

SAN환경에서 NDMP를 이용한 백업소프트웨어

(Backup Software for SAN with NDMP)

복 경 수 [†] 황 흥 연 ^{**} 송 석 일 ^{***} 유 재 수 ^{****}

(Kyoung Soo Bok) (Hong Youn Hwang) (Seok Il Song) (Jae Soo Yoo)

요 약 최근 SAN과 NAS 같은 기술의 발전과 함께 대용량의 데이터를 효과적으로 백업하기 위한 요구들이 증가되고 있다. 본 논문에서는 네트워크 기반의 백업을 위해 제안된 개방형 표준 프로토콜인 NDMP를 이용하여 SAN 환경에 적합한 백업 소프트웨어를 설계하고 구현한다. 제안하는 백업 소프트웨어는 SAN을 기반으로 한 LAN-free 백업을 지원한다. 또한 사용자의 스케줄링에 의한 자동 백업과 다양한 유형의 백업 및 복구 타입을 지원한다. 파일 시스템, 원시 장치 그리고 데이터베이스에 대한 백업 및 복구를 수행한다. 향후 제안하는 백업 소프트웨어는 백업 센터로 사용할 수 있다.

키워드 : 백업 소프트웨어, SAN, NDMP, NAS

Abstract Recently, as new technologies such as SAN and NAS come into wide use to deal with a large amount of data, an efficient backup software for SAN and NAS is very required. In this paper, we design and implement a backup software for SAN that fully supports NDMP. The NDMP is an open standard protocol for network-based backup. The proposed backup software has various unique features such as SAN based lan-free backup, automatic and manual backup, on-line backup by using snap shot, file-system, raw-device, database backup and so on. The proposed backup software also can be configured as a backup center that uses SAN as a backup media.

Key words : Backup Software, SAN, NDMP, NAS

1. 서 론

사용자의 실수 또는 천재지변으로 인한 개인 혹은 기업의 정보 손실은 경제적으로 막대한 손해를 미치게 된다. 만약 기업에서 전산시스템의 고장이나 데이터의 유실로 정지시간이 생긴다면 그 손해는 평균 한 시간동안 100만 달러 이상이라고 한다. 특히 전산업무를 많이 하는 기업의 경우에는 그 액수가 2배에 이른다[1]. 이러한 정지시간을 가능한 최소화하기 위해서는 최신의 백업 데이터를 항상 유지하고 복구 할 수 있어야 한다. 따라

서 각종 데이터 및 파일 시스템을 보호하기 위해서는 데이터를 주기적 또는 비주기적으로 백업하는 것이 필수적이다. 효율적인 백업관리를 위해서는 백업 장치를 통제하며 원하는 시간에 자동 백업 및 복구가 수행되는 자동화된 백업 소프트웨어가 필요하다[2,3].

최근 데이터 통신 기술 및 저장 매체의 발전과 더불어 대용량의 데이터를 가공 처리하기 위한 많은 연구들이 진행되고 있다. 이와 함께 NAS(Network Attached Storage)와 SAN(Storage Area Network) 등의 보급으로 저장 매체의 대용량화는 가속화되고 있다. NAS는 이더넷이나 TCP/IP와 같은 프로토콜을 사용하여 LAN을 통해 다수의 저장장치들을 연결한다. 이러한 NAS는 유닉스의 NFS(Network File System)와 도스/윈도우의 SMB(Server Message Block)와 같은 파일 입출력 요청에 적합하다.

이에 반해 SAN은 대규모 네트워크 클라이언트들을 위하여 서버와 서로 다른 종류의 저장장치들을 광 채널을 통해 연결하는 특수목적용 고속 네트워크 또는 서브네트워크이다. 대체로, SAN은 한 기업의 전체 컴퓨팅 자원을

· 본 논문은 2002년도 산업자원부 차세대신기술개발 사업의 지원과 2003년도 ETRI 컴퓨터소프트웨어연구소 위탁과제 연구비 지원에 의하여 수행되었음

† 비 회 원 : 충북대학교 정보통신공학과
ksbok@netdb.chungbuk.ac.kr

** 비 회 원 : 매크로임팩트 연구원
hwang1021@macroimpact.com

*** 비 회 원 : 한국과학기술원 전자전산학과
prince@netdb.chungbuk.ac.kr

**** 종신회원 : 충북대학교 정보통신공학과 교수
yjs@cbucc.chungbuk.ac.kr

논문접수 : 2002년 10월 23일

심사완료 : 2003년 5월 15일

연결해 놓은 네트워크의 일부가 된다. SAN은 IBM S/390 메인프레임과 같은 이기종 컴퓨팅 자원이 아주 근접하여 밀집해 있게 되는 것이 보통이지만 백업이나 데이터의 저장을 위해 ATM이나 SONET 등과 같은 광역통신망 기술을 이용하여 원거리에 있는 장소로 확장될 수도 있다. SAN은 디스크 미러링, 백업 및 복원, 영구보관 및 영구보관용 데이터의 검색, 다른 저장장치로 데이터 이동 그리고 네트워크 상의 서로 다른 서버들 간에 데이터의 공유 등을 지원한다[4].

이러한 SAN 기반에서 백업은 LAN-free 백업과 Serverless 백업을 지원할 수 있다. LAN-free 백업은 데이터의 흐름이 LAN을 경유하지 않고 광 채널을 통해 구성된 별도의 네트워크를 통해 저장장치로 직접 백업을 수행한다. 따라서 네트워크 트래픽을 줄여 대용량 데이터의 백업을 하는데 유리하다. Serverless 백업은 데이터가 백업 서버를 경유하지 않고 장치들 사이에 직접적인 백업을 수행하는 것으로 서버의 부하를 줄이기 때문에 백업 성능이 향상된다. 또한 SAN은 다수의 테이프 장치의 부착 및 동적인 할당이 가능하여 관리비용을 절감할 수 있다[5,6].

Network Appliance와 IntelliGuard Software는 NAS 기반에서 이기종 간의 백업을 위한 개방형 표준 프로토콜로써 NDMP(Network Data Management Protocol)를 제안하였다. NDMP는 현재까지 IETF(Internet Engineering Task Force)에 버전 5가 제안되었으며 SAN을 지원하기 위한 버전을 개발 중이다[7,8]. 또한 NDMP의 기본적인 기능 및 인터페이스를 구현해 놓은 NDMP SDK(Software Development Kit)가 존재한다. 이러한 NDMP SDK를 이용하여 다양한 플랫폼에서 백업 소프트웨어를 좀더 쉽고 빠르게 개발할 수 있다.

메리랜드 대학에서는 대용량 테이프 장치에 대한 백업을 수행하기 위해 AMANDA(Advanced Maryland Automated Network Disk Archiver)라는 공개용 백업 소프트웨어를 개발하였다. 그러나 AMANDA는 현재 표준 프로토콜로 제안된 NDMP를 지원하지 못하며 SAN 환경에도 적합하지 못하다는 문제점이다. AMANDA 이외에 SAN 기반의 백업을 위한 다수의 백업 소프트웨어들이 개발되었다. 그러나 이러한 백업 소프트웨어는 대부분 외국에서 개발된 제품으로 NAS 기반의 백업을 위해서만 NDMP를 지원하고 있다. 또한 NDMP 또는 SAN을 지원하는 기능을 별도의 옵션으로 개발하여 판매하고 있기 때문에 구축비용이 많이 소요된다. 현재 국내에서는 NAS 기반의 백업을 지원하는 백업 소프트웨어는 일부 개발되었으나 NDMP를 지원하거나 SAN 기

반의 백업을 지원하는 소프트웨어는 거의 없는 상황이다. 따라서 NDMP를 지원하는 것은 물론 SAN 기반의 백업을 지원하기 위한 백업 소프트웨어에 대한 개발이 필요하다.

본 논문에서는 향후 SAN 기반의 백업을 위한 표준 프로토콜로 백업을 위한 프로토콜인 NDMP를 이용하여 NAS뿐만 아니라 SAN에 적합한 백업 소프트웨어를 설계하고 구현한다. 제안하는 백업 소프트웨어는 SAN을 기반으로 LAN-free 백업을 지원하며 원격에 존재하는 데이터의 백업 및 SAN 클러스터 내의 데이터 백업이 가능하다. 또한 여러 클라이언트들의 다양한 백업 요구를 스케줄링하여 처리하여 원하는 시간에 백업을 수행할 수 있도록 하는 기능을 지원한다. 또한 하나의 백업 서버가 백업 및 복구를 중앙 집중적으로 관리하기 때문에 일관된 정책을 유지할 수 있고 수행 중인 작업에 대한 제어를 효과적으로 수행할 수 있다. 또한 테이프 장치의 가상화를 통해 SAN 상에 부착되어 있는 다수의 테이프 장치를 효율적으로 관리하기 위해서 테이프 장치 목록을 유지한다. 따라서 백업에 필요한 파라미터를 결정할 때 테이프 목록에 접근하여 가용한 테이프 장치를 검색할 수 있어 다수의 테이프 장치 중 어느 한 테이프 장치로 과도한 부하가 걸리는 것을 방지한다. 제안하는 백업 소프트웨어는 SAN 상의 방대한 데이터를 보호할 뿐 아니라 향후 SAN 장비를 백업 센터로 사용할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 분야 및 기존에 개발된 백업 소프트웨어의 특징에 대해 기술하고, 3장에서는 제안하는 백업 시스템의 구조에 대해서 크게 백업과 복구로 나누어 설명한다. 4장에서는 제안하는 백업 소프트웨어의 구현한 내용 및 실험 결과와 다른 백업 소프트웨어와 비교 분석한 내용을 기술한다. 마지막으로 5장에서는 결론에 대해서 기술한다.

