DELL Technologies

백서

DELL EMC POWERSCALE ONEFS: 기술 개요

백서 소개

본 백서는 모든 Dell EMC PowerScale 스케일 아웃 NAS 스토리지 솔루션을 구동하는 OneFS 운영 체제의 주요 특징과 기능에 대한 기술적 세부 정보를 제공합니다.

2021 년 9 월

개정 내역

버전	날짜	설명
1.0	2013년 11월	OneFS 7.1의 최초 릴리스
2.0	2014년 6월	OneFS 7.1.1 업데이트
3.0	2014년 11월	OneFS 7.2 업데이트
4.0	2015년 6월	OneFS 7.2.1 업데이트
5.0	2015년 11월	OneFS 8.0 업데이트
6.0	2016년 9월	OneFS 8.0.1 업데이트
7.0	2017년 4월	OneFS 8.1 업데이트
8.0	2017년 11월	OneFS 8.1.1 업데이트
9.0	2019년 2월	OneFS 8.1.3 업데이트
10.0	2019년 4월	OneFS 8.2 업데이트
11.0	2019년 8월	OneFS 8.2.1 업데이트
12.0	2019 년 12 월	OneFS 8.2.2 업데이트
13.0	2020년 6월	OneFS 9.0 업데이트
14.0	2020년 9월	OneFS 9.1 업데이트
15.0	2021년 4월	OneFS 9.2 업데이트
16.0	2021년 9월	OneFS 9.3 업데이트

감사의 말

이 백서는 다음에 의해 작성되었습니다.

작성자: Nick Trimbee

목차

소개

기존 스토리지 모델의 세 가지 계층인 파일 시스템, 볼륨 관리자 및 데이터 보호는 시간이 지나면서 소규모 스토리지 아키텍처의 요구 사항을 충족하도록 발전했지만 복잡성을 크게 증가시키고 페타바이트 규모의 시스템에 적합하지도 않습니다. OneFS 운영 체제는 확장 가능한 데이터 보호 기능이 기본 제공되는 통합된 클러스터 파일 시스템을 제공하고 볼륨 관리의 필요성을 없애 이 세 가지 계층을 모두 대체합니다. 대규모 확장과 뛰어난 효율성을 가능하게 하고 모든 Dell EMC PowerScale NAS 스토리지 솔루션을 구현하는 OneFS는 스케일 아웃 인프라스트럭처의 핵심이 되는 기본 요소입니다.

무엇보다, OneFS는 시스템 측면뿐만 아니라 인력 측면에서도 확장이 용이하도록 설계되었습니다. 따라서 일반적인 스토리지 시스템 관리에 필요한 것보다 적은 수의 인력으로 대규모 시스템을 관리할 수 있습니다. OneFS는 스토리지 관리에 따른 부담을 획기적으로 줄여 주는 자가 복구 및 자가 관리 기능을 포함하고 있어 복잡한 작업이 필요하지 않습니다. 또한 OS의 매우 낮은 레벨에서 병렬 처리 기능이 통합되어 있어 사실상 모든 주요 시스템 서비스가 여러 하드웨어 유닛으로 분산됩니다. 이러한 기능 덕분에 인프라스트럭처 확장 시 OneFS도 어떠한 규모로든 확장할 수 있으므로 데이터 세트가 커져도 현재의 작업 방식을 계속 유지할 수 있습니다.

OneFS는 완벽한 이중화를 통해 균형이 잡힌 파일 시스템입니다. 즉, 클러스터링을 활용해 단지 성능과 용량을 확장하는 데 그치지 않고 RAID의 성능을 훨씬 뛰어넘는 다양한 수준의 이중화와 이기종 시스템 간 페일오버를 지원합니다. 성능은 서서히 증가하는 반면에 스토리지 집적도는 빠르게 증가하는 것이 디스크 하위 시스템의 추세입니다. OneFS는 이중화 규모를 확장하는 동시에 장애 복구 속도를 높여 이러한 현실에 대응하고 있습니다. 따라서 OneFS는 페타바이트급 스토리지 시스템으로 확장하고 기존의 소규모 스토리지 시스템보다 월등한 신뢰성을 제공할 수 있습니다.

PowerScale 하드웨어는 OneFS 실행의 기반이 되는 어플라이언스를 제공합니다. 동급 최고 수준의 상용 하드웨어 구성 요소를 사용하므로 비용 및 효율성이 지속적으로 향상되는 상용 하드웨어의 이점을 이용할 수 있습니다. OneFS는 언제든지 자유롭게 클러스터에서 하드웨어를 통합하거나 제거할 수 있도록 지원하기 때문에 데이터와 애플리케이션을 하드웨어로부터 독립적으로 추상화할 수 있습니다. 따라서 하드웨어의 발전으로 인한 변화와 상관없이 데이터가 보호되고 무한 보존됩니다. 또한 데이터 마이그레이션 및 하드웨어 교체의 비용과 문제도 해소됩니다.

OneFS는 대규모 홈 디렉토리, 파일 공유, 아카이브, 가상화 및 비즈니스 분석을 포함한 엔터프라이즈 환경의 파일 기반 비정형 "빅데이터" 애플리케이션에 가장 적합합니다. 그래서 현재 OneFS는 에너지, 금융 서비스, 인터넷 및 호스팅 서비스, 비즈니스 인텔리전스, 엔지니어링, 제조, 미디어/엔터테인먼트, 생물정보학, 과학 연구 등의 고성능 컴퓨팅 환경을 비롯한 많은 데이터 집약적 산업에서 널리 사용되고 있습니다.

대상

본 백서는 Dell EMC PowerScale 클러스터 배포 및 관리에 대한 정보를 제공하고 OneFS 아키텍처에 대한 포괄적인 배경을 제공합니다.

본 백서는 PowerScale 클러스터 스토리지 환경을 구성하고 관리하는 모든 사용자를 대상으로 합니다. 독자에게 스토리지, 네트워킹, 운영 체제, 데이터 관리에 대한 기본 지식이 있다고 가정합니다.

📖 OneFS 명령 및 기능 구성에 대한 자세한 내용은 OneFS 관리 가이드에서 확인할 수 있습니다.

OneFS 개요

OneFS는 일반적인 스토리지 아키텍처의 세 가지 계층인 파일 시스템, 볼륨 관리자, 데이터 보호를 단일 통합 소프트웨어 계층으로 결합하여 OneFS 기반 스토리지 클러스터에서 실행하는 지능형 단일 분산 파일 시스템을 구축합니다.

그림 1: OneFS 는 파일 시스템, 볼륨 관리자, 데이터 보호 기능을 하나의 지능형 분산 시스템으로 결합.

이러한 특징은 오늘날 기업들이 운영 환경에서 스케일 아웃 NAS를 성공적으로 활용할 수 있게 된 배경이 된 기술 혁신의 핵심입니다. 또한 스케일 아웃, 지능형 소프트웨어, 상용 하드웨어 및 분산 아키텍처의 핵심 원칙을 준수합니다. OneFS는 운영 체제인 동시에, 클러스터로 데이터를 이동하고 저장하는 기본 파일 시스템이기도 합니다.

PowerScale 노드

OneFS는 "클러스터"라고 하는 전용 플랫폼 노드에서만 작동합니다. 단일 클러스터는 메모리, CPU, 네트워킹, 이더넷 또는 레이턴시가 짧은 InfiniBand 상호 연결, 디스크 컨트롤러, 스토리지 미디어 등을 포함하는 랙마운트형 엔터프라이즈 어플라이언스로 구성된 여러 노드로 이루어집니다. 따라서 분산 클러스터의 각 노드는 컴퓨팅뿐만 아니라 스토리지 및 용량 능력을 모두 갖췄습니다.

Gen6 아키텍처를 사용하여 OneFS 8.2 이상에서 최대 252개의 노드까지 확장되는 클러스터를 생성하려면 4RU(랙 유닛) 폼 팩터에 4개의 노드가 있는 단일 섀시가 필요합니다. 개별 노드 플랫폼에서 클러스터를 형성하려면 최소 3개의 노드와 3RU의 랙 공간이 필요합니다. 여러 다른 유형의 노드가 있을 수 있으며, 모든 노드는 처리량 대비 용량 비율 또는 IOPS(Input/Output Operations Per Second)가 각기 다른 노드로 구성된 단일 클러스터로 통합될 수 있습니다. 기존 Gen6 섀시와 PowerScale 올플래시 F900, F600, F200 독립 실행형 노드가 동일한 클러스터 내에서 공존할 수 있습니다.

클러스터에 노드 또는 섀시가 추가될 때마다 디스크, 캐시, CPU 및 네트워크의 총 용량이 늘어납니다. OneFS는 각각의 하드웨어 구성 요소를 모두 활용하므로 전체가 부분의 합보다 큰 효과를 가져옵니다. RAM이 정합성을 보장하는 단일 캐시로 그룹화되므로 클러스터의 모든 부분에서 발생하는 입출력이 모든 위치에서 캐싱된 데이터를 활용할 수 있습니다. 파일 시스템 저널은 전원 장애 발생 시에도 안전한 쓰기 작업을 보장합니다. 디스크와 CPU가 결합되어 클러스터 증가에 따라 처리량, 용량, IOPS도 커지며, 단일 파일 또는 여러 파일에 액세스할 수 있습니다. 클러스터의 스토리지 용량은 수십 TB에서 수십 PB까지 다양합니다. 스토리지 미디어 및 노드 섀시 집적도가 계속 높아짐에 따라 최대 용량은 계속 늘어납니다.

OneFS 기반 플랫폼 노드는 그 기능에 따라 다음과 같이 여러 등급 또는 계층으로 분류되어 제공됩니다.

Tier	I/O Profile	Drive Media	Nodes
Performance	High Perf, Low Latency	Flash NVMe/SAS	F900 F810 F600 F800 F200
Hybrid / Utility	Concurrency & Streaming Throughput	SATA/SAS & SSD	H700 H600 H7000 H5600 H500 H400
Archive	Nearline & Deep Archive	SATA	A300 A200 A3000 A2000

표 1: 하드웨어 계층 및 노드 유형

네트워크

클러스터와 연결되는 네트워크 유형에는 내부, 외부 이렇게 두 가지 유형이 있습니다.

백엔드 네트워크

클러스터의 모든 노드 간 통신은 10Gb, 40Gb 또는 100Gb 이더넷이나 짧은 레이턴시의 QDR IB(InfiniBand)로 구성된 전용 백엔드 네트워크를 통해 수행됩니다. 고가용성을 위해 이중화된 스위치로 구성된 이 백엔드 네트워크는 클러스터의 백플레인 역할을 합니다. 이를 통해 각 노드는 클러스터의 참가 노드 역할을 하고 노드 간 통신을 레이턴시가 짧은 전용 고속 네트워크로 격리시킬 수 있습니다. 이 백엔드 네트워크는 노드 간 통신에 IP(Internet Protocol)를 활용합니다.

프런트엔드 네트워크

클라이언트는 모든 노드에서 사용할 수 있는 이더넷 접속(10GbE, 25GbE, 40GbE 또는 100GbE)을 통해 클러스터에 접속합니다. 노드마다 고유한 이더넷 포트를 제공하기 때문에 클러스터에서 사용할 수 있는 네트워크 대역폭 양은 성능 및 용량과 함께 비례적으로 확장됩니다. 클러스터는 고객 네트워크에 NFS, SMB, HTTP, FTP, HDFS, S3 등의 표준 네트워크 통신 프로토콜을 지원합니다. 또한 OneFS는 IPv4 및 IPv6 환경 모두와 완전히 통합될 수 있도록 지원합니다.

완전한 클러스터 뷰

완전한 클러스터는 다음 뷰에서와 같이 하드웨어, 소프트웨어 및 네트워크와 결합됩니다.

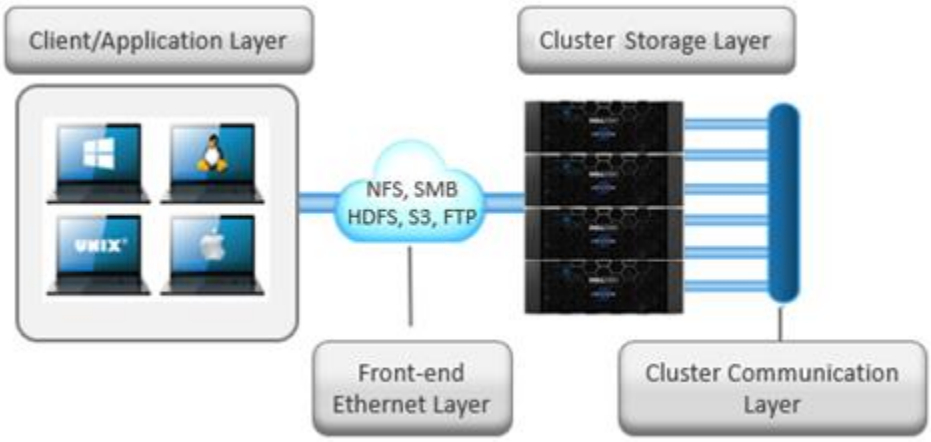

그림 2: 작동 중인 OneFS 의 모든 구성 요소

위 다이어그램은 스케일 아웃 환경에서 워크로드 및 용량 요구 사항이나 처리량 요구 사항이 달라질 때 동적으로 확장할 수 있는 완벽한 분산형 단일 파일 시스템을 제공하기 위해 소프트웨어, 하드웨어 및 네트워크 모두가 서버와 함께 작동하는 아키텍처를 보여 줍니다.

OneFS SmartConnect는 프론트 엔드 이더넷 계층에서 작동하여 클러스터 전체에 클라이언트 연결을 균등하게 분산하는 로드 밸런싱 장치입니다. SmartConnect 소프트웨어는 Linux 및 UNIX 클라이언트의 동적 NFS 페일오버와 페일백을 지원하고 Windows 클라이언트에 SMB3 무중단 가용성을 지원합니다. 노드 장애가 발생하거나 사전 예방적 유지 보수 작업을 수행할 경우 실행 중인 읽기 및 쓰기 작업이 모두 클러스터의 다른 노드로 인계되어 사용자나 애플리케이션의 작업에 영향을 주지 않으면서 읽기/쓰기 작업이 완료됩니다.

페일오버하는 중에는 클러스터에서 정상 작동하는 모든 노드에 걸쳐 클라이언트를 균등하게 분산하여 성능에 미치는 영향을 최소화합니다. 장애 등의 이유로 노드의 작동이 중단되면 노드의 가상 IP 주소가 클러스터의 다른 노드로 원활하게 마이그레이션됩니다. 오프라인 상태의 노드가 다시 온라인 상태가 되면 SmartConnect가 전체 클러스터에서 NFS 및 SMB3 클라이언트를 자동으로 다시 로드 밸런싱하여 스토리지 및 성능 활용도를 극대화합니다. 정기적인 시스템 유지 보수와 소프트웨어 업데이트의 경우에 이 기능은 노드별 점진적 업그레이드를 지원하여 유지 보수 기간 동안 완벽한 가용성을 유지합니다.

📖 자세한 내용은 OneFS SmartConnect 백서에서 확인할 수 있습니다.

OneFS 소프트웨어 개요

운영 체제

OneFS는 BSD 기반 UNIX OS(Operating System)를 토대로 구축됩니다. 하드 링크, 닫기 시 삭제(delete-on-close), atomic rename, ACL 및 확장 특성을 비롯한 Linux/UNIX 및 Windows 의미 체계를 기본적으로 모두 지원합니다. OneFS는 BSD를 기본 OS로 사용합니다. BSD는 오픈 소스 커뮤니티를 통해 혁신을 실현할 수 있는 안정적이고 검증된 OS이기 때문입니다. OneFS 8.2 이후부터 기본 OS 버전은 FreeBSD 11입니다.

클라이언트 서비스

클라이언트가 OneFS와 상호 작용하는 데 사용할 수 있는 프런트엔드 프로토콜을 클라이언트 서비스라고 합니다. 지원되는 프로토콜의 자세한 목록은 "지원되는 프로토콜" 섹션을 참조하십시오. OneFS가 클라이언트와 통신하는 방법을 이해하기 위해 I/O 서브시스템을 상위 절반 또는 '이니시에이터'와 아래쪽 절반 또는 '참가 노드' 이렇게 두 부분으로 나누어 보겠습니다. 클러스터의 모든 노드는 특정 입출력 작업의 참가 노드입니다. 클라이언트가 접속하는 대상 노드는 이니시에이터이고 이 노드는 전체 입출력 작업의 '캡틴' 역할을 합니다. 읽기 및 쓰기 작업은 뒷부분의 섹션에서 자세히 설명합니다.

클러스터 작업

클러스터 아키텍처에는 클러스터 자체의 상태와 유지 보수를 담당하는 클러스터 작업이 있으며 이러한 작업은 모두 OneFS 작업 엔진에서 관리됩니다. 이 작업 엔진은 전체 클러스터에서 실행되며 대용량의 스토리지 관리 작업과 보호 작업을 분할하고 관리합니다. 이를 위해 작업 엔진은 작업을 더 작은 작업 항목으로 줄인 다음 전체 작업의 이러한 일부분을 각 노드의 여러 작업자 스레드에 할당 또는 매핑합니다. 작업 실행 내내 진행 상황이 추적 및 보고되고 작업 완료 또는 종료 시 자세한 보고 및 상태 정보가 제공됩니다.

작업 엔진에는 작업을 시작 및 중지할 뿐 아니라 일시 중지 및 다시 시작할 수 있는 포괄적인 체크포인트 시스템이 포함되어 있습니다. 또한 작업 엔진 프레임워크에는 적응형 영향 관리 시스템도 포함되어 있습니다.

일반적으로 작업 엔진은 스페어 또는 특별 예약된 용량 및 리소스를 사용하여 클러스터 전체에서 백그라운드 작업으로 작업을 실행합니다. 작업은 기본적으로 다음과 같은 세 가지 유형으로 분류할 수 있습니다.

파일 시스템 유지 보수 작업

이 작업은 백그라운드 파일 시스템 유지 보수를 수행하며 대개 모든 노드에 대한 액세스를 필요로 합니다. 이 작업은 기본 구성으로 실행해야 하며 클러스터 성능이 떨어졌을 때 주로 실행됩니다. 파일 시스템 보호 및 드라이브 재구축이 여기에 해당합니다.

기능 지원 작업

기능 지원 작업은 확장된 스토리지 관리 기능을 용이하게 하는 작업을 수행하고 대개 해당 기능이 구성된 경우에만 실행됩니다. 데이터 중복 제거 및 바이러스 백신 검사 등이 여기에 해당합니다.

사용자 작업

이 작업은 데이터 관리 목표를 달성하기 위해 스토리지 관리자가 직접 실행합니다. 병렬식 트리 삭제 및 사용 권한 유지 관리가 여기에 해당합니다.

아래의 표는 제공되는 작업 엔진 작업과 각 작업이 수행하는 내용, 그리고 각각의 파일 시스템 액세스 방법을 포괄적인 목록으로 보여 줍니다.

작업 이름	작업 설명	액세스 방법
AutoBalance	클러스터에서 사용 가능한 용량의 균형을 유지합니다.	**드라이브 + LIN**
AutoBalanceLin	클러스터에서 사용 가능한 용량의 균형을 유지합니다.	**LIN**
AVScan	안티바이러스 서버가 실행하는 바이러스 스캐닝 작업.	**트리**
ChangelistCreate	두 개의 연속 SyncIQ 스냅샷 사이에서 변경된 사항을 목록으로 만들기	**변경 목록**
CloudPoolsLin	파일 풀 정책에 따라, 클라우드 공급자에게 보낸 데이터를 보관합니다.	**LIN**
CloudPoolsTreewalk	파일 풀 정책에 따라, 클라우드 공급자에게 보낸 데이터를 보관합니다.	**트리**
Collect	노드 또는 드라이브에서 다양한 장애 조건이 발생하여 해당 노드 또는 드라이브를 사용할 수 없기 때문에 확보하지 못한 디스크 공간을 재확보합니다.	**드라이브 + LIN**
ComplianceStoreDelete	SmartLock 컴플라이언스 모드 가비지 컬렉션 작업.	**트리**
데이터 중복 제거	파일 시스템에서 동일한 블록에 대해 중복 제거를 수행합니다.	**트리**
DedupeAssessment	데이터 중복 제거의 이점에 대한 진단을 시험 실행합니다.	**트리**
DomainMark	경로 및 해당 컨텐츠를 도메인과 연결합니다.	**트리**
DomainTag	경로 및 해당 컨텐츠를 도메인과 연결합니다.	**트리**
EsrsMftDownload	라이센스 파일에 대해 ESRS에서 관리하는 파일 이전 작업입니다.	
FilePolicy	효율적인 SmartPools 파일 풀 정책 작업.	**변경 목록**
FlexProtect	파일 시스템을 장애 시나리오에서 복구하기 위해 재구축하고 재보호합니다.	**드라이브 + LIN**
FlexProtectLin	파일 시스템을 재보호합니다.	**LIN**
FSAnalyze	InsightIQ와 함께 사용되는 파일 시스템 분석 데이터를 수집합니다.	**변경 목록**
IndexUpdate	FilePolicy와 FSAnalyze 작업에 대한 효율적인 파일 시스템 인덱스를 생성하고 업데이트합니다.	**변경 목록**
IntegrityScan	파일 시스템 비정합성에 대한 온라인 검증 및 수정을 수행합니다.	**LIN**
LinCount	파일 시스템 논리적 inode(LIN)를 검사하고 계산합니다.	**LIN**

작업 이름	작업 설명	액세스 방법
MediaScan	드라이브에서 미디어 레벨 오류를 검사합니다.	**드라이브 + LIN**
MultiScan	Collect 및 AutoBalance 작업을 동시에 실행합니다.	**LIN**
PermissionRepair	파일 및 디렉토리에 대한 사용 권한을 올바르게 수정합니다.	**트리**
QuotaScan	기존 디렉토리 경로에 생성된 도메인에 대한 할당량 계산을 업데이트합니다.	**트리**
SetProtectPlus	기본 파일 정책을 적용합니다. 클러스터에서 SmartPools가 활성화된 경우 이 작업은 해제됩니다.	**LIN**
ShadowStoreDelete	섀도우 저장소와 연결된 공간을 확보합니다.	**LIN**
ShadowStoreProtect	강화된 보안을 통해 LIN이 참조한 섀도우 저장소를 보호합니다.	**LIN**
ShadowStoreRepair	섀도우 저장소를 복구합니다.	**LIN**
SmartPools	동일한 클러스터 내에서 노드 계층 간에 데이터를 실행하고 이동하는 작업입니다. 라이선스 등록 및 구성이 완료되면 CloudPools 기능을 실행합니다.	**LIN**
SmartPoolsTree	하위 트리에서 SmartPools 파일 정책을 적용합니다.	**트리**
SnapRevert	전체 스냅샷을 헤드로 되돌립니다.	**LIN**
SnapshotDelete	삭제된 스냅샷과 관련된 디스크 공간을 확보합니다.	**LIN**
TreeDelete	클러스터에서 직접적으로 연결되는 파일 시스템의 경로를 삭제합니다.	**트리**
Undedupe	파일 시스템 내 동일한 블록의 중복 제거를 제거합니다.	**트리**
Upgrade	이후 OneFS 릴리스에서 클러스터를 업그레이드합니다.	**트리**
WormQueue	SmartLock LIN 큐 검색	**LIN**

그림 1: OneFS 작업 엔진 작업 설명

파일 시스템 유지 보수 작업은 기본적으로 스케줄에 따라 또는 특정 파일 시스템 이벤트에 대한 응답으로 실행되지만, 다른 작업과 비교한 우선 순위 레벨과 영향 정책을 구성하여 작업 엔진 작업을 관리할 수 있습니다.

