

# 데이터 중복 제거 기반의 디스크 아카이브 시스템 설계 및 구현

강성운\*, 정호민, 고영웅\*, 이정근\*

\*한림대학교 컴퓨터공학과

e-mail:{upersbird, chorogyi, yuko, Jeonggun.lee}@hallym.ac.kr

## Design and Implementation of Disk Archive System Exploiting De-duplication Scheme

Sung Woon Kang\*, Ho Min Jung\*, Young Woong Ko\*, Jeong Gun Lee\*

\*Dept. of Computer Engineering, Hallym University

### 요 약

기존의 TAR와 같은 아카이브 포맷은 파일의 중복을 제거하는 기능이 포함되지 않아 리눅스 배포 미러와 같이 버전단위로 저장되는 시스템에서 디스크 공간의 낭비가 발생했다. 본 연구에서는 중복 제거 기능이 포함된 아카이브 포맷인 DTAR와 이를 지원하는 DTM 유틸리티를 제안하였다. 주요 아이디어는 DTAR 헤더에 SHA1 해시를 삽입하고 SHA1 해시를 노드로 하는 R-B Tree를 생성하여 중복을 검색 및 제거하는 것이다. 실험 결과 DTAR가 tar.gz보다 최대 31% 공간을 절약하고, 수행 시간도 줄어드는 것을 확인하여 효율적임을 보였다.

### 1. 서론

컴퓨터를 이용한 문서 작업이 증가하고, 인터넷을 이용하여 데이터를 교환하는 것이 일반화되면서 저장 장치의 용량이 급격하게 증가하고 있다. 따라서 중복 데이터에 대한 처리가 주요한 관심이 되고 있으며 이로 인해 백업 시스템, FTP 미러, 가상화(Virtualization) 시스템 등에서 중복 제거 기술이 적용되고 있는 추세이다. 그러나 이러한 중복 제거 기술들은 특별한 하드웨어와 그에 따른 전용 어플리케이션이 필요하므로 설치비용이 높아지는 단점을 가진다.

자료보관에 쓰이는 유틸리티로 TAR(Tape ARchive) 프로그램이 범용적으로 사용되고 있다. TAR 파일은 1970년대 자기테이프에 백업과 검색 용도를 위해 만들어졌고 현재는 여러 파일을 묶어 전송하는 역할로 많이 쓰이고 있다. TAR 프로그램은 증분 백업을 하기 위해 스냅샷이라는 별도의 파일을 만들어야 하고 풀 백업일 경우 같은 디렉토리를 여러 번 저장할 경우 중복 데이터가 생긴다[1].

본 논문에서는 이러한 TAR 파일의 단점을 극복하기 위해 중복제거 기능을 추가하였다. 주요 아이디어는 TAR 헤더 블록에 파일의 해시 필드를 추가하여 해시 값을 통해 파일을 비교 할 수 있으므로 TAR 파일 내에 또는 바깥에 똑같은 파일이 있는 것을 찾아내어 제거하고 링크를 걸어 중복 데이터를 제거하는 것이다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 TAR

구조에 대해 살펴보고, 3장에서는 DTAR 파일 포맷을 기술하고, 4장에서는 DTAR 포맷의 중복을 제거하는 DTM 시스템에 대한 설계와 구현에 대해 설명한다. 5장에서는 성능평가에 대해 기술하고 6장에서는 결론 및 향후 연구 방향을 제시한다.

### 2. TAR 구조

TAR는 테이프 아카이브를 위해 고안된 파일 형식과 이런 형식의 파일을 다루는데 사용되는 프로그램을 의미한다. 초기에는 테이프 백업 목적으로, 순차적 입출력 장치에 직접 쓰도록 개발되었으나, 현재는, 배포 또는 아카이브 용도로 많은 파일들을 디렉토리 구조, 파일 속성들을 보존하면서 하나의 큰 파일로 묶는 데 주로 쓰인다.

TAR파일은 하나의 파일 또는 그이상의 파일들을 순차적으로 이을 수 있다. 각각의 파일들은 512 바이트 블록의 헤더를 가지고 있으며 파일 내용들은 512 바이트 블록에 담아 저장하고 나머지 512 바이트 블록에는 제로 바이트(zero byte)를 채워 넣어서 파일들을 구분한다. 또한 TAR 파일의 끝에는 두 개의 제로 바이트로 채워진 512 바이트 블록을 채워 넣는다. 기본적으로 512 바이트 블록이지만 테이프 속성에 따라서 블록 크기의 값을 커질 수 있다.