2. 관련 연구

2.1 백업(BACKUP)

백업은 전산장비의 고장이나 다른 불의의 사고에 대비하여 파일 또는 데이터베이스를 복사해 두는 행위를 말한다. 데이터의 백업은 대형 컴퓨터를 운영하는 대규모 사업체에서는 물론 개인 컴퓨터에서도 필수적이고 일상적인 업무이다. 그러나 이러한 백업은 종종 무시되는 경향이 있다.

현재까지 구현된 각종 상용 및 아카데미 백업 소프트웨어는 각각의 소프트웨어마다 다양한 형태의 백업 기능을 제공하고 있다[9]. 백업은 백업의 형태, 기능, 백업

시스템의 구조 등에 따라서 백업 방식을 구분할 수 있다. 우선 어떠한 형태로 백업이 이루어지는지 기능별로 구분하여 본다. 모든 파일을 백업할 것인지 변경된 부분만을 백업할 지에 따라 전체 백업(full backup)과 부분 백업(partial backup)으로 구분할 수 있다. 전체 백업은 백업을 수행할 파일시스템이나 장치 내에 존재하는 모든 파일에 대하여 백업을 수행한다. 부분 백업은 이전 백업에 대해서 변경된 파일만을 백업을 수행한다. 부분 백업은 이전의 마지막 전체 백업 이후에 변경된 데이터를 백업하는 차등 백업(differential backup)과 바로 이전의 백업 이후에 변경된 데이터를 백업하는 점진 백업(incremental backup)으로 나뉜다. 전체 백업은 전체 파일 시스템을 읽고 쓰기 때문에 저속이며 많은 양의 백업 공간을 소비한다. 부분 백업은 새로 생성되거나 변경된 파일만을 변경하므로 백업할 데이터의 양을 줄여 고속의 백업이 가능하다. 실제로 백업이 이루어지는 형태는 전체 백업과 부분 백업이 복합적으로 사용하는 다단계 백업이 많이 이루어지고 있다. 다단계 백업은 전체 백업과 함께 0부터 9까지의 레벨에 따른 점진 백업을 수행하는 것으로 좀 더 적은 비용으로 긴 백업 보장 기간을 갖게 된다.

백업할 대상이 파일 시스템인지 장치인지에 따라 나누어지기도 한다. 파일 기반의 백업(file based backup)은 파일 구조를 파악하여 백업장치에 파일과 디렉토리구조를 저장한다. 따라서 백업 수행시간은 오래 걸리지만 복원은 빠르게 수행할 수 있다. 이에 반해 장치 기반의 백업(device based backup)은 파일구조를 무시하고 디스크 블록을 백업장치에 기록한다. 따라서 백업 성능은 향상되지만 복원시간이 다소 오래 걸리는 단점이 있다. 클라이언트들이 작업을 수행하는 중에도 백업을 수행할 수 있도록 지원하는 온라인 백업(online backup)과 클라이언트의 모든 작업을 중단해야 하는 오프라인 백업(offline backup)이 있다. 이 외에도 데이터의 압축을 지원하는지, 동시에 여러 개의 백업을 수행하는지 그리고 백업 수행 시 데이터에 대한 바이러스 체크를 하는지에 따라 구분할 수가 있다.

또한 백업 시스템의 구성에 따라서 LAN 백업과 LAN free 백업 그리고 Serverless 백업으로 나뉜다. LAN 백업은 LAN을 기반으로 한 백업의 형태로써 업무용 네트워크를 통해 백업을 수행하기 때문에 병목 현상이 발생하는 단점이 있다. 대용량의 데이터 백업을 수행할 때 백업 서버는 많은 CPU, 메모리 그리고 서버 I/O를 수행한다. 모든 서버에 백업 어플리케이션과 백업 장치가 1:1로 연결된 구조를 가진 지역 관리 백업

(locally managed backup)과 이를 하나의 백업서버가 중앙에서 관리하는 중앙 집중적 백업(centralized managed backup)으로 다시 나눌 수 있다[10].

SAN 기반의 백업 시스템은 LAN-free 백업과 Serverless 백업이다. LAN free 백업은 테이프 장치를 LAN에서 SAN 기반에 연결하여 데이터의 흐름을 LAN을 거치지 않고 백업을 수행한다. 따라서 네트워크 트래픽을 줄일 수 있다. LAN free 백업이 데이터의 네트워크 트래픽을 줄이기 위한 것이라면 Serverless 백업은 백업 서버를 거치지 않고 백업 서버의 CPU, 메모리의 사용, 서버 I/O 등을 사용하지 않는다. 이러한 SAN 기반의 백업은 LAN 기반의 8MB/sec 정도의 백업속도에 비해서 상대적으로 매우 빠른 100MB/sec의 백업 및 복구 속도를 가진다. 그리고 다수의 서버와 다수의 백업 저장장치들을 동적으로 공유할 수 있다는 장점을 가지고 있다. 이와 같이 백업 장비를 공유함으로써 백업 장비 도입 및 관리의 비용을 절감할 수도 있다.

2.2 NDMP

NDMP는 Network Appliance NetApp과 IntelliGuard Software가 NAS 기반에서 이기종 간의 백업을 위해 제안된 개방형 표준 프로토콜이다. NDMP는 클라이언트/서버 구조로 DMA(Data Management Application)와 DSP(Data Service Provider)로 구성된다. NDMP에서 백업과 복구를 수행하기 위해서는 먼저 DMA와 DSP들에 NDMP 세션을 설정해야 한다. NDMP 세션이란 데이터 송신자와 수신자들 사이의 데이터 흐름의 집합으로 데이터 서비스와 테이프 서비스들 사이의 데이터 연결(data connection)과 DMA와 각 서비스들 사이의 제어 연결(control connection)을 의미한다. 각 NDMP 서버의 상태를 제어하고 관리하기 위해 DMA는 XDR(eXternal Data Representation) 형태의 메시지 전송한다.

DMA는 NDMP 세션을 생성하고 관리하며 NDMP 서버에 해당하는 NDMP 데이터 서비스와 테이프 서비스 사이의 데이터 전송을 관리하기 위해 하나 이상의 NDMP 서버와의 상호 작용을 수행한다. 또한 DMA는 NDMP의 정책, 클라이언트 인터페이스 그리고 DSP를 관리, 제어한다.

DSP는 실제적인 백업을 수행하기 위한 테이프 서비스(Tape Service), 데이터 서비스(Data Service), SCSI 창구 서비스(SCSI Pass Through Service) 그리고 전송 서비스(Translate Service) 등을 제공하고 있다. 데이터 서비스는 주 저장장치에 대한 NDMP 인터페이스를 제공하며 테이프 서비스는 보조 저장장치에 대한

NDMP 인터페이스를 제공한다. SCSI 창구 서비스는 DMA가 SCSI장치에 SCSI 명령어를 사용할 수 있는 기능을 제공한다. 버전 5에서 새로 등장한 전송 서비스는 데이터 스트림의 멀티플렉싱 및 압축 기능을 제공한다.

[그림 1]은 NDMP의 구조를 나타낸다. DMA와 데이터 서비스와 테이프 서비스를 제공하는 NDMP 서버 사이의 관계성에 따라 NDMP의 위상이 결정된다. [그림 1]의 (a)와 같이 데이터 서비스와 테이프 서비스가 같은 호스트에 존재하는 경우에는 DMA는 NDMP 서버로부터 로컬에 존재하는 테이프 장치에 백업을 수행한다. 이때 NDMP 제어 연결은 네트워크를 통해 연결되지만 NDMP 데이터 연결은 NDMP 서버가 구현되어 있는 호스트 내에 존재한다. 이에 반해 [그림 1]의 (b)는 데이터 서비스와 테이프 서비스가 서로 다른 호스트에 존재하는 경우로 TCP/IP를 통해 서로 다른 호스트에 존재하는 NDMP 서버에 데이터를 전송한다. 이때, NDMP 제어 연결은 서로 다른 두 개의 호스트에 네트워크로 통해 연결하고 NDMP 데이터 연결은 네트워크를 통해 두 개의 NDMP 서버 사이에 이루어진다.

NDMP에서 백업 수행 과정은 다음과 같이 이루어진다. 먼저 DMA는 NDMP 테이프 서버에 통신 채널을 생성하고 백업을 위해 테이프 장치를 오픈하고 위치시킨다. 테이프 장치에 대한 준비가 완료되면 NDMP 데이터 서버를 연결하고 백업된 데이터에 대한 정보를 저장하기 위한 히스토리 파일을 지원할 것인지에 대한 정보를 요청한다. 각각의 NDMP 서버에 대한 준비가 완료되면 DMA는 백업을 위한 준비과정을 수행한다. 백업은 준비 과정은 NDMP 데이터 서버에 백업을 수행할

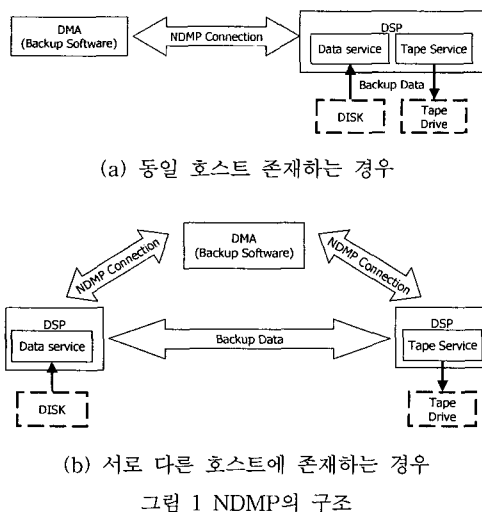
것이라는 메시지를 전송하고 어떠한 형태의 백업이 수행될 것인지를 전송한다. NDMP 데이터 서버는 NDMP 테이프 서버에 데이터 연결을 설정하고 NDMP 테이프 서버는 전송된 데이터를 받기 위해 mover 준비시킨다. mover가 준비되면 데이터 서버는 테이프 서버에 데이터를 전송한다. 백업 과정 중 특별한 이벤트가 발생하면 NDMP 서버는 DMA에게 알림 메시지를 전송하고 필요한 작업을 수행하도록 한다. NDMP의 복구 과정은 NDMP 테이프 서버에서 데이터 서버로 전송된다는 것 이외에는 백업 과정과 매우 유사하다.