영향 정책에는 특정 주의 시간대를 나타내는 하나 이상의 영향 간격이 포함됩니다. 각 영향 간격은 특정 클러스터 작업에 사용할 클러스터 리소스 양을 지정하는 사전 정의된 영향 레벨을 사용하도록 구성할 수 있습니다. 다음과 같은 작업 엔진 영향 레벨을 사용할 수 있습니다.

- Paused
- 낮음
- 보통
- 높음

이러한 세분화를 통해 작업별로 영향 간격 및 레벨을 구성하여 원활한 클러스터 작업을 보장할 수 있습니다. 이와 같이 구성된 영향 정책으로 작업이 실행되는 시간과 작업에서 사용할 수 있는 리소스를 결정합니다.

작업 엔진 작업의 우선 순위는 1부터 10까지의 등급으로 지정할 수 있으며 값이 낮을수록 우선 순위가 더 높습니다. 이 기능은 UNIX 스케줄링 유틸리티 'nice'와 유사합니다.

작업 엔진은 최대 3개의 작업을 동시에 실행하도록 지원합니다. 이 동시 작업 실행은 다음 기준에 의해 제어됩니다.

- 작업 우선 순위
- 제외 세트 - 함께 실행할 수 없는 작업(예: FlexProtect와 AutoBalance)
- 클러스터 상태 - 클러스터가 성능이 저하된 상태일 경우 대부분의 작업은 실행할 수 없습니다.

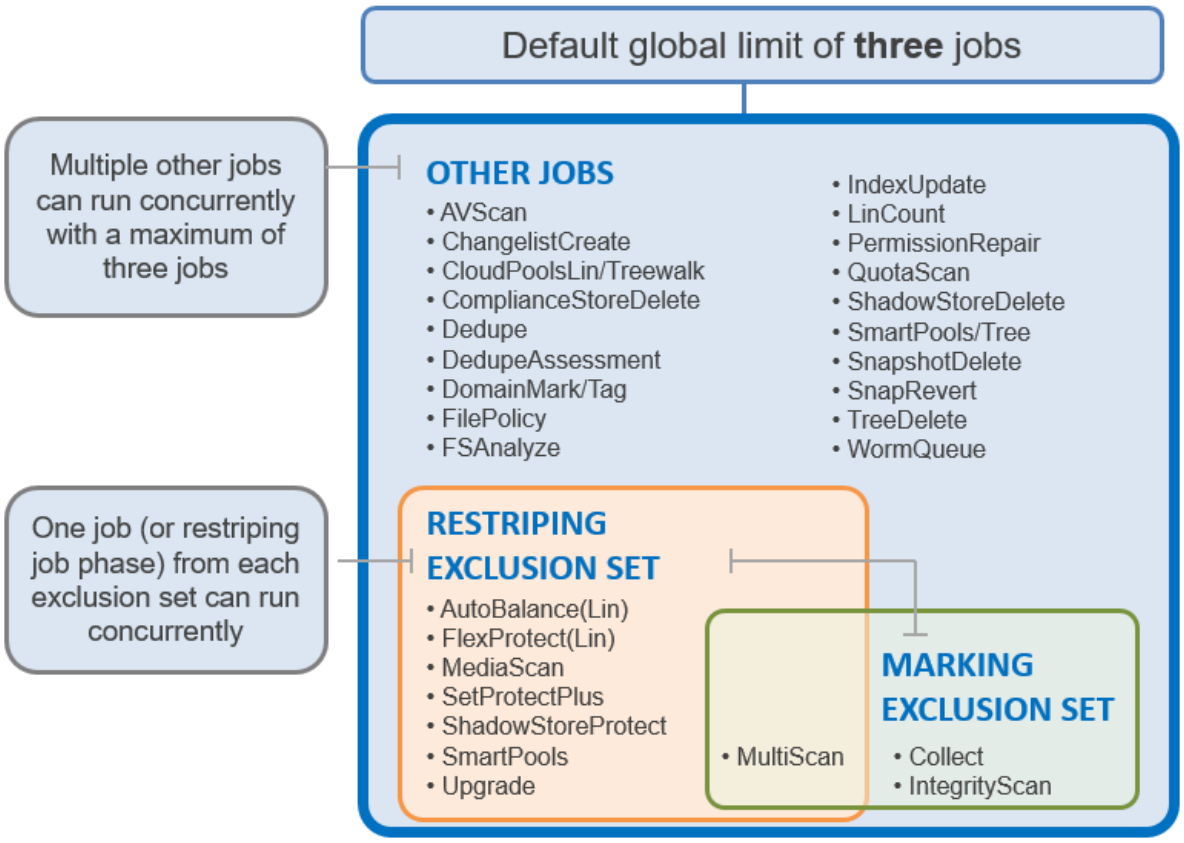

그림 4: OneFS 작업 엔진 제외 세트

📖 자세한 내용은 OneFS 작업 엔진 백서에서 확인할 수 있습니다.

파일 시스템 구조

OneFS 파일 시스템은 UFS(UNIX File System)를 기반으로 하는 매우 빠른 분산 파일 시스템입니다. 각 클러스터는 단일 네임스페이스와 파일 시스템을 생성합니다. 이는 파일 시스템이 클러스터의 모든 노드 간에 분산되고 클러스터의 노드에 접속된 클라이언트에서 액세스가 가능함을 의미합니다. 파티셔닝이 없고 볼륨 생성이 필요하지 않습니다. OneFS는 물리적 볼륨 레벨에서 사용 가능한 용량 및 승인되지 않은 파일에 대한 액세스를 제한하는 것이 아니라, 디렉토리 레벨 할당량 관리를 제공하는 SmartQuotas 서비스와 공유 및 파일 사용 권한을 통해 소프트웨어에서 동일한 기능을 제공합니다.

📖 자세한 내용은 OneFS SmartQuotas 백서에서 확인할 수 있습니다.

모든 정보가 내부 네트워크를 통해 노드 사이에 공유되기 때문에 어떤 노드에서든 데이터를 쓰거나 읽을 수 있으므로 여러 사용자가 동시에 같은 데이터 세트를 읽고 쓸 때 성능이 최적화됩니다.

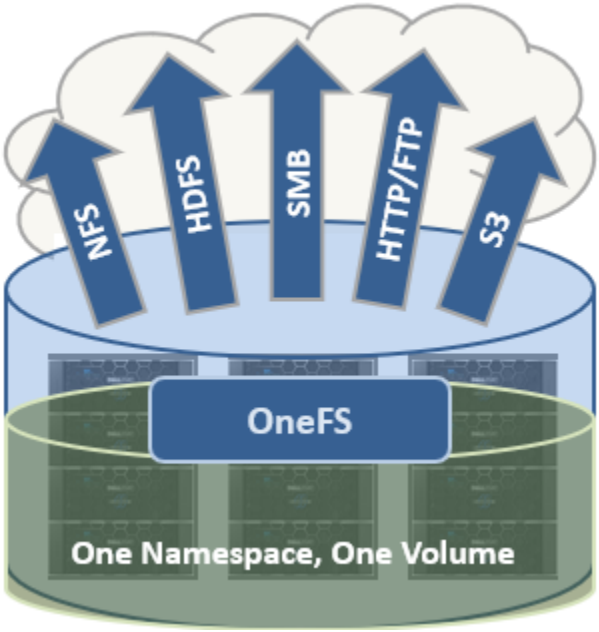

그림 5: 여러 가지 액세스 프로토콜을 지원하는 단일 파일 시스템

OneFS는 하나의 네임스페이스를 사용하는 완전한 단일 파일 시스템입니다. 데이터 및 메타데이터가 여러 노드에 걸쳐 스트라이핑되어 이중화 및 가용성이 보장됩니다. 또한 스토리지가 사용자 및 관리자에 대해 완전히 가상화되었습니다. 따라서 트리 확장 방식이나 사용자의 이용 방식을 계획하거나 감시하지 않고 파일 트리를 유기적으로 늘릴 수 있습니다. OneFS SmartPools가 단일 트리를 방해하지 않고 계층화를 자동으로 처리하므로 관리자가 파일을 적절한 디스크에 계층화하는 방법을 특별히 고민할 필요가 없습니다. 또한, 매우 큰 트리를 어떻게 복제할지 특별히 고민하지 않아도 됩니다. OneFS SyncIQ 서비스가 파일 트리의 구조나 깊이와 관계없이 파일 트리를 하나 이상의 대체 클러스터로 전환하는 작업을 자동으로 병렬 처리하기 때문입니다.

이와 같은 설계는 기존 NAS가 단일 네임스페이스를 가진 것처럼 "보이도록" 하기 위해 널리 사용되는 기술인 네임스페이스 취합과 비교가 됩니다. 네임스페이스 취합을 사용할 경우 파일을 여전히 개별 볼륨에서 관리해야 하지만 간단한 "표시(veneer)" 계층에서 심볼릭 링크를 통해 "최상위 레벨" 트리로 볼륨의 개별 디렉토리를 "통합"할 수 있습니다. 이 모델에서는 LUN과 볼륨 및 볼륨 제한이 여전히 존재합니다. 로드 밸런스를 위해 파일을 수동으로 볼륨 간에 이동해야 합니다. 관리자는 트리의 레이아웃을 신중하게 고려해야 합니다. 계층화가 원활하게 이루어지지 않고 지속적인 수동 작업이 상당히 많이 필요합니다. 페일오버를 위해 볼륨 간에 파일을 미러링해야 하고 이로 인해 효율성은 떨어지며 구입 비용, 전원 및 냉각 비용은 올라갑니다. 전반적으로 네임스페이스 취합을 사용할 때 관리자가 겪는 부담이 단순한 기존 NAS 디바이스에 대한 부담보다 더 큽니다. 따라서 이러한 인프라스트럭처는 대규모로 확장하기가 어렵습니다.

데이터 레이아웃

OneFS는 메타데이터에 대해 물리적 포인터와 익스텐트를 사용하고 파일 및 디렉토리 메타데이터를 inode에 저장합니다. OneFS 논리 inode(LIN)는 일반적으로 크기가 512바이트로 대부분의 하드 드라이브가 포맷된 기본 섹터에 적합합니다. 현재 4KB 섹터로 포맷되어 집적도가 더 높은 하드 드라이브 클래스를 지원하기 위해 8KB inode에 대한 지원도 제공됩니다.

B-트리가 파일 시스템에서 광범위하게 사용되므로 수십억 개의 객체로 확장하고 데이터 또는 메타데이터를 거의 즉각적으로 조회할 수 있습니다. OneFS는 완벽하게 균형이 잡힌 고도로 분산된 파일 시스템입니다. 데이터와 메타데이터는 여러 하드웨어 디바이스에서 항상 이중화됩니다. 또한 클러스터의 모든 노드에서 삭제 코딩을 사용하여 데이터가 보호되기 때문에 클러스터의 효율성이 높아지고 5개 이상의 노드로 구성된 클러스터에서 80% 이상의 물리적 용량 대비 가용 용량을 제공할 수 있습니다. 대개 시스템에서 1% 미만의 용량을 차지하는 메타데이터는 클러스터에서 미러링되어 성능과 가용성이 보장됩니다. OneFS는 RAID에 의존하지 않으므로 관리자가 클러스터의 기본 구성을 벗어나서 파일 또는 디렉토리 레벨에서 이중화 수준을 선택할 수 있습니다. 메타데이터 액세스 작업과 잠금 작업은 P2P(Peer-to-Peer) 아키텍처의 모든 노드에서 균등하게 집합적으로 관리됩니다. 이러한 균형이 이 아키텍처의 단순성과 신뢰성의 핵심입니다. 이 아키텍처에는 단일 메타데이터 서버나 잠금 관리자 또는 게이트웨이 노드가 없습니다.

OneFS는 여러 디바이스에서 동시에 블록에 액세스해야 하기 때문에 데이터 및 메타데이터에 사용되는 주소 지정 체계가 물리적 레벨에서 {노드, 드라이브, 오프셋}의 튜플(tuple)로 인덱싱됩니다. 예를 들어 12345가 노드 3의 디스크 2에 상주하는 블록의 블록 주소일 경우 {3,2,12345}로 읽습니다. 클러스터 내의 모든 메타데이터는 데이터 보호를 위해 관련 파일의 이중화 수준 이상으로 미러링됩니다. 예를 들어 파일이 "+2n"의 삭제 코드로 보호되는 경우, 즉 파일이 두 번의 동시 장애를 감당할 수 있는 경우 이 파일에 액세스하기 위해 필요한 모든 메타데이터는 3x 미러링되어 역시 두 번의 장애를 감당할 수 있습니다. 이 파일 시스템은 기본적으로 모든 구조에서 클러스터의 노드에 있는 모든 블록을 사용할 수 있도록 지원합니다.

다른 스토리지 시스템은 RAID 및 볼륨 관리 계층을 통해 데이터를 보내기 때문에 데이터 레이아웃에서 비효율성을 초래하고 최적화되지 않은 블록 액세스를 제공합니다. OneFS는 클러스터에서 드라이브의 위치에 상관없이 섹터 레벨까지 직접 파일의 배치를 제어합니다. 따라서 최적화된 데이터 배치와 입출력 패턴이 보장되고 불필요한 읽기-수정-쓰기 작업이 방지됩니다. OneFS는 데이터를 디스크에서 파일별로 배치함으로써 스트라이핑 유형은 물론 시스템 레벨과 디렉토리 레벨, 그리고 파일 레벨에 이르기까지 스토리지 시스템의 이중화 레벨을 유연하게 제어할 수 있습니다. 기존의 스토리지 시스템은 특정한 성능 유형 및 보호 설정을 위해 전체 RAID 볼륨을 전용으로 사용해야 합니다. 데이터베이스를 보호하기 위해 RAID 1+0 방식으로 디스크 세트를 배열하는 경우를 예로 들 수 있습니다. 이 방식은 유휴 스핀들을 빌릴 수 없어 전체 스토리지 자산에서 스핀들 사용을 최적화하기 어려울 뿐 아니라 비즈니스 요구 사항에 맞춰 조절할 수 없는 경직된 설계를 야기합니다. OneFS는 언제든지 완전히 온라인 상태에서 개별 튜닝과 유연한 변경을 지원합니다.

파일 쓰기

OneFS 소프트웨어는 모든 노드에서 동일하게 실행되어 각 노드에서 실행되는 단일 파일 시스템을 구축합니다. 어느 한 노드가 클러스터를 제어하거나 "지배"하지 않고 모든 노드가 진정한 피어입니다.

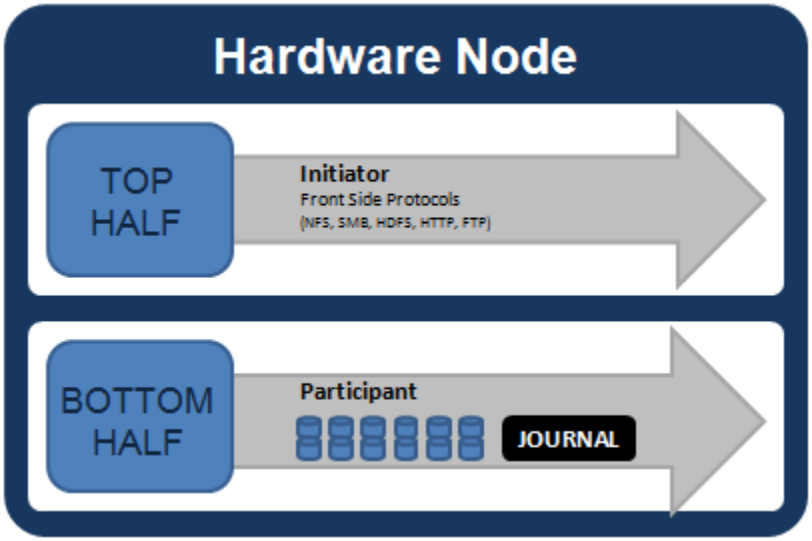

그림 6: I/O 에 관여하는 노드 구성 요소 모델

클러스터에서 입출력에 관여하는 각 노드의 모든 구성 요소를 개괄적으로 살펴보면 위의 그림 6과 같은 모습입니다. 이 스택을 이니시에이터라고 하는 "위쪽" 계층과 참가 노드라고 하는 "아래쪽" 계층으로 분할했습니다. 이러한 분할은 하나의 특정한 읽기 또는 쓰기를 분석하기 위한 "논리적 모델"로 사용됩니다. 물리적 레벨에서 노드의 CPU와 RAM 캐시는 클러스터 전체에서 발생하는 입출력에 대한 이니시에이터 작업과 참가 노드 작업을 동시에 처리합니다. 간단하게 설명하기 위해 위의 다이어그램에서 캐시와 분산 잠금 관리자는 제외되었습니다. 이에 대해서는 백서 뒷부분의 섹션에서 다룹니다.

클라이언트가 파일을 쓰기 위해 노드에 접속하면 노드의 위쪽 절반, 즉 이니시에이터에 접속하게 됩니다. 노드의 아래쪽 절반인 참가 노드(디스크) 부분에 기록되기 전에 파일은 스트라이프라고 하는 더 작은 논리적 청크로 분할됩니다. 쓰기 결합기(write coalescer)를 사용한 장애 방지 버퍼링을 통해 쓰기 효율성이 보장되고 불필요한 읽기-수정-쓰기 작업이 방지됩니다. 각 파일 청크의 크기를 스트라이프 유닛 크기라고 합니다.

OneFS는 단순히 디스크가 아닌 모든 노드에 걸쳐 데이터를 스트라이핑하고 소프트웨어 삭제 코드 또는 미러링 기술을 통해 파일, 디렉토리 및 관련 메타데이터를 보호합니다. 데이터의 경우 관리자의 재량에 따라 데이터 보호를 위해 Reed-Solomon 삭제 코딩 시스템을 사용하거나, 일반적이지는 않지만 미러링을 사용할 수도 있습니다. 사용자 데이터에 적용되는 미러링은 대량의 트랜잭션을 처리하는 성능 사례에 주로 사용됩니다. 대량의 사용자 데이터는 온디스크 효율성을 저해하지 않고 매우 높은 성능을 제공하는 삭제 코딩을 주로 사용합니다. 삭제 코딩은 5개 이상의 노드로 구성된 물리적 디스크에서 80%가 넘는 효율성을 제공하며 대규모 클러스터에서는 이보다 훨씬 뛰어난 효율성을 발휘하는 한편 이중화 수준도 4배 향상시킵니다. 지정한 파일의 스트라이프 너비는 파일이 기록된 노드(드라이브가 아님)의 수입니다. 이러한 너비는 클러스터의 노드 수, 파일의 크기 및 보호 설정(예: +2n)에 의해 결정됩니다.

OneFS는 효율성과 성능을 최대한 고려하여 데이터 레이아웃을 결정하는 고급 알고리듬을 사용합니다. 클라이언트가 노드에 접속하면 이 노드의 이니시에이터가 해당 파일의 쓰기 데이터 레이아웃을 결정하는 "캡틴" 역할을 담당합니다. 데이터, 삭제 코드(ECC) 보호, 메타데이터, inode는 모두 한 클러스터 내의 여러 노드에 분산되고 노드 내 여러 드라이브에도 분산됩니다.

아래의 그림 7은 3노드 클러스터의 모든 노드에서 이루어지는 파일 쓰기를 보여 줍니다.

