

대용량 데이터 서비스를 위한 SANique Smart Vault 백업 시스템의 설계 및 구현

이 규 웅[†]

요 약

다양한 데이터 집중한 응용 서비스의 증가로 인해 사용자 데이터가 급증하고 있으며 방대한 양의 데이터 저장 및 백업에 대한 관심이 높아지고 있다. 특히 기존 일반적 증분백업 기술들이 SAN 기반의 대용량 데이터 서비스 환경에 적합하지 않아 백업 시스템의 성능저하 문제가 심각하다. *SANique Smart Vault* 시스템은 이러한 요구사항을 만족하는 데이터 중복제거 기반의 고성능 백업 솔루션이다. 본 논문에서는 *SANique Smart Vault* 시스템의 개략적 시스템 구조에 대하여 설명하고 특히 백업대상 목록의 탐색과정을 개선한 저널링 기반의 델타 증분백업 기술 및 레코드 레벨의 데이터 중복제거 기능의 구현 기술에 대하여 설명한다. 또한 제안된 무기한 증분백업 기능과 데이터 중복 제거 기능은 타 상용 시스템들과의 성능비교를 통해 백업 오버헤드 측면의 성능 우수성을 보인다.

주제어 : 데이터 백업, 데이터 중복제거, 증분백업, 저널, 데이터 청킹

Design and Implementation of SANique Smart Vault Backup System for Massive Data Services

Kyu Woong Lee

ABSTRACT

There is a lot of interest in the data storage and backup systems according to increasing the data intensive services and related user's data. The overhead of backup performance in massive storage system is a critical issue because the traditional incremental backup strategies causes the time consuming bottleneck in the SAN environment. The *SANique Smart Vault* system is a high performance backup solution with data de-duplication technology and it guarantees these requirements. In this paper, we describe the architecture of *SANique Smart Vault* system and illustrate efficient delta incremental backup method based on journaling files. We also present the record-level data de-duplication method in our proposed backup system. The proposed forever incremental backup and data de-duplication algorithms are analyzed and investigated by performance evaluation of other commercial backup solutions.

Keywords : Backup, Data De-duplication, Incremental Backup, Journal, Data Chucking

[†] 정 회 원: 상지대학교 교수(교신저자)

논문접수: 2013년 12월 23일, 심사완료: 2014년 2월 12일, 게재확정: 2014년 2월 12일

* 본 논문은 2012학년도 상지대학교 교내연구비의 지원으로 수행되었음

1. 서론

대규모 사용자와 방대한 양의 데이터가 지속적으로 생성되는 데이터 집중적 응용 서비스들이 증가하고 있다. 이러한 응용서비스들은 서비스 지속성 및 성능을 중요시하므로 24시간 무중단 서비스와 함께 효율적인 저장관리 및 백업을 제공해야 한다. 테라바이트급 이상의 데이터를 효율적으로 처리하기 위한 저장시스템의 문제와 병행하여 백업 시스템의 오버헤드에 대한 문제가 관심받고 있다. 특히 상용 백업 시스템들은 아직 대용량 데이터에 대한 요구사항과 중복제거기능을 동시에 반영하여 상품화하지 못하고 있다. 인터넷을 통하여 유통될 사용자 데이터는 지속적으로 증가하여 테라바이트를 넘어 엑사(Exa) 바이트급 이상의 규모로 확장될 것으로 전망하고 있다. 데이터의 급격한 증가에 따른 대용량 저장시스템 및 백업 시스템의 개선이 필요한 시점이다.

일반적으로 기존의 파일 시스템 및 백업 시스템들은 DAS(Direct Attached Storage) 기반으로서 데이터를 공유하거나 방대한 크기의 사용자 데이터를 효율적으로 저장하거나 백업하기에 많은 문제점을 안고 있다. 이러한 문제점을 해결하면서 동시에 클라우드 시스템과 같은 대용량 데이터 스토리지 서비스를 지원하기 위해서 NAS(Network Attached Storage) 또는 SAN(Storage Area Network)과 같은 전용 저장장치 구조를 이용한 저장관리 및 백업 시스템이 제안되었다. SAN은 서버와 저장장치 사이를 광채널로 연결한 특수목적의 전용 네트워크이다. 클러스터화된 서버들과 데이터 공유 및 관리가 가능하며, 광채널을 이용한 전용 네트워크를 사용하므로 대용량의 파일전송에 적합하며 대용량 백업시스템에서도 시간을 단축할 수 있다[1].

기존의 백업 기술들은 SAN 구조에 적합하지 않으며 또한 테라바이트급 이상의 대용량 데이터 처리에 비효율적인 단점을 갖고 있다. 데이터의 용량이 급증함에 따라 백업 데이터의 탐색방법 및 전송, 분산 통합관리에 대한 오버헤드가 커지고 있다. 따라서 SAN 기반의 클러스터 파일 시스템을 활용하는 데이터 집중적 응용서비스 시스템은 대용량 데이터에 대해 효율적으로 백업 데이터를 탐색하고 관리하는 새로운 요구사항을 갖는다. SAN 기반 백업 서버

는 수천대의 클라이언트로부터 전송되는 대용량 백업 데이터에 대해 SAN 구조의 특징을 활용하여 병목현상 없이 효율적으로 백업 데이터를 탐색하고 수행해야 한다. 기존의 일반적인 증분백업 기술은 이러한 요구사항을 수용할 수 없으므로 새로운 환경에 적합한 클러스터 백업 시스템의 설계 및 개발이 필요하다.