2.3 백업 소프트웨어

AMANDA는 미국의 메리랜드 대학에서 개발된 공개용 백업 소프트웨어이다. AMANDA가 개발되는 시점에 ExaByte 8mm, DAT 4mm와 같은 대용량 테이프 장치들이 개발되었다. AMANDA는 이러한 테이프 장치들을 이용하여 하나의 백업 서버가 여러 클라이언트의 데이터를 하나의 대용량 테이프 장치에 백업을 수행한다. AMANDA는 전형적인 LAN 기반의 중앙 집중식 백업 형태를 지니고 있다. AMANDA는 테이프 장치 이외에도 테이프 체인저(tape changer)와 같은 백업 장치에 대한 제어 및 관리를 한다. 따라서 한 개의 대용량 테이프 장치만 사용하기 위한 자체적인 백업 프로토콜을 가지고 있다. 그러나 AMANDA는 NDMP를 지원하지 않는다. AMANDA의 프로토콜에는 하루에 수행하여야 할 백업의 양에 따라서 백업의 레벨을 결정하는 독특한 백업 스케줄링 기법을 포함한다. 또한 백업할 데이터를 보내기 전에 선택적으로 compress 명령이나 gzip 명령으로 압축할 수 있다. 백업이 수행된 후에는 각종 에러를 포함한 백업 정보를 관리자 또는 클라이언트의 이메일을 통해 보내준다[11,12,13].

AMANDA는 백업과 복구를 수행하는 명령은 UNIX dump/restore 그리고 GNU tar 등을 이용한다. 가능한 빠른 백업을 위해서 옵션으로 홀딩 디스크(holding disk)를 설정할 수 있다. 하나의 테이프 드라이브가 다수의 클라이언트의 백업 요구를 처리해야 하기 때문에 홀딩 디스크가 없을 경우에는 대기하는 백업요구는 더 많은 시간을 기다려야만 한다. AMANDA의 구조는 [그림 2]와 같다.

클라이언트 측의 amandad는 계속 서버의 요청을 받기 위해 데몬으로 동작하고 있다. 서버 측의 planner는 백업 계획을 만들기 위해서 amandad에게 백업 정보를 요청한다. 이에 따라 서버 측의 dumper는 planner가 만들어 놓은 백업 계획에 의해서 amandad에게 백업을 수행하도록 sendbackup을 요청한다. planner와 dumper



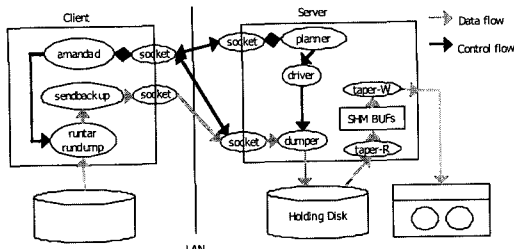


그림 2 AMANDA의 시스템 구성도

사이에는 driver가 존재하며 driver는 dumper에게 dump 수행을 홀딩 디스크(holding disk)로 할 것인지 직접 taper로 할 것인지 결정한다.

taper는 [그림 2]에서 보는 것과 같이 홀딩 디스크(holding disk) 혹은 dumper로부터 직접 데이터를 읽어 들이는 부분(file reader)과 테이프 드라이브에 쓰는 부분(taper writer)이다. reader와 writer 사이에는 32*1024 바이트의 크기를 가진 20개의 버퍼로 구성된 공유메모리 영역을 사용하고 있다. 클라이언트의 amandad는 dumper로부터 백업을 하도록 지시를 받게 되면 tar 명령이나 dump 명령을 수행한다. 만약 AMANDA를 SAN 환경에 적합하도록 하기 위해서는 Holding 디스크나 테이프 장치들을 SAN 클러스터에 연결하고 서버에서 각 장치나 디스크에 백업 명령을 수행하도록 설계해야 한다.

AMANDA 이외에 현재 개방형 시스템 환경에서 백업 및 복구를 위해 수백 개의 상업용 백업 소프트웨어 제품들이 출시되었다. SAN 환경을 지원하는 대표적인 백업 소프트웨어는 NetBackup, Time Navigator, Networker 등이 있다. NetBackup은 네트워크 환경에서 자동화된 백업 및 복구를 위한 시스템을 구축하기 위해 Veritas에서 개발한 백업 소프트웨어이다[14]. Netbackup은 백업 서버 간에 마스터와 슬레이브로 구성할 수 있어 백업의 중앙 집중관리 기능과 대용량의 백업 장치를 공유할 수 있는 기능을 제공한다. 백업 이미지에 대해 산업체 표준인 "tar" 형식을 지원하며 다양한 형태의 데이터베이스에 대한 백업을 지원한다. 또한 백업 이미지에 대해 대소문자를 구별하지 않고 검색하는 기능을 제공한다.

Time Navigator는 Atempo에서 시스템의 고장시간을 최소화하기 위해 개발한 백업 소프트웨어로 기업과 인터넷 데이터 센터를 포함하여 IT 서비스 공급자들을 위하여 설계되었다[15]. Time Navigator는 다양한 플랫폼에서 사용 중인 파일시스템, 데이터베이스 및 응용들을 대단위의 테이프 백업장치에 정해진 스케줄에 따라

자동으로 백업을 수행할 수 있다. 또한 NAS, SAN, IP 스토리지와 같이 급격한 백업 환경의 변화에 시스템의 부하를 최소화하면서 백업을 수행할 수 있다.

Networker는 Legato에서 개발한 백업 소프트웨어로 클라이언트/서버의 구조로 구성된다[16]. Network의 이러한 구조는 복잡한 네트워크 환경에서의 데이터 관리와 보호에 요구되는 유연성과 높은 성능을 가능하게 하며 원격지에서 중앙 집중적인 관리가 가능하다. Network는 클라이언트 병렬처리와 데이터 압축으로 서버와 네트워크의 부하를 최소화할 수 있을 뿐만 아니라 NAS, SAN, LAN 등의 네트워크 환경에서 높은 성능의 백업과 복구 기능을 제공한다. 또한 최소화 전자우편 및 SNMP 모듈을 통한 자동적인 이벤트 통지 기능을 제공하고 있다.

2.4 문제점 및 개발 동기

앞에서 언급한 것과 현재 백업을 위한 많은 소프트웨어들이 개발되었다. 메리랜드 대학에서는 대용량의 테이프 드라이브에 대한 백업을 지원하기 위해 AMANDA라는 공개용 백업 소프트웨어를 개발하였다. 그러나 AMANDA는 표준 프로토콜로 제한된 NDMP를 지원하지 못하며 SAN 기반의 백업을 지원하지 못하는 문제점을 갖고 있다.

현재 SAN 환경을 지원할 지원하는 백업 소프트웨어들은 대부분 외국에서 개발된 제품으로 국내에서는 단순히 외국에서 개발된 제품을 이용하여 시스템을 구축하고 있는 상황이다. 따라서 국내에서는 NDMP와 SAN 기반의 백업 소프트웨어를 개발하기 위한 기술력이 거의 없는 상황이다. SAN 기반의 백업을 위해 개발된 소프트웨어들은 NDMP를 지원하지만 NAS 기반의 백업을 위해서만 NDMP를 지원하고 있다. 또한 SAN 기반의 백업을 위한 기능들을 옵션 형태로 지원하고 그 내용이 공개되지 않기 때문에 연구용으로 활용하기 어렵다는 문제점이 있다. SAN에 연결된 각각의 장치들은 그 용도에 따라 tar, dump, cpio, raw 데이터 백업과 같은 다양한 형태의 백업 타입이 지원되어야 한다. 그러나 대부분의 백업 소프트웨어들은 tar와 같은 특정 백업 타입만을 지원하고 있다.

따라서 NDMP에 대한 기술력을 확보하고 SAN 기반의 백업을 지원하기 위한 백업 소프트웨어에 대한 개발이 필요하다. 또한 그 용도에 맞는 다양한 형태의 백업 타입을 지원하기 위한 백업 소프트웨어가 필요하다.

3. 백업 소프트웨어의 설계

본 논문에서 제안하는 백업 소프트웨어는 SAN 기반

의 백업에 적합하도록 설계되었으며 향후 SAN 기반의 백업을 위한 표준 프로토콜로 사용될 NDMP를 지원하도록 설계하였다. 제안하는 백업 소프트웨어는 Lan-free 백업을 지원하며 백업 또는 복구 수행 과정에서 SAN 클러스터에 존재하는 장치들 사이의 데이터 전송은 NDMP 프로토콜을 통해 수행한다. 또한 다수 클라이언트에 의해 계획된 스케줄링에 의해 자동 백업을 지원하며 온라인과 오프라인 백업을 지원한다. 제안하는 백업 소프트웨어의 특징은 다음과 같다.

- 향후 SAN 기반의 백업을 위한 표준 프로토콜로 사용될 NDMP를 지원
- Lan-free 백업과 스케줄링에 의한 자동 백업 지원
- 파일 시스템과 데이터베이스에 대한 다양한 백업 방식 지원
- 전체 백업과 부분 백업 지원
- 온라인과 오프라인 백업 지원
- SAN 클러스터 존재하는 다양한 장치들 사이의 백업 및 복구 지원
- 부분 복원과 전체 복원 기능 지원
- 백업 및 복구에 대한 중앙 집중식 관리
- 로그 관리 및 작업 관리 기능 지원
- 백업 센터로의 활용이 가능

위에서 언급한 특징들을 지원하기 위해 제안하는 백업 소프트웨어는 [그림 3]과 같이 구성된다. 백업/복구 요청자는 지역 혹은 원격에서 백업/복구 서버에 접속하여 백업/복구 요청을 한다. 클라이언트 프로그램은 백업 서버에 대한 인증 절차를 수행하고 백업/복구에 필요한 인자를 입력받아 필요한 인자를 서버로 전송한다. 백업/복구 서버는 클라이언트로부터 백업/복구 인자를 받아 백업 또는 복구를 수행하기 위한 장치들이 사용 가능 여부를 확인한다. 만약 장치가 가용할 경우 백업 또는 복구에 필요한 파라미터를 설정하고 백업/복구 프로세스를 생성한다. 생성된 프로세스는 NDMP의 DMA에 해당하며 각 장치에 실제적인 백업과 복구를 수행하기 위해 DSP와 NDMP 제어 연결을 수행한다.

