

# WORM(Write Once Read Many) 지원 네트워크 대용량 저장 장치를 위한 파일 시스템

\*김문경, 김선태, 노재춘, 박성순

세종대학교 컴퓨터공학과

e-mail : kmk1030@naver.com, kimst4444@gmail.com, jano@sejong.ac.kr,

sspark@anyang.ac.kr

## A network-attached WORM(Write Once Read Many) storage system

\*MoonKyung Kim, SunTae Kim, Jaechun No, Sungsun Park  
School of Computer Science, Sejong University

### Abstract

This paper presents a network-attached WORM(Write Once Read Many) storage system whose purpose is to protect sensitive data that should not be accessed by accidental/intentional intrusion. In our storage system, the sensitive data would be marked and be stored into a specific WORM volume. Furthermore, with the associated WORM mark, the data whose WORM life cycle has been expired would automatically be moved to the general-purpose disk space, without interfering any other tasks, to save the WORM volume space. Our WORM storage system was integrated to the NAS product.

### I. 서론

본 연구는 변경되어서는 안 되는 중요한 데이터와 비즈니스 기록들을 안전하게 보존할 수 있도록 하는 WORM 기반의 네트워크 스토리지 시스템을 제안한다.

네트워크 스토리지 시스템 상에서는 고의적 또는 실수로 데이터 손실을 가져올 수 있다. 이러한 경우를 대비하여 중요한 데이터들에 대한 신뢰성, 무결성, 가용

성에 대해 검증이 이루어져야 한다.

WORM이란 한번 데이터를 쓰면 이후엔 읽기만 가능하도록 지원하는 개념이며, CD-R과 같이 CD에 한 번 기록된 데이터는 수정 및 삭제가 불가능하고 읽기만 가능하도록 하는 것과 동일하다.

WORM 파일 시스템의 한 종류인 Cached WORM 파일 시스템[1]은 Magnetic Disk Cache를 통해서 WORM 기능을 지원한다. WORM 기능을 지원하는 마그네틱 디스크를 사용하여 구현되었고, 보호대상 데이터에 대한 복사가 수행되기 때문에 중복 저장으로 인한 디스크 공간 낭비가 발생할 수 있다.

본 연구에서는 특정 데이터를 WORM 볼륨 내에 저장하면 유틸리티를 이용해 그 디렉토리 혹은 데이터에 WORM 마크를 찍어 해당 데이터를 영구히 수정할 수 없도록 한다. WORM 마크가 찍힌 후에는 다른 데이터를 추가하는 것은 가능하나 해당 데이터는 절대로 변경될 수 없도록 한다. 또한 WORM 데이터의 관리를 위해 데이터베이스에 WORM 마크 내용을 기록하며, 데이터를 네트워크 스토리지 상에 저장할 수 있도록 함으로써 대용량 WORM 데이터의 활용이 가능하도록 한다.

### II. 본론

#### 2.1 WORM 파일 시스템

본 연구에서 개발된 WORM 파일 시스템은 XFS 파일

시스템[2] 기반으로 구현되었다. 따라서 기존 XFS 파일 시스템과 WORM 기능을 추가한 XFS 파일 시스템 모두 사용이 가능하다. WORM 기능을 구현하기 위해 XFS 아이노드 구조에 WORM 기간 정보 등을 포함한 WORM 마크 구조를 추가하여 설계하였다. WORM 마크에 포함된 기간 정보를 비교하여 일반 데이터와 강력한 보호가 필요한 WORM 데이터를 구별할 수 있으며, 일단 WORM 기능을 부여 받게 되면 해당 데이터에 대한 수정 또는 삭제가 불가능 하도록 구현되었다. 또한 WORM 마크에 포함된 기간 정보를 비교함으로써 보호 기간 (WORM life cycle)이 종료된 데이터를 분별할 수 있으므로 별도의 WORM 해제 기능이 불필요하다.