본 논문에서는 대용량 데이터 처리 요구사항 및 SAN 기반의 특성을 만족하는 매크로임팩트(주)사의 클러스터 기반 백업 시스템 *SANique Smart Vault*의 설계 및 구현내용에 대하여 기술한다. *Smart Vault* 백업 시스템은 대용량 데이터 처리를 위해 저널링(journaling) 기반의 델타 증분(delta incremental) 백업과 무기한 증분(never incremental) 백업 기술을 이용하여 데이터 볼륨의 전체 스캔 없이 백업 목표 데이터 목록을 작성하므로 백업시간을 단축할 수 있다. 또한 대규모 사용자가 사용하는 클러스터 파일 시스템은 데이터 중복성이 높은 특징을 고려하여 데이터 중복 제거 기능을 갖는 백업 시스템을 설계하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 기존 백업 시스템 및 상용 소프트웨어들의 특징을 분석하고 제3장에서는 SAN기반의 클러스터 백업 시스템 *Smart Vault*의 시스템 구조와 효율적 백업을 위한 데몬 프로세스의 구성에 대하여 기술한다. 제4장에서는 제안하는 백업 시스템의 저널링 기반 델타 증분 백업과 데이터 중복 제거 기능의 구현내용에 대하여 설명한다. 제5장에서는 본 논문의 델타 증분백업 및 데이터 중복 제거기능에 대한 성능평가를 기술한다. 끝으로 제6장에서 본 논문의 결론을 맺는다.

2. 기존 백업 시스템의 분석

백업 시스템에서 사용하는 일반적인 백업기술을 백업 범위와 백업대상 기술에 대하여 분석하고 대용량 데이터 집중적 응용서비스에 부적합성을 기술한다. 또한 SAN 기반의 기존 상용 백업 소프트웨어들의 특징과 대용량 데이터 처리를 위해 데이터 중복성에 대해 처리하는 백업시스템에 대해 분석한다.

2.1 기존 백업 기술의 분석

일반적인 백업 기술은 백업의 대상범위에 따라 전체(full) 백업, 증분(incremental) 백업, 차등(differential) 백업으로 분류되며, 백업 데이터의 기술에 따라 파일 기반의 백업과 물리적 블록 기반의 백업, 데이터 압축 등의 방법으로 분류된다 [2]. 전체백업은 파일 시스템 전체를 백업 미디어에 저장하는 것으로 성능 오버헤드가 커서 대용량 데이터 집중적 응용서비스에 적용하기 많은 어려움을 갖고 있다. 증분백업은 이전 백업 시점 이후에 변경된 데이터 블록 등에 대한 부분만 백업하는 기법이다. 백업의 데이터 용량은 작아지지만 변경 데이터를 탐색하여 백업 대상 데이터를 구분해야하는 오버헤드가 크다. 차등 백업은 최초 백업시 증분백업과 동일하다. 그러나 그 이후의 차등백업시 이전 차등 백업에서 수행한 백업을 그대로 다시 백업대상에 포함하는 특징을 갖는다. 전체백업 이후 변경된 모든 파일에 대하여 백업을 수행하므로 증분백업보다 백업의 데이터 용량이 많아진다. 백업을 복구하는 시점에서 살펴보면 전체백업 기법이 가장 복구가 쉽다. 차등백업은 전체백업을 복구하고 그 이후 가장 마지막 수행된 차등백업 부분에 대해 한번만 복구하면 된다. 증분백업은 전체백업 이후 수행된 모든 증분백업에 대해서 차례로 복구 작업을 수행해야 하므로 가장 번거로운 작업을 수행하는 단점을 갖는다. 상용 제품들은 세 가지 백업전략을 선택적으로 적용할 수 있도록 하여, 응용 시스템에 적합한 백업 시나리오를 갖도록 한다. 일반적으로 많이 사용되는 증분백업 기술은 대용량 데이터 처리시 백업 대상 데이터 탐색에 대한 오버헤드가 커져 새로운 증분백업 기술의 개발이 필요하다.

또한 백업 대상 데이터의 분류에 따라 파일기반 백업과 물리적 블록 기반 백업으로 분류한다. 파일 기반 백업은 파일 시스템 구조를 그대로 탐색하여 백업을 수행하므로 각각의 파일에 대해서 복구 및 배제가 가능한 장점을 갖는다. 그러나 파일이 디스크 내에 연속적 블록으로 위치하지 않기 때문에 빈번한 디스크 탐색 오버헤드가 발생하고 파일 일부의 변화가 발생해도 전체 파일을 백업해야 하는 단점을 갖는다. 대용량 데이터 서비스를 위한 백업 시스템에서는 파일의 크기가 상대적으로 크기 때문에 이러한 단점은 백업 성능에 매우

비효율적이다. 물리적 블록기반 백업은 블록 수준으로 증분백업을 수행하므로 디스크 오버헤드가 적어 저장매체의 효율성이 높은 반면 파일 단위의 복구가 불가능한 단점을 갖는다.

본 논문에서 소개하는 백업 시스템에서는 델타 증분백업 기술을 이용하여 데이터 볼륨의 크기에 상관없이 파일의 일부분만 백업할 수 있는 효율적인 레코드 수준의 델타 증분백업 기법을 구현하였다.

2.2 기존 SAN 기반 백업 소프트웨어의 분석

현재 대부분의 백업 시스템은 클라우드 기반의 네트워크 스토리지를 이용한 백업 시스템으로 제안되고 있다. 미국 *Maryland* 대학에서 제안한 AMANDA[3] 공개 백업 소프트웨어는 중앙집중 제어를 담당하는 하나의 마스터 서버를 통해 여러 클라이언트 서버에 해당하는 노드들로부터 전송되는 데이터를 대용량 테이프장치 또는 테이프 유사장비에 백업한다. AMANDA 시스템은 전형적인 범용 LAN기반의 중앙집중식 백업 시스템이며, 백업 데이터를 효율적으로 관리하기 위하여 데이터를 보내기 전에 압축하는 방법을 선택할 수 있다. *Symantec*사에서는 SAN 기반의 백업 시스템으로 *Netbackup*[4] 시스템을 제안하였다. 범용 네트워크로 연결된 수천 개의 클라이언트로부터 전송되는 백업 데이터를 관리하는 중앙 집중식 서버를 두며 백업 스토리지는 SAN 기반 네트워크를 통해 공유가 가능하도록 구축하여 뛰어난 성능을 제공한다. 스냅샷과 서버-프리 기능을 이용하여 디스크에서 테이프를 직접 데이터를 전송할 수 있는 특징을 갖고 있다. EMC에서 인수한 *Legato*의 *NetWorker* 백업 시스템[5]은 DAS, NAS, SAN 환경을 모두 지원하며 범용 네트워크를 사용하지 않고 SAN의 광채널을 경유하여 저장장치에서 백업 미디어로 직접 백업 데이터를 전송하는 *server-less* 및 *LAN-free* 백업 특징을 지원한다. *Rsync*[6]는 네트워크를 통해 파일 시스템 일부의 스냅샷을 백업 미디어로 전송하여 동기화하는 도구이다. 파일에 변화가 발생하면 체크섬(checksum)값을 이용하여 데이터 변화발생 부분만을 식별하고 전송하는 특정한 프로토콜을 이용하여 백업 데이터의 공간을 절약한다. *Rsync*를

