DOI: 10.3745/KIPSTC.2009.16-C.3.363

클러스터 VOD 서버의 부분적 장애에서 QoS 보장

이 좌 형[†] · 정 인 범[††]

요 약

대용량 VOD 서비스를 위한 서버로 높은 성능과 낮은 가격의 클러스터 서버가 주목받고 있다. 일반적으로 클러스터 서버는 하나의 front-end 노드와 여러 back-end 노드로 구성된다. back-end 노드 수를 증가시키면 더 많은 클라이언트들에게 QoS를 보장하는 스트리밍 서비스를 할 수 있지만, back-end 노드의 오류 가능성도 이와 비례하여 증가한다. 서버의 장애는 모든 스트리밍 서비스를 중단시킬 뿐 아니라 현재 재생 위치 정보도 잃어버린다. 본 논문에서는 back-end 노드가 오류 상태가 될 때, 끊이지 않는 스트리밍 서비스를 지원하기 위한 복구 방법을 제안한다. 실제 VOD 서비스 환경을 위해, 일반 PC로 구성된 클러스터 기반의 VOD 서버를 구현하였으며, MPEG 영화를 위한 병렬 처리 기법을 사용하였다. 구현된 VOD 서버에 패리티 연산을 이용한 비디오 블록 복구 방법을 설계하였다. 하지만, 클러스터 기반의 VOD 서버 구조를 고려하지 않으면 복구를 위한 내부 네트워크 성능의 병목현상과 back-end 노드들의 비효율적인 CPU 사용을 야기시킨다. 본 논문에서는 이러한 문제를 해결하기 위해, 파이프라인 개념을 이용한 새로운 장애 복구 방법을 제안한다.

키워드 : 복구, 파이프라인 컴퓨팅, VOD, 클러스터, QoS, 병렬처리

QoS Guarantee in Partial Failure of Clustered VOD Server

Joahyoung Lee[†] · Inbum Jung[††]

ABSTRACT

For large scale VOD service, cluster servers are spotlighted to their high performance and low cost. A cluster server usually consists of a front-end node and multiple back-end nodes. Though increasing the number of back-end nodes can result in the more QoS streams for clients, the possibility of failures in back-end nodes is proportionally increased. The failure causes not only the stop of all streaming service but also the loss of the current playing positions. In this paper, when a back-end node becomes a failed state, the recovery mechanisms are studied to support the unceasing streaming service. For the actual VOD service environment, we implement a cluster-based VOD servers composed of general PCs and adopt the parallel processing for MPEG movies. From the implemented VOD server, a video block recovery mechanism is designed on parity algorithms. However, without considering the architecture of cluster-based VOD server, the application of the basic technique causes the performance bottleneck of the internal network for recovery and also results in the inefficiency CPU usage of back-end nodes. To address these problems, we propose a new failure recovery mechanism based on the pipeline computing concept.

Keywords : Recovery, Pipeline computing, VOD, Clusters, QoS, Parallel processing

1. 서 론

최근 컴퓨터와 네트워크 기술의 발달로 VOD (Video-On-Demand), 전자 도서관, 원격 교육 같은 멀티미디어 서비스를 경제적으로 제공할 수 있게 되었다. VOD 서비스는 가장 대표적인 멀티미디어 애플리케이션이며, QoS를 보장하는 스트리밍 비디오 데이터를 온라인 사용자에게 제공한다. 원본 영상 데이터가 저장장치의 많은 부분을 차지하기 때문에, MPEG 기술과 같은 고효율 압축 방법이 저장 공간을 줄이는데 사용 된다[1]. 그러나 MPEG이 비디오 데이터 크기를 많이 줄일 수 있더라도, VOD 서버가 많은 클라이언트에게 다양한 영화 컨텐츠를 제공하기 위해서 대용량의 저장 공간 사용은 피할 수 없다[2].

스트리밍 비디오의 끊김과 지터는 VOD 클라이언트에게 무의미 하므로, 스트리밍 미디어는 각 클라이언트의 QoS 기준을 만족시킬 수 있어야 한다. QoS를 제공하기 위해 서버는 VOD 클라이언트에게 일정 간격으로 비디오 데이터를 계속해서 전송할 수 있어야 한다. 그리고 또한 서버들 중에 장애가 발생하더라도, 스트리밍 서비스는 사용자가 허용 가능한 MTTR (Mean Time To Repair) 값 안에서 복구되어

※ 본 연구는 산업자원부와 한국산업기술재단의 지역혁신인력양성사업으로 수행된 연구결과임
† 정 회 원 : 강원대학교 컴퓨터정보통신공학과 박사과정
†† 준 회 원 : 강원대학교 컴퓨터정보통신공학전공 교수(교신저자)
논문접수 : 2008년 7월 10일
수 정 일 : 1차 2008년 9월 10일, 2차 2008년 11월 3일
심사완료 : 2008년 11월 24일

야 한다[3, 4].