제안하는 백업 소프트웨어는 모든 백업과 복구에 처리를 중앙 집중식으로 관리하며 클라이언트 또는 관리자에게 백업 및 복구에 관련된 모니터링 기능과 작업관리 기능을 제공한다. 클라이언트 또는 관리자는 필요에 따라 수행 중인 작업에 대한 상태 정보를 확인하고 수행 중인 작업에 대한 중지/정지/재시작과 같은 작업을 수행할 수 있다. 또한 클라이언트가 작업한 스케줄링을 확인하거나 변경할 수 있으며 백업 및 복구 과정에서

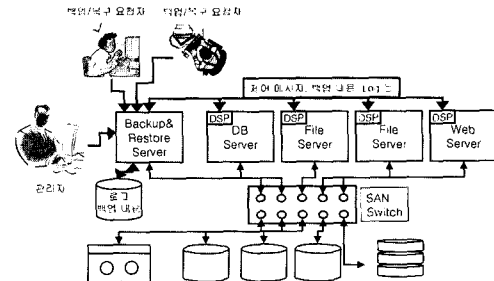


그림 3 제안하는 백업 S/W의 시스템 구성도

생성된 로그 정보를 확인할 수 있다. 관리자는 모든 클라이언트의 작업을 관리할 수 있으며 필요에 따라 클라이언트에 대한 권한을 부여하고 백업 장치들을 관리하는 기능을 수행한다.

3.1 백업

제안하는 백업 소프트웨어는 SAN 환경에서의 백업을 처리하기 위해 다음과 같은 두 가지의 백업을 고려해야 한다. 첫째, SAN 클러스터 내의 모든 데이터를 테이프 등의 보조저장장치로 백업을 하는 경우이다. 여기서 백업되는 데이터는 SAN 환경 내의 일반 파일 및 파일시스템, 데이터베이스 등의 대용량 데이터이다. 둘째, 원격으로 떨어져 있는 클라이언트의 주기억장치에 있는 데이터를 SAN의 데이터영역에 백업하는 것이다. 제안하는 백업 소프트웨어는 SAN을 백업 센터로 확장하여 이용하도록 설계한다. 백업센터란 원격으로 떨어져 있는 여러 클라이언트의 로컬 디스크에 있는 데이터를 백업할 수 있는 시스템이다. 즉, SAN 또는 LAN으로 연결된 사용자의 로컬 디스크에 있는 데이터를 SAN 클러스터에 연결된 백업 장치에 백업을 수행할 수 있는 시스템을 의미한다.

[그림 4]는 백업 처리 과정을 나타낸 것이다. 백업 요청자에 해당하는 클라이언트는 백업을 요청하고 수행 중인 작업에 대한 상태 정보를 확인하거나 필요에 따라 작업에 대한 중지 및 재시작 명령을 수행한다. 백업 서버에 해당하는 백업 서버 데몬과 클라이언트 데몬, 스케줄러 그리고 백업 프로세스는 클라이언트의 요청에 따라 자동 백업 또는 스케줄링에 의해 자동 백업을 수행한다.

클라이언트가 자동 백업을 위해 스케줄링 파일을 작성하면 작성된 스케줄링 파일과 백업에 대한 인자와 함께 서버측으로 전송된다. 백업 서버에 존재하는 클라이언트 데몬은 주기적으로 각각의 클라이언트가 작성한 스케줄링 파일을 검사하여 수행해야 할 백업 요청이 있는지를 확인한다. 만약 수행해야 할 백업 요청이 존재하는 경우

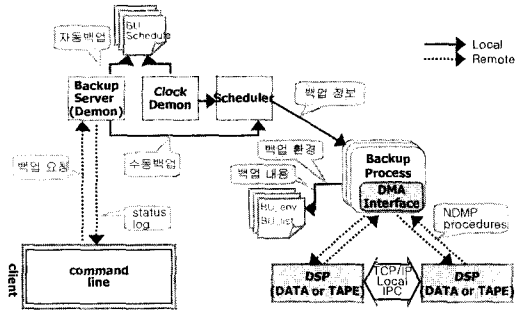


그림 4 백업 처리 과정

클럭 데몬(Clock Demon)은 메시지 큐에 백업 요청을 위한 메시지를 삽입하고 스케줄러를 동작시킨다. 이에 반해 수동 백업은 클라이언트가 백업을 요청하는 즉시 수행되기 때문에 클럭 데몬을 거치지 않고 직접 스케줄러에게 백업을 요청한다.

스케줄러는 메시지 큐를 검사하여 수행해야 할 백업 요청이 전달되었는지를 확인한다. 만약 수행해야 할 백업 요청이 전달되었다면 스케줄러는 백업 장치를 검사하고 백업을 위한 파라미터를 설정한다. 만약 SAN 클러스터 내에 가용한 백업 장치가 없는 경우에 주기적으로 백업 장치 목록을 다시 검사하여 가용한 백업 장치가 있는지를 검사한다. 스케줄러가 백업 파라미터를 생성하면 실제적인 백업을 수행하기 위한 백업 프로세스를 생성한다.

백업 프로세스는 스케줄러에서 전달된 백업 파라미터를 이용하여 NDMP 프로토콜에 따라 백업을 수행하도록 한다. 백업 프로세스는 NDMP의 DMA에 해당하며 실제적인 백업을 관리하고 제어한다. 백업 프로세스는 NDMP에서 제공하는 다양한 인터페이스를 이용하여 DSP에 백업을 요청하고 필요한 메시지를 비동기적으로 처리한다. 백업을 수행하는 과정에서 NDMP의 로그 인터페이스와 파일 히스토리 인터페이스를 통해 로그 정보와 백업된 파일에 대한 정보를 기록한다. 백업 프로세스가 백업을 수행하면 파일 히스토리 인터페이스를 통해 백업 대상 목록에 대한 히스토리 정보와 백업 상태 정보를 기록한다. 이렇게 기록된 히스토리 테이블과 히스토리 파일은 부분 복구를 수행하기 위해 백업된 내용을 검색하기 위해 사용된다.

3.2 복구

클라이언트는 시스템의 손상 또는 데이터의 손실로 인해 백업된 데이터의 일부 또는 전체를 복구하고자 한다. 제안하는 백업 소프트웨어에서는 백업된 데이터의 부분 복구와 전체 복구 기능을 제공하며 백업된 데이터의 상태 정보를 확인하기 위한 검색 기능을 제공하고

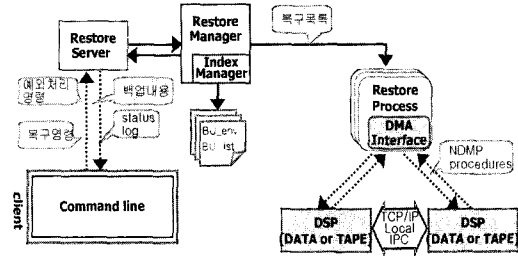


그림 5 복구 처리 과정

있다. [그림 5]는 복구 처리 과정을 나타낸 것이다. 복구 요청자는 필요에 따라 부분 복구 또는 전체 복구를 요청한다. 서버 측에서 복구를 수행하기 위해서는 복구 서버, 인덱스 관리자, 복구 프로세스에 의해 수행된다. 만약 데이터의 일부 손실이나 파손으로 백업된 데이터에 대한 부분 복구를 수행할 때 클라이언트는 이전에 백업된 내용을 검색하고 복구하고자 하는 데이터를 선택하여 복구 리스트를 작성한다. 이때, 클라이언트가 이전에 백업한 내용을 검색하기 위해서는 서버 측에 존재하는 인덱스 관리자를 통해 백업 과정에서 생성된 히스토리 파일과 히스토리 테이블을 검색한다. 클라이언트가 원하는 복구 리스트를 작성하여 서버로 전송하면 서버는 복구 리스트와 복구에 필요한 파라미터를 생성한다. 복구 관리자는 실제적인 복구를 수행하기 위한 복구 프로세스를 생성하고 복구에 필요한 파라미터를 프로세스에게 전달한다. 백업과 유사하게 복구 프로세스는 DMA에 해당하며 NDMP 프로토콜에 따라 DSP 사이의 복구 수행을 관리하고 제어한다.

만약 시스템의 손상으로 백업된 데이터를 이용하여 시스템을 복원하고자 할 경우 클라이언트는 전체 복구를 수행한다. 클라이언트가 전체 복구를 요청하면 복구 서버에 존재하는 인덱스 관리자는 자동적으로 가장 최근에 전체 백업된 날짜를 히스토리 테이블에서 검색한다. 전체 백업된 날짜를 검색하면 이를 기준으로 히스토리 테이블에서 부분 백업된 내용의 리스트를 작성하고 가장 최신의 복구 대상 목록을 생성하여 복구 프로세스에게 전달한다.

