

클러스터링 기반 백업 시스템 설계 및 구현

천영우*, 김진, 고영웅
한림대학교 정보통신공학부
e-mail: {zzzwoo, jinkim, yuko}@hallym.ac.kr

Design and Implementation of Clustering Backup System

Young-Woo Chun*, Jin Kim, Young-Woong Ko
Division of Information Engineering & Telecommunications,
Hallym University

요 약

여러 대의 컴퓨터를 하나의 네트워크로 연결한 클러스터 시스템은 일반적으로 웹 부하 분배, VOD 스트리밍 서비스 그리고 슈퍼컴퓨터, 혹은 메인프레임 급의 성능이 필요한 거대한 작업을 위해서 사용된다. 본 논문에서는 클러스터 기법을 사용하여 안정적이고 높은 성능을 보장하는 백업 시스템을 설계 및 구현하는 내용에 대해서 기술하고 있다. 기존의 데이터 백업 시스템과 비교하여 저가격에 높은 효율성을 제공할 수 있는 장점을 제공하며, 안정성 및 신뢰성을 높이 향상시킬 수 있음을 보이고 있다.

1. 서론

컴퓨터 기술이 발전하고 일상생활과 업무에서 그 사용도가 높아짐에 따라 만들어지고 저장되어지는 데이터의 양이 늘어가게 되었다. 하지만 시스템의 소프트웨어적이나 물리적인 오류로 인해서 한순간에 소중한 데이터가 손실되는 일은 우리 주변에서 어렵지 않게 볼 수 있다. 고가의 백업솔루션이 구축되어 있는 대규모의 사업장, 중요 정부기관이 아닌 중소기업의 사업장이나 가정환경에서 PC를 사용하는 일반 사용자의 경우 데이터 백업에 대해서 취약한 면을 많이 안고 있는 것이 현실이다. 광학매체(CD, DVD 등)를 사용해 필요할 때마다 데이터를 저장하거나 웹 디스크 등의 서비스를 이용해 데이터를 백업하는 방법, 혹은 하드웨어적인 RAID를 사용하는

방법이 있지만, 각 방법들은 기술적, 경제적으로 여러 가지 제한점을 가진다. 물리적인 매체를 통한 비주기적인 데이터 백업은 보관에 대한 애로사항이 있으며 백업한 데이터를 다시 가져오하고자 할 때 번거로운 작업이 발생된다. 웹 디스크를 이용한 백업은 일반적으로 수백 메가바이트, 혹은 수 기가 바이트 정도의 용량적인 한계를 가지고 있으며, RAID 환경의 시스템은 안정적인 측면에서의 장점은 있으나 데이터의 백업과 복원에 대해서 지역적인 한계 및 고비용의 가지고 있다. 본 연구에서 보이고자 하는 클러스터 서버를 이용한 백업 솔루션에서는 위에서 대두되었던 여러 가지 문제를 해결하는 방안을 제시하고 있다. 웹 기반의 백업 환경을 제공하여 지역적인 한계를 극복하게 되며 클러스터 환경의 분할 저장방식을 이용해 RAID 시스템의 안정성을 보장하게 된다. 또한 간단한 소프트웨어 설정을 이용해 백업이 필요할 때마다 일일이 작업해야 했던 번거로운 부분도 해결하는 방안을 제시하고 있다. 사용자가 백업하고자 하는 데이터를 원하는 시간과 주기에 맞춰

This research was supported by the Program for the Training of Graduate Students in Regional Innovation which was conducted by the Ministry of Commerce Industry and Energy of the Korean Government. This work was supported by the Industry University Research Institute Consortium grant from the Small & Medium Business Administration.

시스템이 자동으로 백업 해주기 때문에 사용자는 초기설정만 해주게 되면 데이터 백업에 대해서는 신경 쓰지 않아도 된다.

본 논문은 다음과 같이 구성된다. 2 장에서는 본 연구와 관련된 연구 내용을 기술하고, 3장에서는 클러스터 기반 백업 시스템의 설계 및 구현에 대한 사항을 설명한다. 4장에서는 구현된 시스템의 성능 평가를 기술하고 5장에서 결론을 맺는다.

2. 관련 연구

FAT[1]는 File Allocation Table이라 하여 MS-DOS에서 사용되었던 파일 시스템 기법이다. 디스크의 여러 정보를 축소한 테이블 형태로 슈퍼블록에 저장함으로써 파일 시스템을 관리하게 된다. 본 연구에서 제안하는 클러스터 기법은 여러 대의 시스템에 데이터를 분할해서 저장하기 때문에 각 클러스터 서버 노드의 내부 상태를 메타데이터 형태로 관리해야 할 필요가 있다. 각 클러스터 서버 노드의 저장구조를 통일시켜서 하나의 클러스터 서버 노드의 FAT정보를 가지고 모든 노드를 관리하도록 하였다.