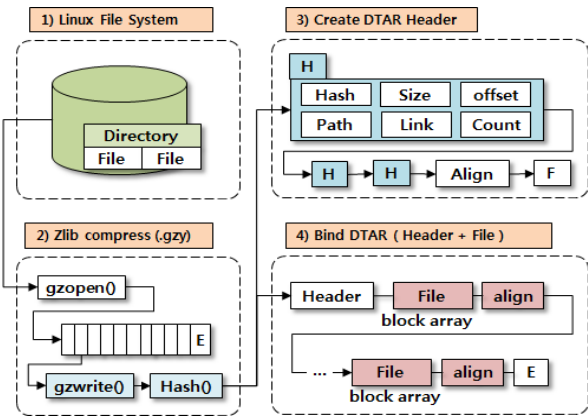
TAR 헤더 블록은 파일의 메타데이터를 포함하고 있으며 ASCII 코드로 인코딩 되어 있다. 파일 포맷은 UNIX 파일 스타일인 파일 이름, 파일 모드, 소유자 아이디, 그룹 아이디, 파일 크기, 수정 날짜, 헤더의 체크섬, 파일 타입, 링크된 파일 이름으로 구성되어 있다.

본 연구는 교육과학기술부와 한국연구재단의 지역혁신인력양성사업과 기초연구사업(No.2010-0016143)의 지원을 받은 결과물임을 밝힙니다.

TAR 외의 아카이브 포맷으로 CD-ROM이나 DVD-ROM에 사용되는 ISO-9660 이미지, 본 셸에서 사용되는 Shell archive, RPM 파일에서 쓰이는 cpio, staic 라이브러리를 만들 때 쓰이는 Unix Archiver 등이 있다[2][3][4][5][6].

### 3. DTAR 구조

DTAR(De-duplication Tape ARchive)는 TAR 포맷과는 달리 헤더부분과 블록부분으로 이루어져 있다. 헤더는 파일들의 정보들을 담고 있으며 TAR 파일 헤더와 같이 파일의 크기, 오프셋(offset), 경로와 같은 속성 값을 담고 있으며 추가적으로 파일의 SHA1 해시 값을 가지고 있어 파일의 중복 여부를 검사할 수 있다. DTAR의 블록 부분은 파일들의 내용을 순차적으로 저장하고 있으며 블록 단위로 정렬하기 위해 파일 끝 부분을 제로 바이트로 채웠다.



(그림 1) DTAR의 생성 과정

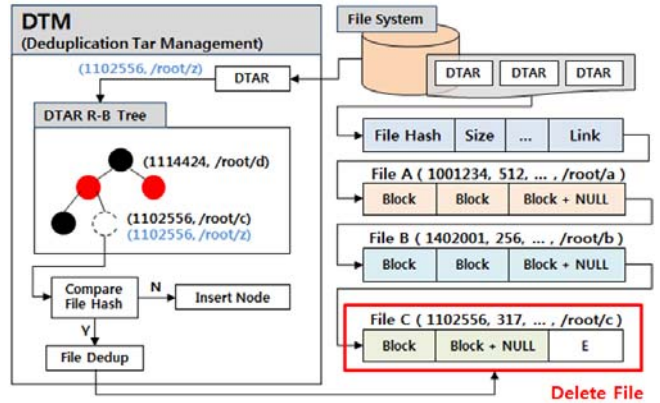
그림 1에서 DTAR의 생성 과정을 보이고 있으며 생성과정은 다음과 같다. 1)선택된 디렉토리 내부의 디렉토리와 파일들을 순차적으로 오픈하여 디렉토리와 파일 목록을 만들고 헤더부분과 블록부분의 크기를 계산하여 미리 공간을 확보한다. 2)파일 목록에 따라 파일 내용을 압축한다. 파일 압축은 zlib API를 사용하며 압축 방법은 *gzopen* 함수를 사용하여 파일을 열고 *read* 함수를 사용해 파일의 내용을 버퍼에 넣고 *gzwrite* 함수로 버퍼내용을 압축하여 DTAR에 쓴다. 또한 동시에 파일의 SHA1 해시를 계산한다. 3) 압축과 해시 계산이 끝나면 헤더부분에 SHA1 해시와 나머지 파일 정보들을 채워 넣는다. 공간을 확보할 때 헤더부분을 미리 제로 바이트로 초기화 했으므로 시스템 블록 크기에 정렬되어 있다. 4) 헤더부분과 압축된 파일 내용이 저장되어 있는 블록부분을 합치며 압축된 파일 부분을 블록 크기에 맞게 조정한다.