기반으로 개발된 *Rsnapshot*[7]은 여러개의 백업 시점에 대한 스냅샷을 백업 디스크에 각 디렉토리별로 저장한다. 변화가 없는 파일들에 대해서는 디렉토리 내에 하드링크 형태로 저장되므로 공간 효율적인 스냅샷을 생성할 수 있다.

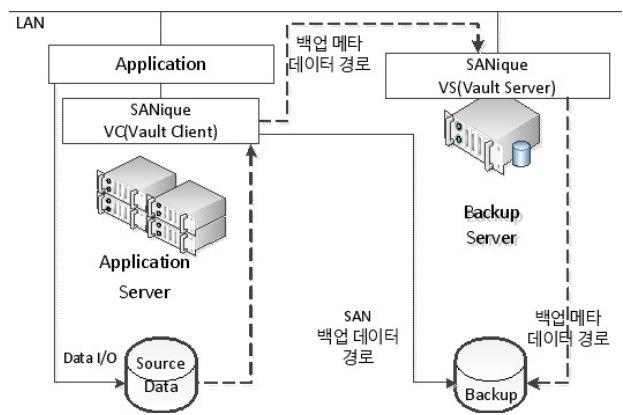
원본 서버의 파일 시스템을 미러링하는 스냅샷 기법에서 또 하나의 새로운 기술은 스냅샷의 내용을 내용주소화(contents addressable) 스토리지 시스템에 저장하는 기법이다. *Venti*[8] 시스템은 데이터 블록을 접근하는 주소기법으로 주소나 ID를 사용하는 대신 데이터 블록의 내용에 대한 해시값을 이용하여 주소로 활용한다. 파일을 고정된 크기의 블록으로 분할하고, 각 블록에 대해 해시값을 생성하여 백업 서버로 전송한다. 데이터 블록에 대한 해시값이 중복되면 동일 스냅샷으로 처리하여 디스크 내에서 합병되어 단일 사본의 스냅샷으로 유지된다. *Venti* 시스템과 유사한 기능을 가지며 성능 개선된 *Data Domain*[9] 시스템은 내용정의 기반(content-defined)의 청크(chunk)를 사용하여 데이터 블록 사이즈보다 작은 크기의 변화에서도 데이터 중복제거를 시행할 수 있다. *PRUN*[15] 시스템은 모듈러-k 알고리즘을 이용하여 파일청킹을 만드는 데이터 중복 제거기능을 제안하였다. 구글 파일 시스템과 맵/리듀스 알고리즘을 이용한 *DiskReduce*[16]은 하둡(Hadoop) 분산 파일 시스템을 개선한 데이터 집적 응용을 위한 백업 시스템을 제안하였다.

기존 백업 소프트웨어의 기능을 분석하면 다음과 같이 요약할 수 있다. 대용량 데이터 처리를 위한 SAN기반 스토리지 시스템 및 파일 시스템을 지원해야 하며, 대규모 데이터에 대한 증분백업의 오버헤드를 줄이기 위해 백업 데이터 대상을 효율적으로 찾을 수 있는 델타 증분백업기술 및 무기한 증분(forever incremental) 백업을 지원해야 한다. 또한 데이터 중복 제거 기능을 통해 데이터 집적 응용 시스템의 중복 백업 데이터를 제거하여 백업 성능을 향상시킬 수 있어야 한다.

3. SANique Smart Vault 시스템

3.1 시스템 구조 및 개요

이 절에서는 저널링 기반의 델타 및 무기한 증분백업 기술을 갖는 *SANique Smart Vault* 백업 시스템의 구조에 대하여 기술한다. 제안하는 백업 시스템은 그림 1과 같이 SAN 기반 클러스터 서버 환경에서 여러 노드의 서버에 구축되는 *Vault Client(VC)*와 *Vault Server(VS)*로 구성된다. *Smart Vault* 백업 시스템은 저널링(journaling)에 기반한 델타 증분백업 방식을 지원하며 파일 단위 및 레코드 단위로 구분하여 수행할 수 있도록 한다. 응용 프로그램에서 원본 데이터에 대한 I/O 요청을 보내면 *VC* 시스템은 원본 요청을 가로채 데이터 변경내용에 대한 물리적 상세내용을 추출하여 변경 레코드에 대한 저널링 기록을 수행한다. 그 후 백업 서버인 *VS*로 통합관리를 위해 백업 메타데이터를 전송하고 백업 데이터는 SAN을 이용하여 백업 미디어에 직접 전송한다.



<그림 1> SANique Smart Vault 백업 시스템 구조도

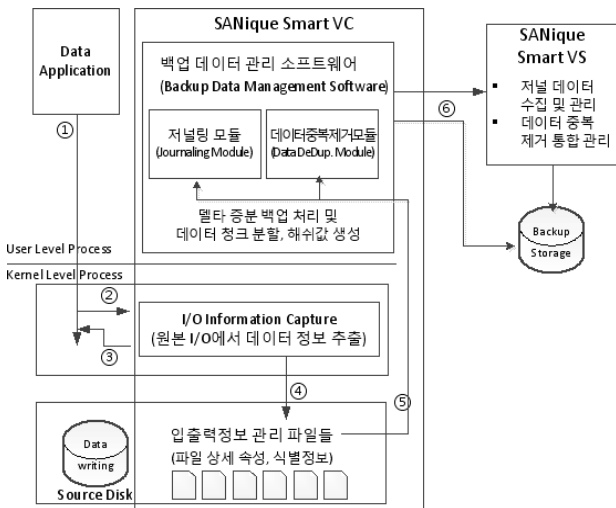
Smart Vault 백업 시스템의 대용량 데이터 응용 시스템을 지원하기 위한 시스템 구조 및 백업 기능상의 특징은 다음 표 1과 같이 요약할 수 있다.