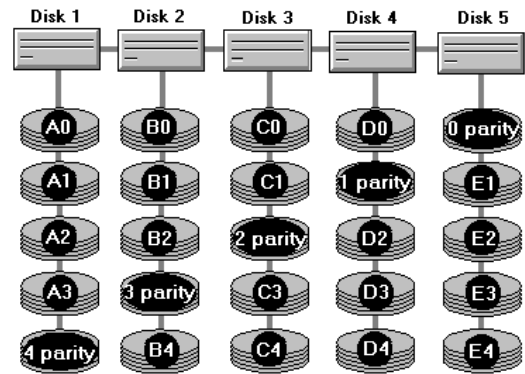
클러스터 기법은 여러 대의 시스템을 하나의 네트워크로 연결하여 커다란 단일 시스템처럼 사용하는 것을 의미하며 큰 규모의 웹 사이트에서 클라이언트 접속에 의해 발생하는 부하를 분배해주는 로드 밸런서로 사용되거나 처리량이 많은 대규모 작업을 위한 고성능 시스템의 대안으로 사용되기도 하며 VOD등의 스트리밍 서비스에서 빠른 처리속도와 네트워크 대역폭을 폭넓게 사용하기 위해서 일반적으로 사용이 되어졌었다. 본 연구에서는 백업하고자 하는 큰 데이터를 블록 단위로 분할하여 클러스터 환경에서 각 클러스터 서버 노드로 분할하여 백업함으로써 빠른 속도와 안정성을 보장받도록 하였다.

Lustre[2]에서는 클러스터 환경에서 데이터의 효과적인 관리를 위해 MDS(Meta Data Server)를 이용한다. 각 백업 데이터 단위에 대응하는 메타데이터를 두고 관리함으로써 클라이언트, 혹은 클러스터 서버노드를 관리하는 마스터 서버에서의 데이터 관리를 효과적으로 할 수 있게 한다. 본 연구에서도 각각의 백업 단위에 대한 메타데이터를 마스터서버에서 관리함으로써 편의성 및 성능적인 측면에서의 향상을 높이도록 하였다.

3. 클러스터 백업 시스템 설계 및 구현

기존의 백업 서버 기법에 있어서 하나의 시스템에 여러 대의 디스크를 연결해 Mirroring이나 확장된 여러 가지 하드웨어적인 RAID 환경을 구축하는 방식은 오래전부터 사용되어 왔었다. 하지만 하드웨어

기반의 RAID 백업 솔루션은 높은 비용을 요구하기 때문에 범용으로 사용되기에는 무리가 있다. 본 연구에서는 저사양의 PC를 이용해서 낮은 비용을 가지고 RAID 백업솔루션의 성능을 소프트웨어적인 방법으로 극복하는 방안을 제시한다. 각 클러스터 서버 노드는 네트워크 스위치를 통해 하나의 네트워크로 연결되어 마스터서버의 관리 하에 하나의 거대한 시스템처럼 사용될 수 있다. 각 노드의 CPU, 메모리 자원을 병렬 프로그래밍을 통해 고성능을 요하는 대규모 연산을 처리할 수도 있으며 병렬적으로 파일 블록을 저장 및 전송할 수 있기 때문에 동영상 스트리밍 서비스 등에서도 높은 성능을 보여준다. 이와 같이 파일 스트라이핑을 이용해 병렬적인 시스템의 특성과 성능을 이용한 백업 솔루션을 설계하였다.