4. 설계된 백업 소프트웨어의 구현 및 특징

4.1 구현 환경

본 논문에서 제안하는 백업 프로그램의 구현 환경은 [표 1]과 같다. 대부분의 백업 프로그램의 구성 모듈은 gcc를 이용하고 실제적인 파일시스템이나 장치의 백업 및 복구를 수행하는 부분은 tar, dump, cpio, dd 명

령을 이용하고 데이터베이스 백업 부분은 ProC와 스크립트 언어를 이용하여 구현한다. DMA를 통해서 데이터 서비스와 테이프 서비스의 NDMP 연결을 관리하기 위해서 NDMP SDK를 이용한다. 본 백업 프로그램에서는 Red Hat Linux release 7.3에서 구현하여 테스트를 하였다. 또한 데이터베이스에 대한 백업은 오라클 9에 국한되어 수행되었으며 오라클에서 제공하는 온라인 백업과 오프 라인 백업 기능을 이용하여 NDMP를 통해 백업과 복구를 수행한다.

표 1 백업 프로그램 구현 환경

운영 체제	Red Hat Linux release 7.3 (Valhalla)
개발 언어	gcc-2.96 bash (GNU Bourne Again SHell)
파일/디바이스 백업	tar, dump, cpio, dd
데이터베이스 백업	tar, dump, cpio, dd, oraback.sh
NDMP	NDMP SDK v3.2

4.2 클라이언트 프로그램

클라이언트 프로그램은 클라이언트에게 명령을 입력 받고 백업서버와 필요한 정보를 송수신하는 역할을 한다. 현재 클라이언트 프로그램은 C언어를 이용하여 CLI (Command Line Interface) 방식으로 구현하였다. 백업 서버와 클라이언트 프로그램은 소켓을 통해 명령을 전달하고 필요에 따라 서버로부터 알람 메시지와 작업에 대한 상태 정보를 받는다. 클라이언트가 백업 또는 복구를 수행하기 위해 서버에 접속하기 위해서는 먼저 로그인 과정을 통해 접속 권한에 대한 인증을 수행한다. 로그인 과정에서 클라이언트의 권한 레벨이 부여되며 클라이언트가 사용할 수 있는 장치 목록이 설정된다. 로그인 과정이 완료되면 클라이언트는 실제적인 백업과 복구에 대한 명령을 서버로 전송한다.

클라이언트 프로그램의 대부분의 작업은 백업 요구 명령에 대한 수행이다. [표 2]는 백업 관련 명령을 보여 준다. aubu 명령은 매일 일정한 시간에 백업이 수행되도록 백업 요청을 할 때 사용된다. 클라이언트 프로그램을 통해서 클라이언트로부터 입력받는 인자는 백업을 수행할 시간, 명령(백업 프로그램), 압축여부, 데이터 서버, 데이터 서버 아이디/패스워드, 백업 파일(혹은 디바이스, 데이터베이스) 그리고 스케줄 파일이다. 백업을 수행할 시간은 일정한 시간 단위로 설정한다. 이와 같이 설정된 시간에 따라 클럭 데몬은 백업 파일 리스트의 유무를 검사한다. 백업 프로그램은 tar, cpio, dd 명령과 데이터베이스 백업을 위한 ora_p_on, ora_p_off, ora_l_tbs,

표 2 백업 관련 명령

명령	기능
aubu(ab)	자동백업(auto backup)
manbu(mb)	수동백업(manual backup)
bpl	백업 프로세스 리스트 보기
showbulist	자동 백업 리스트 보기
abortbu	백업 중지
continue	테이프 교체후 백업 계속하기(관리자)
showschlist	스케줄 파일 리스트 보기
showsch	스케줄 파일 보기
makesch	스케줄 파일 작성

ora_l_tb, ora_l_own, ora_l_full 명령을 한다. 백업 파일은 백업을 수행할 파일 목록, 디렉토리, 디바이스, 데이터베이스 등을 지정한다. 데이터베이스 백업을 위한 명령에서는 DBMS(LOGICAL)인 경우 테이블스페이스 (Tablespaces), 테이블(Tables), 클라이언트(Owner), 전체(Full) 중 택일, DBMS(PHYSICAL)인 경우는 인스턴스를 백업 파일로 설정한다.

SAN의 특성상 클라이언트가 접속하는 데이터 서버가 여러 개 존재한다. 따라서 백업을 수행하고자 하는 데이터 서버를 알려줄 필요가 있으며 각 데이터 서버에 접근할 클라이언트 아이디와 패스워드를 입력해야 한다. 마지막으로 백업 프로그램이 dump나 tar가 선택되었을 경우에는 스케줄 파일을 결정한다. 이 스케줄 파일은 해당 날짜에 어떠한 레벨의 백업을 수행할지 결정하는데 사용된다.

manbu 명령은 수동백업을 수행하는 명령으로 자동백업과 유사한 인터페이스를 갖는다. 다만 백업을 요청한 즉시 수행이 되므로 백업 시간을 입력할 필요가 없다. 또한 단 한 번 수행되는 명령이기 때문에 스케줄 파일을 설정하지 않는다. bpl 명령은 스케줄러에 의해서 생성되어 현재 수행중인 백업 프로세스 리스트를 보여주는 명령이다. 일반 클라이언트의 경우에는 해당 아이디의 백업 프로세스 리스트를 보여주며 관리자인 경우에는 모든 백업 프로세스 리스트를 보여준다. 차후에 구현할 인터럽트 명령으로 어떤 백업 프로세스에게 명령을 보내야 할지 결정할 때 유용하다.

showbulist 명령은 특정 시간에 예약해 놓은 자동 백업 리스트를 보여준다. abortbu 명령은 백업 프로세스를 중지하도록 하는 명령이다. 관리자는 서버에 이상이 발생했거나, 점검을 할 필요가 있을 때 전체 프로세스를 중지하였다가 재시작 할 수 있어야 한다. 일반 클라이언트는 자신이 요청한 백업 프로세스만을 중지시킬 수 있

으며, 한 번 중지시킨 백업 프로세스를 재시작할 수는 없다. 현재 NDMP DMA는 이러한 인터럽트 명령을 받도록 수정하였다. co 명령은 테이프 장치에 문제가 발생하였을 때 이를 관리자가 처리하고, 멈춰 있던 프로세스에 작업을 재개한다. 이외에도 스케줄 파일을 만들고 (makesch 명령), 내용이나 목록을 보여주는 명령 (showsch 명령, schlist명령)이 있다.

다음 [표 3]은 복구 명령과 복구할 목록을 검색하기 위한 명령을 보여준다. set 명령은 복구하고자 하는 데이터 서버를 선택한다. 백업 테이프 드라이브에 저장되어 있는 백업 파일들은 여러 곳의 데이터 서버로부터 전송되어 온 것이다. 따라서 가장 먼저 필수적으로 이 명령을 수행하여 복구할 파일이 어느 데이터 서버로 복구될 것인지 설정한다. 일단 set 명령으로 복구할 데이터 서버를 설정하면 검색 및 복구 작업을 진행한다.

show, search, cd, ls 명령은 히스토리 파일을 검색하기 위한 명령이다. 검색 명령에 대한 설명은 4.4절 검색 부분에서 한다. 히스토리 파일을 검색하면 복구할 백업 파일이나 디렉토리를 add 명령으로 복구 목록에 추가하고, 필요에 따라 del 명령으로 복구 목록에서 삭제 가능하다. 모든 검색 작업과 복구 목록 작성이 끝나면 마지막으로 addressore 명령을 수행하여 복구 요청을 수행한다. 그리고 restore 명령은 선택된 데이터 서버의 마지막 전체 백업부터 최근의 부분 백업까지 복구를 수행한다.

백업이나 복구를 수행할 경우 DMA는 각종 에러 및 로그 정보를 기록한다. 이러한 정보를 클라이언트가 요

표 3 복구/검색 관련 명령

명령	기능
view	복구할 데이터서버 목록 보기
set	복구할 데이터서버 설정
restore	전체 복구
add	복구 리스트 추가
del	복구 리스트 삭제
addressrore	부분 복구
abortrestore	실행중인 복구 작업 중지
continue	테이프 교체 후 복구 계속(관리자)
show, search, cd, ls	인덱스 파일 검색 관련 명령

청하기 위해서 loglist 명령과 showlog 명령을 수행한다. 클라이언트는 서버에 저장되어 있는 로그의 리스트를 가져와서(loglist 명령) 원하는 로그의 정보를 레벨에 따라서 선택적으로 요구한다(showlog 명령). [그림 6]은 전송되는 로그정보의 일부이다.

4.3 백업 및 복구서버

백업 및 복구서버는 클라이언트의 요청에 따라 백업, 복구 그리고 모니터링 기능을 수행한다. 서버는 클라이언트에서 전송된 명령을 처리하고 이에 대한 성공 여부를 전송한다. 또한 현재 수행 중인 작업 또는 수행 예정인 작업에 대한 로그 정보나 상태 정보를 관리하고 클라이언트의 요청에 따라 원하는 정보를 가공 처리하여 클라이언트에게 전송한다.