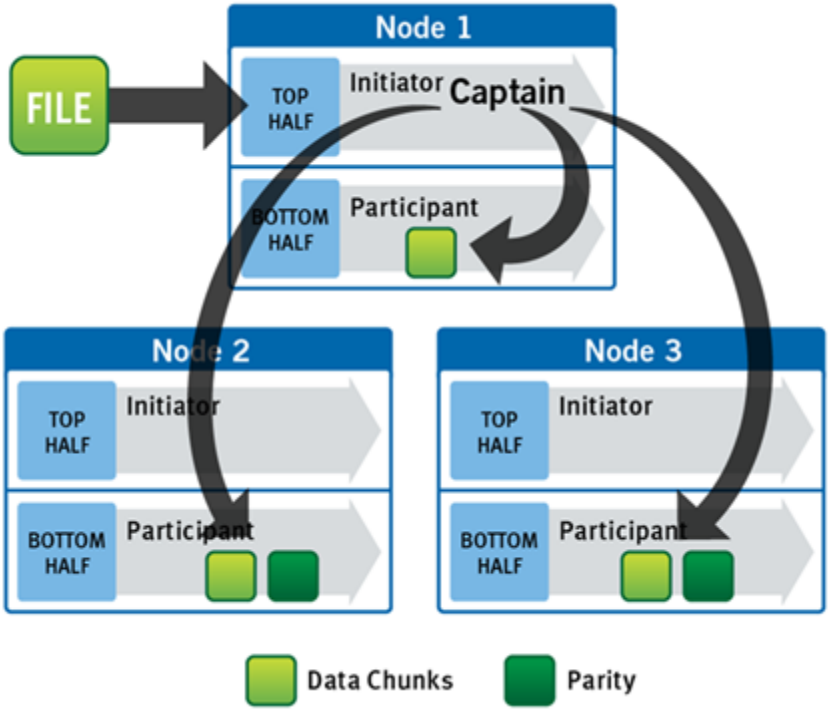

그림 7: 3 노드 클러스터의 파일 쓰기 작업

OneFS는 백엔드 네트워크를 사용하여 클러스터의 모든 노드에 데이터를 자동으로 할당하고 스트라이핑하므로 별도의 처리가 필요하지 않습니다. 데이터가 기록될 때 지정된 수준으로 보호됩니다. 쓰기가 수행될 때 OneFS는 데이터를 보호 그룹이라고 하는 원자 단위로 분할합니다. 모든 보호 그룹이 안전하면 전체 파일의 안전성이 구현되도록 보호 그룹 내에서 이중화가 기본 제공됩니다. 삭제 코드로 보호되는 파일의 경우 보호 그룹은 일련의 데이터 블록 및 이러한 데이터 블록에 대한 삭제 코드 세트로 구성되고, 미러링으로 보호되는 파일의 경우 보호 그룹은 블록 세트에 대한 모든 미러본으로 구성됩니다. OneFS는 파일을 쓸 때 해당 파일에서 사용된 보호 그룹의 유형을 동적으로 전환할 수 있습니다. 이 기능은 많은 추가적인 기능을 가능하게 합니다. 예를 들어 시스템이 클러스터의 일시적인 노드 장애로 인해 원하는 수의 삭제 코드를 사용하지 못하게 되는 상황에서 교착 상태에 빠지지 않고 계속 작동할 수 있습니다. 이 같은 경우 미러링을 일시적으로 사용하여 쓰기를 계속할 수 있습니다. 노드가 클러스터에 복구되면 관리자의 개입 없이 이 미러링 방식 보호 그룹이 다시 삭제 코드로 보호되는 방식으로 원활하게 자동으로 전환됩니다.

OneFS 파일 시스템의 블록 크기는 8KB입니다. 8KB 미만의 파일은 완전한 8KB 블록을 사용합니다. 데이터 보호 수준에 따라 이 8KB 파일은 결국 8KB보다 큰 데이터 공간을 사용할 수도 있습니다. 데이터 보호 설정에 대해서는 본 백서 뒷부분의 섹션에서 자세히 설명합니다. OneFS는 수십억 개의 소규모 파일로 이루어진 파일 시스템을 매우 높은 성능으로 지원할 수 있습니다. 모든 온디스크 구조가 이러한 크기로 확장되고 총 객체 수와 관계없이 모든 객체에 대한 거의 즉각적인 액세스를 제공하도록 설계되었기 때문입니다. 더 큰 파일에 대해서는 여러 개의 인접한 8KB 블록을 이용할 수 있습니다. 이 경우 최대 16개의 인접 블록을 단일 노드의 디스크로 스트라이핑할 수 있습니다. 파일의 크기가 32KB일 경우 4개의 인접한 8KB 블록이 사용됩니다.

이보다 더 큰 파일의 경우에는 16개의 인접 블록으로 구성된, 스트라이프 유닛당 총 128KB의 스트라이프 유닛을 이용하여 순차적 성능을 극대화할 수 있습니다. 쓰기 중에 데이터는 스트라이프 유닛으로 분할되고 이 유닛은 보호 그룹으로 여러 노드 간에 분산됩니다. 또한 데이터가 클러스터에서 배치될 때 필요한 경우 파일이 항상 보호되도록 삭제 코드 또는 미러본이 각 보호 그룹 내에 분산됩니다.

OneFS가 제공하는 AutoBalance 기능의 핵심 역할 중 하나는 가능한 경우 데이터를 재할당 및 재조정하여 스토리지 공간의 가용성과 효율성을 높이는 것입니다. 대부분의 경우 큰 파일의 스트라이프 너비를 증가시켜 노드가 추가될 때 사용 가능한 새로운 공간을 활용하고 온디스크 스트라이핑을 보다 효율적으로 만들 수 있습니다. AutoBalance는 높은 온디스크 효율성을 유지하고 온디스크 "핫 스팟"을 자동으로 제거합니다.

"캡틴" 노드가 있는 위쪽 절반의 이니시에이터는 아래의 그림 8에 나와 있는 것처럼 수정된 2단계 커밋 트랜잭션을 사용하여 클러스터에서 여러 NVRAM에 안전하게 쓰기를 분산합니다.

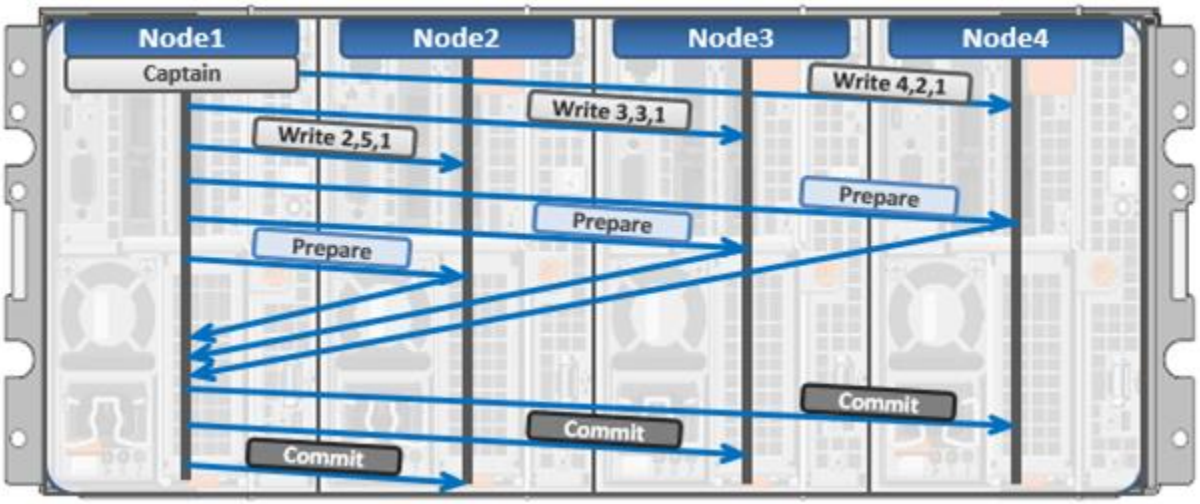

그림 8: 분산된 트랜잭션과 2 단계 커밋

특정 쓰기에서 블록을 소유하는 모든 노드는 2단계 커밋에 관여합니다. 이 메커니즘은 NVRAM을 기반으로 스토리지 클러스터 내 모든 노드에서 이루어지는 모든 트랜잭션을 저널링합니다. 여러 NVRAM을 병렬로 사용함으로써 쓰기 처리 성능을 극대화하는 동시에 전원 장애를 비롯한 모든 방식의 장애로부터 안전하게 데이터를 유지합니다. 트랜잭션 중간에 노드 장애가 발생할 경우 이 노드의 참여 없이 즉각적으로 트랜잭션이 재개됩니다. 노드가 복구되면 노드가 NVRAM의 저널을 재생하고(몇 초 또는 몇 분 소요), 때때로 AutoBalance를 통해 트랜잭션에 참여한 파일을 재조정하기만 하면 됩니다. 많은 리소스가 소모되는 'fsck' 또는 '디스크 검사' 프로세스가 전혀 필요하지 않습니다. 그뿐만 아니라 장시간의 재동기화를 수행할 필요가 없습니다. 또한 장애로 인해 쓰기가 차단되지 않습니다. 특허받은 이 트랜잭션 시스템은 OneFS가 단일 또는 여러 장애 지점을 제거하는 방식 중 하나입니다.

쓰기 작업에서 이니시에이터는 데이터 및 메타데이터의 레이아웃은 물론 삭제 코드의 생성과 잠금 관리 및 사용 권한 제어의 일반적인 작업을 조정하는 "캡틴" 역할을 합니다. 관리자는 언제든지 웹 관리 또는 CLI 인터페이스를 통해 해당 워크플로에 더 적절하게 OneFS의 레이아웃 결정을 최적화할 수 있습니다. 관리자는 파일별 또는 디렉토리별로 다음과 같은 액세스 패턴을 선택할 수 있습니다.

- 동시성: 클러스터의 현재 로드를 최적화하고 많은 동시 클라이언트를 지원합니다. 이 설정은 혼합 워크로드에 최상의 성능을 제공합니다.
- 스트리밍: 단일 파일의 고속 스트리밍을 최적화합니다. 예를 들어 단일 클라이언트의 읽기 속도를 크게 향상시킵니다.

- 랜덤: 스트라이핑을 조정하고 프리페치 캐시의 사용을 금지하여 파일에 대한 예측 불가능한 액세스를 최적화합니다.

또한 OneFS에는 실시간 적응형 프리페치가 포함되어 있어 관리자의 개입 없이 인식 가능한 액세스 패턴이 있는 파일에 대한 최적의 읽기 성능을 제공합니다.

ⓘ 현재 OneFS가 지원하는 최대 파일 크기는 OneFS 8.2.2 이상에서 16TB로 증가하여 이전 릴리스의 최대 4TB에서 대폭 증가되었습니다.

OneFS 캐싱

OneFS 캐싱 인프라스트럭처는 클러스터의 각 노드에 있는 캐시를 글로벌 액세스가 가능한 메모리 풀로 통합하도록 설계되어 있습니다. 이를 위해 OneFS에서는 NUMA(Non-Uniform Memory Access)와 유사한 효율적인 메시징 시스템을 사용합니다. NUMA를 사용하면 모든 노드의 메모리 캐시를 클러스터에 구성된 각각의 노드가 사용할 수 있습니다. 원격 메모리는 내부 상호 연결을 통해 액세스되며 하드 디스크 드라이브에 액세스하는 것보다 레이턴시가 훨씬 짧습니다.

원격 메모리 액세스를 위해 OneFS는 기본적으로 분산 시스템 버스와 같은 이중화된 평면 과소구독 이더넷 네트워크를 활용합니다. 원격 메모리 액세스는 로컬 메모리처럼 빠르지는 않지만 여전히 40Gb 이더넷의 짧은 레이턴시로 매우 빠른 편입니다.

OneFS 캐시 서브시스템은 클러스터 전반에 걸쳐 정합성이 보장됩니다. 즉, 동일한 컨텐츠가 여러 노드의 전용 캐시에 존재할 경우 이러한 캐싱된 데이터는 모든 인스턴스 전체에서 정합성이 보장됩니다. OneFS는 캐시 정합성을 유지하기 위해 MESI 프로토콜을 활용합니다. 이 프로토콜은 "쓰기 시 무효화(invalidate-on-write)" 정책을 구현하여 모든 데이터가 공유 캐시 전체에서 정합성이 보장되도록 합니다.

OneFS에서는 최대 3개 레벨의 읽기 캐시와 하나의 NVRAM 지원 쓰기 캐시 또는 결합기(Coalescer)를 사용합니다. 이러한 항목과 이들 항목의 상호 작용에 대한 간략한 설명이 다음 그림에 나와 있습니다.

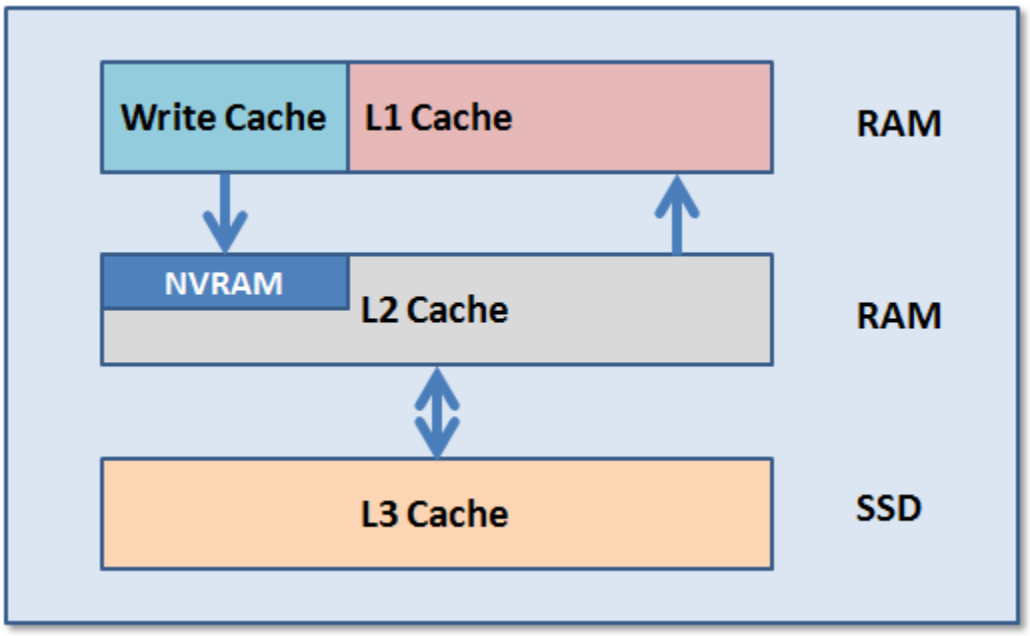

그림 9: OneFS 캐싱 계층 구조

처음 두 유형의 읽기 캐시 L1(Level 1) 및 L2(Level 2)는 메모리(RAM) 기반이며, 프로세서(CPU)에 사용되는 캐시와 유사합니다. 이러한 두 가지 캐시 계층은 모든 플랫폼 스토리지 노드에 존재합니다.

이름	유형	지속성	설명
L1 캐시	RAM	휘발성	프런트엔드 캐시라고도 하며, 클라이언트가 프런트엔드 네트워크를 통해 요청하는 파일 시스템 데이터 및 메타데이터 블록의 클러스터 정합성이 보장되는 클린(clean) 복제본을 유지
L2 캐시	RAM	휘발성	백엔드 캐시, 로컬 노드의 파일 시스템 데이터 및 메타데이터에 대한 클린(clean) 복제본을 포함
SmartCache / 쓰기 결합기(Write Coalescer)	NVRAM	비휘발성	디스크에 커밋되지 않은 보류 중인 모든 쓰기를 프런트엔드 파일에 버퍼링하는 지속적인 배터리 기반 NVRAM 저널 캐시
SmartFlash **L3 캐시**	SSD	비휘발성	L2 캐시에서 제거된 파일 데이터 및 메타데이터 블록을 포함하며 L2 캐시 용량을 효과적으로 늘림

OneFS 캐시 정합성 보장

OneFS 캐시 서브시스템은 클러스터 전반에 걸쳐 정합성이 보장됩니다. 즉, 동일한 컨텐츠가 여러 노드의 전용 캐시에 존재할 경우 이러한 캐싱된 데이터는 모든 인스턴스 전체에서 정합성이 보장됩니다. 예를 들어, 다음과 같은 초기 상태와 일련의 이벤트를 고려해 보십시오.

1. 노드 1 및 노드 5가 각각 공유 캐시 내 주소에 위치한 데이터 복제본을 가지고 있습니다.
2. 쓰기 요청에 대한 응답으로 노드 5가 노드 1의 복제본을 무효화합니다.
3. 노드 5가 값을 업데이트합니다. (아래 참조)
4. 노드 1이 업데이트된 값을 가져오려면 공유 캐시에서 데이터를 다시 읽어야 합니다.

OneFS는 캐시 정합성을 유지하기 위해 MESI 프로토콜을 활용합니다. 이 프로토콜은 "쓰기 시 무효화(invalidate-on-write)" 정책을 구현하여 모든 데이터가 공유 캐시 전체에서 정합성이 보장되도록 합니다. 다음 그림에서는 캐시 내 데이터가 취할 수 있는 다양한 상태와 이러한 상태 간 전환에 대해 보여 줍니다. 그림에 표시된 다양한 상태는 다음과 같습니다.

- M – 수정: 데이터가 로컬 캐시에만 있으며, 공유 캐시의 값으로부터 변경되었습니다. 수정된 데이터는 일반적으로 더티(Dirty)라고 합니다.
- E – 전용: 데이터가 로컬 캐시에만 있지만 공유 캐시의 데이터와 일치합니다. 이 데이터는 대개 클린(Clean)이라고 합니다.
- S – 공유: 로컬 캐시의 데이터가 클러스터에 포함된 다른 로컬 캐시에 있을 수도 있습니다.
- I – 유효하지 않음: 데이터에서 잠금(전용 또는 공유)이 손실되었습니다.

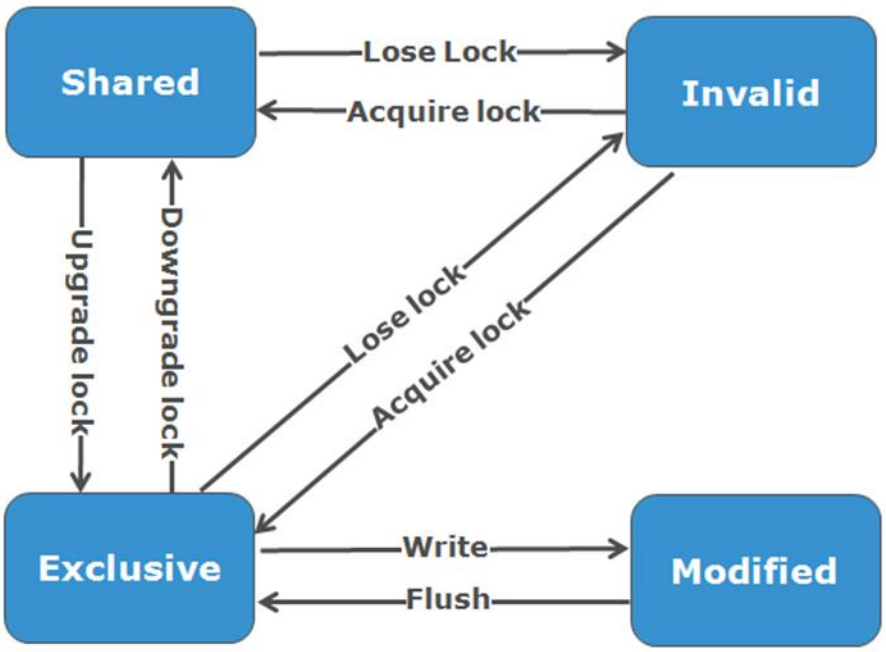

그림 10: OneFS 캐시 정합성 보장 상태 그림

L1 캐시

L1(Level 1) 캐시, 즉 프런트엔드 캐시는 해당 노드에 연결된 클라이언트, 즉 이니시에이터가 사용하는 프로토콜 계층(예: NFS, SMB 등)에 가장 가까이 위치한 메모리입니다. L1 캐시의 주된 용도는 원격 노드에서 데이터를 프리페치하는 것입니다. 데이터는 파일 단위로 프리페치되며, 노드의 백엔드 네트워크 관련 레이턴시를 줄이기 위해 최적화됩니다. 백엔드 상호 연결 레이턴시는 상대적으로 짧으며, L1 캐시 크기 및 요청에 따라 저장되는 일반적인 데이터 크기는 L2 캐시보다 작습니다.

L1은 클러스터의 다른 노드에서 검색되는 데이터를 포함하므로 원격 캐시라고도 합니다. L1은 클러스터 전체에서 정합성이 보장되지만 해당 데이터가 상주하는 노드에서만 사용하고 다른 노드에서는 액세스할 수 없습니다. 스토리지 노드의 L1 캐시에 저장된 데이터는 사용 후 바로 폐기됩니다. L1 캐시는 파일 객체에 대한 오프셋을 통해 데이터가 액세스되는 파일 기반 주소 지정 방식을 사용합니다.

L1 캐시는 이니시에이터와 동일한 노드의 메모리를 나타냅니다. 로컬 노드만 액세스할 수 있으며, 대개 캐시는 데이터의 마스터 복제본이 아닙니다. 다른 코어가 주 메모리에 쓰면 무효화되는 CPU 코어의 L1 캐시와 유사합니다.

L1 캐시 정합성은 위에 설명된 것과 같이 분산 잠금을 사용하는 MESI 유사 프로토콜을 통해 관리됩니다.

OneFS에서는 최근 요청된 inode가 유지되는 전용 inode 캐시도 사용합니다. 클라이언트가 종종 데이터를 캐싱하므로 inode 캐시는 성능에 큰 영향을 미치기 쉽습니다. 또한 많은 네트워크 입출력 작업은 주로 캐싱된 inode에서 신속하게 반환될 수 있는 파일 속성 및 메타데이터에 대한 요청입니다.

ⓘ 디스크 드라이브가 없는 클러스터 Accelerator 노드에서는 L1 캐시가 다르게 활용됩니다. 대신, 모든 데이터를 다른 스토리지 노드에서 가져오므로 L1 캐시가 전체 읽기 캐시로 작동합니다. 또한, 캐시 에이징은 스토리지 노드의 L1 캐시에서 일반적으로 사용되는 삭제(drop-behind) 알고리듬과 달리 LRU(Least Recently Used) 제거(eviction) 정책을 기반으로 합니다. Accelerator의 L1 캐시는 대규모이고 여기에 저장된 데이터는 다시 요청될 가능성이 훨씬 크므로 데이터 블록이 사용 후 캐시에서 즉시 제거되지 않습니다. 그러나 메타데이터 및 업데이트가 많은 워크로드에는 그다지 유용하지 않으며 Accelerator의 캐시는 노드에 직접 연결된 클라이언트에만 유용합니다.

L2 캐시

L2(Level 2) 캐시, 즉 백엔드 캐시는 특정 데이터 블록이 저장되는 노드의 로컬 메모리를 나타냅니다. L2 캐시는 클러스터의 모든 노드에서 전역적으로 액세스할 수 있으며 디스크 드라이브에서 직접 "검색"할 필요가 없으므로 읽기 작업의 지연 시간을 줄이기 위해 사용됩니다. 따라서 원격 노드에서 사용하기 위해 L2 캐시로 프리페치되는 데이터의 크기는 L1 캐시에서보다 훨씬 더 큽니다.

L2 캐시는 또한 해당 노드에 위치한 디스크 드라이브에서 검색되어 원격 노드에서 발생하는 요청에 사용할 수 있는 데이터를 포함하므로 로컬 캐시라고도 합니다. L2 캐시의 데이터는 LRU(Least Recently Used) 알고리듬에 따라 제거됩니다.