### 4. DTM 시스템 설계 및 구현

DTM(De-duplication Tar Management)은 DTAR로 구성된 파일들을 관리해주는 툴이다. DTM의 주요 기능은 DTAR의 중복 제거와 DTAR 헤더의 정보 수정을 통한 파일 시스템 관리이다. 생성된 DTAR는 DTM의 명령에

따라 특정 디스크 공간에 저장한다.

DTM의 파일 중복 제거, 백업 및 복원 등의 기능들은 사용자의 요청에 의해 이루어지거나 일정한 시간을 두고 자동으로 처리 될 수 있다. 이러한 방식은 DB 접근이나 테이블 유지를 위한 메모리 점유를 필요로 하지 않기 때문에 입출력 성능 향상에 영향을 주는 요소가 된다.



(그림 2) DTM의 중복 제거 과정

그림 2는 DTM의 중복 제거 과정을 보여주고 있다. 파일 중복 제거는 DTM에서 지정한 디렉토리의 DTAR 파일의 헤더를 읽어 구성된 파일들의 SHA1 해시 비교를 위해 R-B Tree를 작성하게 된다. R-B Tree에서 중복된 해시가 아니라면 R-B Tree에 해당 해시를 삽입하고, 중복된 해시라면 해당 블록을 삭제하고 헤더에는 중복된 파일 있는 DTAR에 심플릭 링크를 걸어둔다. R-B Tree는 트리의 높이가  $2\lg(n+1)$  보다는 무조건 작고 AVL 트리보다 insert, delete에 대한 비용이 적은 특징을 가진다. 파일에 대한 비교, 삽입이 빈번한 DTM에서 R-B Tree는 효과적인 알고리즘이다. R-B Tree의 노드의 키 값은 SHA1 파일 해시가 되며 노드에 포함된 정보로는 DTAR 링크를 가지고 있다.

### 5. 성능 평가

실험에 사용한 컴퓨터는 CPU 클럭 3.0 GHz, 512RAM의 하드웨어 스펙을 가지며 운영체제로는 커널 버전 2.6.18의 Fedora Core 9이다.

<표 1> 원본 데이터 정보

	Set 1	Set 2	Set 3
용량(GB)	0.5	1.0	1.6
파일 수	47,991	96,880	147,101

실험 데이터로 사용한 디렉토리는 미리 사이트인 kernel.org에 배포되어 있는 커널을 버전 단위로 9개를 다운로드 하였다. 각 커널 디렉토리의 용량은 170~180MB이며 파일 수는 15,000~17,000개 정도가 포함되어 있다.

Set 1, Set 2, Set 3는 여러 파일이 존재하는 9개의 디렉

토리를 3개, 6개, 9개씩 사용하는 데이터 수의 그룹을 말한다. 각 실험 세트에 존재하는 kernel 파일들은 버전이 다르게 배포될 때마다 수정되는 부분이 많지 않았기 때문에 많은 중복이 발생할 것이다.

단위(sec)

<표 3> DTAR, tar.gz 압축 시간

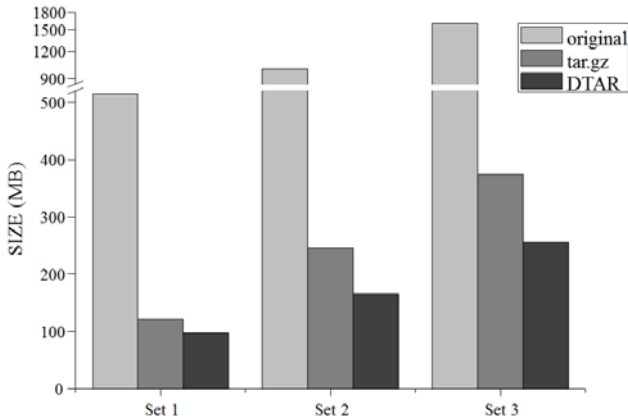
	Set 1			Set 2			Set 3		
	user	sys	total	user	sys	total	user	sys	total
tar.gz	74.2	10.8	85.0	148.6	21.6	170.2	224.9	34.2	259.1
DTAR	52.1	29.3	81.4	107.9	56.6	164.5	164.5	89.2	253.7

5.1. DTAR, tar.gz 압축의 공간 비교 실험

단위(MB)

