

# 클라우드 스토리지 시스템을 위한 중복제거 기술 연구

김태훈, 김지홍, 엄영익  
성균관대학교 정보통신대학  
e-mail: arton1224@naver.com, {jjilong, yieom}@ece.skku.ac.kr

## A Study on De-Duplication for Cloud Storage Systems

Taehun Kim, Jee Hong Kim and Young Ik Eom  
Sungkyunkwan University

### 요 약

클라우드 스토리지에 저장되는 데이터가 급증하면서 데이터 중복으로 인한 스토리지의 저장 비용이 증가하고 있다. 이러한 데이터 중복 문제를 해결하기 위해서 기존의 스토리지 환경에서는 다양한 데이터 중복제거 기술이 제안되었다. 이에 본 논문에서는 클라우드 스토리지를 위한 데이터 중복제거 기술의 분석을 통하여 대규모 클라우드 스토리지에서의 데이터 중복 제거율을 높이기 위해 데이터 유형의 특성을 고려한 스토리지 중복제거 기술의 필요성을 제시한다.

### 1. 서론

최근 클라우드 스토리지 환경에서 데이터 사용량이 급증함에 따라 스토리지의 저장비용이 빠르게 증가하고 있다. 이에 대용량의 클라우드 데이터를 제공하는 아마존의 S3, OpenStack의 Swift, Eucalyptus 등에서는 스토리지 관리 비용 증가 및 이에 따른 문제점이 발생하고 있다. 클라우드 스토리지에서는 늘어나는 사용자에게 따른 데이터 사용량의 급증과 더불어 데이터의 무결성을 보장하기 위한 대량의 데이터의 복제본 관리 비용 또한 급증하고 있다. 이러한 급증하는 데이터를 관리하기 위한 데이터 중복 저장은 스토리지의 안정성을 보장하는 반면 스토리지의 확장성 및 관리 비용을 증가시키는 원인이 된다[1].

이와 같은 데이터의 중복 문제를 개선하기 위해서 기존의 LiveDFS, ZFS, Btrfs에서는 블록 중복 제거기술을 제안하였으며 SDFS에서는 파일 중복 제거기술 제안하여 스토리지 가용 공간을 확보하고자 하였다[3][6][8][9]. 각각의 중복제거 기술은 블록과 파일단위의 중복제거 기술이다.

이에 본 논문에서는 클라우드 스토리지를 위한 데이터 중복 제거기술의 특징을 분석하여 향후 대규모 클라우드 스토리지에서의 데이터 중복 제거율을 높이기 위한 데이터 유형 분석을 반영한 스토리지 중복제거 기술의 필요성을 제시한다.

본 논문의 2장에서는 데이터 중복제거에 관한 배경지식을 기술하고, 3장에서는 관련 연구로 기존의 데이터 중복제거 기술들에 대해 분석한다. 4장에서는 각각의 데이터 중복제거 기술들을 평가하고, 마지막으로 결론 및 향후 연구 방향을 제시한다.

### 2. 배경지식

데이터 중복제거 기술은 중복되는 데이터를 제거하고 고유한 내용만을 저장함으로써 데이터의 저장 비용을 절감한다. 데이터 중복제거 기술의 시스템 분류는 (그림1)과 같이 크게 소스 기반과 타겟 기반으로 나뉘며, 다양한 중복제거 기술이 연구 되고 있다[2]. 클라우드 스토리지에서 데이터 중복제거 기술을 적용하면 40~90% 사이의 저장 공간을 절약할 수 있다[3][6]. 데이터의 중복제거 절차는 파일 분리, 인덱싱, 어드레싱, 데이터저장 등의 단계로 구성된다.

파일 분리 단계에서는 중복제거 할 데이터를 나누고 데이터 간에 중복되는 부분을 검사할 수 있도록 준비한다.