L2 캐시의 데이터는 해당 노드에 로컬인 디스크 드라이브에 대한 오프셋을 사용하여 로컬 노드를 기준으로 주소 지정됩니다. 원격 노드에서 요청되는 데이터가 디스크에서 어디에 있는지를 노드가 인식하기 때문에 원격 노드에 대해 지정된 데이터가 훨씬 빠르게 검색됩니다. 원격 노드는 특정 파일 객체에 대한 블록 주소를 조회함으로써 L2 캐시에 액세스합니다. 위에 설명한 바와 같이 이 작업에는 MESI 무효화 작업이 필요하지 않으며 캐시는 쓰기가 진행되는 동안 자동으로 업데이트되고 트랜잭션 시스템 및 NVRAM에 의해 정합성이 유지됩니다.

L3 캐시

선택 사항인 세 번째 읽기 캐시 계층은 SmartFlash 또는 L3(Level 3) 캐시라고 하며, 이 또한 SSD(Solid State Drive)를 포함하는 노드에서 구성 가능합니다. SmartFlash(L3)는 L2 캐시 블록이 오래되어 메모리에서 제거됨에 따라 채워지는 제거 캐시입니다. 캐싱에 기존 파일 시스템 스토리지 디바이스 대신 SSD를 사용하면 여러 가지 이점을 얻을 수 있습니다. 예를 들어, 캐싱에 예약된 경우 전체 SSD가 사용되며 상당히 명확하고 예측 가능한 방식으로 쓰기가 수행됩니다. 따라서 일반적인 파일 시스템을 사용할 때보다 활용률이 훨씬 높고 마모도는 상당히 줄어들며 지속성은 증가합니다. 이는 랜덤 쓰기 워크로드에서 특히 그렇습니다. 또한 SSD를 캐시로 사용하면 스토리지 계층으로 사용할 때보다 SSD 용량을 사이징하기가 훨씬 간단하고 오류 발생 가능성이 줄어듭니다.

다음 그림에서는 클라이언트가 OneFS 읽기 캐시 인프라스트럭처 및 쓰기 결합기(Write Coalescer)와 상호 작용하는 방법을 보여 줍니다. L1 캐시는 필요한 모든 노드의 L2 캐시와 계속 상호 작용하며, L2 캐시는 스토리지 서브시스템 및 L3 캐시 모두와 상호 작용합니다. L3 캐시는 노드 내에서 SSD에 저장되며 동일한 노드 풀에 있는 각 노드에서 L3 캐시가 사용하도록 설정됩니다.

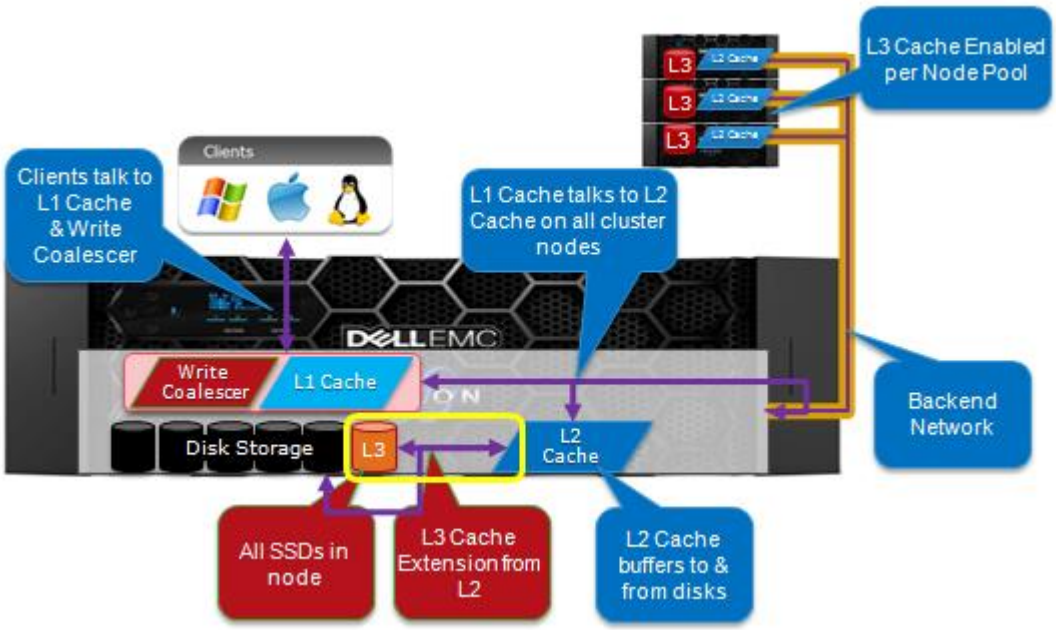

그림 11: OneFS L1, L2 및 L3 캐싱 아키텍처

OneFS에서는 파일을 클러스터의 여러 노드에 쓰고, 한 노드의 여러 드라이브에 쓰기도 하므로 모든 읽기 요청 시 원격(및 경우에 따라 로컬) 데이터 읽기가 수행됩니다. 클라이언트에서 읽기 요청이 도달하면 OneFS에서 요청된 데이터가 로컬 캐시에 있는지 여부를 확인합니다. 로컬 캐시에 있는 모든 데이터를 즉시 읽습니다. 요청된 데이터가 로컬 캐시에 없는 경우 디스크에서 읽어옵니다. 로컬 노드에 없는 데이터의 경우 데이터가 상주하는 원격 노드에 요청이 수행됩니다. 다른 노드 각각에서는 다른 캐시 조회가 수행됩니다. 캐시에 있는 모든 데이터가 즉시 반환되며, 캐시에 없는 데이터는 디스크에서 검색됩니다.

데이터가 로컬 캐시와 원격 캐시(및 가능한 경우 디스크)에서 검색되면 클라이언트로 다시 반환됩니다.

로컬 노드와 원격 노드 모두에서 읽기 요청이 수행되는 단계를 간략히 설명하면 다음과 같습니다.

로컬 노드(요청을 수신하는 노드):

1. 요청된 데이터의 일부가 로컬 L1 캐시에 있는지 여부를 확인합니다. 있는 경우 클라이언트로 반환합니다.
2. 로컬 캐시에 없는 경우 원격 노드에 데이터를 요청합니다.

원격 노드:

1. 요청된 데이터가 로컬 L2 또는 L3 캐시에 있는지 여부를 확인합니다. 있는 경우 요청한 노드로 반환합니다.
2. 로컬 캐시에 없는 경우 디스크로부터 데이터를 읽고 요청한 노드로 반환합니다.

쓰기 캐싱은 클러스터에 데이터를 쓰는 프로세스 속도를 높여줍니다. 작은 쓰기 요청을 큰 청크로 함께 결합하고 디스크에 보내 디스크 쓰기 지연 시간을 대폭 줄임으로써 프로세스 속도를 높입니다. 클라이언트가 클러스터에 데이터를 쓰면, OneFS는 데이터를 즉시 디스크에 쓰지 않고 잠시 이니시에이터 노드의 NVRAM 기반 저널 캐시에 씁니다. 그런 다음 나중에 좀 더 편리한 시간에 이러한 캐싱된 쓰기를 디스크에 플러시합니다. 또한 이와 같은 쓰기는 참가 노드의 NVRAM 저널에도 미러링되어 파일의 보호 요구 사항을 충족합니다. 따라서 클러스터 분할 또는 예기치 않은 노드 운영 중단이 발생할 경우 커밋되지 않은 캐싱된 쓰기가 완전히 보호됩니다.

쓰기 캐시는 다음과 같이 작동합니다.

- NFS 클라이언트가 노드 1에 +2n으로 보호된 파일에 대한 쓰기 요청을 보냅니다.
- 노드 1은 이 쓰기 요청을 NVRAM 쓰기 캐시(빠른 경로)로 받아들인 다음 보호를 위해 참가 노드의 로그 파일에 미러링합니다.
- 쓰기 확인 메시지가 NFS 클라이언트로 즉시 전송되고, 그 결과 디스크 쓰기 지연 시간이 방지됩니다.
- 노드 1의 쓰기 캐시가 가득 차면 캐시가 주기적으로 플러시되고 위에서 설명한 것처럼 적절한 삭제 코드(ECC) 보호(+2n)가 적용된 2단계 커밋 프로세스를 통해 쓰기가 디스크에 커밋됩니다.
- 쓰기 캐시와 참가 노드 로그 파일이 지워지고 새로운 쓰기를 수용할 수 있습니다.

📖 자세한 내용은 OneFS SmartFlash 백서에서 확인할 수 있습니다.

파일 읽기

데이터, 메타데이터, inode는 모두 한 클러스터 내의 여러 노드로 분산되고 노드들 내의 여러 드라이브로도 분산됩니다. 클러스터에서 데이터를 읽거나 쓸 때 클라이언트가 연결된 노드는 이러한 작업의 "캡틴" 역할을 합니다.

읽기 작업에서 "캡틴" 노드는 클러스터의 다양한 노드에서 모든 데이터를 수집하고 일관된 방식으로 요청자에게 제공합니다.

경제적인 업계 표준 하드웨어를 사용하는 클러스터는 필요한 경우 읽기 및 쓰기 작업에 대해 동적으로 할당되는 높은 디스크 대비 캐시 비율(노드당 여러 GB)을 제공합니다. 이러한 RAM 기반 캐시는 클러스터의 모든 노드에서 통합적이고 정합성이 보장되기 때문에 클라이언트가 한 노드에 대해 읽기를 요청할 때 다른 노드에서 이미 처리된 입출력을 활용할 수 있습니다. 이와 같은 캐싱된 블록은 레이턴시가 짧은 백플레인을 통해 모든 노드에서 빠르게 액세스할 수 있으므로 대용량의 효율적인 RAM 캐시가 구축되고 이를 통해 읽기 성능을 대폭 향상시킬 수 있습니다.

클러스터 규모가 확장되면 캐시의 이점도 늘어납니다. 따라서 클러스터에서 디스크의 입출력 양이 대체로 기존 플랫폼보다 상당히 낮기 때문에 레이턴시가 줄어들고 사용자 경험이 개선됩니다.

동시성 또는 스트리밍 액세스 패턴으로 표시된 파일의 경우 OneFS는 SmartRead 구성 요소에서 사용하는 추론을 기반으로 데이터 프리페치를 이용할 수 있습니다. SmartRead는 L2 캐시에서 데이터 "파이프라인"을 생성하여 "캡틴" 노드에서 로컬 "L1" 캐시에 프리페치합니다. 이 기능을 통해 모든 프로토콜에서 순차적 읽기 성능이 향상되고 사실상 단 몇 밀리초 만에 RAM에서 직접 데이터를 읽어 옵니다. 순차성이 높을 경우 SmartRead는 매우 적극적으로 미리 프리페치하여 상당히 높은 데이터 전송 속도로 개별 파일의 읽기 또는 쓰기를 지원합니다.

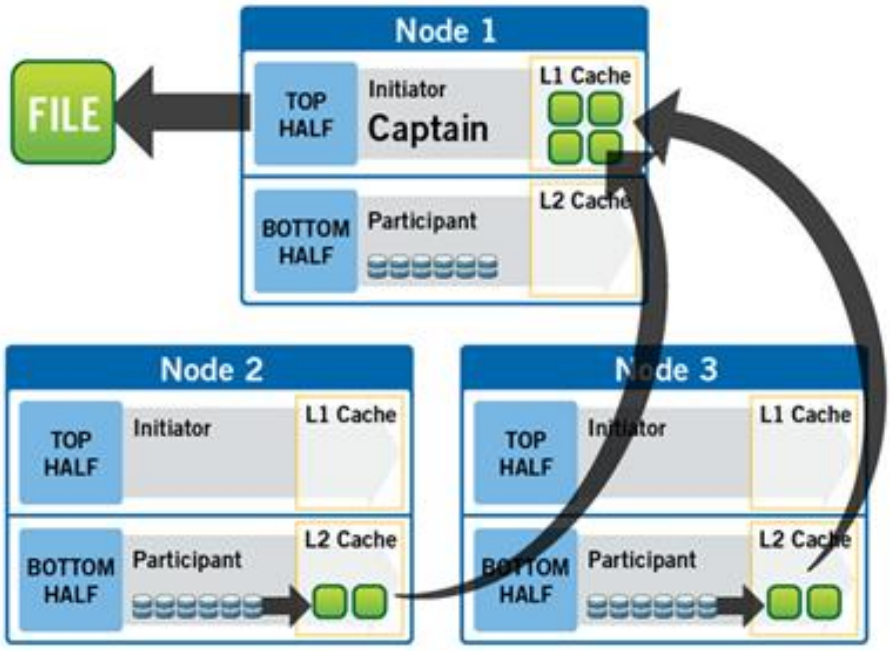

그림 12: 3 노드 클러스터의 파일 읽기 작업

그림 10에서는 SmartRead가 3노드 클러스터의 노드 1에 연결된 클라이언트 요청에 따라 캐싱되지 않고 순차적으로 액세스되는 파일을 읽는 방법을 보여줍니다.

1. 노드 1이 메타데이터를 읽고 파일 데이터의 모든 블록이 있는 위치를 식별합니다.
2. 그리고 L1 캐시에서 요청받은 파일 데이터가 있는지 확인합니다.
3. 노드 1은 디스크에서 파일 데이터를 가져오기 위해 해당 파일 데이터의 일부분을 포함하는 모든 노드로 동시 요청을 보내는 읽기 파이프라인을 구축합니다.
4. 각 노드는 디스크에서 해당 L2 캐시(또는 가능한 경우 L3 SmartFlash 캐시)로 파일 데이터의 블록을 풀링하고 파일 데이터를 노드 1로 전송합니다.
5. 노드 1은 들어오는 데이터를 L1 캐시에 기록하는 동시에 파일을 클라이언트에 제공합니다. 그 동안 프리페치 프로세스가 계속됩니다.
6. 순차성이 높을 경우 필요하다면 L1 캐시의 데이터를 "삭제"하여 다른 L1 또는 L2 캐시 요구를 위해 RAM을 확보할 수 있습니다.

SmartRead의 지능형 캐시 기능은 높은 수준의 동시 액세스에서 매우 높은 읽기 성능을 제공합니다. 무엇보다, 지연 시간이 짧은 클러스터 상호 연결을 통해 노드 1이 로컬 디스크를 액세스하는 것보다 더 빠르게 노드 2의 캐시에서 파일 데이터를 가져올 수 있습니다. SmartRead의 알고리듬은 프리페치의 적극도(랜덤 액세스의 경우 프리페치 해제)와 데이터가 캐시에서 유지되는 기간을 제어하고 데이터가 캐싱되는 위치를 최적화합니다.

잠금 및 동시성

OneFS는 스토리지 클러스터의 모든 노드에서 데이터 잠금을 통제하는 완전히 분산화된 잠금 관리자를 포함합니다. 확장성이 뛰어난 이 잠금 관리자는 다중 잠금 "특성"을 통해 SMB 공유형 잠금 또는 NFS 권고형(Advisory) 잠금 같은 클러스터 정합성이 보장된 프로토콜 레벨 잠금과 파일 시스템 잠금을 모두 지원할 수 있습니다. 또한 OneFS는 CIFS oplocks 및 NFSv4 위임과 같은 위임된 잠금도 지원합니다.

클러스터의 모든 노드는 리소스를 잠그기 위한 코디네이터이고 코디네이터는 고급 해시 알고리듬을 기반으로 잠금 가능한 리소스에 할당됩니다. 이 알고리듬은 코디네이터가 요청의 이니시에이터가 아닌 다른 노드에 위치하도록 설계되었습니다. 파일에 대한 잠금이 요청될 경우 이 잠금은 여러 사용자가 대개 읽기 용도로 잠금을 동시에 공유할 수 있는 공유 잠금 또는 주로 쓰기 용도로 특정 시간에 한 사용자만 허용되는 배타적 잠금일 수 있습니다.

아래의 그림 13은 여러 노드의 스레드가 코디네이터에게 잠금을 요청하는 방식의 예를 보여 줍니다.

1. 노드 2가 이러한 리소스의 코디네이터로 지정됩니다.
2. 노드 4의 스레드 1과 노드 3의 스레드 2가 노드 2의 파일에 대한 공유 잠금을 동시에 요청합니다.
3. 노드 2가 요청된 파일에 대한 배타적 잠금이 있는지 확인합니다.
4. 배타적 잠금이 없을 경우 노드 2는 요청된 파일에 대한 공유 잠금을 노드 4의 스레드 1과 노드 3의 스레드 2에 부여합니다.
5. 이제 노드 3과 노드 4가 요청된 파일에 대한 읽기를 수행합니다.
6. 노드 1의 스레드 3이 노드 3과 노드 4가 읽고 있는 동일한 파일에 대한 배타적 잠금을 요청합니다.
7. 노드 2는 노드 3과 노드 4에서 공유 잠금을 회수할 수 있는지 확인합니다.
8. 노드 3과 노드 4가 계속 읽고 있으므로 노드 2는 노드 1의 스레드 3에게 잠시 기다리도록 요청합니다.
9. 노드 1의 스레드 3은 차단되었다가 노드 2에서 배타적 잠금이 부여된 후 쓰기 작업을 완료합니다.

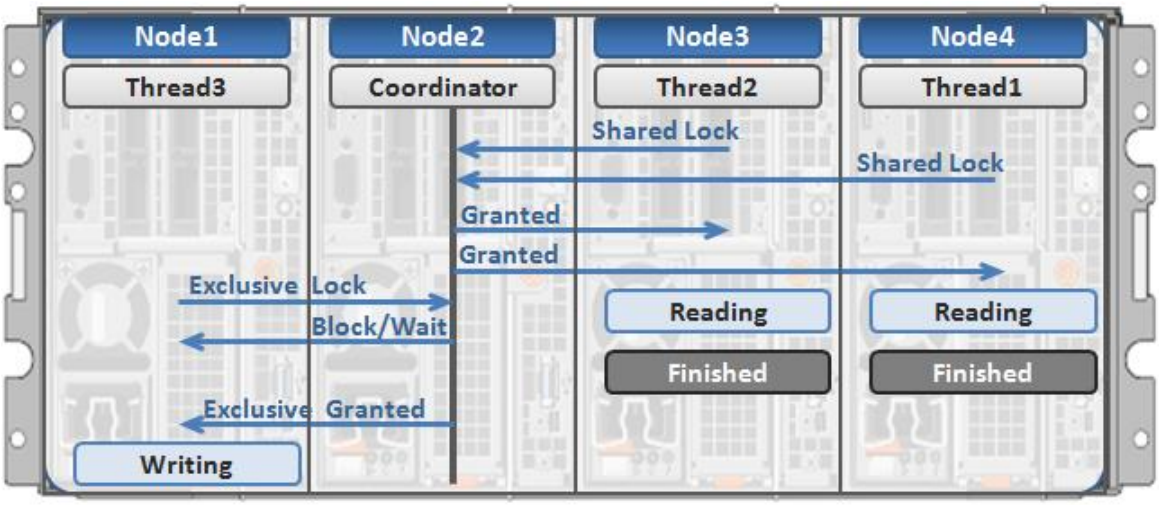

그림 13: 분산 잠금 관리자

멀티 스레드 방식의 입출력

서버 가상화 및 엔터프라이즈 애플리케이션 지원을 위해 대용량 NFS 데이터 저장소를 점점 더 많이 사용하게 되면서 대용량 파일에 대한 높은 처리 성능과 짧은 지연 시간이 요구되고 있습니다. 이러한 요구를 수용하기 위해 OneFS 멀티 작성기는 여러 개의 스레드를 동시에 개별 파일에 쓸 수 있도록 지원합니다.

위의 예에서 대용량 파일에 대한 동시 쓰기 액세스는 전체 파일 레벨에서 적용되는 배타적 잠금 메커니즘에 의해 제한될 수 있습니다. 이 같은 잠재적인 병목 현상을 막기 위해 OneFS 멀티 작성기는 파일을 여러 개의 개별 영역으로 다시 분할하여 전체 파일 레벨이 아닌 개별 영역에 배타적 쓰기 잠금을 부여하는 한층 더 세분화된 쓰기 잠금 기능을 제공합니다. 따라서 여러 클라이언트가 같은 파일의 여러 부분에 동시에 쓸 수 있습니다.

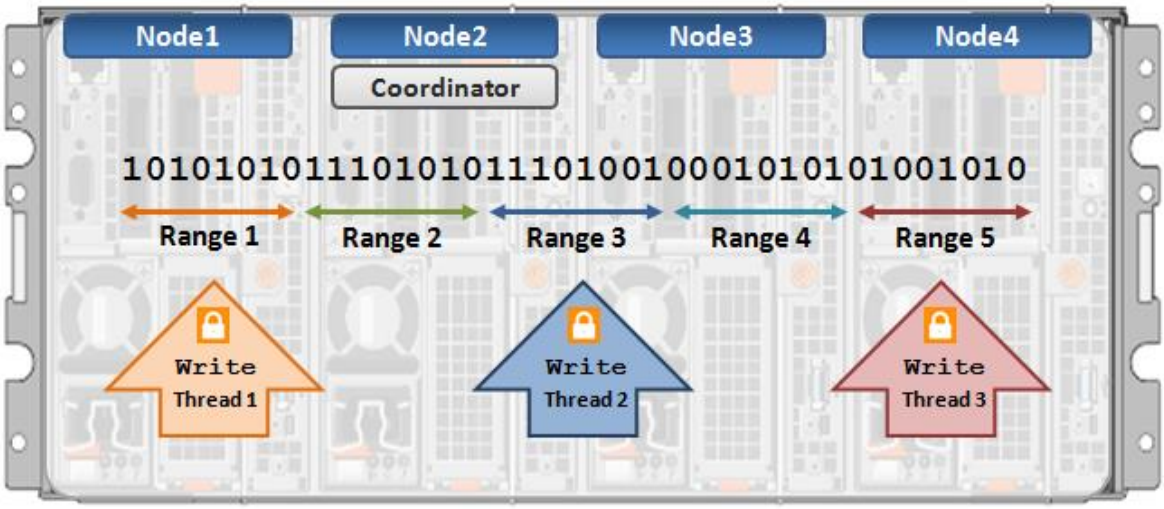

그림 14: 멀티 스레드 방식의 IO 작성기

데이터 보호

전원 손실

파일 시스템의 변경 사항에 대한 정보를 저장하는 파일 시스템 저널은 전원 손실과 같은 시스템 장애 또는 충돌 후 정합성이 보장되는 신속한 복구를 지원하기 위해 설계되었습니다. 파일 시스템은 노드 또는 클러스터가 전원 손실 또는 기타 운영 중단에서 복구된 후 저널 항목을 재생합니다. 저널을 사용하지 않을 경우 파일 시스템은 장애 후 "fsck" 또는 "chkdsk" 작업을 통해 모든 가능한 변경 사항을 일일이 조사하고 검토해야 하며 대규모 파일 시스템에서는 이 작업이 오래 걸릴 수 있습니다.