<표 1> *SANique Smart Vault* 시스템의 기능적 특징

구분	특징
백업 시스템 구조	<ul style="list-style-type: none"> SAN 기반 클러스터 환경 지원 DAS/NAS/SAN 구조 지원 Serverless 백업 LAN-Free 백업
백업 방법	<ul style="list-style-type: none"> 델타(delta) 증분 백업 무기한(forever) 증분 백업 파일 및 레코드 기반 백업 저널링 기반 델타 백업
데이터 중복제거	<ul style="list-style-type: none"> 고성능 host기반 중복제거 동일 데이터 파일내 중복 제거 동일 서버 및 다중 서버내 중복제거 기능 청크 해싱 함수 선택 적용

3.2 Smart Vault VC 및 VS의 구성

Smart Vault VC 시스템은 백업 성능을 효율적으로 향상시키기 위하여 커널 레벨의 데몬 모듈들과 사용자 레벨의 백업 데이터 관리 소프트웨어 모듈 그리고 관리 정보를 저장 및 유지하기 위한 입출력 정보 관리 파일로 구성된다. 백업 프로세스들에서 활용되는 입출력 정보 관리 파일은 디렉토리 정보 파일과 백업 대상 정보 파일로 구성된다. 기본적인 백업의 수행과정은 그림 2와 같다. 응용 프로그램(①)은 디스크에 입출력을 요청하고, ②에서 I/O Information Capture 데몬 프로그램에 의해 원본 I/O를 가로채 필요한 물리적 입출력 정보를 추출한다. 원본 I/O요청은 그대로 수행되도록 디스크(③)로 보낸다. I/O Information Capture 데몬은 물리적 입출력 정보와 함께 델타 증분백업을 위해 파일의 상세 속성을 관리 정보 파일에 체계적으로 저장 및 관리한다(④). 백업 데이터 관리 소프트웨어는 ⑤에서 관리되고 있는 파일을 기반으로 해당 파일에 대한 백업 데이터의 저널링 기록 및 데이터 중복 제거를 위한 데이터 체크 분할, 해시값 생성 등의 작업을 수행하여 SANique Smart VS 및 백업 스토리지로 전송하는 작업을 수행하게 된다.



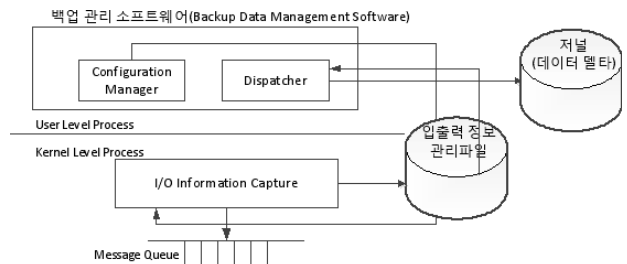
<그림 2> Smart VC/VS 구조 및 백업 데이터 전송 흐름도

3.3 백업 데이터 관리 소프트웨어

커널 수준 모듈 I/O Information Capture 데몬과 직접적으로 상호 동작하는 백업 데이터 관리 소프트웨어는 Configuration Manager와 Dispatcher 모듈로 구성

되며 사용자 수준의 프로세스이다. 두 세부 모듈의 목적은 그림 3과 같이 입출력 정보관리 파일들의 내용을 기반으로 이전 백업 이후 변경된 파일들에 대한 파일의 이름, 오프셋 등 속성을 추출하여 변경 파일 단위의 데이터 델타 분에 대하여 저널링을 수행하는 것이 목적이다. 델타 증분백업 수행을 위한 저널링 데이터는 SANique VS로 전송된다. 데이터 중복 제거 기능을 위한 데이터 체크 분할 및 해시값 생성 등의 작업 또한 백업 데이터 관리 소프트웨어에서 수행된다.

추출된 파일 정보의 일부는 메시지 큐에 저장되어 관리 파일에 전송되고, 백업 데이터 관리 소프트웨어의 Dispatcher가 이 정보를 이용하여 백업 대상 파일들을 저널 레코드로 생성하고 별도의 파일로 저장한다. 커널 데몬과 Configuration Manager 모듈 사이의 통신은 디렉토리 정보 파일을 통하여 수행한다. 사용자가 백업대상으로 정하는 디렉토리 관련 정보는 Configuration Manager를 통하여 디렉토리 정보를 저장된다. 커널 데몬은 디렉토리 정보 파일에 저장된 정보를 이용하여 백업 대상에 속하는 관련 정보만 추출해낸다.



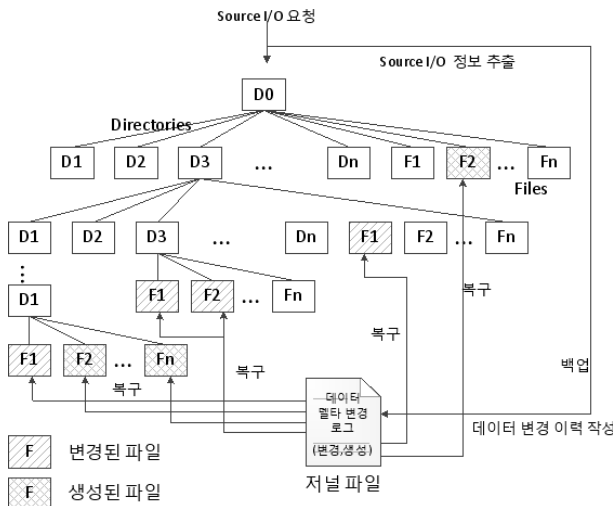
<그림 3> 백업 데이터 관리 소프트웨어 세부 모듈

4. 저널링 기반 델타 증분백업 및 중복제거 기술의 구현

4.1 저널링 기반 델타 증분백업의 구현

Smart Vault 백업 시스템에 구현한 주요 기술 중 하나는 저널링 기반의 델타 증분백업기술이다. 델타 증분백업은 이전 백업 메타 데이터를 기준으로 변경된 부분에 대한 마크업 데이터를 생성하여 효율적으로 변경 데이터를 백업할 수 있도록 하는 기법이다. 백업 대상 파일을 찾기 위하여 파일 시스템 전체를 탐색할 필요가 없어 성능을 향상시킬 수 있다. Smart Vault 시스템은 그림 4와 같이 변