노드의 장애는 모든 스트리밍 서비스를 중단시킬 뿐 아니라 모든 상영되는 영화의 서비스되는 위치 정보를 잃어버리게 된다. 노드의 장애에도 VOD 서버가 QoS를 보장해야 하기 때문에, 실제 VOD 서비스를 다루기 위해 장애 복구 기법이 필요하다[6, 7]. 본 논문에서는, 클러스터 기반의 VOD 서버에서 back-end 노드의 장애 발생시 QoS를 보장하기 위한 복구 방법을 제안한다. 실제 VOD 서버의 장애 발생을 알아보기 위해서, 일반 PC로 구성된 클러스터 기반의 VOD 서버를 구현하였고 많은 수의 클라이언트에게 서비스를 할 수 있도록 MPEG 미디어를 위한 병렬 처리 기법을 채택하였다[9].

구현된 VOD 서버 상에서 복구시스템은 복구 노드의 입력 네트워크에 병목현상을 야기시키며 back-end 노드들이 비효율적으로 CPU를 사용한다. 이러한 문제를 해결하기 위해 모든 back-end 노드들 간에 파이프라인 컴퓨팅 기반의 새로운 장애 복구 시스템을 제안한다. 제안된 시스템은 배타적 OR 를 이용하여 컴퓨팅 부하 뿐 아니라 back-end 노드 간의 네트워크 트래픽을 분산시킨다. 정상적인 back-end 노드들은 모두 복구 과정에 참여하여 클러스터 기반의 VOD 서버에서 back-end 노드 장애 발생시 끊기지 않는 스트리밍 서비스를 하는 개선된 성능을 제공한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 연구를 설명하고 클러스터 VOD 서버인 VODCA와 클러스터 아키텍처에서 비디오 블록 관리에 대하여 설명한다. 3장에서는 기본적인 복구 시스템을 제안한다. 구현한 VOD 서버에서 기본적인 복구 시스템의 성능을 측정하고 시스템의 제약을 알아본다. 4장에서는 back-end 노드들을 이용하기 위한 파이프라인 개념을 이용한 새로운 복구 방법을 제안한다. 5장에서 본 논문의 결론을 맺는다.

2. 관련연구

2.1 복구기법들

다양한 클라이언트의 요구와 제한된 자원 하에서 좀더 많은 클라이언트에게 안정적인 서비스를 제공하기 위한 VOD 시스템에 관한 많은 연구가 이루어지고 있다[17, 18, 19]. 상업적인 VOD 서비스에서 부분적인 오류 상태가 발생하는 경우에 QoS 스트림은 제한된 MTTR값 안에서 클라이언트에게 제공되어야 한다. 인간에게 허용 가능한 MTTR값은 VOD 서비스를 위한 필수 조건이다. 이것은 우수한 VOD 서비스를 위한 중요한 QoS의 평가 요소이다. 파일, 데이터베이스, 웹 서버에서의 fault tolerance분야에 많은 연구가 진행중이다. 하지만 미디어 스트리밍은 실시간성과 같은 특징이 있다. VOD 서버의 부분적인 오류 상태에서 끊김이나 지터 현상 없는 QoS 스트림을 보장하기 위한 충분한 연구가 이루어지고 있지 않다.

오류가 발생한 저장 시스템을 복구하기 위해 미러(Mirror) 개념 기반의 연구가 수행되었었다[8, 20]. Tiger 비디오 서버는 VOD 서비스를 위한 미러(Mirror) 기반의 저장시스템에 구현되었다[21]. RMD(Rotational Mirrored Declustering) 기술은 오류 발생 디스크나 개별 노드를 복구하기 위해 제안되었다[5]. 하지만 이들 미러(Mirror) 기반의 접근은 디스크를 비효율적으로 사용할 뿐 아니라 복구 노드의 부하를 가중시킨다.

