로드밸런싱을 수행할 수 있어야 autofs 합니다. `autofs` 구성은 SVM 간 및 내부 SVM에 따라 다릅니다.

시작하기 전에

적절한 종속성이 설치되어 있어야 autofs 합니다. 이에 대한 자세한 내용은 Linux 설명서를 참조하십시오.

이 작업에 대해

설명된 단계에서는 다음 항목이 포함된 예제 /etc/auto_master 파일을 사용합니다.

```
/flexcache auto_hotspot
```

`autofs` `/etc` 프로세스가 디렉터리에 액세스하려고 할 때마다 디렉터리에서 `/flexcache` 호출된 파일을 찾도록 `auto_hotspot` 구성합니다. 파일의 내용은 `auto_hotspot` 디렉토리 내에 마운트할 NFS 서버 및 접속 경로를 `/flexcache` 지정합니다. 설명된 예는 파일에 대한 서로 다른 `auto_hotspot` 설정입니다.

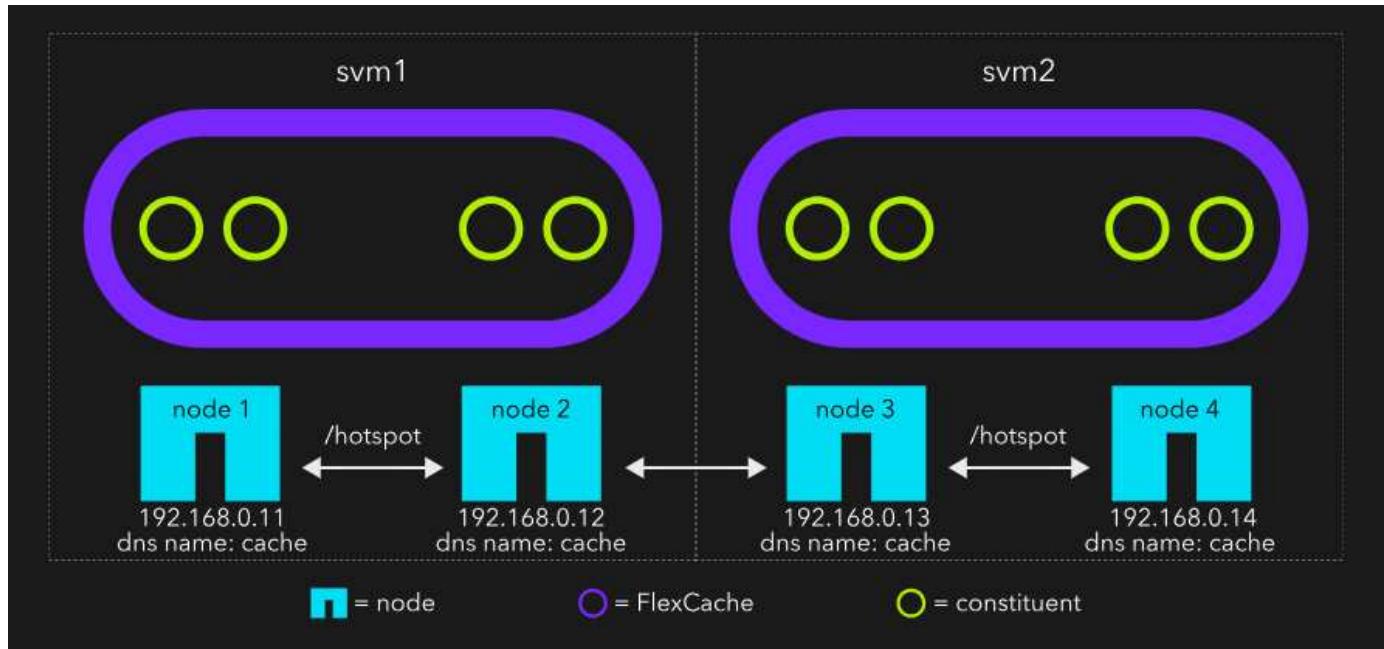
내부 SVM HDFA 자동 전송 구성

다음 예제에서는 의 디어그램에 대한 지도를 만듭니다 autofs [그림 1](#). 각 캐시에는 동일한 접합 경로가 있고 호스트 이름에 4개의 DNS A 레코드가 있으므로 cache 한 줄만 있으면 됩니다.

```
hotspot cache:/hotspot
```

이 간단한 한 줄은 NFS 클라이언트가 호스트 이름에 대한 DNS 조회를 cache 수행하도록 합니다. DNS가 라운드 로빈 방식으로 IP를 반환하도록 설정됩니다. 이렇게 하면 프런트엔드 NAS 연결이 고르게 분산됩니다. 클라이언트는 IP를 수신한 후에서 `/flexcache/hotspot` 접합 경로를 마운트합니다 `/hotspot`. SVM1, SVM2, SVM3 또는 SVM4에 연결할 수 있지만 특정 SVM은 중요하지 않습니다.