경된 데이터 델타 부분에 대한 최소한의 물리적 속성 메타 데이터를 추출하여 저널 로그파일을 생성한다. 저널 파일은 파일단위 백업과 레코드단위 백업에 따라 저널 레코드를 구분하여 작성한다. 백업복구 시에 데이터 변경 대상 검색을 위해 파일 시스템을 검색할 필요 없이 오직 저널 파일의 로그만 분석하여 복구가 가능하다.



<그림 4> 저널링 기반 데이터 변경 로그 작성

대부분의 상용 백업 시스템들은 증분백업을 수행할 때 데이터 변경 부분을 찾는 오버헤드가 매우 크다. 특히 대용량 데이터 집중적 응용 시스템을 사용하는 서비스에서는 데이터 변경부분을 찾기 위한 오버헤드가 커서 순조로운 서비스가 어렵다. 본 논문에서 구현한 백업 시스템은 앞 절에서 기술한 I/O Information Capture 때문에 의해 원본 디스크 I/O 요청을 가로채 데이터 접근에 대한 상세 정보를 직접 추출하여 데이터 델타 변경 부분의 저널 로그를 작성하므로 백업 대상 검색에 대한 오버헤드가 없다.

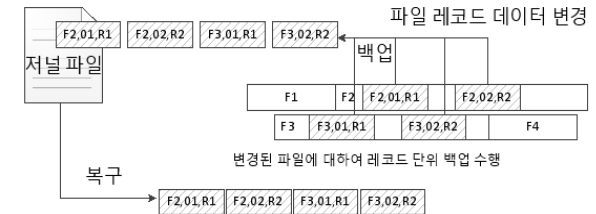
Smart Vault 백업 시스템의 델타 증분백업은 파일 단위 백업과 레코드 단위 백업을 지원한다. 파일 단위 백업은 대규모 사용자가 이용하는 SNS 서비스와 같이 파일 단위의 생성 및 삭제가 빈번한 응용에 적합한 백업 단위이며 레코드 단위 백업은 그룹웨어 서버와 같이 부분적 파일 변경이 빈번한 사무환경에서 우수한 성능을 보인다. 본 백업 시스템에서는 두 가지 백업 단위를 모두 지원하며 저널 파일에서 백업의 단위 기준에 따라 그림 5와 같이 원본 I/O요청 정보로부터 데이터

델타 변경부분에 대한 물리적 정보를 가로채 파일 저널과 레코드 저널로 구분하여 로그 레코드를 기록한다. 백업복구 시에는 파일 시스템의 추가적 탐색 없이 저널 파일 정보만 이용하여 파일 또는 레코드 단위의 복구를 수행한다.

1) 델타파일 백업



2) 델타 레코드 백업



<그림 5> 델타 파일/레코드 백업에 따른 저널 파일 구성 방법

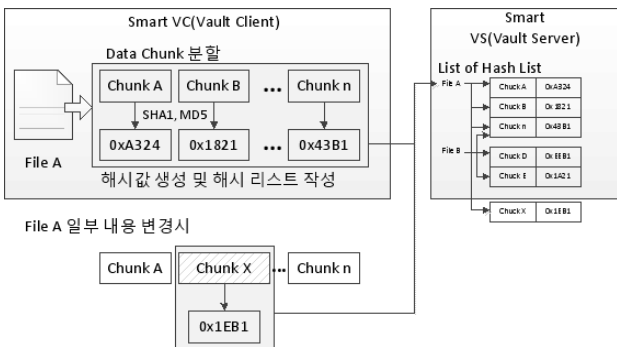
본 백업 시스템의 저널링 기반 델타 증분백업은 백업복구시 필수 소요시간인 백업 대상 탐색 시간을 줄여 단순 데이터 복제 수준의 성능을 보일 수 있으며 또한 변경 부분만을 백업하는 레코드 레벨 백업을 지원하여 백업 데이터양을 대폭 줄이게 되므로 시공간적 성능향상을 제공한다.

4.2 데이터 중복제거 기술의 구현

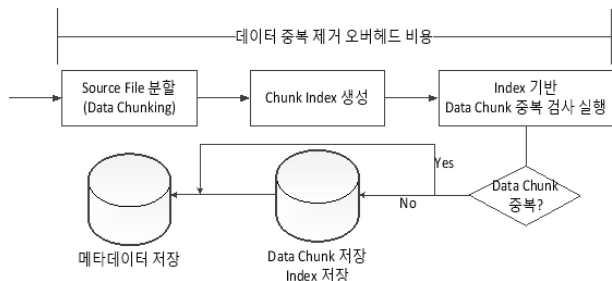
대규모 사용자가 사용하는 대용량 스토리지 시스템은 사용자별 중복 파일을 저장하는 경우가 많으며 이런 경우 대규모 중복 파일을 관리함으로써 백업의 성능을 향상시킬 수 있다[10][11]. 고전적인 데이터 중복제거 기술은 DeDe[12] 시스템과 같이 클러스터 파일 시스템에서 중복된 파일을 저장하지 않기 위해 제안된 기술이다. 이러한 중복제거 기술을 클러스터 기반 백업시스템에 활용하여 성능을 향상할 수 있다. 데이터 중복 여부를 판단하는 가장 일반적인 방법은 파일을 일정한 크기의 데이터 청크(chunk)로 분할하여 데이터 중복 여부를 검사하는 방법을 사용한다. 분할 방법에 따라 고정크기 분할과 가변 크기 분할로 나눌 수 있다. 고정 크기 분할 방법은 데이터 청

크가 모두 변경되므로 변경 이후 부분에 대한 데이터 중복여부를 감지할 수 없게 된다. 가변크기 분할방식은 데이터 변경이 발생하는 경우 데이터 청크의 크기를 증가시키거나, 분할, 삭제하여 데이터 청크 변화 범위를 최소화함으로써 데이터 중복을 가장 많이 찾을 수 있도록 한다.

