iSCSI 프로토콜 기반의 멀티미디어 콘텐츠 서비스지원을 위한 클러스터링 저장시스템

김문경, 김선태, 노재춘, 박성순

세종대학교 컴퓨터공학과
e-mail : kmk1030@naver.com, kimst4444@gmail.com, jano@sejong.ac.kr,
sspark@anyang.ac.kr

iSCSI Protocol-based Clustering Storage System for supporting Multimedia Contents

Moonkyung Kim, Suntae Kim, Jaechun No, Sungsun Park

School of Computer Science, Sejong University

요 약

본 논문은 블록단위 데이터 접근이 가능하며, 같은 데이터로의 동시 접근을 제어할 수 있는 록 서비스 기능을 지원하는 iSCSI 기반의 클러스터링 저장 시스템을 제안한다. 본 논문에서 제시되는 iSCSI기반의 클러스터링 시스템은 중.소 규모의 저장 시스템 구축에 유용하게 활용될 수 있고, 동시에 빠른 성능의 멀티미디어 데이터 서비스를 제공할 수 있다.

1. 서 론

멀티미디어 서비스 지원을 위한 클러스터링 저장시스템을 구성하기 위해서는 데이터 저장을 위한 스토리지와 사용자들에게 투명한 접근 및 공유를 허용할 수 있는 미들웨어[1]가 필요하다. 또한, 일단 저장이 된 콘텐츠들은 주로 읽기 요청을 서비스하기 때문에 클러스터에 참여하는 노드들의 일부만 쓰기를 담당하는 생산자 노드로 설정하고, 나머지 노드들은 읽기를 위한 소비자 노드 역할을 수행 하는 등의 차별화된 노드 관리 정책이 필요하다.

멀티미디어 데이터를 처리하기 위한 방식으로 전형적인 클라이언트-서버 구조인 NFS(Network File System)[2]를 사용할 수 있다. 그러나 NFS는 멀티미디어 서비스가 요구하는 빠른 성능을 지원할 수 없고, 위에서 언급한 차별화된 노드 관리 정책을 제공할 수 없다. 네트워크 공유를 위한 또 다른 방식으로 SAN(storage Area Network)[3][4]을 들 수 있다.

SAN(Storage Area Network)은 스토리지들을 파이버 채널[5]로 연결하여 구성하게 되는데, 파이버 채널을 이용한 구성은 투자비용이 많이 들고, 10Km의 거리 제한이 있어 채널 확장기를 통해 거리제한을 연장해야 하는 문제점이 있다.

이러한 SAN의 문제점을 보완하는 방법으로 보편화된 이더넷 IP를 이용하는 방식이 있다. 이더넷 IP (Internet Protocol)가 LAN과 WAN의 중심 요소로 등장하고, 데이터 스토리지 요구조건이 지속적으로 강화됨에 따라 스토리지와 IP 네트워킹을 통합시키기 위한 다각적인 방안들이 모색되었다.[6]

iSCSI (Internet Small Computer Systems Interface) 프로토콜[7]은 스토리지와 IP 네트워크를 통합시키는 방안중 하나로 제시되었다. iSCSI는 IP 네트워크에서 블록 단위로 스토리지 트래픽을 전송해 주는 프로토콜로서, 스토리지의 SCSI 명령(SCSI Command)과 네트워킹의 IP 프로토콜을 기반으로 하고 있다.

iSCSI는 IP 네트워크에서 스토리지 I/O 블록 데이터를 전송하기 위한 단대단(End-to-End) 프로토콜로서, 서버(Initiator), 스토리지 장비(Targets), 프로토콜 전송 게이트웨이 장비 등에 사용되고 있다. iSCSI는 서버에서 스토리지로의 데이터 전송을 위해 표준 이더넷 스위치와 라우터를 사용하며, IP와 이더넷 인프라를 사용하여 SAN 스토리지 접속성을 강화하고 SAN 연결성을 무한히 확장할 수 있다. 또한 파이버 채널을 이용하지 않

본 논문은 Seoul R&BD Program(10557) 지원을 받아 수행되었음

고도 구성이 가능하기 때문에 비용이 저렴하고, 기존 이더넷 IP 네트워크를 사용하기 때문에 거리 제한이 없는 등의 장점을 가지고 있다[8].

반면에 현재 iSCSI는 NFS같이 다중 사용자들의 동시 데이터 접근을 제어하기 위한 록 서비스가 존재 하지 않아 다중 사용자 접근 시에는 데이터의 읽기만 가능한 문제점이 있다.