OneFS는 파일 시스템에 대한 커밋되지 않은 쓰기를 보호하기 위해 사용되는 배터리 구동 NVRAM 카드가 각 노드에 내장되어 있는 저널링 파일 시스템입니다. 충전된 NVRAM 카드 배터리는 며칠 동안 재충전하지 않고 유지됩니다. 노드가 부팅될 때 저널을 확인하고 저널링 시스템에서 필요하다고 판단한 디스크에 대해 선별적으로 트랜잭션을 재생합니다.

OneFS는 시스템에 아직 없는 모든 트랜잭션이 기록되었음을 보장할 수 있는 경우에만 마운트됩니다. 예를 들어 적절한 종료 절차를 따르지 않고 NVRAM 배터리가 방전된 경우 트랜잭션이 손실되었을 수도 있습니다. 이 경우 잠재적인 문제를 방지하기 위해 노드가 파일 시스템을 마운트하지 않습니다.

하드웨어 장애와 쿼럼

클러스터가 제대로 작동하고 데이터 쓰기를 받아들이기 위해서는 노드의 쿼럼(quorum)이 활성 상태이고 응답해야 합니다. 쿼럼은 다음과 같이 단순 다수로 정의됩니다. 노드가 있는 클러스터는 쓰기를 허용하려면 [/2]+1개의 노드가 온라인 상태여야 합니다. 예를 들어 7노드 클러스터에서는 쿼럼을 구성하기 위해 4개의 노드가 필요합니다. 노드 또는 노드 그룹이 작동하고 응답하지만 쿼럼의 구성원이 아니면 읽기 전용 상태로 실행됩니다.

OneFS는 클러스터가 일시적으로 두 개의 클러스터로 분할되는 경우 초래될 수 있는 "스플릿 브레인"(split-brain) 현상을 방지하기 위해 쿼럼을 사용합니다. 쿼럼 규칙을 따름으로써 Isilon 아키텍처는 장애가 발생하거나 온라인으로 복구된 노드 수와 관계없이 쓰기가 수행되는 경우 이전에 수행된 모든 쓰기와의 정합성을 보장할 수 있습니다. 또한 쿼럼은 특정 데이터 보호 수준으로 전환하는 데 필요한 노드 수를 결정합니다. +의 삭제 코드 기반 보호 수준의 경우 클러스터는 2+1개 이상의 노드를 포함해야 합니다. 예를 들어 +3n 구성에서는 최소 7개의 노드가 필요합니다. 이 구성에서는 3개의 노드가 동시에 손실되어도 4개 노드의 쿼럼이 여전히 유지되므로 클러스터가 계속 정상 작동할 수 있습니다. 클러스터가 쿼럼 미만으로 떨어지면 파일 시스템이 자동으로 읽기 전용의 보호되는 상태가 되어 쓰기를 거부하지만 사용 가능한 데이터에 대한 읽기 액세스는 계속 허용합니다.

하드웨어 장애 - 노드 추가/제거

GMP(Group Management Protocol)이라고 하는 시스템이 언제든지 클러스터의 상태를 전역적으로 파악할 수 있도록 해 주고 전체 클러스터에서 모든 다른 노드의 상태를 보여 주는 일관된 뷰를 보장합니다. 하나 이상의 노드가 클러스터 상호 연결을 통해 연결할 수 없게 되면 이 그룹은 클러스터에서 "분리" 또는 제거됩니다. 모든 노드는 해당 클러스터에 대한 새로운 일관된 뷰로 확인됩니다. 클러스터가 두 개의 별도 노드 그룹으로 분할된 것으로 생각하면 됩니다. 단, 오직 한 개의 그룹만이 쿼럼을 가질 수 있습니다. 이렇게 분할된 상태에서는 파일 시스템의 모든 데이터에 액세스할 수 있고 쿼럼을 포함하는 그룹의 경우 수정이 가능합니다. "다운"된 디바이스에 저장된 데이터는 클러스터에 저장된 이중화 데이터를 사용하여 재구축됩니다.

이 노드가 다시 연결할 수 있게 되면 "병합" 또는 추가 작업이 수행되고 노드가 클러스터로 복구됩니다. 두 그룹이 다시 하나로 병합됩니다. 재구축 및 재구성하지 않아도 노드가 클러스터에 다시 연결될 수 있습니다. 이 점이 드라이브를 재구축해야 하는 하드웨어 RAID 스토리지와 다른 점입니다. 분할 중에 보호 그룹 중 일부가 덮어써지고 더 좁은 간격의 스트라이프로 변환되었을 경우 AutoBalance가 효율성 향상을 위해 일부 파일을 다시 스트라이핑할 수 있습니다.

또한 OneFS 작업 엔진에는 고립 항목 수집기 역할을 하는 Collect 프로세스가 포함되어 있습니다. 쓰기 작업 동안 클러스터가 분할될 경우 해당 파일에 대해 할당된 일부 블록을 쿼럼 쪽에서 다시 할당해야 할 수 있습니다. 이 작업은 비 쿼럼 쪽에서 할당된 블록을 "고립"시킵니다. 클러스터가 다시 병합될 때 Collect 작업은 병렬 처리되는 표시 및 스윕(mark and sweep) 검색을 통해 이러한 고립된 블록을 찾아내고 클러스터의 사용 가능한 용량으로 재확보합니다.

확장 가능한 재구축

OneFS는 데이터를 할당하거나 장애 후 데이터를 재구성할 때 하드웨어 RAID에 의존하지 않습니다. 대신 OneFS는 파일 데이터 보호를 직접 관리하고 장애가 발생할 경우 병렬 처리 방식으로 데이터를 재구축합니다. OneFS는 디스크 외부에서 직접 선형 영역의 inode 데이터를 읽어 일정 시간에 장애의 영향을 받는 파일을 확인할 수 있습니다. 영향을 받는 파일은 작업 엔진에 의해 클러스터 노드 사이에 분산되는 작업자 스레드 세트에 할당됩니다. 작업자 노드는 파일을 병렬로 복구합니다. 이는 클러스터 크기가 증가하면 장애에서 재구축하는 데 걸리는 시간이 감소함을 의미합니다. 이와 같은 특징은 클러스터의 크기가 늘어날 때 클러스터의 복구 성능을 유지하는 측면에서 월등히 높은 효율성을 제공합니다.

가상 핫 스페어

RAID를 기반으로 하는 대부분의 기존 스토리지 시스템은 장애가 발생한 드라이브를 독립적으로 복구하기 위해 하나 이상의 "핫 스페어" 드라이브를 프로비저닝해야 합니다. 이 핫 스페어 드라이브는 RAID 세트에서 장애가 발생한 드라이브를 대체합니다. 추가 장애가 나타나기 전에 이러한 핫 스페어가 대체되지 않을 경우 시스템이 치명적인 데이터 손실 위험에 빠집니다. OneFS는 핫 스페어 드라이브를 사용하지 않으며, 장애에서 복구하기 위해 시스템에서 사용 가능한 용량으로부터 단순히 용량을 빌려옵니다. 이 기술을 가상 핫 스페어라고 합니다. 이렇게 하면 사용자 개입 없이 클러스터가 완전히 자가 복구될 수 있습니다. 관리자가 가상 핫 스페어 예약을 생성하면 사용자가 지속적으로 쓰는 상태에서도 시스템 자가 복구가 가능합니다.

삭제 코딩을 통한 파일 레벨 데이터 보호

클러스터는 클러스터의 데이터 서비스를 막지 않고 한 번 이상의 동시 구성 요소 장애를 감당하도록 설계되었습니다. 이를 위해 OneFS는 Reed-Solomon 오류 수정을 통한 삭제 코드 기반 보호(N+M 보호) 또는 미러링 시스템을 사용하여 파일을 보호합니다. 데이터 보호가 소프트웨어에서 파일 레벨로 적용되므로 시스템이 전체 파일 세트 또는 볼륨을 검사하고 복구할 필요 없이 장애로 손상된 파일만 복구하는 데 집중할 수 있습니다. OneFS 메타데이터와 inode는 Reed-Solomon 코딩이 아닌 미러링을 통해 이들 항목이 참조하는 데이터의 보호 수준 이상으로 항상 보호됩니다.

모든 데이터, 메타데이터 및 보호 정보가 클러스터의 모든 노드 사이에 분산되므로 클러스터에는 메타데이터를 관리할 전용 패리티 노드 또는 드라이브나 전용 디바이스 또는 디바이스 세트가 필요하지 않습니다. 따라서 어떤 노드도 단일 장애 지점이 되지 않습니다. 모든 노드가 수행할 작업을 균등하게 분담하여 P2P 아키텍처에서 완벽하게 균형이 잡힌 로드 밸런싱을 제공합니다.

OneFS에서는 구성 가능한 데이터 보호 설정을 여러 레벨로 제공하며, 이 레벨은 클러스터나 파일 시스템을 오프라인 상태로 전환할 필요 없이 언제든지 수정할 수 있습니다.

삭제 코드로 보호되는 파일의 경우 각 해당 보호 그룹은 N+M/b 레벨로 보호됩니다. 여기서 N>M이고 M>=b입니다. N과 M 값은 보호 그룹 내에서 데이터에 사용된 드라이브의 수와 삭제 코드에 사용된 드라이브의 수를 각각 나타냅니다. b 값은 이 보호 그룹을 배치하는 데 사용된 데이터 스트라이프의 수와 관련이 있으며 이에 대해서는 아래에서 설명합니다. 쉽게 이해할 수 있는 일반적인 케이스는 b=1이며 보호 그룹이 다음을 포함함을 시사합니다. N개 드라이브 분량의 데이터와 삭제 코드에 저장된 M개 드라이브 분량의 이중화 항목을 포함하고 이 보호 그룹이 전체 노드 세트에서 정확히 하나의 스트라이프에 배치되어야 함을 의미합니다. 이렇게 하면 보호 그룹의 M개 구성원에서 동시에 장애가 발생해도 문제가 없고 여전히 100% 데이터 가용성을 보장할 수 있습니다. M 삭제 코드 구성원 수는 N 데이터 구성원 수를 기준으로 계산됩니다. 아래의 그림 13은 일반적인 4+2 보호 그룹(N=4, M=2, b=1)의 경우를 보여 줍니다.

OneFS는 파일을 여러 노드에 걸쳐 스트라이핑하기 때문에 이는 N+M으로 스트라이핑된 파일은 번의 동시 노드 장애를 가용성 손실 없이 견딜 수 있음을 나타냅니다. 따라서 OneFS는 장애가 드라이브, 노드 또는 노드 내 구성 요소(예: 카드) 중 어디에서 발생하든 모든 장애 유형에서 복구 성능을 제공합니다. 또한 노드는 장애가 발생한 내부 구성 요소의 수 또는 유형과 관계없이 단일 장애로 계산됩니다. 즉, 노드 내에서 5개의 드라이브에 장애가 발생해도 N+M 보호를 위해 단일 장애로만 계산됩니다.

OneFS는 최대 4개의 가변 레벨 M을 이용해 4중 장애 보호를 제공하는 고유한 기능이 있습니다. 이는 현재 널리 사용되는 RAID의 최대 수준을 현저히 앞서는 것으로, RAID-6 장애 보호 기능의 두 배입니다. 이와 같은 이중화 수준을 통해 스토리지의 신뢰성이 대폭 향상되기 때문에 +4n 보호는 기존 하드웨어 RAID보다 몇 배 뛰어난 신뢰성을 제공할 수 있습니다. 이러한 추가적인 보호 기능 덕분에 4TB 드라이브 및 6TB 드라이브 같은 대용량 SATA 드라이브를 안심하고 추가할 수 있습니다.

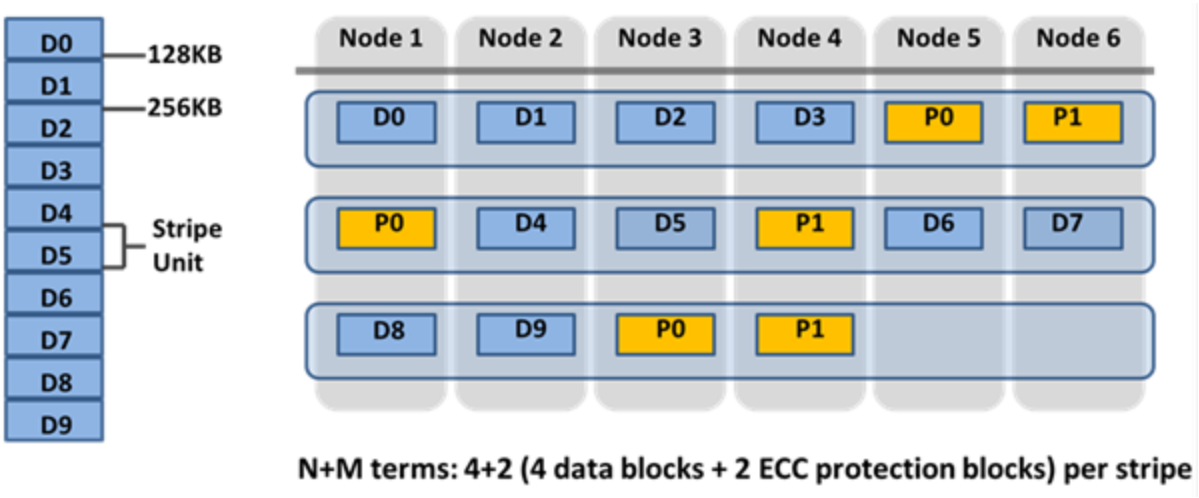

그림 15: OneFS 이중화 – N+M 삭제 코드 보호

더 작은 클러스터는 +1n으로 보호할 수 있지만, 이 경우 단일 드라이브 또는 노드는 복구할 수 있는 반면 서로 다른 두 노드에 있는 두 개의 드라이브는 복구할 수 없습니다. 드라이브 장애는 노드 장애보다 발생할 가능성이 몇 배 더 큽니다. 대용량 드라이브가 포함된 클러스터의 경우 단일 노드 복구 성능이 양호한 수준이기는 하지만 여러 드라이브 장애를 수용할 수 있는 보호 수준을 제공하는 것이 좋습니다.

이중 디스크 이중화와 단일 노드 이중화를 사용하기 위해 두 배 또는 세 배 너비의 크기로 보호 그룹을 구축할 수 있습니다. 이러한 두 배 또는 세 배 너비의 보호 그룹이 구성되면 이는 동일한 노트 세트를 구성된 대로 한두 번 더 보호합니다. 각 보호 그룹이 정확히 두 개 디스크 분량의 이중화를 포함하므로 이 메커니즘을 통해 클러스터는 2~3개 드라이브 장애 또는 전체 노드 장애를 데이터 가용성 손실 없이 감당할 수 있습니다.

무엇보다 소규모 클러스터의 경우 이 같은 스트라이핑 방식은 M/(N+M)의 온디스크 효율성을 기반으로 매우 뛰어난 효과를 발휘합니다. 예를 들어 이중 장애 보호가 적용된 5노드 클러스터에서 N=3, M=2를 사용할 경우 효율성이 1−2/5 즉 60%인 3+2 보호 그룹을 확보하게 됩니다. 동일한 5노드 클러스터를 사용하지만 각 보호 그룹을 2개의 스트라이프에 배치하면 N=8이 되고 M=2이므로 1-2/(8+2) 즉 80%의 온디스크 효율성을 얻을 수 있으며 이중 노드 장애 보호만 포기하고 이중 드라이브 장애 보호를 계속 유지할 수 있습니다.

OneFS는 보호 체계를 몇 가지 지원합니다. 여기에는 두 개의 드라이브 오류 또는 하나의 노드 오류에 대해 보호 기능을 제공하는 유비쿼터스 +2d:1n이 포함됩니다.

ⓘ 특정 클러스터 구성에 권장되는 보호 수준을 사용하는 것이 모범 사례로 가장 좋습니다. 이 권장 보호 수준은 OneFS WebUI 스토리지 풀 구성 페이지에서 'suggested'로 명확하게 표시되며 일반적으로 기본값으로 설정되어 있습니다. 현재 모든 Gen6 하드웨어 구성의 경우 권장되는 보호 수준은 "+2d:1n"입니다.

하이브리드 보호 체계는 Gen6 섀시 고집적 노드 구성에 특히 유용하며, 이러한 구성에서는 여러 드라이브에 장애가 발생할 확률이 전체 노드 장애가 발생할 확률보다 훨씬 높습니다. 드물지만 여러 디바이스가 동시에 장애를 일으킬 경우, 즉 파일이 "적용된 보호 수준"을 벗어날 경우 OneFS는 가능한 모든 데이터에 보호 수준을 다시 적용하고 장애로 인해 영향을 받은 개별 파일의 오류를 클러스터의 로그에 기록합니다.

OneFS는 2배부터 8배까지 다양한 미러링 옵션을 제공합니다. 따라서 지정된 컨텐츠의 미러본을 2개에서 8개까지 생성할 수 있습니다. 예를 들어 메타데이터는 기본적으로 FEC보다 한 수준 위에 미러링됩니다. 예를 들어 +2n 수준으로 보호되는 파일의 경우 관련 메타데이터 객체가 3x 미러링됩니다.

OneFS 보호 수준의 모든 범위는 다음 표에 요약되어 있습니다.

보호 수준	설명
+1n	드라이브 1개 또는 노드 1개의 장애가 허용됩니다.
+2d:1n	드라이브 2개 또는 노드 1개의 장애가 허용됩니다.
+2n	드라이브 2개 또는 노드 2개의 장애가 허용됩니다.
+3d:1n	드라이브 3개 또는 노드 1개의 장애가 허용됩니다.
+3d:1n1d	드라이브 3개 또는 노드 1개와 드라이브 1개의 장애가 허용됩니다.
+3n	드라이브 3개 또는 노드 3개의 장애가 허용됩니다.
+4d:1n	드라이브 4개 또는 노드 1개의 장애가 허용됩니다.
+4d:2n	드라이브 4개 또는 노드 2개의 장애가 허용됩니다.
+4n	노드 4개의 장애가 허용됩니다.
2x ~ 8x	구성에 따라 2개~8개 노드가 미러링됩니다.

OneFS를 통해 관리자는 클라이언트가 연결되어 데이터를 읽고 쓰는 상태에서 실시간으로 보호 정책을 수정할 수 있습니다.

ⓘ 클러스터의 보호 수준이 높아지면 클러스터의 데이터가 사용하는 용량도 증가할 수 있습니다.

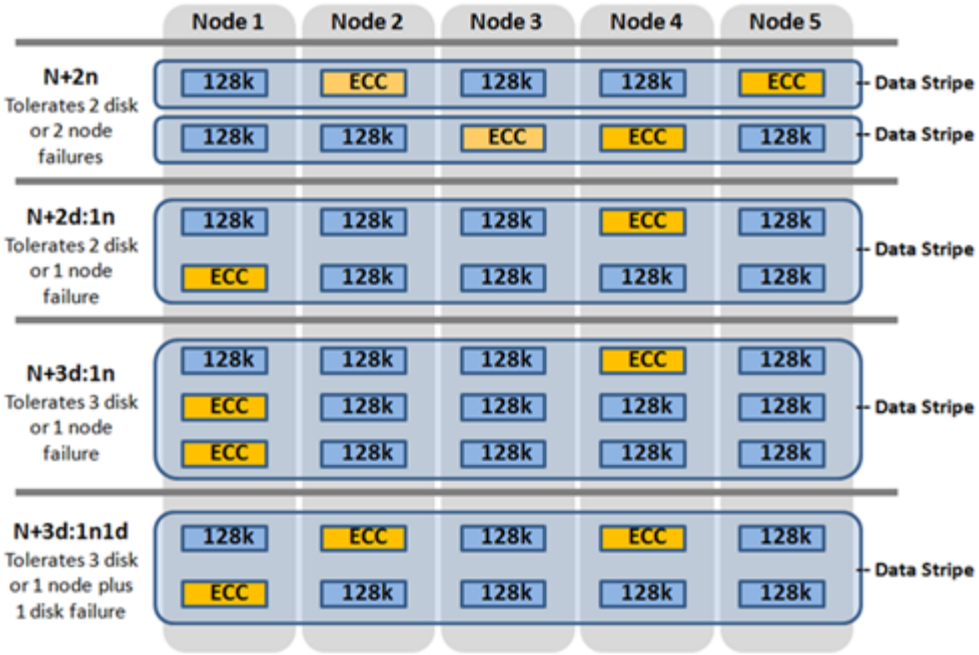

그림 16: OneFS 하이브리드 삭제 코드 보호 체계

ⓘ 또한 OneFS는 새 클러스터 설치에 대한 보호 수준이 낮을 경우 알림을 제공합니다. 클러스터의 보호 수준이 낮을 경우 CELOG(Cluster Event Logging) 시스템은 알림을 생성하여 관리자에게 보호 부족을 경고하고 특정 클러스터 구성의 보호 수준을 적절한 수준으로 변경하도록 권장합니다.

📖 자세한 내용은 OneFS 고가용성 및 데이터 보호 백서에서 확인할 수 있습니다.