(그림 1) 중복 제거 시스템의 분류

† 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0025971)

그다음 인덱싱과 어드레싱 단계에서는 해시 알고리즘을

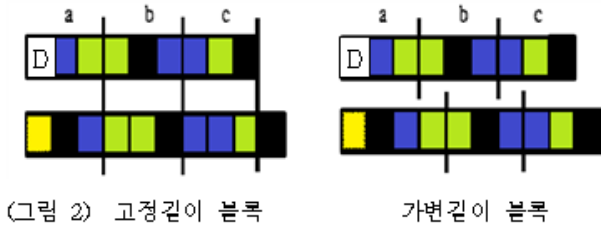
통하여 각 데이터를 식별할 수 있는 고유한 식별 번호를 부여한다. 마지막으로 데이터저장 단계에서는 중복되지 않은 고유한 데이터를 스토리지에 저장한다.

**2.1 파일 중복제거**

파일 중복제거는 파일 단위로 중복제거를 한다. 파일 중복제거의 경우 같은 파일의 데이터가 한 비트만 달라도 서로 다른 해시 값이 생성되며, 그렇기 때문에 중복제거율이 블록, 바이트 등의 중복 제거율보다 낮다[4].

**2.2. 블록 중복제거**

블록 중복제거는 파일을 블록단위로 분할하여 중복 제거하며, 고유한 블록만을 스토리지에 저장한다. 또한 중복 블록은 해시 알고리즘을 통해 판별한다. 블록 중복제거는 길이에 따라서 (그림2)와 같이 고정길이 블록과 가변길이 블록으로 구분된다.



(그림 2) 고정길이 블록

가변길이 블록

(그림2)는 a 블록에 데이터(D)가 추가될 때 고정, 가변길이 블록의 차이점을 나타낸다. 고정길이 블록에 데이터(D)가 추가되면 각 블록의 길이가 변경되지 않는다. 그러나 가변 길이 블록은 추가블록의 길이가 변경되며, 그 외 블록들의 길이변경은 없다[5].

**2.3 바이트 중복제거**

바이트 중복제거는 파일을 바이트 단위로 분할하여 중복제거를 수행한다. 중복제거에서 파일 중복이나, 블록 중복보다 중복제거가 높으며, 각각의 바이트에 대해 해시 알고리즘을 계산하여 핑거 프린트를 많이 만들기 때문에 오버헤드가 많다는 단점을 가지고 있다[4].

**2.4 사후 처리(Post-Processing) 중복제거**

사후 처리 중복제거는 데이터를 수신하고 즉시 중복제거 하지 않고, 데이터를 임시 디스크에 저장해 두었다가 추후시점에 데이터 중복제거를 진행하는 방식이다. 단 중복제거가 진행되기 전까지 데이터를 저장할 충분한 공간이 필요하며, 중복데이터를 효율적으로 관리하지 못하면 가용공간이 부족하게 된다. 그러나 데이터를 수신하는 시점에 별도의 오버헤드가 존재하지 않아 빠른 송수신이 가능하다는 장점이 있다[4].

**2.5 즉시(In-line) 중복제거**

즉시 중복제거는 타켓 스토리지에 수신된 데이터를 저장하기 전에 데이터의 중복을 제거한다. 별도의 임시 저장

공간이 필요하지는 않지만, 데이터의 송수신 시 오버헤드가 존재한다[4].

**3. 관련 연구**

**3.1 기존 파일 시스템**

전통적인 파일 시스템은 사용자가 데이터를 요청할 때, 원본 데이터를 복사해서 동일한 사본의 데이터를 생성하였다. 그 결과 상당량의 데이터가 중복되었다.

**3.2 LiveDFS**

LiveDFS는 사용자가 데이터를 요청할 때 새로운 블록을 생성 하지 않고 기존의 블록을 참조 한다. LiveDFS는 Ext3을 기반으로 즉시 중복제거를 하는 기술이다. 또한, 40%가량의 저장공간을 절감 할 수 있다.

LiveDFS의 주요 특징으로 Spatial locality, Prefetching metadata, Journaling, Kernel-space design등이 있다. Spatial locality은 메타 데이터를 관리하는 메타 레퍼런스를 동일한 그룹블록에 저장한다. Prefetching metadata는 페이지 캐시안의 동일한 블록그룹에 대하여 블록들을 검색할 때, 메타 데이터와 데이터 블록 등의 디스크 탐색 시간이 감소한다. Journaling은 데이터의 이동을 추적하고, 데이터가 손상되었을 때 복구한다. 또한, 디스크에 데이터를 쓸 때, 데이터를 한 번에 기록함으로써 디스크의 탐색 시간과 쓰기 수행시간이 향상된다. Kernel-space design는 Ext3FS를 참조하여 만들어졌으며, 리눅스의 파일 시스템을 따른다.