그림 1: 2x2x2 SVM 간 HDFA



내부 SVM HDFA 자동 전송 구성

다음 예제에서는 의 디어그램에 대한 지도를 만듭니다 [autoofs](#) 그림 2. NFS 클라이언트가 HDF 접합 경로 배포의 일부인 IP를 마운트해야 합니다. 즉, IP 192.168.0.11이 아닌 다른 것을 탑재하고 싶지 않습니다. `/hotspot1` 이를 위해 맵에서 하나의 로컬 마운트 위치에 대한 IP/접합 경로 쌍 4개를 모두 나열할 수 `auto_hotspot` 있습니다.



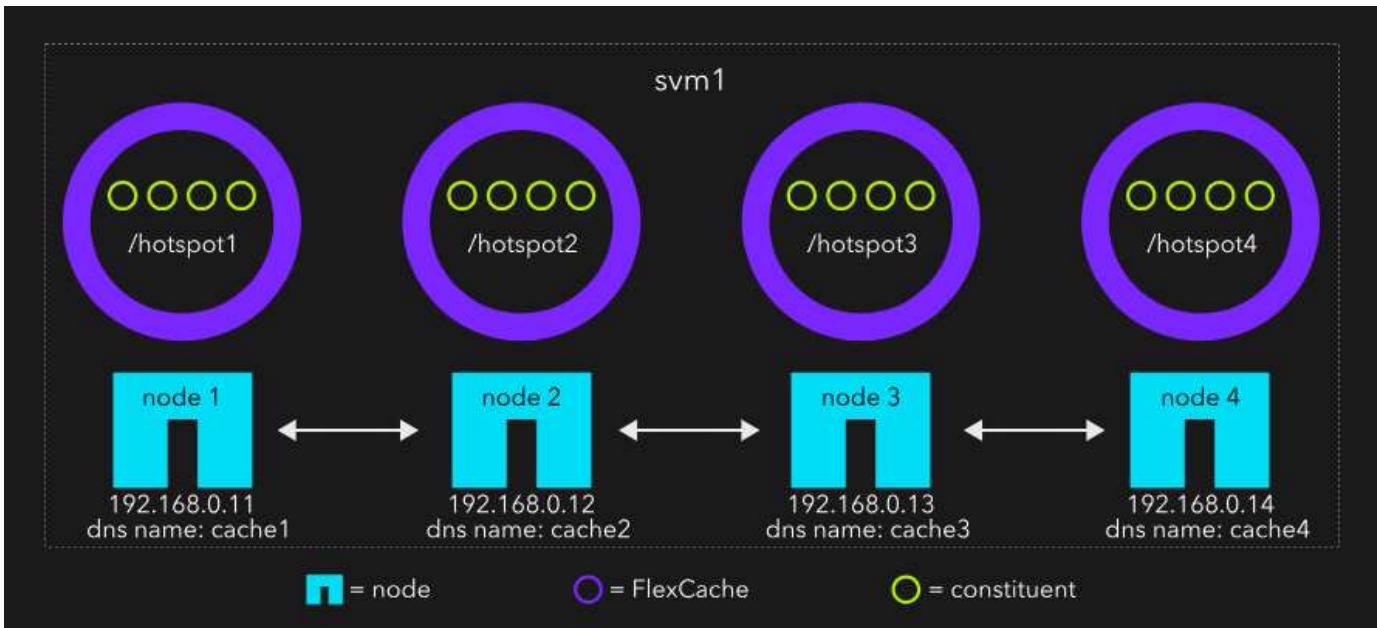
(`(` 다음 예제의 백슬래시)는 항목을 다음 줄로 이어가며 읽기 쉽도록 합니다.

```
hotspot      cache1:/hostspot1 \
              cache2:/hostspot2 \
              cache3:/hostspot3 \
              cache4:/hostspot4
```

클라이언트가 액세스를 시도할 때 `/flexcache/hotspot`은 `autoofs` 네 개의 호스트 이름 모두에 대해 정방향 조회를 수행합니다. 4개의 IP가 모두 클라이언트와 동일한 서브넷에 있거나 다른 서브넷에 있다고 가정하면 `autoofs` 각 IP에 대해 NFS NULL ping을 실행합니다.

이 NULL ping을 수행하려면 ONTAP의 NFS 서비스에서 패킷을 처리해야 하지만 디스크 액세스가 필요하지 않습니다. 반환할 첫 번째 ping은 마운트하도록 선택한 IP 및 접합 경로입니다. `autoofs`

그림 2: 4x1x4 내부 SVM HDFA



Windows 클라이언트 구성

Windows 클라이언트에서는 내부 SVM HDFA를 사용해야 합니다. SVM의 여러 HDFS 간에 로드 밸런싱을 수행하려면 각 HDF에 고유한 공유 이름을 추가해야 합니다. 그런 다음 의 단계를 따라 "[Microsoft 설명서](#)"동일한 폴더에 대해 여러 DFS 대상을 구현합니다.

저작권 정보

Copyright © 2026 NetApp, Inc. All Rights Reserved. 미국에서 인쇄됨 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그레픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 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 라이센스도 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 있으며 재사용이 불가능하며 취소 불가능한 라이센스를 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 라이센스는 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.