본 논문은 블록단위 데이터 접근이 가능하며, 같은 데이터로의 동시 접근을 제어할 수 있는 록 서비스 기능을 지원하는 iSCSI 기반의 클러스터링 저장 시스템을 제안한다. 본 논문에서 제시되는 iSCSI기반의 클러스터링 시스템은 중.소 규모의 저장 시스템 구축에 유용하게 활용될 수 있고, 동시에 빠른 성능의 멀티미디어 데이터 서비스를 제공할 수 있다.

2. 관련 연구

2.1 iSCSI

Internet Small Computer Systems Interface(iSCSI)는 target과 initiator로 구성되어 있으며, 이중 target은 데이터 제공을 위한 서버의 역할을 수행하고, initiator는 target에 SCSI[9][10] 명령을 요청하는 클라이언트의 역할을 하게 된다. Initiator는 Target을 통해 공유된 디스크에 SCSI 명령을 요청하고 Target은 요청에 대한 응답 및 처리를 담당한다.

iSCSI는 target와 initiator 간의 연결을 설정한 후, 통신을 위해 PDU(Protocol Data Unit)라는 프로토콜을 사용한다. PDU는 기존 TCP/IP의 네트워크 전송에 필요한 라우팅 정보를 가지는 IP Header와 전송 정보에 대한 신뢰성 유지를 위한 TCP Header, 그리고 iSCSI의 target의 위치를 위한 iSCSI Header와 SCSI Command, 전송될 데이터의 조합으로 이루어져 있다. PDU를 이용하여 initiator는 target에 SCSI Command를 요청하고 target은 PDU를 이용하여 응답 및 처리를 한다.

현재 iSCSI는 target과 initiator간에 1대1 관계를 형성하여 데이터 통신이 일어나도록 설계되어 있으며, 동시에 여러 initiator간 읽기 통신만 가능하도록 구성되어 있다.

2.2 Frangipani

Frangipani[11]는 확장성 있는 분산 파일 시스템으로서, 여러 Petal 가상 디스크[12]를 결합한 스토리지 풀(storage pool)을 구성하여 사용자들에게 단일 스토리지 환경을 제공하도록 구현되어있다.

Petal은 대용량 스토리지를 구축하기 위해 리모트 노드들의 독립적인 디스크들을 가상 디스크로 설정하고, 설정된 가상 디스크들을 그룹화 하여 하나의 단일 시스템 이미지를 지원하도록 한다.

2.3 Global File System(GFS)

GFS[13]는 Frangipani와 같이 대용량 데이터 처리를 위한 분산 파일 시스템이며, 여러 서브풀(subpool) 들을 SAN으로 연결한 NSP(Network Storage Pool)을 구축하도록 하고 있다. NSP는 여러 서브풀들을 그룹화 하여 단일 시스템 이미지를 형성하도록 지원하는 역할을 수행한다. NSP에 속해 있는 각 서브풀은 디스크나 볼륨 당 하나의 RG(Resource Group)을 가지고 있다. 각 RG는 SGI XFS의 allocation group와 같은 역할을 수행하며, 블록 정보, 데이터 비트맵, dinode, 데이터 블록 등으로 구성되어 있다.

3. iSCSI 기반 클러스터링 시스템

3.1 구조

[그림 1]는 iSCSI기반 클러스터링 파일 시스템의 구성도를 보여 준다. 각 클라이언트노드 상에서는 iSCSI initiator 모듈이 수행되고, 실제 데이터를 저장할 스토리지에는 iSCSI target 모듈이 수행되며, 단일 시스템 이미지 지원과 분산 록 관리 및 메타데이터의 처리를 위한 MDS(Meta-Data Server)로 구성된다.

MDS상에서는 전체 iSCSI target device의 정보를 관리하기 위한 CLVM (Clustering Logical Volume Manager) 모듈이 수행된다. CLVM은 타겟 디바이스들을 단일 시스템 이미지로 만들어 주고, 블록 할당, inode 비트맵 검사, 실제 데이터가 존재하는 타겟 디바이스를 가리키는 실제 주소를 관리한다. MDS의 또 다른 역할은 메타데이터 관리 이다. 메타데이터 관리에는 시스템 고장에 대비한 로그 생성 및 데이터의 생성시간, 수정시간, 권한, 사용자, 그룹 등의 요소 관리를 포함한다. 또한, 사용 중인 데이터를 보호하기 위하여 앞서 설명한 분산 록 관리 모듈이 수행된다.