본 논문에서 구현한 데이터 중복 제거기술은 클라이언트 VC의 파일 분할 및 해시리스트 작성 모듈과 서버 VS의 전역적 해시 리스트 통합관리 모듈로 구성되어 구현된다. 파일 및 레코드 수준의 중복제거를 하기 위해 VC 모듈은 파일을 내용기반 가변 크기의 청크로 분할한다. 분할된 각 청크에 대해 VC 모듈은 그림 6과 같이 해시값을 생성하고 이에 대한 해시 리스트를 구축한다. 구현된 해시 함수는 SHA1[13]와 MD5[14]를 선택적으로 이용할 수 있도록 설정할 수 있다. 각 VC 클라이언트에서 생성된 해시 리스트는 VS 서버로 전송된 후 데이터 중복 제거를 위한 통합 해시 리스트로 구축된다.



<그림 6> 데이터 중복제거를 위한 데이터 청크 분할 및 해시리스트 작성



<그림 7> 데이터 중복제거 기법의 개략적 흐름도

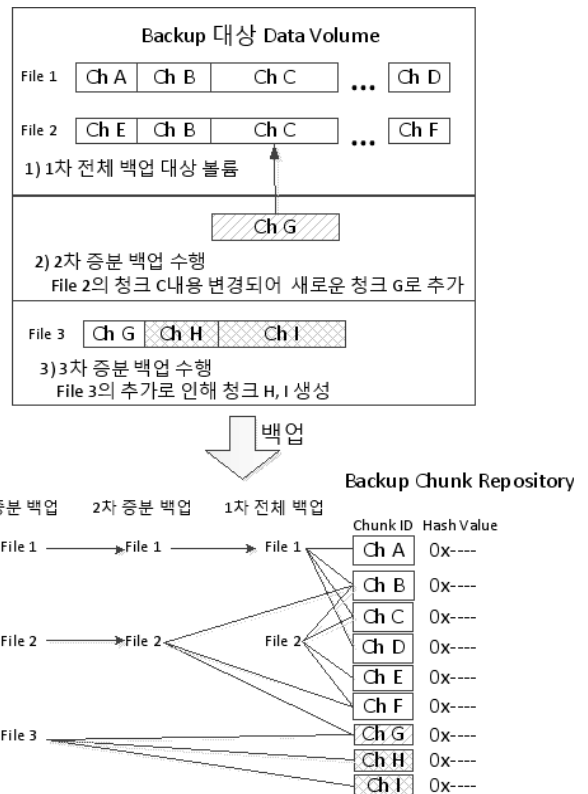
데이터 변경이 발생하는 경우 변경 부분에 대한 가변 청크로 변경하여 기존 청크에 대한 해시값이 변하지 않도록 최대한 유지한다. 그림 6에서 보인 바와 같이 파일 A의 데이터 청크 B의 내용이 변

경되는 경우 새롭게 생성된 가변청크 X에 대한 해시값을 생성하여 해시리스트를 변경하고 변경내용을 VS 서버로 전송하여 통합관리 리스트의 내용을 갱신한다. VS 서버는 다른 VC 클라이언트들의 해시리스트를 통합관리하여 추가 변경되는 데이터 청크들의 데이터 중복여부를 탐지한다.

데이터 중복제거를 위한 청크 분할 방법 및 데이터 중복 여부를 검사하는 개략적인 과정은 그림 7과 같다.

4.3 델타 무기한 증분백업 데이터의 생성 및 관리 예제

이 절에서는 데이터 중복 제거기능을 갖는 델타 무기한 증분백업 방식에 따른 백업 데이터의 생성 및 저장관리 예제를 보인다. 그림 8에 주어진 것과 같은 백업 대상 데이터 볼륨에 대하여 1차 전체 백업을 진행하고 데이터 변화에 따른 2차, 3차 증분백업을 수행할 때의 예를 설명한다.



<그림 8> 데이터 중복과정을 갖는 델타 무기한 증분백업 과정의 예제

그림 8의 예에서 백업 대상 볼륨은 두 개의 파일로 구성되며 데이터 청크 B, C를 공유한다. 1차 전체 백업에서 데이터 중복 제거작업을 선행하여

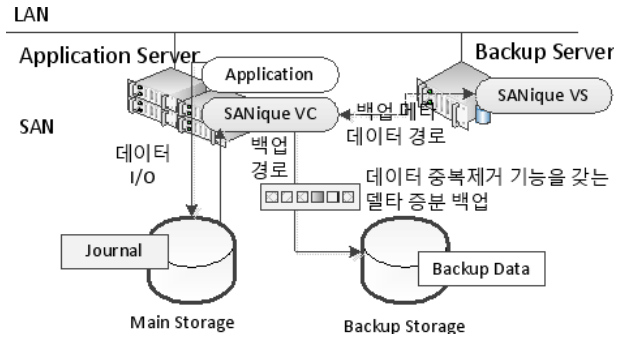
수행한다. 중복 데이터 체크는 하나의 복사본만 백업하고 파일 목록에서 링크로 공유할 수 있도록 한다. 2차 증분백업에서는 파일 2의 데이터 체크 C의 내용이 변경된 후이므로, 새로운 데이터 체크 G를 생성하여 백업하고 백업 체크 스토리지에 추가한다. 추가된 체크의 링크는 파일2의 체크목록에 포함시키고 기존 데이터 체크C는 언링크된다. 파일 1은 변화가 없으므로 그대로 링크를 유지하여 복구시 기존 백업 체크를 사용할 수 있도록 한다. 3차 증분백업에서는 새롭게 생성된 파일 3에 대해 데이터 체크를 생성하고 백업한다. 기존에 중복된 데이터 체크 G는 백업에서 제외하고 파일 3의 데이터 체크 목록에서 공유할 수 있도록 한다. 파일 1, 2는 변화가 없으므로 기존에 파일 링크를 그대로 유지하여 백업 데이터를 최소화한다.