[그림 7]은 구현한 서버의 전체적인 구조를 나타낸 것이다. 클라이언트의 요청에 따라 서버는 작업을 수행하

```
[NORMAL]:Received notify_connected request: version:3 reason:CONNECTED
[NORMAL]:connected 3
[NORMAL]:resid_count : 0
[NORMAL]:length : 31
[NORMAL]:data : [table]09092002_181352_dev_nst0
[NORMAL]:tape table : [table]09092002_181352_dev_nst0
.... 종략
[NORMAL]:block_size : 4096
[NORMAL]:blockno : 0
[NORMAL]:total_space : 0x0
[NORMAL]:space_remain : 0x0
[NORMAL]:partition : 0
[NORMAL]:data_connection_addr : LOCAL
[NORMAL]:bu_type:1, user_name: oracle, passwd:dhfkzmf, compress:1, bu_mode:0, bu_objstor:9
[NORMAL]:Received notify_data_halted: reason:INTERNAL_ERROR
.... 종략
[NORMAL]:data : [table]09092002_181352_dev_nst0
[NORMAL]:tape table : [table]09092002_181352_dev_nst0
[NORMAL]:resid_count : 0
[NORMAL]:NOREWIND : False
[NORMAL]:WR_PROT : False
.... 종략
[NORMAL]:Received
[NORMAL]:mover_halted
```

그림 6 로그 정보

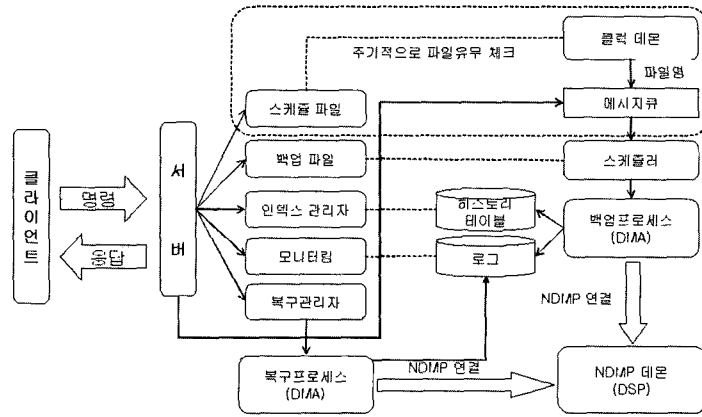


그림 7 백업 및 복구 서버 구조

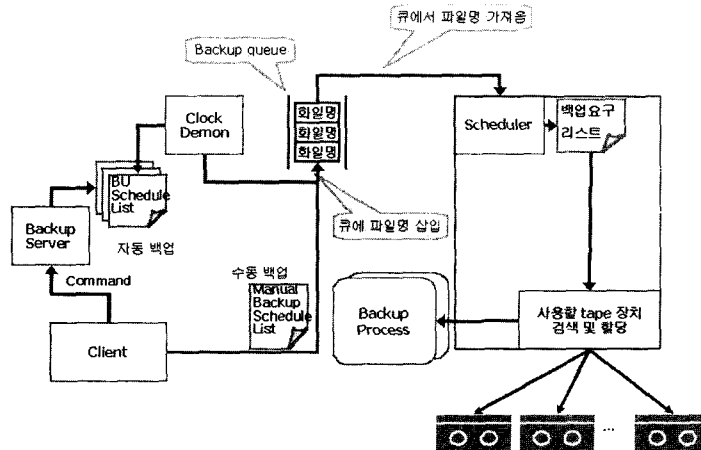


그림 8 클럭 데몬과 스케줄러

기 위해 클럭 데몬, 스케줄러, 모니터링, 인덱스 관리자 그리고 복구 관리자로 구성되어 있다. 주기적인 자동 백업을 수행하기 위해 작성한 스케줄 파일은 서버 측에서 관리되며 실제적인 백업을 위해 필요한 백업 타입, 백업 레벨, 백업 대상과 같은 정보들은 백업 파일에 저장되어 있다. 모니터링은 또는 복구 과정에서 생성된 로그 정보와 프로세스의 상태 정보를 확인하고 필요에 따라 검색하는 기능을 제공한다. 인덱스 관리자는 백업 프로세스에 의해 백업된 정보를 저장하고 클라이언트가 원하는 복구 데이터를 검색할 수 있도록 한다. 또한 복구 관리자는 부분 또는 전체 복구를 수행하기 위한 프로세스를 생성하고 복구에 필요한 다양한 정보들을 관리한다.

4.4 클럭 데몬 및 스케줄러

클럭 데몬은 서버 측에서 동작하는 데몬으로 클라이

언트의 자동 백업 요청을 주기적으로 검사하여 스케줄러에게 백업 요청을 전달하는 역할을 수행한다. 자동 백업을 위해 클라이언트가 작성한 스케줄 파일은 클라이언트 별로 서버 측에 저장되게 된다. 클럭 데몬은 특정 주기마다 각 클라이언트의 스케줄 파일을 검사하여 현재 시점에 수행해야 할 백업 요청을 스케줄러에게 전달하는 역할을 수행한다. 스케줄러는 클럭 데몬 또는 서버 직접 전달된 백업 요청을 수행하기 위해 필요한 백업 파라미터를 생성하여 백업 프로세스에게 전달하는 역할을 수행한다.

[그림 8]은 서버 측에서 수행하는 클럭 데몬과 스케줄러의 수행 과정을 나타낸 것이다. 클럭 데몬은 서버 측에서 정해진 시간마다 스케줄 파일을 검사하여 수행해야 할 백업 요청에 대한 스케줄 파일을 메시지 큐에 삽

입한다. 만약 클라이언트가 즉시 실행해야 할 수동 백업을 요청하는 경우에는 클럭 데몬을 거치지 않고 메시지 큐에 삽입되게 된다. 스케줄러는 메시지 큐를 검사하여 백업 프로세스에게 전달할 백업 파라미터를 생성한다. 백업 파라미터는 수행해야 할 백업 레벨, 백업 장치, 백업 타입, 백업 대상 목록 그리고 기타 백업에 필요한 옵션들로 구성된다. 스케줄러는 백업 장치를 선택하기 위해 로그인 과정에서 클라이언트에게 설정된 백업 장치 목록을 검사하고 가용한 백업 장치가 있는지를 검사한다. 만약 가용한 백업 장치가 존재하지 않는 경우에는 주기적으로 다시 검사를 수행한다.

스케줄러는 백업을 수행하기 위해 필요한 파라미터를 생성하면 백업 프로세스에게 백업 파라미터를 전달하여 실제적인 백업을 수행하도록 한다. [그림 9]은 스케줄러에서 백업프로세서로 전달하는 백업 파라미터를 보여준다.

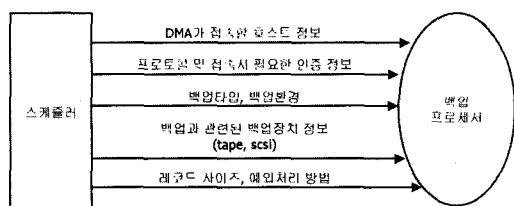


그림 9 백업 파라미터 전달

[그림 10]은 스케줄러의 알고리즘으로 백업 파라미터를 생성하여 백업 프로세스를 생성하기까지의 과정을 나타낸 것이다. 스케줄러는 set_metadata_path()를 통해

```

알고리즘 : Scheduler
입력 : 없음
출력 : 성공하면 0, 실패하면 1
{
  init_file_name_queue(); //배지지를 초기화
  set_metadata_path(); //메타 데이터의 저장 위치를 설정
  while(1){
    msgrcv(); // 메시지 큐에 입력된 메시지를 판별
    백업 정보가 들어 있는 파일명 설정;
    for(){
      get_buinfo(); //백업 정보를 읽음
      if(자동백업)
        set_bulevel(); // 백업 레벨을 설정
      tape_search(); // 가용한 백업 장치를 검색
      make_bupara(); // 백업 파라미터를 생성
      send_burequest(); // 백업 프로세스를 생성
    }
  }
}
    
```

그림 10 스케줄러 알고리즘

백업에 관련된 메타 데이터가 저장된 위치를 설정하고 msgrcv()을 통해 메시지 큐에 입력된 메시지를 판별하여 백업 정보가 들어 있는 파일명을 설정한다. 백업 정보에는 백업 대상 목록, 백업 타입 그리고 기타 옵션들로 구성되어 있다. get_buinfo()는 백업 정보가 저장되어 있는 파일에서 백업에 필요한 정보를 읽고 tape_search()을 통해 백업을 수행할 백업 장치를 선택한다. send_burequest()는 make_bupara()을 통해 생성된 백업 파라미터를 통해 실제적인 백업을 수행할 프로세스를 생성하여 백업 파라미터를 전달한다.

4.5 인덱스 관리자(Index Manager)

인덱스 관리자는 백업을 수행하는 과정에서 백업 프로세스에 의해 기록된 히스토리 정보를 관리하고 클라이언트의 질의에 따라 백업 내용을 검색할 수 있도록 하는 기능을 제공한다. 백업 프로세스는 백업에 대한 정보와 백업된 데이터에 대한 목록을 클라이언트 별로 인덱스 디렉토리에 저장한다. [그림 11]은 인덱스 관리자가 관리하는 인덱스 디렉토리의 구조를 나타낸다. 제안하는 백업 소프트웨어를 설치할 때 시스템 환경 변수에 클라이언트의 백업 정보를 기록하기 위한 인덱스 디렉토리가 설정되어 있다. 백업 프로세스는 인덱스 디렉토리에 각 클라이언트 별로 히스토리 테이블과 히스토리 파일을 관리한다.

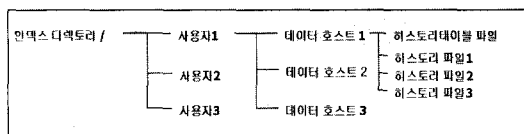


그림 11 인덱스 디렉토리의 구조

히스토리 테이블은 각 클라이언트의 백업 요청에 따라 DSP의 데이터 서버에 해당하는 호스트 별로 저장되며 클라이언트가 특정 데이터 서버에 존재하는 데이터를 백업한 전체적인 정보를 관리한다. 히스토리 테이블에는 백업을 수행될 때마다 생성되는 히스토리 파일, 백업 환경, 백업 장치 목록 정보, 백업 날짜, 백업 목록의 최상위 디렉토리, 백업된 데이터의 크기 등을 저장한다. 히스토리 파일은 백업이 수행될 때마다 하나씩 생성되며 백업 목록에 대한 히스토리 정보를 기록한다. 백업 목록에 대한 히스토리 정보에는 백업된 데이터의 경로, 크기, 클라이언트, 권한 등과 같이 파일 또는 디렉토리의 상태 정보를 기록한다.

