

최근 파일 시스템 메타데이터 성능 최적화 관련 연구 동향

안태환, 손용석

중앙대학교

atahwan10@cau.ac.kr, sysganda@cau.ac.kr

A Study on the File System Metadata Performance Optimization

Taehwan Ahn, Yongseok Son

Chung-Ang University

요약

본 논문은 최근 파일 시스템 메타데이터 성능 최적화 관련 연구를 파악한다. 파일 시스템 메타데이터 성능은 파일 시스템 전체 성능에 상당한 영향을 미치며 다수의 파일 관리가 요구되는 서버 및 스토리지 환경에서 중요한 성능 지표이다. 이에 따라 최근 다양한 연구들이 메타데이터 성능 향상을 위한 기법들을 제시하고 있다. 예를들어, IndexFS는 여러 가지 기술을 결합하여 기존 솔루션보다 뛰어난 성능을 발휘하는 메타데이터 처리량을 보여준다. Soft updates는 writeback cache block에 대한 세분화된 업데이트의 저비용 시퀀싱을 위해 구현되었다. Versioning FileSystem은 저널 기반 메타데이터를 사용하여 inode 및 간접 블록과 관련된 공간의 줄인다. Multiversion b-tree는 timestamp를 사용하여 디렉터리 관련 공간 요구사항을 줄인다. 분산 파일시스템에서 메타데이터의 load를 분산시키고, 시스템 규모 증가에 따른 메타데이터 서비스의 신뢰성을 향상시킨다. 이와 같이 본 논문에서는 해당 연구들을 바탕으로 최근 메타데이터 구조 및 메타데이터 성능 최적화 관련 연구동향을 살핀다.

1. 최근 파일 시스템 메타데이터 성능 최적화 연구 동향

본 논문에서는 최근 파일 시스템 메타데이터 성능 최적화 관련 연구를 분석하고, 어떠한 방식으로 성능을 향상시키는지 살펴본다.

1-1. IndexFS

IndexFS [1]는 메타데이터 및 작은 파일에서 확장 가능한 고성능 작업을 위해 PVFS, Lustre, HDFS와 같은 기존 파일 시스템에 지원을 추가하는 미들웨어 설계이다. 디렉터리별로 네임스페이스를 점진적으로 분할하는 테이블 기반 아키텍처를 사용하여 작은 디렉터리에 대한 서버 및 디스크 로컬리티를 보존한다. 최적화된 로그 구조 레이아웃은 메타데이터 및 작은 파일을 효율적으로 저장하는데 사용된다. 또한 N-N 체크포인팅과 같은 집중적인 워크로드를 생성하기 위한 벌크 네임스페이스 삽입과 핫스팟 완화를 위한 상태 비저장 일관 메타데이터 캐싱의 두가지 클라이언트 기반 스톱 프리 캐싱 기술을 제안하였다. 이러한 기술을 결합하여 IndexFS [1]는 PVFS, Lustre 및 HDFS와 같은 기존 솔루션보다 50%에서 2배 더 뛰어난 성능을 발휘하는 아웃코어 메타데이터 처리량을 보여준다.

1-2. Soft updates

Soft updates [2]는 writeback cache block에 대한 세분화된 업데이트의 저비용 시퀀싱을 위해 구현된 기술이다. 이 기술을 사용하여 메타데이터 업데이트 종속성을 추적하고 적용하면 File System은 거의 모든 파일 작업에 대해 writeback을 안전하게 사용할 수 있다. Soft updates [2]를 사용하는 디스크 기반 파일 시스템이 대부분의 디스크 기반 파일 시스템보다 더 강력한 무결성과 보안 보장을 제공하면서 메모리 기반 파일 시스템 성능을 달성한 다는 것을 보여준다. 메타데이터 업데이트(예:파일 생성 및 삭제)를 자주 수행하는 workload의 경우 기존 동기식 쓰기 접근 방식과 비교할 때 성능이 2배 이상, 최대 20배까지 향상된다. 또한 Soft updates

[2]는 충돌 복구 지원을 옵션 및 백그라운드 역할로 강등하여서 파일 시스템 가용성을 향상시켜 파일 시스템 복구 시간을 1초 미만으로 단축할 수 있다.

1-3. Versioning FileSystem

Versioning FileSystem [3]은 이전 버전의 수정된 파일을 유지하므로 사용자의 실수나 시스템 손상으로부터 복구하는 것이 가능하다. 공간 효율적인 메타데이터인 저널 기반 메타데이터는 각 메타데이터 버전을 단일 저널 항목으로 인코딩한다. CVFS는 inode 및 간접 블록에 이 구조를 사용하여 관련 공간 요구 사항을 80%까지 줄인다. Multiversion b-trees는 timestamp를 사용하여 각 항목의 키를 확장하고 현재 및 과거 항목을 단일 트리에 유지한다. CVFS는 디렉터리에 이 구조를 사용하여 관련 공간 요구 사항을 99% 줄인다.