자동 파티셔닝

OneFS에서 데이터 계층화와 데이터 관리는 SmartPools 프레임워크를 통해 처리됩니다. 데이터 보호 및 레이아웃 효율성의 관점에서 보면 SmartPools는 대용량의 동종 노드를 'MTTDL(Mean Time To Data Loss)'에 더 적합한 작은 디스크 풀로 분할할 수 있도록 지원합니다. 예를 들어 노드 80개로 구성된 H500 클러스터는 대개 +3d:1n1d의 보호 수준으로 실행됩니다. 그런데 이 클러스터를 20개의 노드로 구성된 디스크 풀 4개로 분할하면 각각의 풀을 +2d:1n 보호 수준으로 실행할 수 있기 때문에 관리 부담을 가중시키지 않으면서도 보호를 위한 오버헤드를 절감하고 공간 활용도를 높일 수 있습니다.

스토리지 관리를 간소화한다는 설계 목표에 따라 OneFS는 MTTDL과 효율적인 공간 활용에 최적화된 디스크 풀 또는 '노드 풀' 수를 자동으로 계산하여 클러스터를 분할합니다. 즉, 위에서 언급한 예의 80개 노드로 구성된 클러스터와 같은 보호 수준을 고객이 직접 결정할 필요가 없습니다.

자동 프로비저닝을 통해 상응하는 모든 노드 하드웨어 세트가 최대 40개의 노드와 노드당 드라이브 6개로 구성된 디스크 풀로 자동 분할됩니다. 이렇게 분할된 노드 풀은 기본적으로 +2d:1n 보호 수준으로 보호되며, 여러 풀을 논리적 계층으로 결합하여 SmartPools 파일 풀 정책을 통해 관리할 수 있습니다. 노드의 디스크를 보호되는 여러 개의 개별 풀로 세분화하기 때문에 복수의 디스크 장애가 발생했을 때 노드의 복구 성능이 이전보다 크게 향상됩니다.

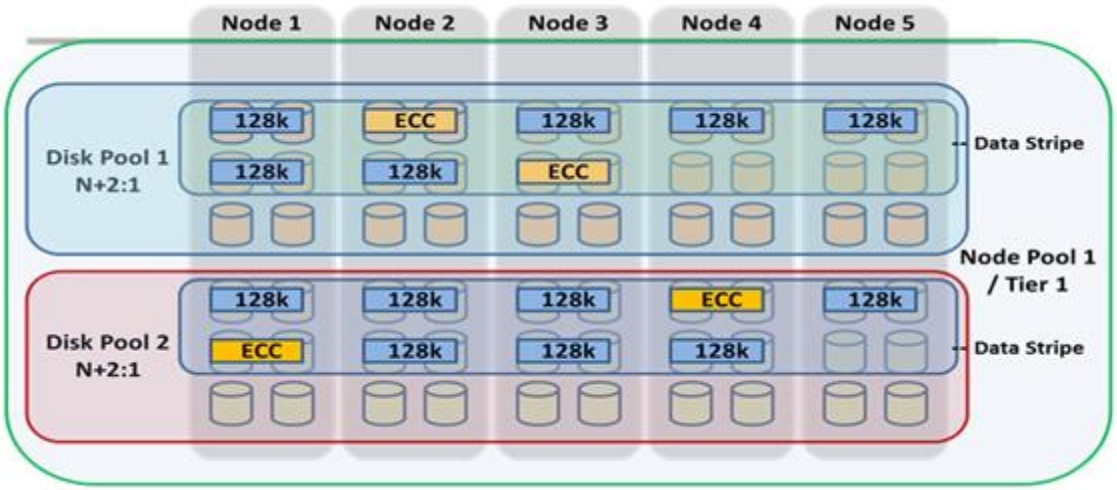

그림 17: SmartPools 를 통한 자동 파티셔닝

📖 자세한 내용은 SmartPools 백서에서 확인할 수 있습니다.

PowerScale Gen6 모듈형 하드웨어 플랫폼은 단일 4RU 섀시에 4개의 노드가 포함된 고집적 모듈형 설계가 특징입니다. 이 방법은 OneFS 장애 도메인 개념에 또 다른 차원의 복원력을 추가하는 디스크 풀, 노드 풀 및 '이웃(neighborhood)'의 개념을 강화합니다. 각 Gen6 섀시에는 4개의 컴퓨팅 모듈(노드당 1개) 및 노드당 5개의 드라이브 컨테이너 또는 슬레드가 포함되어 있습니다.

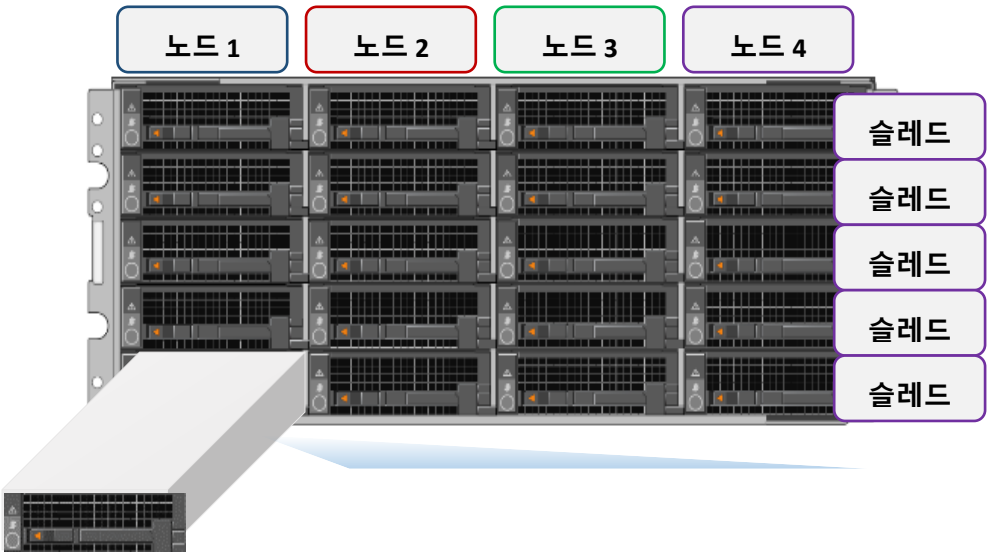

그림 18. 드라이브 슬레드를 보여주는 Gen6 플랫폼 섀시 전면

각 슬레드는 섀시의 전면에 장착되는 트레이로서 특정 섀시의 구성에 따라 3~6개의 드라이브를 포함합니다. 디스크 풀은 스토리지 풀 계층 구조 내에서 가장 작은 단위입니다. OneFS 프로비저닝의 기본 원리는 유사한 노드의 드라이브를 여러 세트 또는 디스크 풀로 나누는 것으로, 각 풀이 별도의 장애 도메인을 나타냅니다. 이러한 디스크 풀은 기본적으로 +2d:1n(2개의 드라이브 장애 또는 1개의 전체 노드 장애 허용)으로 보호됩니다.

디스크 풀은 각 Gen6 노드의 5개 슬레드 모두에 배치됩니다. 예를 들어, 슬레드당 드라이브가 3개인 노드에는 다음과 같은 디스크 풀 구성이 있습니다.

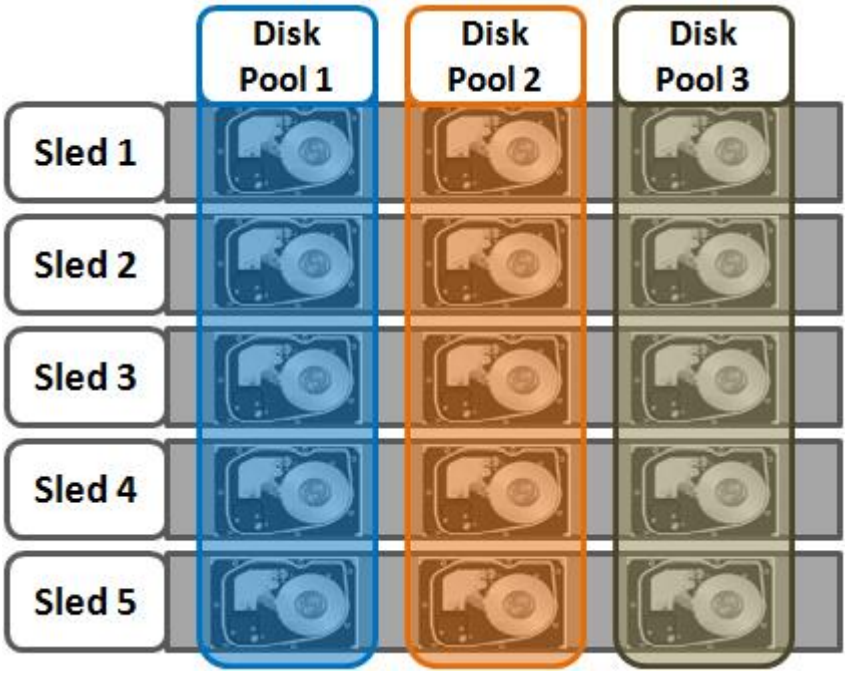

그림 19. OneFS 디스크 풀

노드 풀은 유사한 스토리지 노드(호환성 클래스)에 분산된 디스크 풀 그룹입니다. 아래 그림 20을 참조하십시오. 서로 다른 노드 유형의 여러 그룹이 하나의 이기종 클러스터에서 함께 작동할 수 있습니다. 예를 들어 I/OPS 작업이 많은 애플리케이션에 사용되는 F-Series 노드 풀과 동시 및 순차 워크로드가 많은 작업에 주로 사용되는 H-Series 노드 풀과 주로 니어라인 및 장기 아카이브 워크로드용으로 사용되는 A-Series 노드 풀이 하나씩 있을 수 있습니다.

따라서 OneFS에서 SSD, 고속 SAS, 대용량 SATA 등 여러 드라이브 미디어 유형으로 구성된 단일 스토리지 리소스 풀을 생성하여 다양한 성능, 보호 및 용량 특성을 제공할 수 있고, 중앙 집중식 관리 기능을 통해 이러한 이기종 스토리지 풀에서 광범위한 애플리케이션 및 워크로드 요구 사항을 지원할 수 있습니다. 또한 구형 및 신형 하드웨어를 조합함으로써 여러 세대의 제품에서 상당한 투자 보호 효과를 쉽게 얻을 수 있고 원활하게 하드웨어를 교체할 수 있습니다.

각 노드 풀은 동일한 스토리지 노드 유형의 디스크 풀만 포함하고 있으며, 디스크 풀은 하나의 노드 풀에만 속할 수 있습니다. 예를 들어 한 노드에 1.6TB SSD 드라이브가 있는 F-Series 노드는 한 풀에 있는 반면, 10TB SATA 드라이브가 있는 A-Series 노드는 다른 풀에 있게 됩니다. 현재, PowerScale H700과 같은 Gen6 하드웨어의 경우 노드 풀당 최소 4개의 노드(1개의 섀시)가 필요하고 PowerScale F900과 같은 독립 노드의 경우 풀당 3개의 노드가 필요합니다.

OneFS '이웃(neighborhood)'은 노드 풀 내에 있는 장애 도메인이며, 그 목적은 전반적인 안정성을 개선하고 드라이브 슬레드를 실수로 제거하여 데이터를 사용할 수 없게 되는 상황을 방지하는 것입니다. PowerScale F200과 같은 독립형 노드의 경우 OneFS에는 노드 풀당 20개 노드가 이상적인 규모이며, 최대 노드 규모는 39개입니다. 40번째 노드가 추가되면 노드는 20개의 노드로 구성된 이웃 2개로 분할됩니다.

Gen6 플랫폼을 사용하면 이웃의 이상적인 규모가 20개 노드에서 10개 노드로 변경됩니다. 이렇게 하면 동시에 발생하는 노드 쌍의 저널 장애 및 전체 섀시 장애를 보호할 수 있습니다.

파트너 노드는 저널이 미러링된 노드입니다. Gen6 플랫폼을 사용하면 각 노드가 이전 플랫폼과 마찬가지로 NVRAM에 저널을 저장하는 대신 노드의 저널이 SSD에 저장되고 모든 저널에 다른 노드의 미러 복사본이 있습니다. 미러링된 저널을 포함하는 노드를 파트너 노드라고 합니다. 저널 변경을 통해 얻을 수 있는 안정성 측면의 이점이 몇 가지 있습니다. 예를 들어, SSD는 상태 유지를 위해 충전된 배터리가 필요한 NVRAM보다 영구적이고 안정성이 높습니다. 또한 미러링된 저널을 사용하면 저널이 손실된 것으로 간주되기 전에 두 저널 드라이브가 모두 종료되어야 합니다. 따라서 미러링된 저널 드라이브 모두에서 장애가 발생하지 않는 한, 두 파트너 노드가 모두 정상적으로 작동할 수 있습니다.

가능한 경우 파트너 노드 보호를 통해 노드가 서로 다른 이웃에 배정되므로 장애 도메인이 달라집니다. 클러스터가 5개의 전체 섀시(20개 노드)에 도달하면 OneFS가 첫 번째 이웃 분할 후 파트너 노드를 다른 이웃에 배정하여 파트너 노드 보호가 가능합니다.

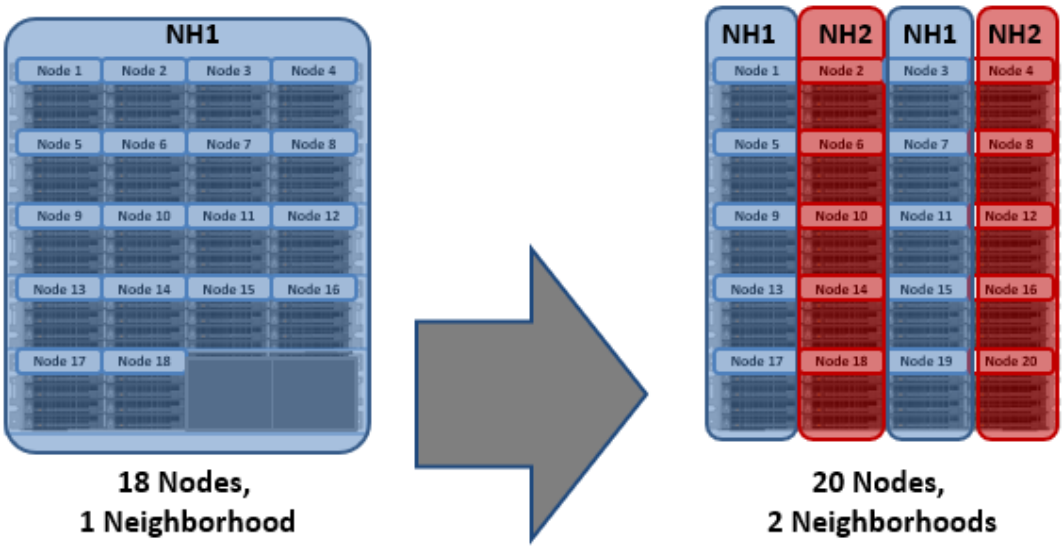

그림 20. 20개의 노드가 2개의 이웃으로 분할

두 노드가 모두 작동을 멈추더라도 서로 다른 장애 도메인에 있어 장애 도메인에는 단일 노드의 손실만 발생하므로 파트너 노드 보호를 통해 안정성이 향상됩니다.

섀시 보호를 통해 가능한 경우 섀시 내의 네 개의 노드가 각각 서로 다른 이웃에 배정됩니다. 40개의 노드가 이웃으로 분할되면서 섀시 내의 모든 노드가 서로 다른 이웃에 배정되어 40개의 노드에서 섀시 보호가 가능합니다. 따라서 Gen6 클러스터를 38개 노드에서 40개 노드로 확장하면 두 개의 기존 이웃이 10개의 노드로 구성된 이웃 4개로 분할됩니다.

섀시 보호는 전체 섀시에 장애가 발생 시 각 장애 도메인에 하나의 노드만 손실되도록 합니다.

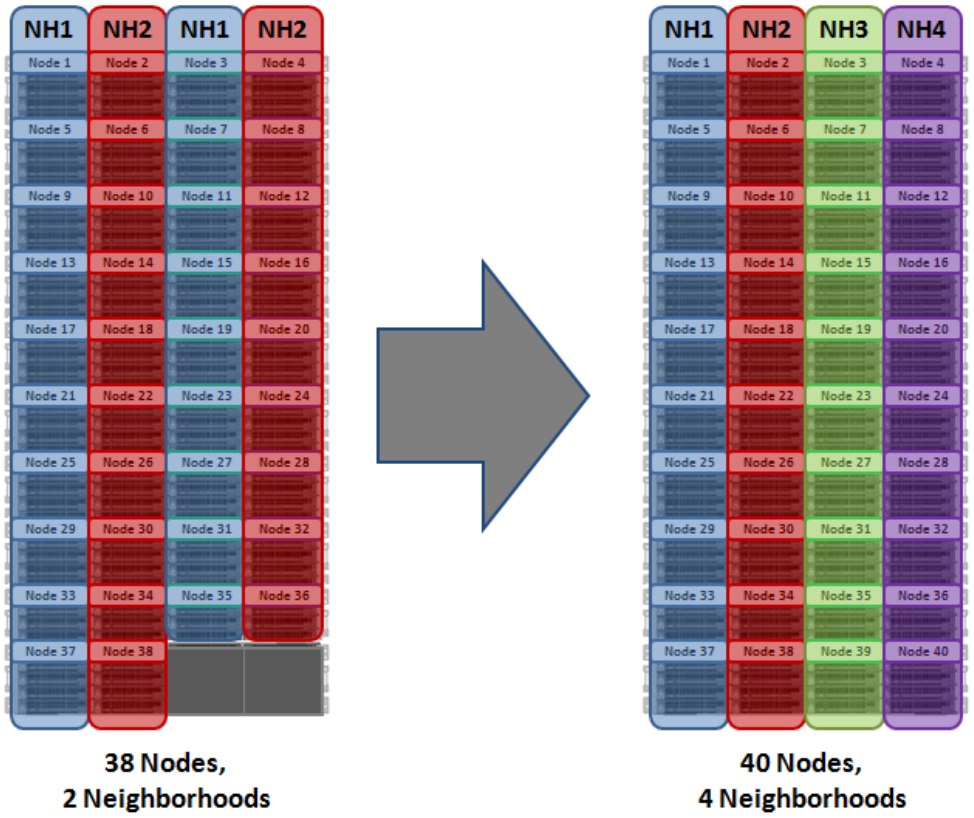

그림 21. OneFS 이웃 – 4 개의 이웃으로 분할.

ⓘ 기본 수준인 +2d:1n으로 보호되는 40개 이상의 노드로 구성된 클러스터는 이웃당 하나의 노드 장애를 감당할 수 있습니다. 이렇게 하면 단일 Gen6 섀시 장애로부터 클러스터를 보호할 수 있습니다.

전반적으로, Gen6 플랫폼 클러스터는 다음과 같은 향상된 기능의 직접적인 결과로 유사한 용량의 이전 세대 클러스터보다 안정성이 적어도 몇 배 이상 뛰어납니다.

- 미러링된 저널
- 더 작아진 이웃 규모
- 미러링된 부팅 드라이브

호환성

유사하지만 동일하지 않은 특정 노드 유형은 노드 호환성에 따라 기존 노드 풀에 프로비저닝될 수 있습니다. OneFS에서는 노드 풀에 최소 3개의 노드가 포함되어야 합니다.

ⓘ 아키텍처가 크게 다르기 때문에 Gen6 플랫폼, 이전 하드웨어 세대 또는 PowerScale 노드 사이에는 노드가 호환되지 않습니다.

OneFS에는 SSD 용량이 다른 노드를 단일 노드 풀로 프로비저닝할 수 있는 SSD 호환성 옵션도 포함되어 있습니다.

SSD 호환성은 OneFS WebUI SmartPools 호환성 목록에 생성되고 명시되며 계층 및 노드 풀 목록에도 표시됩니다.

ⓘ 이 SSD 호환성을 생성할 때 OneFS는 병합할 두 풀에 동일한 수의 SSD, 계층, 요청된 보호 및 L3 캐시 설정이 있는지 자동으로 확인합니다. 이러한 설정이 다른 경우 OneFS WebUI에 이러한 설정의 통합 및 정렬을 묻는 메시지가 표시됩니다.

📖 자세한 내용은 SmartPools 백서에서 확인할 수 있습니다.

지원되는 프로토콜

적절한 자격 증명과 권한이 있는 클라이언트는 클러스터와의 통신을 위해 지원되는 다음 표준 방법 중 하나를 사용하여 데이터를 생성하고 수정하고 읽을 수 있습니다.

- NFS(Network File System)
- SMB/CIFS(Server Message Block/Common Internet File System)
- FTP(File Transfer Protocol)
- HTTP(Hypertext Transfer Protocol)
- HDFS(Hadoop Distributed File System)
- REST API(Representational State Transfer Application Programming Interface)
- S3(오브젝트 스토리지 API)

NFS 프로토콜의 경우 OneFS는 NFSv3, NFSv4와 OneFS 9.3의 NFSv4.1을 모두 지원합니다. 추가로 OneFS 9.2 이상에서는 NFSv3overRDMA를 지원합니다.

Microsoft Windows 측면에서, SMB 프로토콜 버전 3까지 지원됩니다. SMB3 언어의 일부로 OneFS는 다음과 같은 기능을 지원합니다.

- SMB3 다중 경로
- SMB3 무중단 가용성 및 Witness
- SMB3 암호화

SMB3 암호화 기능은 공유별로, 존(zone)별로 또는 클러스터 전체에 대해 구성할 수 있습니다. SMB3 암호화를 지원하는 운영 체제만 암호화된 공유를 사용할 수 있습니다. 이러한 운영 체제는 클러스터가 암호화되지 않은 연결을 허용하도록 구성된 경우 암호화되지 않는 공유도 사용할 수 있습니다. 다른 운영 체제는 클러스터가 암호화되지 않은 연결을 허용하도록 구성된 경우에만 암호화되지 않은 공유에 액세스할 수 있습니다.

클러스터에 있는 모든 데이터의 파일 시스템 루트는 /ifs(OneFS 파일 시스템)입니다. 이 경로는 SMB 프로토콜을 통해서는 'ifs' 공유(\\<cluster_name\ifs)로, NFS 프로토콜을 통해서는 '/ifs' 내보내기(<cluster_name>:/ifs)로 제공될 수 있습니다.

ⓘ 데이터가 모든 프로토콜 간에 공통적이므로 한 액세스 프로토콜을 사용하여 파일 콘텐츠를 변경하면 즉시 다른 액세스 프로토콜에서 표시할 수 있습니다.