LiveDFS의 경우 리눅스 커널 소스코드의 변경 없이 커널의 모듈로 넣을 수 있으며, 블록의 공유를 허용한다. LiveDFS의 주요 구성 요소로는 핑거 프린트와 레퍼런스 카운트 등이 있다. 핑거 프린트는 MD-5, SHA-1등의 해시알고리즘으로 각 블록에 대한 연산을 통해 고유한 블록으로 디스크에 저장된다. 레퍼런스 카운트는 각 블록에 대한 참조 횟수이며, 공유한 블록들에 대해 갱신이 일어날 때 COW(Copy-on-Write)가 발생하며, 레퍼런스 카운트가 감소한다[6].

**3.3 OpenDedup SDFS**

SDFS는 클라우드 스토리지에서 파일을 저장할 때 하나의 파일을 여러 개의 청크로 나눈 후에, 청크를 하나의 오브젝트로 처리해서 저장한다. 이렇게 청크로 파일을 나누면 동시에 여러 개의 청크를 전송할 수 있으며, 접근 속도가 향상된다. 그러나 오브젝트를 청크로 나누는 계산을 하는 과정에 CPU를 많이 소모하는 단점을 가지고 있다.

SDFS는 즉시, 예약 중복제거를 지원하며 약 80~90% 스토리지를 절약할 수 있는 파일 중복제거 기술이다. SDFS의 경우Volume, File-system Service, DSE(Deduplication Storage Engine), Data chunks로 구성된다. 또한, SDFS는 FUSE(FileSystem in Userspace)를 사용하여 커널 소스의 수정없이 추가가 가능하다. SDFS는 마스터와 슬레이브의

구조를 통해, 데이터가 손상되었을 때 신속히 복구할 수 있다. 기본적으로 SDFS는 파일 레벨 중복제거이지만, 아마존 S3 오브젝트 스토리지에는 블록형태로 저장한다[3].

### 3.4 OpenSolaris ZFS

ZFS는 여러 저장 공간을 논리적으로 공유해서 하나의 저장 공간으로 만든다. ZFS는 COW와 슈퍼 블록을 사용하며, 블록 중복제거를 하는 즉시 제거기반의 기술이다. 그리고 SDFS와 같이 FUSE를 사용하여 커널의 수정 없이 시스템을 구축한다. ZFS는 파일의 메타 데이터에 대한 중복제거는 하지 않는다. SHA-256해시 알고리즘을 사용하여 블록을 암호화하면, 256bit의 체크섬을 사용하여 고유한 블록으로 매핑 시킨다. ZFS는 메모리안의 같은 페이지들에 대해 공유한다. 오라클의 데이터파일은 “logbias”의 속성을 설정하여 동일한 데이터를 디스크 상에 두 번 쓰는 것을 방지한다[8].

### 3.5 Btrfs (B-tree file system)

Btrfs는 즉시 중복제거 기반이다. Ext3와 Btrfs간에 상호간 변환이 가능하다는 특징을 가지며 Zlib를 통한 자체 압축을 지원한다. 또한, Btrfs는 ZFS가 기반이 되어 비슷한 특징들을 가지고 있다. 해시 알고리즘은 TEA, CRC-32C을 사용한다. 또한, 오브젝트 레벨로 Raid0, 1, 10을 지원한다.

## 4. 중복제거 기술 분석

본 4장에서는 기존의 중복제거 기술을 분석하고 비교를 통해 데이터 중복제거의 필요성을 기술한다. 디스크에서 메모리로 데이터를 읽어 올 때, 이미 메모리에 로드된 데이터가 있는 경우 중복되는 부분을 제외하고 데이터를 읽는다. 그러므로 중복이 많이 된 파일일수록 중복제거를 통해 읽기성능이 향상된다[7].