백업 프로세스에 의해 기록된 히스토리 테이블과 히스토리 파일은 백업된 내용에 대한 확인 및 복구 목록

의 생성하기 위해 백업된 내용에 대한 목록을 검색하기 위해 사용된다. 백업된 내용의 확인 및 복구 목록의 생성을 위해 날짜, 디렉토리, 파일을 기준으로 백업된 내용을 검색할 수 있다. 또한 날짜를 기준으로 한 검색에서는 플래그와 함께 주어진 날짜, 이전, 이후에 대한 검색을 지정할 수 있다. 인덱스에 대한 검색은 먼저 사용자의 아이디를 통해 히스토리 테이블이 저장된 경로를 지정하여 파일을 오픈한다.

[그림 12]는 인덱스 관리자에 의해 관리되는 백업 정보를 검색하는 알고리즘이다. 백업 정보를 확인하기 위해서는 히스토리가 오픈되면 백업 날짜, 디렉토리, 파일을 통해 히스토리 정보를 검색할 수 있다. 만약 클라이언트가 날짜에 대한 값을 입력하였다면 날짜에 대한 플래그를 설정하고 read_fh_table()을 통해 히스토리 테이블 파일에서 날짜의 조건에 맞는 히스토리 테이블을 검색한다. 사용자가 특정 파일이나 특정 디렉토리에 대한 내용을 검색하고 하는 경우에는 search_directory()를 통해 사용자가 원하는 디렉토리 또는 파일이 백업되었는지를 검사한다. 만약 사용자가 특정 디렉토리에 대한 검색을 수행하는 경우 search_directory()는 히스토리 테이블에서 백업된 최상위 디렉토리를 비교하여 사용자가 원하는 디렉토리와 관련된 디렉토리인 경우에 히스토리 파일을 검색한다. 히스토리 파일에는 백업된 모든 디렉토리와 파일에 대한 정보가 기록되어 있기 때문에 이를 통해 사용자가 원하는 디렉토리 또는 파일이 백업되었는지를 판별할 수 있다.

백업된 히스토리에 대한 검색을 수행하면 서버는 조건에 맞는 히스토리 테이블의 정보와 백업된 최상위 디렉토리 리스트를 클라이언트에게 전송한다. 클라이언트는 유닉스나 리눅스에서 사용하는 ls, cd와 유사한 명령을 이용하여 백업 내용을 확인하거나 특정 디렉토리로 이동할 수 있다.

```

알고리즘 : search_index
입력 : date, directory, file, flag
출력 : result_history
{
  히스토리 테이블을 저장한 파일 오픈;
  if(date){
    날짜 플래그 설정; //날짜를 기준으로 검색할 플래그를 설정
    read_fh_table(); //날짜를 기준으로 히스토리 테이블 검색
    search_directory(); // 디렉토리 또는 파일을 검색
  }
  else{
    read_fh_table();
    search_directory();
  }
}

```

그림 12 백업 히스토리 검색

인덱스 관리자는 필요에 따라 클라이언트가 특정 파일이나 디렉토리에 대한 부분 복구를 수행하고자 복구 리스트를 작성할 수 있는 기능을 제공한다. 클라이언트는 검색의 결과로 반환된 디렉토리를 순회하면서 add 명령을 통해 디렉토리나 파일을 복구 리스트에 추가할 수 있다. 또한 list 명령을 통해 복구 리스트에 있는 내용을 확인하고 del 명령을 통해 복구 리스트에 있는 내용을 삭제할 수 있다.

4.6 복구 관리자(Restore Manager)

백업 과정에서 스케줄러는 클라이언트의 백업 요청에 따라 백업 파라미터를 생성하고 이를 복구 프로세스에게 전달하는 역할을 수행하였다. 이와 유사하게 복구 관리자는 복구 파라미터를 생성하고 이를 복구 프로세스에게 전달하는 역할을 수행한다. 제안하는 백업 소프트웨어는 부분 복구를 위한 addressore와 시스템의 전체 복구를 위한 restore를 기능을 제공한다. addressore는 백업된 내용을 검색하여 사용자가 작성한 백업 리스트를 통해 특정 디렉토리 또는 파일을 복구한다. 이에 반해 restore는 시스템의 손상이나 디스크 파손으로 손실된 시스템을 복원하는 기능을 수행한다.

[그림 13]은 전체 복구를 수행하는 restore 알고리즘이다. 전체 복구를 수행하기 위해 먼저 백업된 히스토리 정보를 기록한 히스토리 파일을 오픈하고 read_fh_table()를 통해 히스토리 테이블의 내용을 읽는다. 만약 클라이언트가 시스템의 특정 디렉토리에 손실된 내용을 복구하고자 할 경우 search_dir()를 통해 히스토리 테이블에서 해당 디렉토리와 관련된 디렉토리가 가장 최근에 전체 백업된 디렉토리를 검색한다. 가장 최근에 전체 백업된 디렉토리가 존재한다면 make_restore_para()를 통해 디렉토리에 대한 전체 백업을 수행한 이후에 백업된 내용을 통해 복구 목록을 생성한다.

```

알고리즘 : restore
입력 : dir, user_id, auth_id, auth_passwd
출력 : 성공하면 0, 실패하면 1
{
  히스토리 테이블을 저장한 파일 오픈;
  read_fh_table(); // 히스토리 파일을 읽음
  if(특정디렉토리복구){
    히스토리 파일에서 복구할 디렉토리 설정;
    search_dir(); // 해당 디렉토리를 백업한 마지막 전체 백업을 검색
    make_restore_para(); // 복구 목록을 생성
  }
  else{
    현재 파일 시스템 전체를 복구 디렉토리로 설정;
    make_restore_para();
  }
  send_restore_request();
}

```

그림 13 restore 알고리즘

이에 반해 시스템 전체에 대한 복구를 수행하고 할 경우 시스템 전체가 전체 백업 내용을 검색하여 전체 백업 이후에 백업된 내용을 통해 복구 목록을 생성한다. 복구 목록이 생성되면 히스토리 테이블에 있는 백업 장치 정보와 최상위 디렉토리 정보를 이용하여 복구 파라미터를 생성한다. 복구 파라미터가 생성되면 send_restore_request()를 통해 복구 프로세스를 생성하고 복구 프로세스에 복구 파라미터를 전달한다. addressotre의 수행 과정은 미리 작성된 복구 리스트를 이용하여 복구파라미터를 생성한다는 것 이외에는 restore와 유사하다.

4.7 백업 및 복구 프로세스

스케줄러 또는 복구 관리자는 실제적인 백업을 수행하기 위한 프로세스를 생성한다. 생성된 프로세스는 NDMP의 DMA에 해당되며 NDMP 프로토콜에 따라 DSP들을 관리하고 제어하는 기능을 수행한다. [그림 14]는 백업 프로세스의 기능을 나타낸 것이다. 스케줄러로부터 백업 파라미터를 전달받은 백업 프로세스는 DMA 인터페이스를 호출한다. DMA 인터페이스는 DSP에 대한 NDMP 제어 연결을 설정하여 백업을 수행하고 DSP들 사이의 NDMP 데이터 연결을 설정하도록 한다. 이때, DSP에 대한 NDMP 제어 연결은 백업 파라미터로 전달된 백업 장치와 백업 목록을 이용하여 실제적인 장치들 사이의 연결을 수행한다.

백업 프로세스는 백업 과정에서 백업된 파일 또는 디렉토리에 대한 정보를 히스토리 파일에 기록하고 백업이 완료되면 백업 장치, 백업 크기, 백업 타입, 백업된 최상위 디렉토리, 생성된 히스토리 파일에 대한 정보를 히스토리 테이블에 기록한다. 또한 백업 과정에서 발생하는 로그 정보와 알림 메시지를 백업 서버에게 전달하여 클라이언트가 수행 중인 작업 또는 완료된 작업에 대한 모니터링을 수행할 수 있도록 한다.

[그림 15]는 복구 프로세스의 구조이다. 복구 관리자로부터 복구 파라미터를 전달받은 복구 프로세스는 DMA 인터페이스를 호출하고 NDMP 프로토콜에 따라

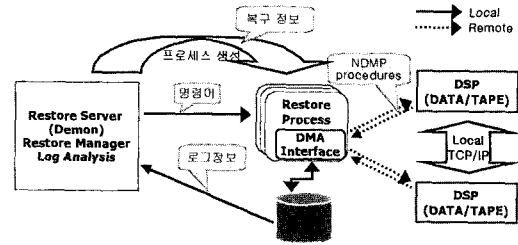


그림 15 복구 프로세스 구조

복구를 수행한다. 백업과 마찬가지로 복구를 수행할 DSP에 대한 제어 연결은 복구 파라미터로 전달된 백업 장치와 복구 리스트를 이용하여 설정한다. 또한 로그와 알림 메시지를 복구 서버에 전달하고 클라이언트가 모니터링 할 수 있도록 한다.

4.8 데이터베이스 백업

제안하는 백업 소프트웨어는 데이터베이스의 백업을 지원한다. 현재 데이터베이스에 대한 백업은 오라클만을 지원하고 있으며 물리적 백업과 논리적 백업을 지원한다[17]. 데이터베이스의 물리적 백업은 쉘 스크립트와, Pro C로 구현되어 있으며 오라클에서 제공하는 기능을 이용하여 백업 해야할 파일들의 목록을 구하고 온라인/오프라인 백업의 여부에 따라서 이를 청상적으로 처리할 수 있도록 하는 작업을 수행한다. 최종적으로 실제 파일들의 백업은 tar나 dd를 이용하여 수행된다[18,19]. 온라인 백업을 지원하는 방법으로는 ORACLE의 백업 모드를 이용하는 방법과 스냅샷(snapshot)을 이용하는 방법이 있다. ORACLE의 백업모드에서는 백업하는 동안에 테이블스페이스의 내용을 변경하지 않고 재수행 로그에만 기록하고, 백업이 끝난 이후에 재수행 로그를 데이터베이스에 반영한다. 반면에 스냅샷을 이용하는 경우에는 백업 모드를 이용할 필요가 없어 데이터베이스의 성능을 감소시키지 않는다. 논리적 백업은 ORACLE의 exp, imp 명령을 이용해서 클라이언트, 테이블, 전체 데이터베이스를 백업한다.