OneFS는 프런트 엔드 이더넷 네트워크, SmartConnect 및 포괄적인 스토리지 프로토콜 및 관리 툴을 통해 IPv4 및 IPv6 환경 모두를 완벽하게 지원합니다.

또한 OneFS CloudPools는 다음과 같은 클라우드 공급업체의 스토리지 API를 지원하므로 다음과 같은 여러 스토리지 타겟에 파일을 스텁할 수 있습니다.

- Amazon Web Services S3
- Microsoft Azure
- Google Cloud Service
- Alibaba Cloud

- Dell EMC ECS
- OneFS RAN(네임스페이스에 대한 RESTful 액세스)

📖 자세한 내용은 CloudPools 관리 가이드**에서 확인할 수 있습니다.**

운영 환경에 영향을 미치지 않는 작업 - 프로토콜 지원

OneFS는 Linux 및 UNIX 클라이언트의 동적 NFSv3 및 NFSv4 페일오버와 페일백을 지원하고 Windows 클라이언트에 SMB3 무중단 가용성을 지원함으로써 데이터 가용성을 높입니다. 노드 장애가 발생하거나 사전 예방적 유지 보수 작업을 수행할 경우 실행 중인 읽기 및 쓰기 작업이 모두 클러스터의 다른 노드로 인계되어 사용자나 애플리케이션의 작업에 영향을 주지 않으면서 읽기/쓰기 작업이 완료됩니다.

페일오버하는 중에는 클러스터에서 정상 작동하는 모든 노드에 걸쳐 클라이언트를 균등하게 분산하여 성능에 미치는 영향을 최소화합니다. 장애 등의 이유로 노드의 작동이 중단되면 노드의 가상 IP 주소가 클러스터의 다른 노드로 원활하게 마이그레이션됩니다.

오프라인 상태의 노드가 다시 온라인 상태가 되면 SmartConnect가 전체 클러스터에서 NFS 및 SMB3 클라이언트를 자동으로 다시 로드 밸런싱하여 스토리지 및 성능 활용도를 극대화합니다. 정기적인 시스템 유지 보수와 소프트웨어 업데이트의 경우에 이 기능은 노드별 점진적 업그레이드를 지원하여 유지 보수 기간 동안 완벽한 가용성을 유지합니다.

파일 필터링

OneFS 파일 필터링은 NFS 및 SMB 클라이언트에서 내보내기, 공유 또는 액세스 존에 대한 쓰기를 허용하거나 허용하지 않도록 하는 데 사용할 수 있습니다. 이 기능을 사용하면 보안 문제, 생산성 저하, 처리량 문제, 또는 스토리지 환경 복잡화 등의 현상을 일으킬 수 있는 특정 형식의 파일 확장명이 차단되는 것을 방지할 수 있습니다. 구성은 명시적 파일 확장명을 차단하는 제외 목록 또는 특정 파일 형식의 쓰기를 명시적으로 허용하는 포함 목록을 사용해 구성할 수 있습니다.

데이터 중복 제거 - SmartDedupe

SmartDedupe 제품은 조직의 데이터를 보관하는 데 필요한 물리적 스토리지의 양을 줄여 클러스터의 스토리지 효율성을 극대화합니다. 온디스크 데이터에서 동일한 블록을 검색한 후 중복 블록을 제거하여 효율성을 향상시킵니다. 이 방법을 일반적으로 사후 처리 또는 비동기식 중복 제거라고 합니다.

중복 블록이 발견되면 SmartDedupe는 이러한 블록의 단일 복사본을 섀도우 저장소라고 하는 특수한 파일 세트로 이동시킵니다. 이 과정에서 중복 블록은 실제 파일에서 제거되고 섀도우 저장소에 대한 포인터로 교체됩니다.

사후 처리 중복 제거를 통해 신규 데이터가 먼저 스토리지 디바이스에 저장된 다음 후속 작업으로 공통성을 검색하는 데이터를 분석합니다. 이는 쓰기 경로에서 추가적인 계산이 필요하지 않기 때문에 초기 파일 쓰기 또는 수정 성능에는 영향을 미치지 않음을 나타냅니다.

SmartDedupe 아키텍처

OneFS SmartDedupe 아키텍처는 다음과 같은 다섯 가지 주요 모듈로 구성됩니다.

- 중복 제거 제어 경로
- 중복 제거 작업
- 중복 제거 엔진
- 섀도우 저장소
- 중복 제거 인프라스트럭처

SmartDedupe 제어 경로는 OneFS 웹 관리 인터페이스(WebUI), CLI(Command Line Interface) 및 RESTful 플랫폼 API로 구성되며 중복 제거 작업의 구성 관리, 예약 및 제어를 담당합니다. 작업 자체는 클러스터의 모든 노드에서 중복 제거의 조정을 관리하는 고도로 분산된 백그라운드 프로세스입니다. 작업 제어는 중복 제거 엔진과 함께 작동하여 파일 시스템 검색, 감지, 일치하는 데이터 블록의 공유를 담당합니다. 중복 제거 인프라스트럭처 계층은 공유 데이터 블록을 섀도우 저장소로 통합하는 커널 모듈로, 물리적 데이터 블록과 참조 또는 포인터를 모두 공유 블록에 보관하는 파일 시스템 컨테이너입니다. 이는 아래에 보다 자세히 설명됩니다.

그림 22: OneFS SmartDedupe 모듈형 아키텍처

📖 자세한 내용은 OneFS SmartDedupe 백서에서 확인할 수 있습니다.

섀도우 저장소

OneFS 섀도우 저장소는 데이터를 공유 가능한 방식으로 저장할 수 있는 파일 시스템 컨테이너입니다. 따라서 OneFS의 파일에는 섀도우 저장소의 공유 블록에 대한 물리적 데이터와 포인터 또는 참조가 모두 포함될 수 있습니다.

섀도우 저장소는 일반 파일과 유사하지만 일반적으로 일반 파일 inode와 관련된 모든 메타데이터를 포함하지는 않습니다. 특히 시간 기반 속성(생성 시간, 수정 시간 등)은 명시적으로 유지되지 않습니다. 각 섀도우 저장소에는 최대 256개의 블록이 포함될 수 있으며 각 블록은 32,000개의 파일에서 참조할 수 있습니다. 이 32K 참조 제한을 초과하면 새 섀도우 저장소가 생성됩니다. 또한 섀도우 저장소는 다른 섀도우 저장소를 참조하지 않습니다. 섀도우 저장소에는 하드 링크가 없으므로 섀도우 저장소의 스냅샷은 허용되지 않습니다.

ⓘ 섀도우 저장소는 중복 제거 외에 OneFS 파일 클론과 SFSE(Small File Storage Efficiency)에도 활용됩니다.

소규모 파일 스토리지 효율화

섀도우 저장소를 사용하는 또 다른 이유는 OneFS 소규모 파일 스토리지 효율화 때문입니다. 이 기능은 의료 서비스 PACS 워크플로와 같이 아카이브 데이터 세트를 구성하는 소규모 파일을 보관하는 데 필요한 물리적 스토리지의 양을 줄여 클러스터의 공간 활용도를 극대화합니다.

온디스크 데이터에서 전체 복제본 미러의 보호를 받는 소규모 파일을 검색한 후 섀도우 저장소에 패킹함으로써 효율성을 향상시킵니다. 이러한 섀도우 저장소는 미러링이 되는 대신 패리티로 보호되며 일반적으로 80% 이상의 스토리지 효율성을 얻을 수 있습니다.

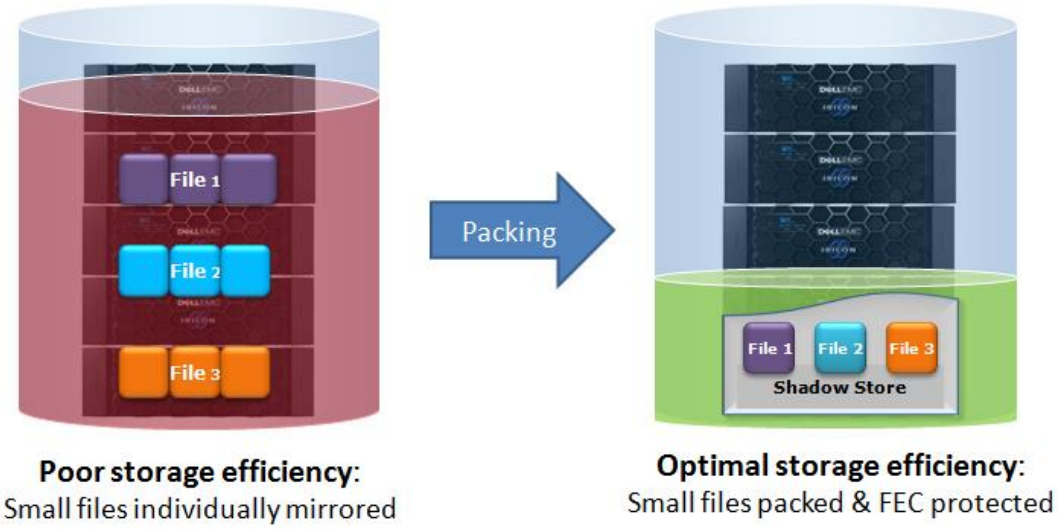

그림 23: 소규모 파일 컨테이너화

소규모 파일 스토리지 효율화는 스토리지 활용도 향상을 위해 약간의 읽기 레이턴시 성능 저하를 감수합니다. 아카이빙된 파일은 물론 쓰기 가능한 상태로 유지되지만, 섀도우 참조가 있는 컨테이너화된 파일을 삭제하거나 자르거나 덮어쓰는 경우, 참조되지 않은 블록이 섀도우 저장소에 남을 수 있습니다. 이러한 블록은 추후에 확보되므로 스토리지 효율성을 저하시키는 여지를 만들 수도 있습니다.

실제 효율성 손실 정도는 섀도우 저장소에서 사용하는 보호 수준 레이아웃에 따라 달라집니다. 컨테이너의 모든 블록에는 참조 파일이 최대 하나가 있고, 패킹된 크기(파일 크기)가 작기 때문에 보호 그룹 크기가 작을수록 컨테이너화된 파일이 더 취약해집니다.

덮어쓰기 및 삭제로 인한 파일 조각화를 줄이기 위해 조각 모음이 추가되었습니다. 이 섀도우 저장소 조각 모음은 ShadowStoreDelete 작업에 통합됩니다. 조각 모음 프로세스는 컨테이너화된 각 파일을 논리적 청크(각 32MB이하)로 나누고, 각 청크의 조각화를 진단하는 방식으로 작동합니다.

조각난 청크의 스토리지 효율성이 목표보다 낮으면 데이터를 다른 위치로 이동시켜 해당 청크를 처리합니다. 기본 목표 효율성은 섀도우 저장소에 사용되는 보호 수준에 따라 가능한 최대 스토리지 효율성의 90%입니다. 보호 그룹의 크기가 클수록 스토리지 효율성이 이 임계값 미만으로 떨어지기 전까지 더 높은 수준의 조각화를 허용할 수 있습니다.

인라인 데이터 감소

OneFS 인라인 데이터 감소는 F900, F810, F600, F200 올플래시 노드, H700/7000 및 H5600 하이브리드 섀시, A300/3000 아카이브 플랫폼에서 사용할 수 있습니다. OneFS 아키텍처는 다음과 같은 기본 구성 요소로 구성됩니다.

- 데이터 감소 플랫폼
- 압축 엔진 및 청크 맵
- 제로 블록 제거 단계
- 중복 제거 인메모리 인덱스 및 섀도우 저장소 인프라스트럭처
- 데이터 감소 알림 및 보고 프레임워크
- 데이터 감소 제어 경로

인라인 데이터 감소 쓰기 경로는 세 가지 주요 단계로 구성됩니다.

- 제로 블록 제거
- 인라인 중복 제거
- 인라인 압축

인라인 압축과 중복 제거 모두 클러스터에서 활성화되어 있는 경우 제로 블록 제거가 먼저 수행된 후 중복 제거가 수행되고, 그 다음에 압축이 수행됩니다. 이 순서로 각 단계가 수행됨에 따라 해당 후속 단계의 작업 범위가 줄어들 수 있습니다.

그림 24: 인라인 데이터 감소 워크플로

F810 에는 하드웨어 압축 오프로드 기능이 포함되어 있는데 각 F810 섀시의 각 노드에는 Mellanox Innova-2 Flex 어댑터가 있습니다. 즉, 압축 및 압축 해제는 최소한의 레이턴시로 Mellanox 어댑터에서 운영 환경에 영향을 미치지 않고 수행되므로 노드의 값비싼 CPU 및 메모리 리소스를 사용할 필요가 없습니다.

OneFS 하드웨어 압축 엔진은 zlib 을 사용하며 PowerScale F900, F810, F600, F200, H700/7000, H5600, A300/3000 노드를 위한 igzip 의 소프트웨어 구현입니다. 또한 소프트웨어 압축은 압축 하드웨어 장애가 발생한 경우와 혼합 클러스터인 경우, 하드웨어 압축 기능이 없는 비 F810 노드에 예비 수단(fallback)으로 사용되며 압축 하드웨어 장애 발생 시 예비 수단으로도 사용됩니다. OneFS 는 128KB 의 압축 청크 크기를 사용하며 각 청크는 16 개의 8KB 데이터 블록으로 구성됩니다. 또한 이 크기는 OneFS 가 데이터 보호 스트라이프 단위에 사용하는 크기와 같기 때문에 추가 청크 패킹의 오버헤드를 방지하여 단순성과 효율성을 제공합니다.

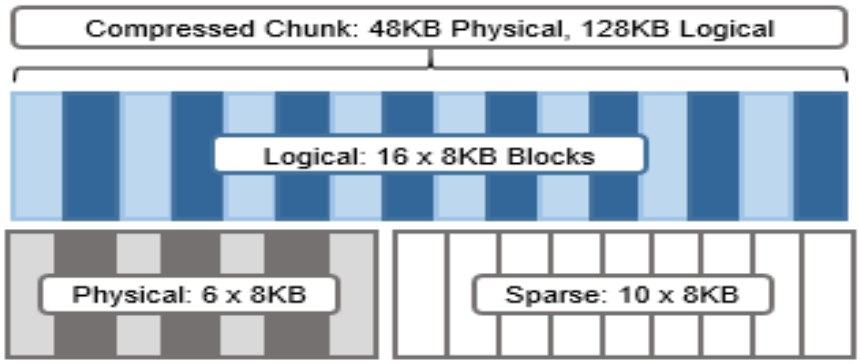

그림 25: 압축 청크 및 OneFS 의 운영 중단 없는 오버레이.

위의 다이어그램을 참고하십시오. 압축 후 이 청크는 크기가 16KB 에서 8KB 블록으로 줄어듭니다. 즉, 이 청크는 이제 물리적으로 크기가 48KB 입니다. OneFS 는 물리적 속성에 운영 중단 없는 논리적 오버레이를 제공합니다. 이 오버레이는 백업 데이터가 압축되었는지의 여부와 청크의 어느 블록이 물리적 또는 스파스(sparse)인지에 대해 설명하며, 따라서 파일 시스템 사용자는 압축의 영향을 받지 않습니다. 따라서 압축된 청크는 실제 물리적 크기와 관계없이 128KB 의 논리적 크기로 표시됩니다.

최소 8KB(한 블록) 이상이어야 압축되어 효율성을 확보할 수 있으며, 그렇지 않으면 해당 청크 또는 파일은 무시되어 압축되지 않은 원래 상태로 유지됩니다. 예를 들어 8KB(한 블록)가 절감되는 16KB 파일이 압축됩니다. 파일이 압축되면 FEC 로 보호됩니다.

압축 청크는 노드 풀을 교차하지 않습니다. 따라서 보호 수준을 변경하거나 복구된 쓰기를 수행하거나 보호 그룹 경계를 이동하기 위해 데이터를 압축 해제하거나 다시 압축할 필요가 없습니다.

동적 확장/필요 시 확장

성능 및 용량

성능을 높이거나 용량을 추가할 때 "스케일 업"이 필수인 기존 스토리지 시스템과 달리, OneFS 운영 체제가 실행되는 스토리지 시스템에서는 기존의 파일 시스템 또는 볼륨을 페타바이트급 용량으로 원활하게 늘리면서 "스케일 아웃"하는 동시에 그에 비례하게 성능도 향상시킬 수 있습니다.

클러스터에 용량을 추가하고 성능을 높이는 작업은 다른 스토리지 시스템보다 매우 쉽습니다. 스토리지 관리자가 랙에 다른 노드를 추가하고, 노드를 백엔드 네트워크에 연결하고, 클러스터에 추가 노드를 추가할 것을 지시하는 간단한 세 가지 단계만 수행하면 됩니다. 각 노드에는 CPU, 메모리, 캐시, 네트워크, NVRAM 및 입출력 제어 경로가 포함되어 있으므로 새로운 노드가 추가되면 그만큼 용량이 늘어나고 성능도 향상됩니다.

OneFS의 AutoBalance 기능은 백엔드 네트워크를 통해 정합성을 유지하면서 자동으로 데이터를 이동하므로 클러스터에 상주하는 기존의 데이터는 새로 추가된 스토리지 노드로 이동합니다. 이러한 자동 재조정 기능 덕분에 새로운 노드는 새 데이터로 인한 핫 스팟이 되지 않고 기존의 데이터는 더욱 강력해진 스토리지 시스템에서 더욱 효과적으로 운용됩니다. 또한 OneFS의 AutoBalance 기능은 최종 사용자에게 아무런 영향을 주지 않으며 고성능 워크로드에 대한 영향을 최소화하도록 조정이 가능합니다. OneFS에서 이 기능만을 사용하더라도 필요 시 클라이언트에 영향을 미치지 않으면서 TB에서 PB 단위로 즉각적으로 확장할 수 있으며, 이로 인한 관리 부담이나 스토리지 시스템 구성의 복잡성은 더 커지지 않습니다.

대규모 스토리지 시스템은 순차적, 동시적, 랜덤 방식 등 다양한 워크플로에 필요한 성능을 제공해야 합니다. 서로 다른 워크플로는 여러 애플리케이션 간이나 개별 애플리케이션 내에서도 존재할 수 있습니다. OneFS 운영 체제는 지능형 소프트웨어를 통해 이러한 요구를 동시에 모두 해결합니다. 보다 중요한 것은 OneFS에서의 처리량과 IOPS가 단일 시스템의 노드 수에 비례하여 확장된다는 점입니다. 또한 OneFS는 균등한 데이터 분산, 자동 재조정, 그리고 분산 프로세싱 등을 통해 시스템의 확장에 따라 추가되는 CPU, 네트워크 포트, 메모리 등을 활용할 수 있습니다.

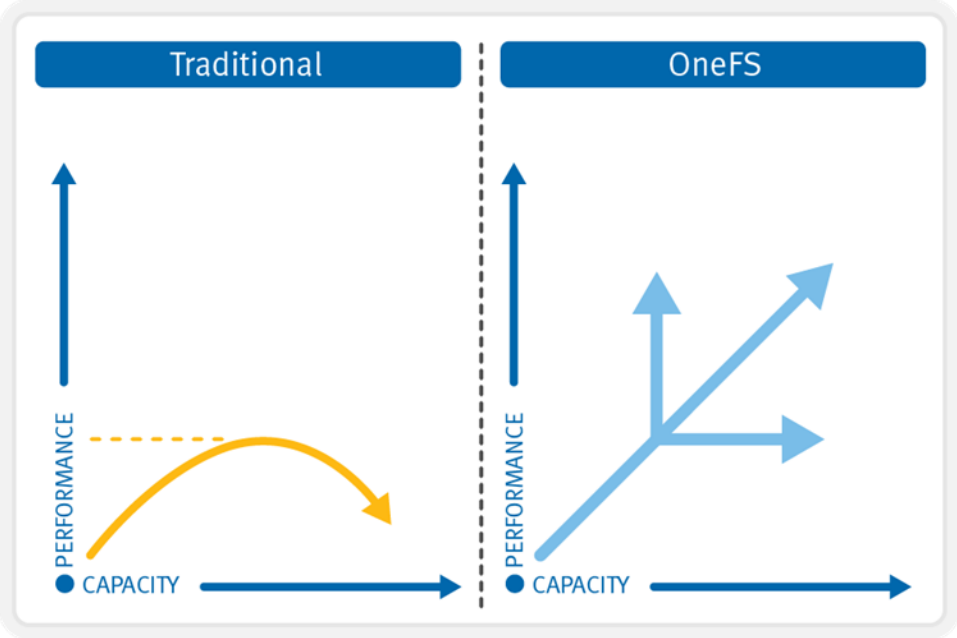

그림 26: OneFS 선형 확장성

Interfaces

관리자는 다음과 같은 여러 가지 인터페이스를 사용하여 해당 환경에서 스토리지 클러스터를 관리할 수 있습니다.