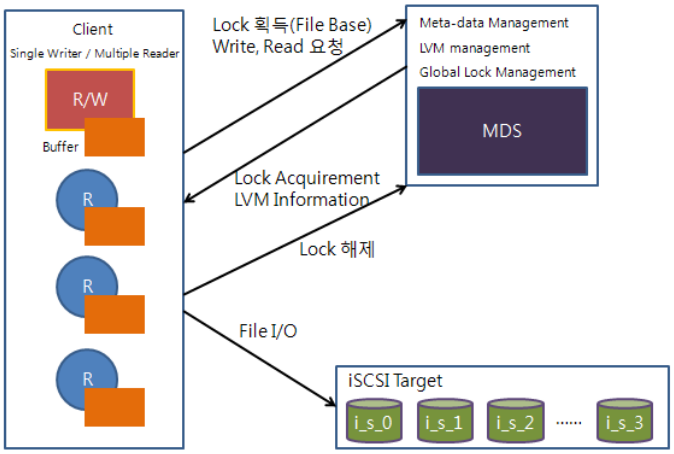


그림 1. iSCSI 기반 클러스터링 파일 시스템

iSCSI Target에서는 실제 데이터를 저장하고 있는 각각의 스토리지는 RAID[14]로 구성되어 있으며, iSCSI Enterprise Target[15]을 실행 시켜 RAID로 묶인 iSCSI 타겟 디바이스를 공유할 수 있도록 구축되어 있다. 클라이언트는 MDS에서 얻은 정보를 토대로 iSCSI 타겟 디바이스와 데이터 통신을 수행할 수 있도록 설정되어 있다.

클러스터에 참여하는 클라이언트노드들은 하나의 생산자와 다중의 소비자로 차별화되어 설정될 수 있다. 이는 데이터의 생성은 적고 반면에 데이터의 소비가 많은 UCC 와 같은 멀티미디어 콘텐츠 데이터를 사용할 경우 좋은 성능을 발휘할 수 있다. [그림 2]에서 보여주듯이 클라이언트노드 상에는 Intel iSCSI initiator[16] 모듈이 실행된다. 각 클라이언트는 읽은 데이터를 보관할 수 있는 버퍼캐시를 가지고 있어 이미 읽은 데이터를 다시 읽지 않도록 하고 있다.

파일작업을 수행하기 위해서는 먼저 MDS에 접근하여 록과 CLVM 정보를 획득하고, CLVM 정보에 따라 iSCSI 타겟 디바이스에 접근하여 데이터 통신을 수행한다. 요청한 작업이 종료되면 MDS에 록 해제 메시지를 보내어 데이터 사용이 끝났음을 알린다.

3.2 파일 생성

3.2.1 Distributed Lock Manager(DLM)

분산 록 관리(DLM)[17]는 분산 파일 시스템에서 클라이언트의 록 요청에 대한 처리를 수행하고, 클라이언트에 대한 상태 정보를 관리한다.

록의 상태는 다른 클라이언트가 쓰기 작업을 수행하지 못하도록 배제하는 읽기 록, 다른 클라이언트가 읽기나 쓰기 작업을 못하게 제어하는 쓰기 록, 록을 보유한 클라이언트가 없음을 가리키는 록 해제 등 세 가지로 구분된다.

A	P1	Read	Lock

그림 2. 록 테이블

[그림 2]은 록 테이블을 보여준다. 록 테이블은 파일 이름, 클라이언트 아이디, 록 종류, 록 상태 순서로 구성되어 있다.

록의 해제 조건은 록을 요청한 클라이언트의 해제 요청이 있거나, 수행중인 프로세스의 종료로 이루어진다.

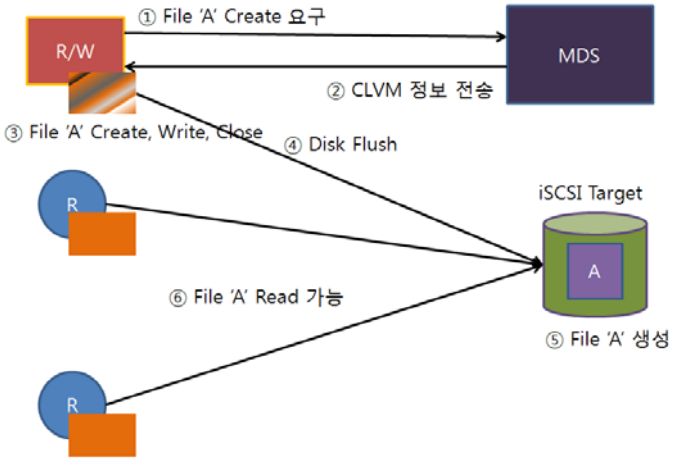


그림 3. 파일 생성 순서

[그림 3]은 파일 생성 순서를 보여 준다. 파일의 생성은 읽기/쓰기 권한을 가지는 생산자 클라이언트만이 가능하다. 파일생성을 위해 생산 클라이언트는 MDS에게 록 요청을 수행한다.