1-4. 분산 메타데이터 파일시스템의 파일 생성 전략

많은 client를 수용하려면 메타데이터 load를 분산시키는 것이 중요하다. 메타데이터 성능 확장은 원시 I/O 성능 확장보다 더 복잡하며, 분산 메타데이터를 사용하면 복잡성이 더욱 증가합니다. PVFS [4] 분산 메타데이터 파일 시스템을 testbed로 사용하여 생성 작업의 복잡성을 줄이고, 성능을 높일 수 있는 설계가 제시되었다. 해당 설계는 시스템이 확장됨에 따라 거의 일정한 작업 대기시간을 제공하고 경합이 심한 상황에서도 저하되지 않으며 메타데이터 서버의 수가 증가함에 따라 처리량을 선형적으로 증가시킨다.

1-5. 대규모 분산 파일시스템을 위한 신뢰성 높은 메타데이터 서비스

시스템 규모 증가에 따른 하드웨어 장애 및 소프트웨어 문제 증가로 인해 메타데이터 서비스 신뢰성은 파일 및 디렉터리 작업에 직접적인 영향을

미치기 때문에 중요한 문제가 된다. 기존 메타데이터 관리 메커니즘은 어느 정도 내결함성 기능을 제공할 수 있지만 충분하지 않다. 제안된 신뢰할 수 있는 이 메타데이터 서비스는 내결함성을 위한 새로운 활성-대기 아키텍처를 채택하고 전체적인 접근 방식을 사용하여 파일 시스템 가용성을 향상시킨다. 새로운 공유 스토리지 풀(SSP)은 활성 서버와 대기 서버 간의 투명한 메타데이터 동기화 및 복제를 위해 설계되었다. SSP를 기반으로 장애 발생 시 메타데이터 서비스 복구를 수행하기 위해 MAMS(Multiple Actives Multiple Standby) [5]라는 새로운 정책이 제시된다. 메타데이터 서비스의 연속성을 유지하기 위해 새로운 전역 상태 복구 전략과 스마트 클라이언트 내결함성 메커니즘이 달성되었다. 실험 결과에 따르면 성능에 미치는 영향은 미미하면서 다양한 오류 시나리오에서 빠른 장애 조치를 통해 파일 시스템 안정성을 크게 향상시킬 수 있음이 확인되었다. Hadoop Avatar, Hadoop HA 및 Boom-FS 파일 시스템의 일반적인 안정성 설계와 비교하여 안정성이 뛰어난 메타데이터 서비스를 사용한 평균 복구 시간(MTTR)은 각각 80.23%, 65.46%, 28.13% 감소했다.

2. 결론

본 논문에서는 최근 파일 시스템 메타데이터 성능 최적화를 위한 다양한 연구들을 분석하여 각각의 기술이 어떤식으로 성능을 향상시키고 있는지 살펴보았다. IndexFS는 메타데이터 및 작은 파일에 대한 확장 가능한 고성능 작업을 위해 기존 파일 시스템에 여러 가지 지원을 추가하는 방식으로 성능을 향상시켰다. Soft updates는 writeback cache block에 대한 세분화된 업데이트 저비용 시퀀싱을 위해 구현된 기술로 메타데이터 업데이트를 자주 수행하는 workload 환경에서 성능을 최대 20배까지 향상시켰다. Versioning FileSystem은 inode 및 간접블록에 저널 기반 메타데이터 구조를 사용하여 관련 공간 요구사항을 80%까지 줄였다. Multiversion b-tree는 timestamp를 사용하여 디렉터리 관련 공간 요구사항을 줄였다. 메타데이터의 load를 분산시키기 위해 PVFS 분산 메타데이터 파일 시스템을 testbed로 사용하여 생성 작업의 복잡성을 줄이고, 성능을 높였다. 시스템 규모 증가에 따른 하드웨어 장애 및 소프트웨어 문제 증가를 해결하기 위해 새로운 전역 상태 복구 전략과 스마트 클라이언트 내결함성 메커니즘이 달성하였다.

이러한 다양한 접근법들은 각각의 환경과 목적에 맞추어 설계되었으며, 파일시스템의 메타데이터 성능을 크게 향상시키는데 기여하였다. 앞으로의 연구에서는 이러한 기술들을 종합적으로 분석하고, 각 기술의 장점을 결합하여 보다 효율적인 메타데이터 최적화 방법을 모색하는 것이 필요하다.

ACKNOWLEDGMENT

이 성과는 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2022R1A4A5034130)
(교신저자: 손용석).

참 고 문 헌

[1] K. Ren, Q. Zheng, S. Patil and G. Gibson, "IndexFS: Scaling File System Metadata Performance with Stateless Caching and Bulk Insertion," SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis,

New Orleans, LA, USA, 2014, pp. 237-248, doi: 10.1109/SC.2014.25.
 [2] Ganger, Gregory R., et al. "Soft updates: a solution to the metadata update problem in file systems." *ACM Transactions on Computer Systems (TOCS)* 18.2 (2000): 127-153.
 [3] Soules, Craig AN, and Garth R. Goodson. "Metadata efficiency in versioning file systems." 2nd USENIX Conference on File and Storage Technologies (FAST 03). 2003.
 [4] Soules, Craig AN, and Garth R. Goodson. "Metadata efficiency in versioning file systems." 2nd USENIX Conference on File and Storage Technologies (FAST 03). 2003.
 [5] Devulapalli, Ananth, and Pete Wyckoff. "File creation strategies in a distributed metadata file system." 2007 IEEE International Parallel and Distributed Processing Symposium. IEEE, 2007.