- "WebUI"(Web Administration User Interface)
- SSH 네트워크 액세스 또는 RS232 직렬 연결 기반 명령줄 인터페이스
- 간단한 추가/제거 기능을 지원하는 각 노드의 LCD 패널
- 클러스터 구성 및 관리를 프로그래밍 방식으로 제어하고 자동화하는 REST 기반 플랫폼 API

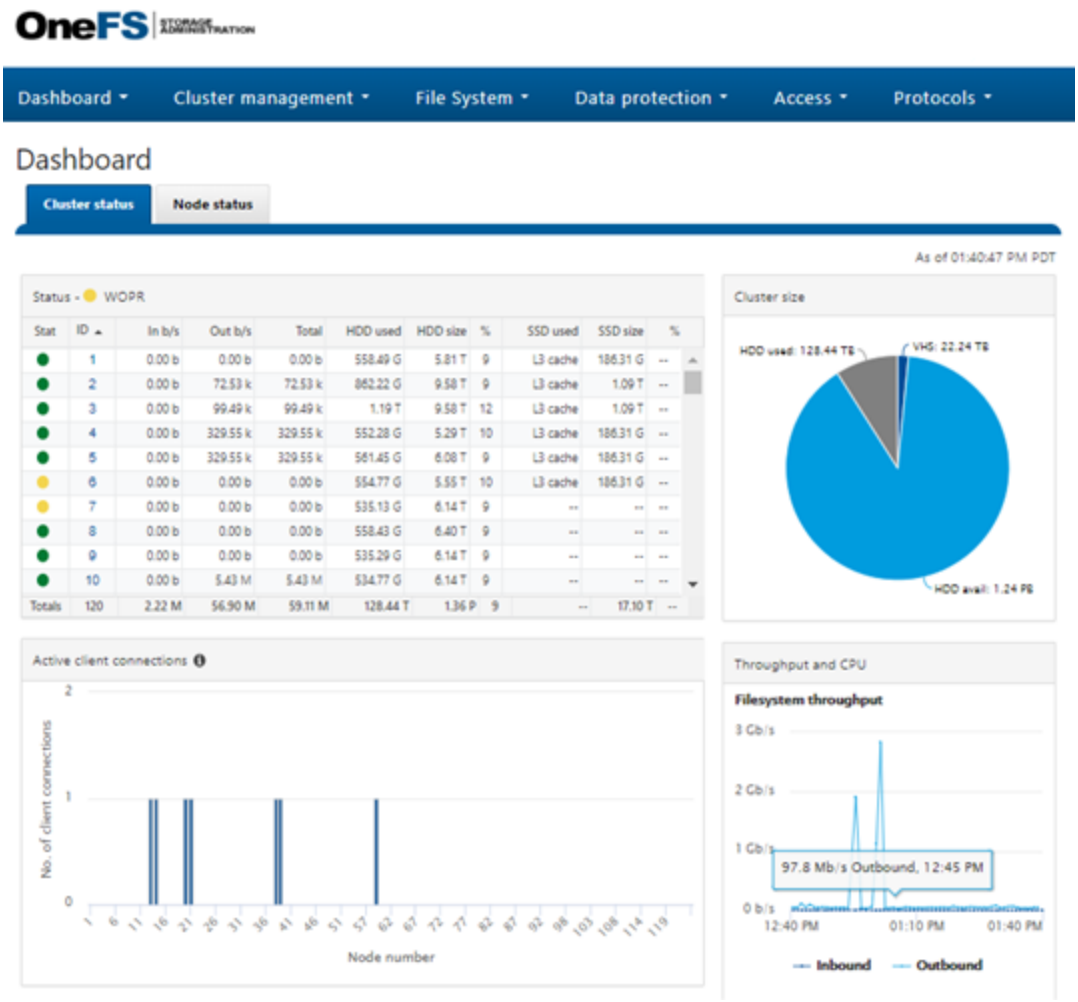

그림 27: OneFS 웹 사용자 인터페이스

📖 OneFS 명령 및 기능 구성에 대한 자세한 내용은 OneFS 관리 가이드에서 확인할 수 있습니다.

인증 및 액세스 제어

인증 서비스는 사용자의 자격 증명을 확인한 후 파일 액세스 및 수정을 허용하는 보안 계층을 제공합니다. OneFS는 다음과 같은 4가지 사용자 인증 방식을 지원합니다.

- AD(Active Directory)
- LDAP(Lightweight Directory Access Protocol)
- NIS(Network Information Service)
- 로컬 사용자 및 그룹

또한 인증 유형을 두 개 이상 사용할 수 있도록 지원합니다. 그러나 클러스터에서 여러 인증 방식을 사용하기 전에 인증 유형 간의 상호 작용에 대해 자세히 알아보는 것이 좋습니다. 여러 가지 인증 모드를 적절하게 구성하는 방법에 대한 정보는 제품 설명서를 참조하십시오.

Active Directory

Active Directory는 Microsoft의 LDAP 구현 방식으로, 네트워크 리소스에 대한 정보를 저장할 수 있는 디렉토리 서비스입니다. Active Directory는 다양한 기능을 제공하지만 도메인에 클러스터를 연결하는 주된 이유는 사용자 및 그룹 인증을 수행하기 위해서입니다.

웹 관리 인터페이스 또는 명령줄 인터페이스를 통해 클러스터의 Active Directory 설정을 구성하고 관리할 수 있지만 되도록이면 웹 관리 인터페이스를 사용하는 것이 좋습니다.

클러스터의 각 노드는 동일한 Active Directory 시스템 계정을 공유하므로 관리하기가 매우 용이합니다.

LDAP

LDAP(Lightweight Directory Access Protocol)는 서비스 및 리소스를 정의하고 쿼리하고 수정하는 데 사용되는 네트워킹 프로토콜입니다. LDAP의 주요 이점은 디렉토리 서비스의 개방적인 특성과 여러 플랫폼에서 LDAP를 사용할 수 있다는 점입니다. 클러스터 스토리지 시스템은 LDAP를 통해 사용자 및 그룹을 인증하여 클러스터에 대한 액세스를 허용할 수 있습니다.

NIS

Sun Microsystems가 설계한 NIS(Network Information Service)는 OneFS에서 클러스터에 액세스하는 사용자 및 그룹을 인증하는 데 사용할 수 있는 디렉토리 서비스 프로토콜입니다. YP(Yellow Pages)라고도 하는 NIS는 OneFS가 지원하지 않는 NIS+와는 다릅니다.

로컬 사용자

OneFS는 로컬 사용자 및 그룹 인증을 지원합니다. WebUI 인터페이스를 사용하여 클러스터에서 직접 로컬 사용자 및 그룹 계정을 생성할 수 있습니다. 로컬 인증은 Active Directory, LDAP 또는 NIS와 같은 디렉토리 서비스가 사용되지 않는 경우 또는 특정 사용자나 애플리케이션이 클러스터를 액세스해야 하는 경우에 유용할 수 있습니다.

액세스 존

액세스 존(Access Zone)은 클러스터 액세스를 논리적으로 분할하고 독립형 유닛에 리소스를 할당하여 공유 테넌트 또는 멀티 테넌트 환경을 구축하는 방법을 제공합니다. 이를 용이하게 하기 위해 액세스 존은 다음과 같은 세 가지 핵심 외부 액세스 구성 요소를 결합합니다.

- 클러스터 네트워크 구성
- 파일 프로토콜 액세스
- 인증

따라서, SmartConnect 존은 액세스 제어를 위해 SMB 공유 세트, NFS 내보내기, HDFS 랙, 존별 하나 이상의 인증 공급자와 연결됩니다. 이는 중앙 집중식으로 관리되는 단일 파일 시스템의 이점을 제공하며 이러한 파일 시스템은 여러 테넌트에 대해 프로비저닝 및 보호될 수 있습니다. 이 같은 기능은 중앙 IT 부서에서 여러 개별 사업부에 서비스를 제공하는 엔터프라이즈 환경에서 특히 유용합니다. 서버 통합 이니셔티브에서 신뢰할 수 없는 개별 Active Directory 포리스트에 가입된 여러 Windows 파일 서버를 병합할 때도 도움이 됩니다.

액세스 존을 사용하는 경우 기본 제공되는 System 액세스 존에는 지원되는 각 인증 공급자의 인스턴스, 모든 사용 가능한 SMB 공유 및 모든 사용 가능한 NFS 내보내기가 기본적으로 포함됩니다.

이러한 인증 공급자는 Microsoft Active Directory, LDAP, NIS 및 로컬 사용자 또는 그룹 데이터베이스 인스턴스를 여러 개 포함할 수 있습니다.

역할 기반 관리

역할 기반 관리는 "루트" 및 "관리자" 사용자의 권한을 보다 세분화된 권한으로 분할하고 특정 역할별로 이러한 권한을 할당할 수 있도록 해주는 클러스터 관리 RBAC(Role Based Access Control) 시스템입니다. 이 세분화된 역할을 권한이 없는 다른 사용자에게 부여할 수 있습니다. 예를 들어 데이터 센터 운영 직원에게 전체 클러스터에 대한 읽기 전용 권한을 부여하여 전체 모니터링 액세스는 허용하지만 구성은 변경하지 못하게 할 수 있습니다. OneFS는 감사, 시스템 및 보안 관리자를 비롯한 기본 제공 역할 컬렉션을 제공하며, 액세스 존별 또는 클러스터 전체에 사용자 정의 역할을 생성할 수 있습니다. 역할 기반 관리는 OneFS 명령줄 인터페이스, WebUI 및 플랫폼 API와 통합됩니다.

📖 멀티 프로토콜 환경에서의 ID 관리, 인증, 액세스 제어에 대한 자세한 내용은 OneFS 멀티 프로토콜 보안 가이드를 참조하십시오.

OneFS 감사

OneFS는 클러스터에서 시스템 구성, NFS, SMB, HDFS 프로토콜 활동을 감사하는 기능을 제공합니다. 이를 통해 조직은 적용되는 다양한 데이터 거버넌스 및 규제 준수 의무 요건을 충족할 수 있습니다.

모든 감사 데이터는 클러스터 파일 시스템에 저장되어 보호되며 감사 항목별로 구성됩니다. 여기에서 감사 데이터를 Dell EMC CEE(Common Event Enabler) 프레임워크를 통해 Varonis DatAdvantage 및 Symantec Data Insight와 같은 타사 애플리케이션으로 내보낼 수 있습니다. OneFS 프로토콜 감사를 액세스 존별로 활성화할 수 있으므로 클러스터 전체에서 세부적인 제어가 가능합니다.

클러스터가 병렬 로드 밸런스 구성을 사용하여 노드당 최대 5개의 CEE 서버에 걸쳐 감사 이벤트를 작성할 수 있습니다. 이를 통해 OneFS는 완벽한 엔터프라이즈급 감사 솔루션을 제공할 수 있습니다.

📖 자세한 내용은 OneFS 감사 백서에서 확인할 수 있습니다.

소프트웨어 업그레이드

최신 버전의 OneFS로 업그레이드하면 새 기능, 수정 사항 및 기존 기능을 모두 사용할 수 있습니다. 클러스터는 동시 업그레이드 또는 점진적 업그레이드의 두 가지 방법으로 업그레이드할 수 있습니다.

동시 업그레이드

동시 업그레이드는 클러스터의 모든 노드에 새 운영 체제를 동시에 설치하고 재부팅합니다. 동시 업그레이드에서는 업그레이드 프로세스 중에 노드를 다시 시작하는 동안 2분 미만의 일시적인 서비스 중단이 필요합니다.

점진적 업그레이드

점진적 업그레이드는 클러스터를 순차적으로 하나씩 업그레이드하고 재시작합니다. 점진적 업그레이드를 진행하는 동안 클러스터는 온라인 상태를 유지하고 서비스 중단 없이 계속 클라이언트에 데이터 서비스를 제공합니다. OneFS 8.0 이전에는 점진적 업그레이드를 OneFS 코드 버전 제품군 내에서만 수행할 수 있었고 OneFS 주 코드 버전 개정 간에는 지원되지 않았습니다. OneFS 8.0 이후부터 모든 새 릴리스는 이전 버전으로부터 점진적 업그레이드가 가능합니다.

운영 중단 없는 업그레이드

NDU(Non-Distruptive Upgrade)를 사용하면 클러스터 관리자가 스토리지 OS를 업그레이드할 수 있으며 최종 사용자는 오류나 중단 없이 데이터에 계속 액세스할 수 있습니다. 클러스터에서 운영 체제를 업데이트하는 것은 점진적 업그레이드이며 간단하게 할 수 있습니다. 이 프로세스 중에는 한 번에 하나의 노드가 새 코드로 업그레이드되고 해당 노드에 연결된 활성 NFS 및 SMB3 클라이언트가 클러스터의 다른 노드로 자동 마이그레이션됩니다. 부분 업그레이드도 허용되어 클러스터 노드의 하위 세트를 업그레이드할 수 있습니다. 노드의 하위 세트는 업그레이드하는 동안 증가될 수도 있습니다. 업그레이드를 일시 중지했다가 재개할 수 있으므로 고객이 여러 번의 소규모 유지 보수 기간을 거치며 업그레이드를 마칠 수 있습니다. 또한 OneFS 8.2.2 이상에서는 병렬 업그레이드를 도입하여 클러스터가 한 번에 전체 이웃 또는 장애 도메인을 업그레이드할 수 있으므로 대규모 클러스터 업그레이드 기간이 대폭 단축됩니다. OneFS 9.2 이상에서는 OS와 펌웨어 업그레이드를 결합하여 동시에 진행될 수 있도록 하여 업그레이드에 미치는 영향과 기간을 상당히 줄입니다. 9.2 이상에는 모든 SMB 클라이언트가 노드에서 연결을 끊을 때까지 노드를 재부팅하거나 프로토콜 서비스를 재시작할 수 없도록 하는 드레인 기반 업그레이드도 포함되어 있습니다.

롤백 지원

OneFS는 업그레이드 롤백을 지원하여 업그레이드가 커밋되지 않은 클러스터를 이전 버전의 OneFS로 복원하는 기능을 제공합니다.

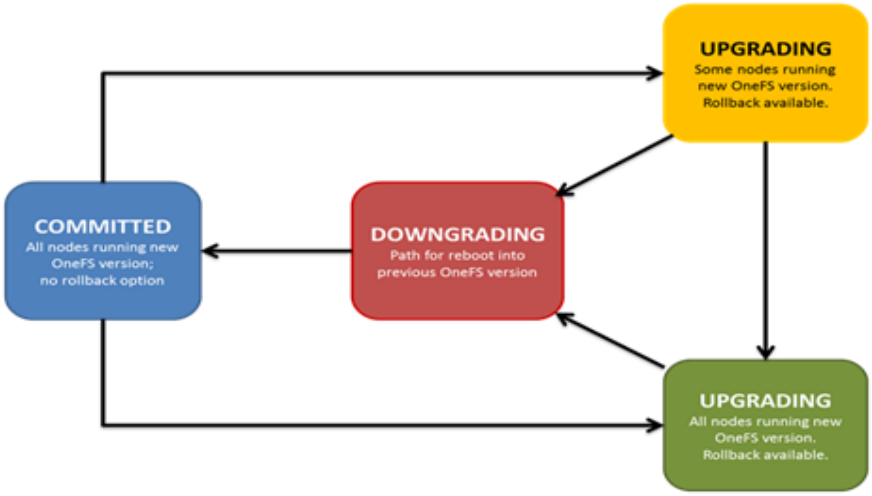

그림 28: OneFS 운영 중단 없는 업그레이드 상태

자동 펌웨어 업데이트

OneFS 기반 클러스터는 운영 중단 없는 펌웨어 업데이트 프로세스의 일부로 새 드라이브 및 교체 드라이브에 대한 자동 드라이브 펌웨어 업데이트를 지원합니다. 펌웨어 업데이트는 드라이브 지원 패키지를 통해 제공되며, 이 패키지는 클러스터 전체에서 기존 드라이브와 새 드라이브의 관리를 간소화 및 효율화합니다. 이렇게 하면 드라이브 펌웨어가 최신 상태로 유지되고 알려진 드라이브 문제로 인한 장애 발생 가능성을 줄일 수 있습니다. 따라서 자동 드라이브 펌웨어 업데이트는 OneFS의 고가용성 및 운영 환경에 영향을 미치지 않는 운영 전략의 중요한 구성 요소입니다. 드라이브 및 노드 펌웨어는 점진적 업그레이드 또는 전체 클러스터 재부팅을 통해 적용할 수 있습니다.

OneFS 8.2 이전에는 노드 펌웨어 업데이트를 한 번에 하나의 노드에 설치해야 했는데, 이는 대규모 클러스터에서 특히나 시간이 많이 소요되는 작업이었습니다. 노드 펌웨어 업데이트는 이제 동시에 업데이트할 노드 목록을 제공하여 클러스터 전체에서 구성될 수 있습니다. 업그레이드 지원 툴을 사용하여 동시에 업데이트할 수 있는 노드 중 원하는 조합 및 업데이트하면 안 되는 노드의 명시적 목록(예: 노드 쌍을 이루고 있는 노드)을 선택할 수 있습니다.

업그레이드 수행

업그레이드의 일환으로 OneFS는 설치 전 검증 확인을 자동으로 실행합니다. 이를 통해 기존 OneFS 설치 환경의 구성이 업그레이드할 OneFS 버전과 호환되는지 확인합니다. 지원되지 않는 구성이 발견되면 업그레이드가 중지되고 문제 해결 지침이 표시됩니다. 업그레이드를 시작하기 전에 사전 예방적으로 설치 전 업그레이드 확인을 실행하면 호환되지 않는 구성으로 인한 중단을 방지할 수 있습니다.

OneFS 데이터 보호 및 관리 소프트웨어

OneFS는 다양한 요구 사항을 해결할 수 있도록 포괄적인 데이터 보호 및 관리 소프트웨어 포트폴리오를 제공합니다.

소프트웨어 모듈	기능	설명
CloudIQ™	**클러스터 상태 모니터링**	지능적인 예측 분석을 실시하여 클러스터 상태를 사전에 모니터링합니다.
InsightIQ™	**성능 관리**	혁신적인 성능 모니터링 및 보고 툴을 사용하여 클러스터의 성능 극대화
DataIQ™	**데이터 분석 및 관리**	운영 환경 내부 또는 클라우드에서, 파일 및 오브젝트 스토리지 등 어디에 있든지 데이터를 몇 초 만에 찾아 액세스하고 관리할 수 있습니다. 단일 인터페이스를 통해 이기종 스토리지 시스템을 포괄적으로 볼 수 있으며, 사일로에 갇힌 데이터를 효과적으로 제거할 수 있습니다.
SmartPools™	**리소스 관리**	매우 효율적인 자동화된 계층형 스토리지 전략을 실행하여 스토리지 성능 및 비용을 최적화할 수 있습니다
SmartQuotas™	**데이터 관리**	스토리지를 관리하기 쉬운 클러스터, 디렉토리, 하위 디렉토리, 사용자 및 그룹 레벨의 세그먼트로 원활하게 분할하고 용량을 할당하는 할당량 지정 및 관리 기능 제공
SmartConnect™	**데이터 액세스**	클라이언트 접속 로드 밸런싱을 구현하고 스토리지 노드 전체에 걸쳐 클라이언트 접속의 동적 NFS 페일오버 및 페일백을 지원하여 클러스터 리소스 활용도를 최적화
SnapshotIQ™	**Data Protection**	성능 오버헤드를 거의 또는 전혀 일으키지 않으면서 거의 즉각적인 보안 스냅샷을 통해 데이터를 효율적이고 안정적으로 보호하며, 필요에 따라 스냅샷 백업을 즉시 복구하여 중요 데이터를 빠르게 복구 OneFS 쓰기 가능 스냅샷을 사용하여 읽기 전용 스냅샷의 공간 및 시간 효율적이고 수정 가능한 복사본을 생성합니다.
SyncIQ™	**데이터 복제**	신뢰할 수 있는 재해 복구 기능을 위해 대규모 미션 크리티컬 데이터 세트를 여러 사이트의 여러 공유 스토리지 시스템에 비동기식으로 복제 및 배포하고, 한 번에 간편하게 페일오버/페일백을 수행함으로써 미션 크리티컬 데이터의 가용성 극대화
SmartLock™	**데이터 보존**	소프트웨어 기반의 WORM(Write Once Read Many) 방식을 이용해 중요 데이터가 실수로, 성급한 판단으로 또는 악의적으로 변경/삭제되는 것을 방지하고 SEC 17a-4 규정을 포함한 모든 엄격한 기업 내부 및 정부 규정 준수 요건을 충족
SmartDedupe™	**데이터 중복 제거**	클러스터에서 동일한 블록을 검색한 다음 중복 항목을 제거하여 필요한 물리적 스토리지 양을 절감함으로써 스토리지 효율성 극대화
CloudPools™	**클라우드 계층화**	CloudPools를 사용하면 클러스터 내 어떤 데이터를 클라우드 저장소에 아카이빙해야 하는지 정의할 수 있습니다. 클라우드 공급업체에는 Microsoft Azure, Google Cloud, Amazon S3, Dell EMC ECS, Virtustream 및 기본 OneFS가 포함됩니다.

표 3: Dell EMC Power Scale 데이터 서비스 포트폴리오

자세한 내용은 제품 설명서를 참조하십시오.

결론

OneFS 운영 체제를 기반으로 하는 Dell EMC 스케일 아웃 NAS 솔루션을 사용하면 조직은 단일 관리 지점을 통해 단일 파일 시스템, 단일 볼륨 내 TB에서 PB 단위로 확장할 수 있습니다. 또한 OneFS 운영 체제를 사용하면 높은 성능과 대규모의 처리량을 실현하면서 관리 복잡성은 최소화할 수 있습니다.

차세대 데이터 센터는 지속적인 확장성을 충족하도록 구축되어야 합니다. 이러한 데이터 센터는 자동화의 이점을 활용하고, 하드웨어의 상용화를 이용하며, 네트워크 Fabric을 완벽하게 이용하고, 끊임없이 변화하는 요구 사항을 해결할 수 있는 유연성을 갖추고 있어야 합니다.

OneFS는 이러한 당면 과제를 해결하도록 설계된 차세대 파일 시스템입니다. OneFS의 차별화된 특징은 다음과 같습니다.

- 완벽하게 분산된 단일 파일 시스템
- 완전하게 균형이 잡힌 고성능 클러스터
- 클러스터 안에 있는 모든 노드에서 파일 스트라이핑
- 자동화된 소프트웨어로 복잡성 제거
- 동적 컨텐츠 밸런싱
- 유연한 데이터 보호 기능
- 고가용성
- 웹 기반 및 명령줄 관리

OneFS는 대규모 홈 디렉토리, 파일 공유, 아카이브, 가상화, 비즈니스 분석 등 엔터프라이즈 Data Lake 환경의 파일 기반 및 비정형 "빅데이터" 애플리케이션에 이상적입니다. 또한 에너지 탐사, 금융 서비스, 인터넷 및 호스팅 서비스, 비즈니스 인텔리전스, 엔지니어링, 제조, 미디어 및 엔터테인먼트, 생물정보학, 과학 연구 등 다양한 데이터 집약적 고성능 컴퓨팅 환경에도 적합합니다.

다음 단계로 진행

Dell EMC 영업 담당자 또는 공인 리셀러에 연락하여 PowerScale NAS 스토리지 솔루션으로 얻을 수 있는 이점을 자세히 알아보십시오.

기능을 비교하고 자세한 정보를 확인하려면 Dell EMC PowerScale 방문을 참조하십시오.

Dell EMC PowerScale 솔루션에 대한 자세한 정보

Dell EMC 전문가에게 문의하기

추가 리소스 보기

대화에 참여: #DellEMCStorage