MDS는 요청된 록을 생산자 클라이언트에 제공한 후, 요청된 파일 이름을 갖는 객체를 생성한다. 또한, 파일 데이터가 저장될 iSCSI 타겟 디바이스 정보를 포함하고 있는 CLVM를 생성한 후, 이를 클라이언트에게 전송한다.

읽기 권한만을 가지는 소비자 클라이언트는 생산자 클라이언트의 버퍼 안에 저장된 최근 데이터가 디스크로 저장될 때까지 기다려야 하며, 저장 후에만 그 최근 데이터로의 접근이 가능하다.

3.2.2 록 수행 순서

[그림 3]과 같이 생산자 클라이언트가 파일을 생성

중일 경우, 소비자 클라이언트는 접근을 할 수 없다. 파일 생성을 위해 생산자 클라이언트는 MDS에게 록 요청을 수행하고, MDS는 록 요청에 대해 생산자 클라이언트가 쓰기 록을 획득했음을 록 테이블에 기록한 후, 록 획득을 생산자 클라이언트에게 알려 준다.

생산자 클라이언트가 쓰기 록을 획득하면 iSCSI 타겟 디바이스에 데이터를 저장하게 되고, 쓰기 록을 해제할 때까지 소비자 클라이언트는 생성중인 데이터에 접근할 수 없다. 생산자 클라이언트가 데이터를 모두 쓴 후 프로세스가 종료되면, MDS는 록 테이블에 저장된 쓰기 록 내용을 쓰기 록 해제 상태로 수정 한다.

소비자 클라이언트가 데이터 읽기 요청을 하게 되면 MDS는 록 테이블을 확인하여 생산자 클라이언트의 쓰기 록이 해제 상태임을 확인한다. 또한, 소비자 클라이언트의 읽기 록을 테이블에 등록하고 데이터의 CLVM 정보를 전송하여 데이터를 읽을 수 있도록 한다.

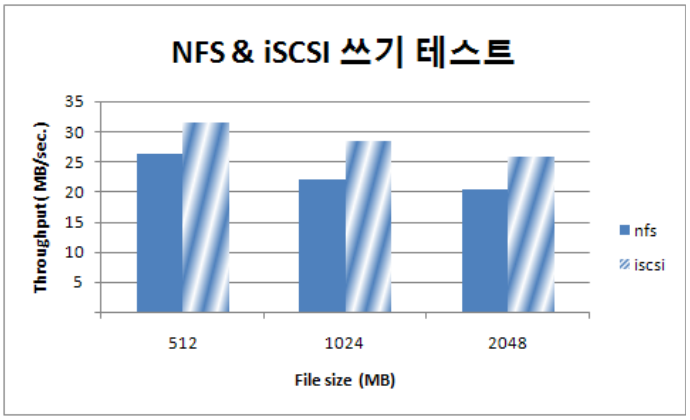
4. 성능 테스트

CPU	AMD Athlon(tm) XP 2500+
램	512MB
HDD	80GB
OS	Fedora Core 3
네트워크카드	Intel(R) PRO/1000
허브	3com Baseline Switch 2824

표 1. 테스트 환경

성능 테스트를 위해 [표 1]에 기술된 환경을 갖추고 있는 두 대의 노드를 사용하였고, 그 중 서버로 구성된 노드에 커널 2.6.20 기반의 iSCSI Enterprise Target 모듈을 설치하였다. 클라이언트 노드에는 커널 2.6.12 기반의 Intel iSCSI initiator 모듈을 설치하였으며, NFS와의 성능 비교를 수행하였다. 각 테스트는 벤치마크 도구인 IOzone[18]을 수행시킨 후 평균값을 표사한 것이다.

[그림 4]는 쓰기 및 읽기 성능을 측정한 결과를 보여준다. [그림 4]에서 보듯이 쓰기 및 읽기 성능은 iSCSI가 NFS의 입출력 성능 보다 더 빠름을 알 수 있다. 또한, 512MB이상의 데이터를 저장할 경우, NFS와 iSCSI의 성능 차이가 더 커짐을 알 수 있다. [그림 4]는 512MB 이상의 데이터를 읽을 경우에도 같은 결과를 보여줌을 알 수 있다.