중복제거 기술을 적용시키면 전체적인 파일 크기의 감소와 버퍼 캐시 적중률 향상된다. 또한, 파일 읽기 및 쓰기의 속도가 증대된다. 즉, 같은 크기의 버퍼 캐시를 사용하게 되지만 중복제거로 파일의 크기가 작아지므로 상대적으로 적중률이 향상된다. 이는 중복제거율이 높을수록 더욱 효과가 높다.

<표1>은 중복제거 기술 간의 특징을 비교하였다. SDFS의 경우 LiveDFS에 비해 중복제거 속도가 빠르고, 오버헤드가 적지만 10%가량 중복 개선율이 낮다. LiveDFS의 경우 SDFS에 비해 중복제거 단위가 낮으므로 평균 프린트의 크기가 더 크다. LiveDFS는 FUSE를 사용하지 않고 리눅스 파일 시스템에 추가 가능하지만, LiveDFS를 제외한 다른 중복제거 기술들은 FUSE를 사용하여 시스템에 추가한다. LiveDFS를 제외한 다른 중복제거 기술은 FUSE를 사용하여 문맥 교환 비용이 발생할 수 있다. 각 중복제거 시스템에서 메모리 사용 면에서는 LiveDFS가 메타 레퍼런스를 사용해 메타 데이터 일부만

을 올려서 사용하기 때문에 효율적이다.

<표1> 중복제거 기술 비교분석

	LiveDFS	SDFS	ZFS	btrfs
제거 기술	blok dedup	file dedup	blok dedup	blok dedup
제거 방식	in-line	in-line Post-process	in-line	in-line
블록 종류	fixed size	x	variable size	variable size
지원 os	Linux	windows, Linux	Linux, Mac OS,etc,	Linux, Mac OS,etc,
클라우드 스토리지	OpenStack “Swift”	아마존 “S3”	RAID-Z	RAID-0,1, 10
메타데이터 중복제거	o	o	x	x
메타데이터 메모리로드	partially load	entire load	entire load	entire load
해시 함수	MD-5, SHA-1	Tiger	SHA-256	CRC-32, TEA
FUSE사용	x	o	o	o
Cow사용	o	x	o	o

## 5. 결론

본 논문에서는 데이터 중복제거 기술들에 대한 분석과 비교를 통해 클라우드 스토리지에서 적합한 중복제거 기술이 어떤 것인지 고찰하였다. 클라우드 스토리지에는 멀티미디어, VMimage, 문서 등의 다양한 데이터들이 저장된다. 하지만 기존의 중복제거 기술은 스토리지 환경에서의 다양한 데이터의 유형이 고려되지 않았다. 이를 고려하여 향후에는 클라우드 스토리지에 저장되는 데이터의 유형을 고려한 중복제거 방안에 대해 연구할 계획이다.

### 참고문헌

- [1] Openstack Swift, <http://www.openstack.org>
- [2] Meister D., and Brinkmann A., “Multi-level comparison of data deduplication a backup scenario,” in Proceedings of 2<sup>nd</sup> The Israeli Experimental Systems Conference(SYSTOR’09), 2009.
- [3] OpenDedup SDFS, <http://www.opendedup.org>
- [4] Qinlu H., Zhanhuai L., and Xiao Z., “Data Deduplication Techniques,” International Conference on Future Information Technology, 2010.
- [5] ExaGrid Systems, [http://www.exagrid.com/ExaGrid\\_Systems\\_8\\_Considerations\\_for\\_Evaluating\\_Disk-Based\\_Backup\\_Solutions.pdf](http://www.exagrid.com/ExaGrid_Systems_8_Considerations_for_Evaluating_Disk-Based_Backup_Solutions.pdf).
- [6] Chun-Ho N., Mincao M., Tsz-Yeung W., and Patrick P., “Live Deduplication Storage of Virtual Machine Images in an Open-Source Cloud,” Proceedings of ACM/IFIP /USENIX 12<sup>th</sup> International Middleware Conference, 2011.
- [7] 이철민, 민영혜, 김재훈, 김영규, “선택적 중복 제거(De-Duplication)기법,” 한국정보과학회, 2011.
- [8] ZFS, [https://blogs.oracle.com/bonwick/entry/zfs\\_dedup](https://blogs.oracle.com/bonwick/entry/zfs_dedup)
- [9] Btrfs, <http://en.wikipedia.org/wiki/Btrfs>