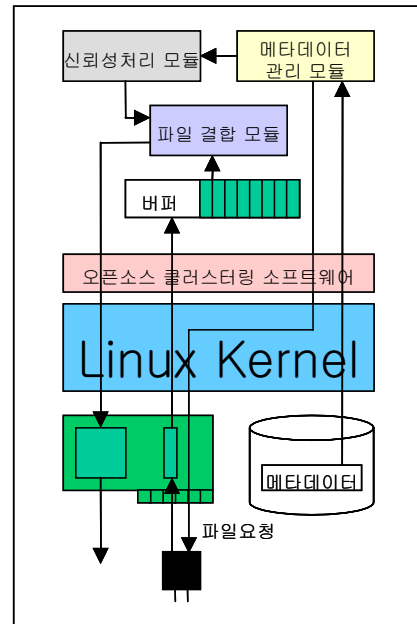


[그림 1] RAID 레벨5의 파일 스트라이핑

Lustre 파일 시스템에서 사용되었던 MDS에서는 메타데이터 서버를 이용해서 백업된 파일에 대한 관리를 해주게 된다. 본 연구에서는 단일 파일에 1회 백업이 아닌 디렉토리 수준의 백업을 지원하며 설정된 주기를 통해 반복적인 백업이 자동으로 이루어지므로 메타데이터 관리 부분에서 더 많은 요구를 필요로 한다. 백업되어지는 파일, 혹은 디렉토리의 이름과 백업 시간, 클러스터 서버 노드의 어느 위치에 스트라이핑 되어서 저장되어있는지에 대한 정보가 필요하며 주기적인 백업 시에 백업 데이터가 겹치는 것을 방지하기 위해 버전 관리가 이루어진다. 이러한 연산은 사용자 클라이언트와 마스터 서버에서 이루어지며 메타데이터는 마스터 서버에 의해서 저장 및 관리가 이루어지고 마스터서버 때문에 실행될 때 연결리스트 형태로 메모리에 적재되기 때문에 백업 및 복구과정에서 빠른 검색과 저장을 가능하게 한다.

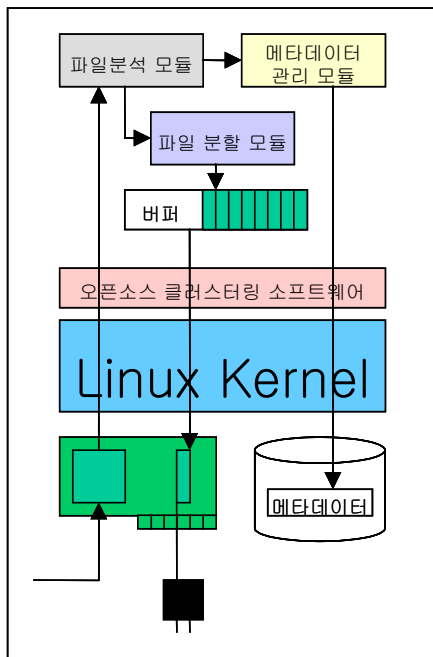
같은 데이터에 대한 주기를 이용한 반복적인 백업이 이루어지다보면 백업 데이터의 중복이 일어나게

된다. 이는 시스템 자원의 낭비이며 불필요한 오버헤드를 발생시키는 것이다. 이에 대해 백업 데이터의 버전 관리를 통한 목록을 메타데이터화 시켜서 필요한 부분에 대해서만 백업 작업을 수행하게 되며 파일 및 디렉토리를 복구 할 때 최근 백업 순서대로 복구를 하여 사용자가 가장 최근에 백업을 했던 데이터를 복구 시켜주게 된다. 이는 LSF(Log Structured File system)[3]에서 로그를 이용한 파일 백업, 복구에 대한 아이디어를 확장시킨 것이다. 반복 백업 작업 시에 사용자 클라이언트 시스템에서는 마스터 서버에 저장되어있는 백업 목록의 버전을 가지고 연산을 통해 어떠한 파일들을 백업해야하는지를 판단하며 필요한 파일들만을 저장함으로 인해서 성능 및 자원적인 측면에서의 효과적인 사용이 가능하게 된다.



[그림 3] 데이터 복구 단계

위의 [그림 3]은 데이터 복구 단계에서의 동작하는 모습을 보이고 있다. 사용자 클라이언트에서 복구를 요청하게 되면 메타데이터를 검색해서 저장되어있는 클러스터 서버 노드의 주소를 가져오게 되며 그 주소를 각 클러스터 서버 노드에 전송해서 스트라이핑된 데이터 블록을 전송받아서 결합 후에 사용자 클라이언트로 재전송 해주는 단계를 거쳐서 마무리 된다.



[그림 2] 데이터 백업 단계

[그림 2]에서는 데이터 백업 단계에서의 동작을 나타내고 있다. NIC를 통해 클라이언트로부터 마스터 서버에 백업 데이터가 들어오게 되면 파일분석 모듈에서 스트라이핑 및 메타데이터에 사용할 정보들을 추출하는 연산을 한 후에 스트라이핑 된 백업 데이터는 버퍼에 512Byte단위의 블록으로 나뉘어 소켓으로 연결되어있는 클러스터 서버 노드에 보내지게 되며 메타데이터는 마스터서버의 디스크에 저장되어지게 된다.

4. 실험 결과

제안하는 클러스터 백업 솔루션의 프로토타입은 1대의 마스터 서버와 3대의 클러스터 서버 노드로 이루어져 있으며 사양은 아래와 같다.