<표 2> DTAR, tar.gz의 압축 용량

	Set 1	Set 2	Set 3
tar.gz	121.1	246.0	373.8
DTAR	96.7	165.5	256.0

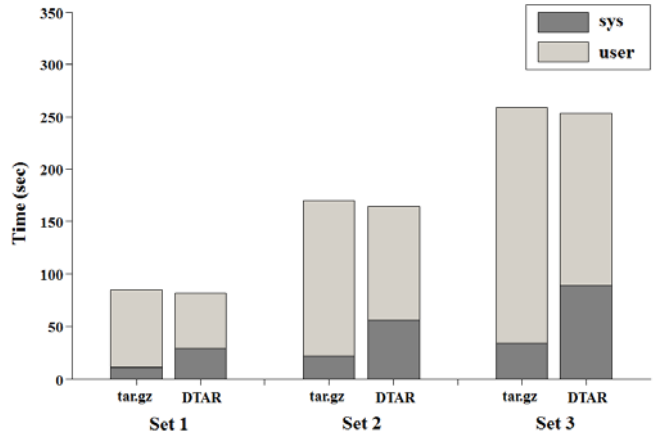


(그림 3) DTAR, tar.gz의 압축 용량 비교

그림 3은 DTAR, tar.gz으로 각각 압축하여 파일 크기를 비교한 그래프이다. 각 'Set'에서 나타내는 것은 원본 데이터 크기, tar.gz로 압축한 크기, DTAR로 중복제거 압축한 크기를 나타낸다. 'Set 1'을 DTAR와 tar.gz으로 각각 압축하였을 때, 원본 데이터 크기에서 DTAR는 18.7%, tar.gz은 23.5%의 압축률을 보였다. 'Set 2, 3'은 DTAR로 압축하였을 때, 16.5%, 16.0%의 압축률을 보였고 tar.gz으로 압축하였을 때, 24.6%, 23.3%의 압축률을 보였다. 실험을 압축 용량으로 살펴보면 DTAR는 tar.gz보다 각각 24.4MB, 80.5MB, 117.8MB의 공간 절약을 보였다. 실험을 통해 원본 데이터의 수가 많아질수록 DTAR가 tar.gz보다 공간이 절약되는 것을 보였다. 이러한 효과는 현재 실험에 사용한 커널 미러사이트와 같은 잦은 수정과 백업이 요구되는 시스템에서 큰 효과가 있을 것으로 보였다.

5.1. DTAR, tar.gz 압축의 시간 비교 실험

표 3은 DTAR와 tar.gz의 실험 수행 시간을 측정한 표이다. 나타난 시간의 정보는 user와 sys로 나뉜다. user는 사용자 모드에서 소비한 CPU 소비 시간을 말하며 sys는 실제 커널에서 할애한 시간을 말한다. 그림 4는 DTAR와 tar.gz의 수행 시간을 그래프로 보였다. 실험 결과로 볼 때 DTAR의 수행은 tar.gz와 비슷한 시간대에서 종료되지만 DTAR가 tar.gz보다 조금 더 빠르게 동작함을 확인하였다.



(그림 4) DTAR, tar.gz 압축 시간 비교

6. 결론 및 향후연구

본 논문에서는 현재 널리 사용되고 있는 TAR 유틸리티에 중복 제거 기능을 추가하여 DTAR 포맷과 DTM을 설계 및 구현을 기술하였다. 주요 아이디어는 DTAR 헤더에 SHA1 해시를 삽입하여 파일의 유일성을 추가하고 DTM에서 SHA1 해시를 노드로 하는 R-B Tree를 생성하여 중복을 검색 및 제거한다.

실험 결과 본 논문에서 제안한 DTAR가 tar.gz보다 최대 31% 공간을 절약하고, 수행 시간 측면에서도 성능 향상이 있다. 즉, DTAR가 효율적인 기법임을 보였다. 향후 연구로는 DTM의 파일 시스템에서 블록 단위 파일에 대한 백업 및 복구에 대한 연구와 네트워크를 통하여 송수신 가능한 DTAR를 제어하는 DTM 서버를 설계 및 구현할 계획이다.

참고문헌

- [1] POSIX.1003.1-1988, POSIX.1003.1-2001
- [2] Standard ECMA-119, Volume and File Structure of CDROM for Information Interchange, 2nd edition (December 1987)
- [3] <http://linux.die.net/man/1/shar>
- [4] Peek J, O'Reilly T, Loukides M. 1997. Unix Power Tools. O'Reilly & Associates, Inc.
- [5] The 32-bit PA-RISC Run-time Architecture Document, HP-UX 11.0 Version 1.0, Hewlett-Packard, 1997.
- [6] Hiroyuki Kawano, Management of Storage Devices and FileFormats in Web Archive Systems, ISORA'10, 2010