5. Smart Vault 백업시스템의 성능 분석

Smart Vault 백업 시스템의 기능에 따라 타사 솔루션 및 일반 백업 방법과의 용량과 시간적 성능에 대해 비교한다. 성능평가를 위한 SAN기반 백업 서버의 시스템 구조는 그림 9와 같이 구성한다. 각 서버는 x86계열 듀얼코어 CPU와 8GB RAM, 1TB STA HDD 두 개로 구성된다.

5.1 저널링 기반 레코드 레벨 델타 증분백업

저널링 기반 레코드 레벨 델타 증분백업 기능에 대해 타사 솔루션들이 일반적으로 제공하는

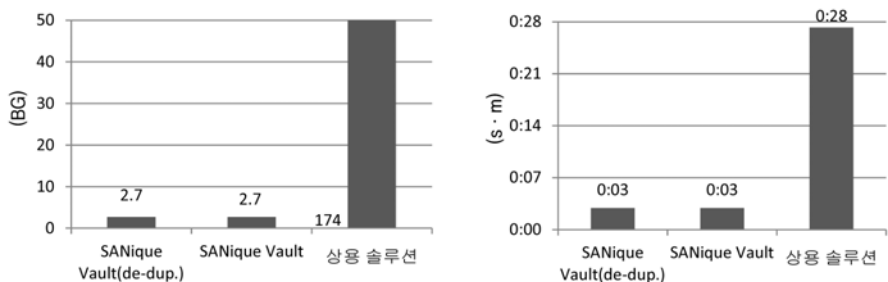


<그림 9> 성능평가 시스템 구조

기본적 증분백업 기능의 성능과 비교한다. 실제 백업 데이터의 크기 및 시간을 가늠하기 위해서 tar 도구와도 비교한다. 성능평가 시나리오는 초기 전체 백업과 2차 증분백업을 수행한 후 백업 데이터 용량과 시간 측정을 하는 것이다. 초기 전체 백업 데이터 볼륨은 약 700GB 데이터 크기이며 약 2백만 개의 파일로 구성된다. 초기 전체 백업을 진행한 후 약 1,000개의 파일 변화가 진행된 후 1차 증분백업을 수행하고 약 2,600개의 파일변화가 발생한 후 2차 증분백업을 수행하였다. 같은 환경에서 동시에 수행하여 일반 증분백업 방식을 갖는 상용 솔루션과 tar방식 그리고 SANique Vault 백업 시스템의 백업시간과 데이터 크기를 측정한 결과는 그림 10과 같다.

일반적으로 사용되는 상용 솔루션의 증분백업은 백업 대상을 탐색하는 소요시간이 많이 소요되고 파일 수준 또는 블록 수준의 백업을 지원하므로 그룹웨어나 메일 서버와 같이 데이터 변경이 빈번한 경우 데이터 백업 크기가 커져 성능상 오버헤드를 갖는다.

백업 방식	백업 용량	변화량	Tar 도구		상용 솔루션		SANique Vault	
			용량	시간	용량	시간	용량	시간
1차 증분백업	약 300MB	약 1,000개 파일	650MB	14m37s	298MB	6m44s	9MB	49s
2차 증분백업	약 400MB	약 2,600개 파일	758MB	22m16s	415MB	7m01s	80MB	48s



<그림 10> 레코드 레벨 델타 증분백업 성능 비교

반면 제안하는 백업 시스템은 저널링에 기반한 델타 증분백업을 지원하여 백업 탐색시간을 줄이고 레코드 레벨의 백업을 지원하여 백업 데이터의 크기도 줄일 수 있다.

5.2 데이터 중복제거 기능을 갖는 백업 시스템의 성능분석

이 절에서는 데이터 중복제거기능을 가는 백업 시스템의 성능 오버헤드를 분석한다. 성능시험을 위한 시스템 구조는 동일하게 사용한다. 백업 대상의 데이터 볼륨은 약 450GB이며, 약 220만개의 파일, 평균파일 크기는 약 300KB로 구성한다. 성능평가 시나리오는 다음과 같이 구성한다. 초기 전체백업 수행전에 데이터 중복제거를 수행하고, 초기 전체백업 후에 약 4,700건의 파일변경 동안 4차 증분백업을 수행하고 백업 데이터의 크기 및 시간을 측정한다. 이 때 발생하는 데이터 변화는 파일 신규생성 약 400건, 삭제 약 300건, 파일 속성 변경 약 500건, 파일 내용변경 약 3,500건으로 구성하여 실험한다. 데이터 중복 제거에 대한 오버헤드를 평가하기 위해 중복제거 기능을 갖지 않는 버전과의 측정값을 비교하고 또한 일반적 증분백업 기능을 갖는 상용 시스템과도 비교한다. 동일조건의 성능평가 테스트베드를 2세트씩 구성한 후 동시에 증분백업을 수행하여 백업 성능시간을 측정한 결과는 그림 11과 같다.