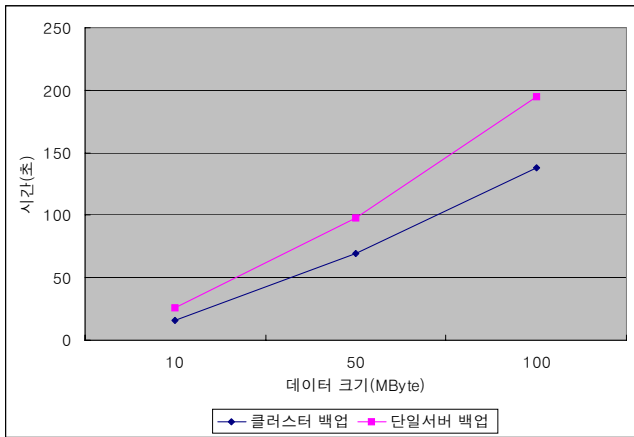
CPU	펜티엄4 3.0GHz
Memory	512 MB
HDD	120 GB
OS	Fedora Core 4

[표 1] 마스터 서버 사양

CPU	셀러론 1.0GHz
Memory	128 MB
HDD	40 GB
OS	RedHat Linux 9

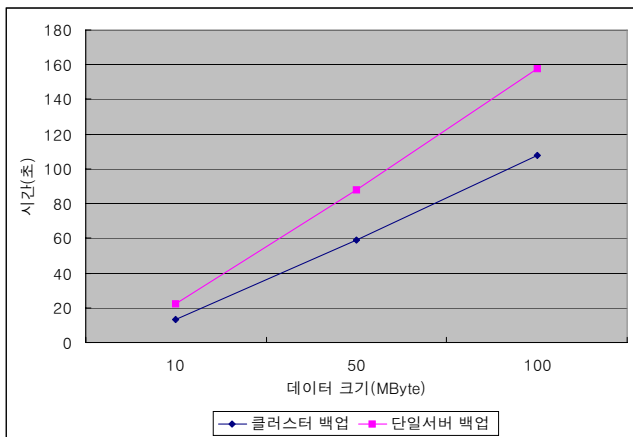
[표 2] 클러스터 서버 노드 사양

실험 및 성능평가에서는 단일 서버와 클라이언트의 일대일 연결에서 데이터 백업과 본 연구에서 제안한 3개의 클러스터 서버 노드를 이용한 백업에 걸리는 시간을 측정해서 비교해 보았다. 각 10, 50, 100MByte의 데이터를 백업하고 복구하는 과정을 각 5회씩 반복해서 실행시킨 후 그 평균값을 사용하였다. 구현된 프로토타입에서는 올바르게 전송이 되는지의 여부를 테스트하기 위해서 스트라이핑 데이터가 전송될 때마다 출력문을 삽입하였기 때문에 성능의 저하가 있을 수 있겠지만 모든 경우에 적용되었으므로 결과 값의 절대적 차이가 있을 뿐 상대적 차이에는 변함이 없을 것으로 예상된다.



[그림 4] 데이터 백업 성능평가

위의 [그림 4]의 그래프를 살펴보면 본 연구에서 제안한 클러스터 방식의 백업 솔루션이 단일 서버 백업에 비해서 높은 성능을 보여주고 있음을 알 수 있다. 또한 백업 데이터가 커짐에 따라서 그 차이가 커지고 있음을 확인할 수 있는데 이는 대용량 백업을 하는 경우에 더 큰 이점이 있음을 보여준다.



[그림 5] 데이터 복원 성능평가

[그림 5]의 그래프는 백업한 데이터를 복원하는 부분에 대한 실험 및 성능평가의 결과를 보여주고 있다. 백업과정에 비해서 상대적으로 연산이 적기 때문에 전체적인 시간이 적게 걸렸으나 단일서버 백업과 클러스터 서버 백업간의 차이는 [그림 4]에서 나타난 결과와 큰 차이가 없음을 알 수 있다.

5. 결론 및 향후 연구 내용

본 연구에서는 저사양의 PC를 이용한 클러스터 기법을 통해 적은 비용을 가지고 높은 성능을 가진 백업 시스템을 제안하고 기존에 제시되었던 클러스터 기법을 확장해서 높은 성능과 안정성을 제공하는 시스템을 설계 및 구현하였다. 향후, 고가용성(High Availability)[4]을 지원함으로 예기치 못한 클러스터 서버 노드의 시스템에 이상이 생겼을 시 남아있는 클러스터 서버 노드를 이용하여 작업을 이어나갈 수 있는 방법을 연구할 계획이다. 또한 RAW Disk 기업을 적용하여 서버 데몬 어플리케이션에서 커널을 거치지 않고서도 파일에 대한 접근이 가능하도록 하여 성능 향상을 한다.

참고문헌

- [1] Tom Waghorn, "Testing the Data Maintenance of the File Allocation Table File System", 1st Australian Computer, Network & Information Forensics conference 2003, 25 November 2003, Perth, Wester Australia
- [2] Cluster File System, Inc, "Lustre: A Scalable, High-Performance File System", November 11th, 2002
- [3] Mendel Rosenblum and John K. Ousterhout, "The Design and Implementation of a Log-Structured File System", 13th ACM Symposium on Operating Systems Principles, July 24th, 1991
- [4] Enrique Vargas, "High Availability Fundamentals", Enterprise Engineering Sun BluePrints OnLine, November, 2000