4.9 실험 및 비교 분석

본 절에서는 제안하는 백업 소프트웨어에 대한 실험 결과 및 제안하는 백업 소프트웨어의 우수성을 입증하기 위해 기존에 제안된 백업 소프트웨어의 비교 분석을 수행한다. [표 4]는 제안하는 소프트웨어에 대한 백업한 시간을 나타낸 것이다. 실험을 위한 50M 바이트, 100M 바이트, 200M 바이트의 데이터에 대해 3번의 백업을 수행한 평균 값을 구한 것이다. 동일한 크기에 대한 백업을 수행하는 과정에 실험을 수행할 때마다 약간의 시간

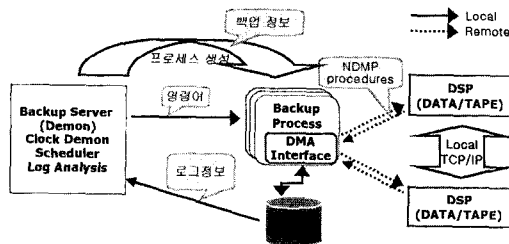


그림 14 백업 프로세스 구조

표 4 실험 결과

백업 데이터	백업 시간
50M 바이트	약 40 초
100M 바이트	약 55 초
200M 바이트	약 68 초
500M 바이트	약 91 초

을 차이를 보였는데 이는 테이프 장치를 인식하고 백업할 위치로 헤더가 이동하는 시간 때문에 발생하는 것이다. 그러나 [표 4]에서 보는 것과 같이 백업 데이터의 크기의 증가에 비해 백업되는 시간의 증가는 거의 없다. 복구 시간은 복구할 데이터를 검색하는 시간을 제외하고 백업 시간과 거의 동일하다.

제안하는 백업 소프트웨어는 NDMP 프로토콜을 이용해서 SAN 환경에서 적합하도록 설계되었다. 여러 데이터 서버 및 테이프 장치, 그 외의 많은 백업 장치가 한 SAN 구성 내에 존재할 수 있다. 이러한 경우 소스 데이터의 위치, 백업 데이터의 위치, 백업 시간, 백업 종류 등 많은 정보를 유지해야 한다. NDMP 프로토콜을 이용해서 이러한 정보를 모두 유지할 수 있다. 필요에 따라 백업 히스토리 정보를 이용해서 백업 내용의 일부분이나 전체를 복구를 수행할 수 있다.

현재 구현되어 있는 백업 소프트웨어를 정상적으로 동작시키기 위해서는 서버의 데몬들이 root 권한으로 실행되어야 한다. TAR, CPIO를 이용한 백업 및 복구가 가능하며, 또한 데이터베이스의 백업 및 복구가 가능하다. 현재 구현되어 있는 기능은 오라클만을 지원하고 있다. 물리적 백업에서는 데이터, 재수행 로그, 제어파일을 tar, cpio, dd, restore를 이용해서 백업 및 복구를 수행

할 수 있다. 온라인 및 오프라인 백업이 가능하다. 논리적 백업을 지원하기 위해서 exp/imp 명령을 이용하였다 [20]. 이로써 전체 데이터베이스 혹은 특정 클라이언트의 데이터베이스 및 테이블을 백업할 수 있다.

[표 5]는 AMANDA 및 현재 출시된 상용제품과 제안하는 백업 소프트웨어를 비교 분석한 것이다. 제안하는 백업 소프트웨어는 상용제품의 대부분의 기능을 수용하였으며 AMANDA에 비해 SAN 환경에 적합할 뿐만 아니라 NDMP 및 다양한 백업 타입을 지원하는 특징을 가진다. 중앙 집중적 관리를 통해 원활한 백업/복구가 가능하며 파일시스템의 백업, 파일시스템 및 원시 데이터의 백업 그리고 데이터베이스 백업을 지원한다. 또한 NDMP와 완벽한 호환이 이루어지며 데이터 서비스와 테이프 서비스의 위치에 따라 LAN 백업과 LAN-free 백업을 할 수 있다.

5. 결론

본 논문에서는 SAN 상의 방대한 데이터를 보호뿐만 아니라, SAN 장비를 백업 센터로 사용할 수 있는 백업 소프트웨어를 설계하고 구현했다. 제안하는 백업 소프트웨어는 다양한 백업 타입을 제공하며 SAN 환경에 적합하도록 LAN-free 백업과 온라인 백업을 수행할 수 있다. 제공하는 백업 타입으로는 파일 백업, 장치 백업, Raw 장치 백업, 데이터베이스 백업이 있다. 또한 백업을 위한 개방형 표준 프로토콜인 NDMP를 지원한다. 여러 클라이언트의 백업 요구를 스케줄링하여 처리하고 원하는 시간에 복구 및 백업을 수행하기 위해 하나의 서버가 중앙집중적으로 관리하며, 테이프 장치의 가상화 및 할당 정책으로 하나의 테이프장치로 과부하가 걸리는 것을 방지한다. 원격으로 떨어져 있는 클라이언트의

표 5 백업 소프트웨어의 특징 비교

기능	AMANDA	NetBackup	Time Navigator	Networker	제안하는 백업 S/W
중앙집중식 관리	○	○	○	○	○
NDMP 호환	×	○	○	○	○
파일시스템 백업	○	○	○	○	○
데이터베이스 백업 (NDMP 이용)	×	×	×	×	○
raw 장치 백업 (NDMP 이용)	×	×	×	×	○
압축	○	○	○	○	○
온라인 백업	×	○	○	○	○
LAN-free 백업	×	○	○	○	○
LAN 백업	○	○	○	○	○

주 저장장치에 있는 데이터를 백업하기 위하여 SAN을 백업 장치로 활용 가능하다.

참 고 문 헌

[1] "Backup/Recovery Tutorial," Storage Networking Industry Association, 2001.

[2] A. L. Chervenak, V. Vellanki, Z. Kurmas and V. Gupta. "Protecting File System, A Survey of Backup Techniques." Pro. Joint NASA and IEEE Mass Storage Conference, 1998.

[3] "왜 백업이 필요한가?," <http://www.veritas.co.kr/whitepaper.asp>

[4] 조성훈, 김성주, 이준호, 이주영, 박석천, "SAN의 구조와 기술 요소", 한국정보처리 학회지, 제8권, 제4호, pp.19-28, 2001.

[5] "Storage Area Network - Backup and SAN," http://www.atempo.com/library/pdf/US/wp_tinasan_us.pdf

[6] Technical Brief Time Navigator, "Backup and Restore On Storage Area Network," http://www.atempo.com/library/pdf/US/wp_tinasan_us.pdf

[7] "Network Data Management Protocol (NDMP)," <http://www.ndmp.org/wp/wp.shtml>

[8] White Paper Time Navigator for NDMP, "Backup of Network Attached Storage," http://www.atempo.com/library/pdf/US/tina_ndmp.pdf

[9] W. Curtis Preston and Gigi Estabrook, "UNIX Backup and Recovery," O'REILLY, 730 pages, 1st Edition November 1999.

[10] "Backup Software RFI," <http://www.backupcentral.com/rfi.html>

[11] J. da Silva and O. Guomundsson, "The Amanda Network Backup Manager," Pro. USENIX Systems Administration (LISA VII), pp.171-182, 1993.

[12] J. da Silva, O. Guomundsson and D. Mosse, "Performance of a Parallel Network Backup Manager," Pro. USENIX, pp.17-26, 1992.

[13] "Using Amanda," <http://www.backupcentral.com/amanda.html>

[14] "Veritas NetBackup," <http://www.veritas.com/products/>

[15] "Time Navigator," <http://www.atempo.com/products/html/US/>

[16] "Networker," <http://www.legato.com/>

[17] "Database Backup Overview," http://www.oracle.com/ip/std_infrastructure/blueprints/emailonhp/blueprint_email/Operations/Backup/Database_Backup_Procedures_and_Scripts/Database_Backup_Overview.htm

[18] TAR, "Red Hat Linux release 7.3 (Valhalla) man page"

[19] CPIO, "Red Hat Linux release 7.3 (Valhalla) man

page"

[20] The oraback.sh Script, "<http://www.geocities.com/orabackup/>"



박 경 수

1998년 충북대학교 수학과(이학사). 2000년 충북대학교 정보통신공학과(공학석사) 2000년~현재 충북대학교 정보통신공학과 박사과정. 관심분야는 데이터베이스 시스템, 내용기반 멀티미디어 검색, 저장 시스템, 고차원 색인 구조, 시공간 색인

구조 등



황 홍 연

1999년 충북대학교 정보통신공학과(공학사). 2003년 충북대학교 정보통신공학과(공학석사). 2003년~현재 매크로임팩트주임연구원. 관심분야는 XML 저장 관리 시스템, 자료 저장 시스템, 멀티미디어 데이터베이스 등



송 석 일

1998년 충북대학교 정보통신공학과(공학사). 2000년 충북대학교 정보통신공학과(공학석사). 2003년 충북대학교 정보통신공학과(공학박사). 2003년~현재 KAIST 전자전산학과 박사후연구원. 관심분야는 데이터베이스 시스템, 트랜잭션, 저장 시스템, 멀티미디어 정보검색, XML, 정보검색 프로토콜 등



유 재 수

1989년 전북대학교 공과대학 컴퓨터공학과(학사). 1991년 한국과학기술원 전산학과(공학석사). 1995년 한국과학기술원 전산학과(공학박사). 1995년~1996년 목포대학교 전산통계학과 전임강사. 1996년~현재 충북대학교 전기전자 및 컴퓨터공학부 부교수. 관심분야는 데이터베이스 시스템, 정보검색, 멀티미디어 데이터베이스, 분산 객체 컴퓨팅