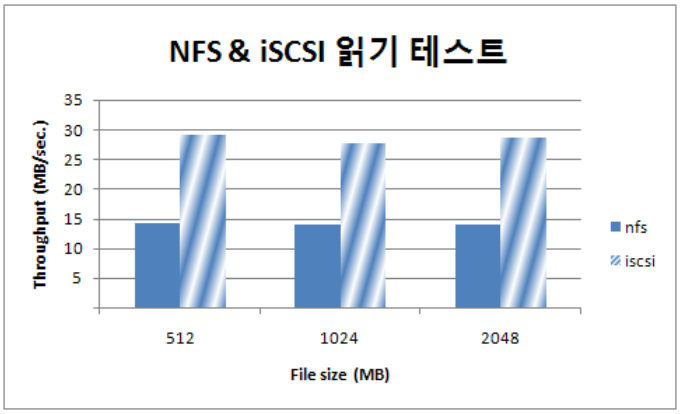


그림 4. NFS & iSCSI 입출력 성능 테스트

5. 결론 및 향후 과제

본 논문에서 제시한 iSCSI 기반의 클러스터링 저장시스템은 클러스터를 구성하는 노드들의 차별화된 관리 정책을 통해 빠른 멀티미디어 서비스를 지원할 수 있도록 구축되었다. 이를 위해 고성능 및 저가의 장점을 제공하는 iSCSI 프로토콜을 이용하였으며, 사용자들에게 투명한 접근 및 공유를 허용할 수 있도록 설계되었다.

현재 노드들의 차별화 정책을 추가한 iSCSI 기반 클러스터링 저장시스템 구현 중에 있다.

참고문헌

[1] Andrew S. Tanenbaum, Maarten Van Steen, "Distributed Systems Principles and Paradigms", Second Edition, 2007

[2] Brian Pawlowski, Chet Juszczak, Peter Staubach, Carl Smith, Diane Lebel, David Hitz, "NFS Version 3 Design and Implementation", In Proceedings of the Summer USENIX Conference, pages 137-152, June 1994

[3] Jon Tate, Fabiano Lucchese, Richard Moore,

"Introduction to Storage Area Networks", IBM, Redbooks, July 2006
[4] Omar Barazza, Ted Uhler, "Storage Area Networks: The Superior Storage Solution", A Dot Hill Paper, Octover 2000
[5] John Wiley & Sons, "Storage Networks Explained", 2004
[6] P. Sarkar, K. Voruganti, K. Meth, O. Biran, J. Satran, "Internet Protocol storage area networks", IBM SYSTEM JOURNAL, VOL 42, NO 2, 2003
[7] J. Satran, K. Meth, C. Sapuntzakis, M. Chadalapaka, E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC-3720, April. 2004
[8] Tony Asaro, "iSCSI-based IP Storage Area Networks", The Enterprise Strategy Group, April 2006
[9] Ralph O. Weber, John B. Lohmeyer, George O. Penokie, "SCSI Architecture Model - 2", SAM-2, September 2002
[10] Ralph O. Weber, John B. Lohmeyer, George O. Penokie, "SCSI Primary Commands - 3", SPC-3, January 2002
[11] Chandramohan A. Thekkath, Timothy Mann, Edward K. Lee, "Frangipani: A Scalable Distributed FIle System", ACM Special Interest Group on Operating Systems, v.31 no.5, pp.224-237, 1997
[12] Edward K. Lee and Chandramohan A. Thekkath, "Petal: Distributed Virtual Disks", ACM Press, v.31 no.9, pp.84-93, 1996
[13] Steven R. Soltis, Grant M. Erickson, Kenneth W. Preslan, Matthew T. O'Keefe, and Thomas M. Ruwart, "The Global File System: A File System for Shared Disk Storage", NASA conference publication, v.3340 no.2, pp.319-342, 1996
[14] Sameshan Perumal and Pieter Kritzinger, "A Tutorial on RAID Storage Systems", Data Network Architectures Group Department of Computer Science University of Cape Town Private, May 6, 2004
[15] The iSCSI Enterprise Target Project, http://iscsitarget.sourceforge.net/
[16] Intel's Open Storage Toolkit, http://sourceforge.net/projects/intel-iscsi
[17] Kristin Thomas, "Programming Locking Applications", IBM Corporation, http://opendlm.sourceforge.net/
[18] IOzone Filesystem Benchmark, http://www.iozone.org