2.2 WORM 파일 시스템의 동작

WORM 데이터의 경우 변경, 삭제가 불가능해야 하므로 일반 데이터에서 WORM 데이터로 변환되는 과정은 [그림 1]과 같이 개발된 Commit 명령을 통해 이루어진다. Commit 명령은 해당 데이터의 위치 정보와 시작 날짜, 종료 날짜, 사후처리를 위한 옵션 정보와 함께 수행되며, 관련 정보는 WORM 데이터들의 관리를 위해 SQLite[3] 데이터베이스에 저장된다. 해당 데이터가 가지는 시간 정보와 현재 시간을 비교하여 일반 데이터일 경우 수정 및 삭제가 가능하게 하고, WORM 데이터인 경우 수정 및 삭제가 불가능 하다.

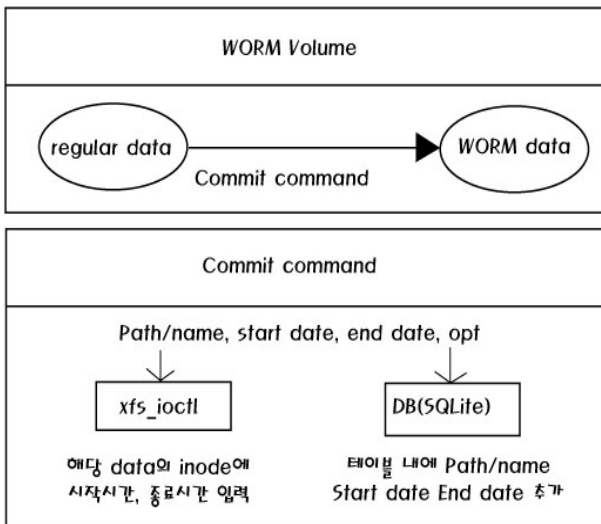


그림 1. 일반 데이터에서 WORM 데이터로의 변환 과정

2.3 WORM 파일 시스템의 성능 평가

성능 테스트는 XFS 파일 시스템과 WORM 파일 시스템을 비교하였다. 성능 테스트에 사용된 틀은 파일

시스템 벤치마크 틀인 Bonnie++[4]이다. CPU는 AMD 64 processor 3000+, RAM은 1GB, OS는 Debian, Kernel은 2.6.20, xfsprogs[5]는 2.8.16을 이용하여 측정 하였다.

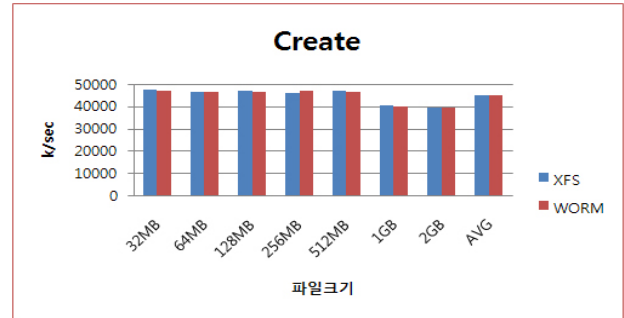


그림 2. XFS와 WORM의 Create 비교

Bonnie++를 이용한 XFS와 WORM의 비교한 결과를 [그림 2]에 표시하였다. 비교를 위해 32MB부터 2GB까지의 크기를 갖는 파일들을 생성하였다. [그림 2]는 기존 XFS 파일 시스템과 WORM 파일 시스템의 성능이 크게 차이가 크게 나지 않음을 보여준다.

III. 결론

본 연구의 성과물인 WORM 파일 시스템은 기존 XFS의 특징과 성능을 그대로 가져갈 수 있기 때문에 XFS 파일 시스템이 갖는 저널링, 신속한 트랜잭션, 높은 확장성, 뛰어난 처리량 등을 그대로 보유할 수 있다는 장점이 있다. 또한 특정 기간 동안 문서의 삭제 및 수정이 불가능 할 수 있도록 지원함으로써 강력한 보호가 필요한 기록물들의 관리에 용이하다. 더욱이 대용량 저장 장치에서 구현되었기 때문에 어디서든 접근 및 검색이 가능하다.

참고문헌

[1] Sean Quinlan, A Cached WORM File System, AT&T Bell Laboratories, Murray Hill, New Jersey 07974, U.S.A  
 [2] Adam Sweency, Scalability in the XFS File System, Silicon Graphics, January, 1996  
 [3] <http://www.sqlite.org/>  
 [4] <http://sourceforge.net/projects/bonnie/>  
 [5] <http://oss.sgi.com/projects/xfs/>