대용량 데이터 서비스의 경우 대규모 사용자가 중복 데이터를 보유하는 경향이 많아 파일 기반 중복제거를 수행 후 백업 데이터를 탐색하면 성능측

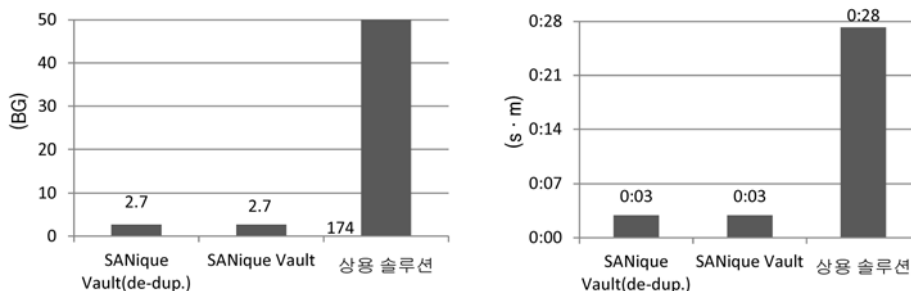
면에서 큰 이익을 얻을 수 있다. 따라서 중복제거 기능이 있는 백업시스템과 일반 상용 시스템의 성능은 큰 차이를 보임이 당연하다.

앞 장에서 언급한 바와 같이 데이터 중복제거 기능은 초기 데이터 분할 및 해시 인덱스 값 생성에 대한 오버헤드가 크다. 본 성능평가에서는 초기 오버헤드를 제외하고 델타 증분백업 방식을 수행할 때 데이터 변경 부분의 데이터 중복제거에 대한 오버헤드를 평가하였다. 초기 오버헤드를 제외하면 중복제거기능을 갖지 않는 시스템과 유사한 성능을 항상 보였다. 이 실험으로 델타 증분백업에 대한 데이터 중복 제거기능의 오버헤드는 백업 시스템의 성능에 영향을 주지 않음을 알 수 있다.

6. 결론

SANique Smart Vault 백업 시스템은 전통적인 백업 기능은 물론 저널링 기반의 레코드 레벨 델타 증분백업을 지원하여 기존 상용화 제품에서 지원하는 파일 기반 백업 시스템 보다 한층 우수한 성능을 기대할 수 있다. 또한 데이터 중복 제거 기법에서도 레코드 레벨 증분백업시 활용하는 데이터 추출 정보를 활용하여 변경 데이터 부분에 대해서만 중복 검사를 실행하는 새로운 기법을 상용화하여 시간적 및 공간적 성능 향상을 기대할 수 있다. 데이터 중복제거 기능의 성능오버헤드 대비 효율성을 증대하기 위하여 초기 데이터 분할 작업성능 개선에 대한 연구를 진행 중이며 다양한

백업 방식	백업 대상 데이터	변화량	SANique Vault (데이터 중복제거)		SANique Vault		상용 솔루션	
			용량	시간	용량	시간	용량	시간
4차 증분백업	약 450GB	약 4,700개 파일	2.7GB	3min.	2.7GB	3min.	174GB	28min.



<그림 11> 데이터 중복제거 기능의 성능평가

상용 백업 시스템과의 성능평가도 계획 중이다.

참고문헌

[1] Schmuck, F. B., & Haskin, R. L. (2002, January). GPFS: A Shared-Disk File System for Large Computing Clusters. In FAST (Vol. 2, p. 19).

[2] Kaczmarek, Michael, Tricia Jiang, and David A. Pease. (2003). "Beyond backup toward storage management." IBM Systems Journal 42.2 322-337.

[3] The Advanced Maryland Automatic Network Disk Archiver. <http://www.amanda.org/>.

[4] Guofu, S. T. Q. (2006). A Study on the Application of Veritas NetBackup Software in Library [J]. New Technology of Library and Information Service, 6.

[5] Back, E. C. B. S. S. (2003). up-to-Disk Guide with LEGATO Networker Diskbackup Option (DBO). Engineering White Paper, 1-28.

[6] Tridgell, A. (1999). Efficient algorithms for sorting and synchronization. Canberra: Australian National University.

[7] Rubel, M. Rsnapshot: A Remote Filesystem Snapshot Utility Based on Rsync. 2005. URL <http://rsnapshot.org>.

[8] Quinlan, S., & Dorward, S. (2002, January). Venti: A New Approach to Archival Storage. In FAST (Vol. 2, pp. 89-101).

[9] Zhu, B., Li, K., & Patterson, R. H. (2008, February). Avoiding the Disk Bottleneck in the Data Domain Deduplication File System. In Fast (Vol. 8, pp. 269-282).

[10] Hong, B., Plantenberg, D., Long, D. D., & Sivan-Zimet, M. (2004, April). Duplicate Data Elimination in a SAN File System. In MSST (pp. 301-314).

[11] Bobbarjung, D. R., Jagannathan, S., & Dubnicki, C. (2006). Improving duplicate elimination in storage systems. ACM Transactions on Storage (TOS), 2(4), 424-448.

[12] Clements, A. T., Ahmad, I., Vilayannur, M., & Li, J. (2009, June). Decentralized deduplication in SAN cluster file systems. In Proceedings of the 2009 conference on USENIX Annual technical conference (pp. 8-8). USENIX Association

[13] Eastlake, D., & Jones, P. (2001). US secure hash algorithm 1 (SHA1).

[14] den Boer, B., & Bosselaers, A. (1994, January). Collisions for the compression function of MD5. In Advances in Cryptology—EUROCRYPT'93 (pp. 293-304). Springer Berlin Heidelberg.

[15] Won, Y., Kim, R., Ban, J., Hur, J., Oh, S., & Lee, J. (2008, June). Prun: Eliminating information redundancy for large scale data backup system. In Computational Sciences and Its Applications, 2008. ICCSA'08. International Conference on (pp. 139-144). IEEE.

[16] Fan, B., Tantisiriroj, W., Xiao, L., & Gibson, G. (2009, November). DiskReduce: RAID for data-intensive scalable computing. In Proceedings of the 4th Annual Workshop on Petascale Data Storage (pp. 6-10). ACM



이 규 응

1990 한국외국어대학교
컴퓨터학과(이학사)
1992 서강대학교
컴퓨터학과(공학석사)
1998 서강대학교
컴퓨터학과(공학박사)

1998-2000 한국전자통신연구원 선임연구원
2000-현재 상지대학교 컴퓨터정보공학부 교수
관심분야: 데이터베이스 시스템, 클러스터 시스템
E-Mail: leekw@sangji.ac.kr