RAID는 일반적으로 클러스터 서버의 병렬 노드나 오류가 발생한 디스크를 복구하기 위해 사용된다. 특히, RAID-3, 4, 5는 패리티 기반의 복구 알고리즘을 사용한다[11, 12, 20]. RAID-3는 데이터 블록이 나눠져서 데이터 디스크에 기록된다. 작은 단위로 나누기 때문에, 큰 비디오 블록을 인출하기 위해서 수많은 디스크 접근이 필요하다. RAID-4는 각각의 완전한 블록을 디스크에 기록한다. 이 방법은 큰 단위로 나누어진 유닛을 제공하기 때문에 한 번에 디스크로부터 한 번에 큰 블록을 인출함으로써 디스크 성능이 개선될 수 있다. RAID-3와 RAID-4 모두 복구 과정에 사용되는 패리티 디스크를 가지고 있다. RAID-5에서는 같은 수준(rank)의 패리티 블록이 분산된 디스크에 나누어져 있다. back-end 노드 오류시, 모든 남아있는 노드는 모두 개별적으로 복구 과정을 수행한다. 패리티 블록이 모든 디스크에 분산되어 있어서 블록들을 교환하기 위해서 노드들 간의 네트워크 트래픽이 크게 증가한다. 이러한 동작 환경 하에서, back-end 노드의 실제 부하를 실시간으로 측정하기 어렵기 때문에 VOD 클라이언트들에게 꾸준한 QoS 스트림을 보장할 수 없다.

최근에, 클러스터 서버 구조는 저비용과 높은 성능으로 많은 분야에서 사용되어 왔다. 실제로 VOD 서비스는 실시간 환경에서 모든 클라이언트에게 스트리밍 미디어를 제공해야 하는 고유의 특징이 있다. 디스크나 back-end 노드의 오류가 발생하더라도, 불규칙한 끊김이나 지터가 인간이 허용하는 시간 안에서 해결되어야 한다[3, 4]. 하지만 현재까지 클러스터 기반의 VOD 서버에서 복구 방법에 관한 연구가 깊이 있게 이루어지지 않았다. 좀 더 상업적인 VOD 서비스를 위해서 미디어 스트리밍의 특징에 맞게 복구 시스템이 연구되어야만 한다.

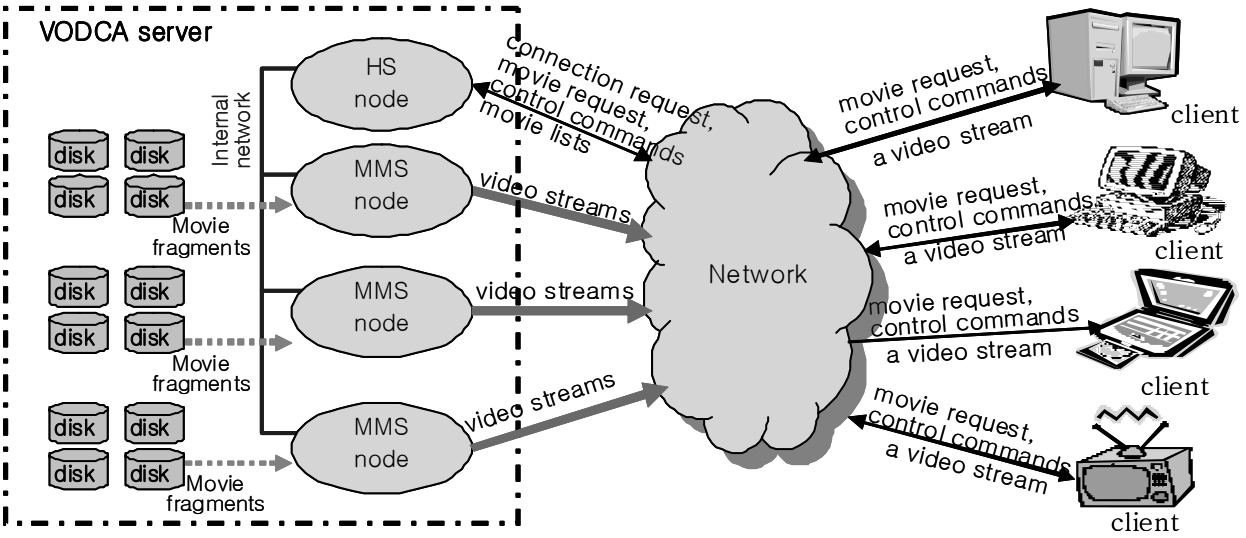
2.2 클러스터 VOD 서버

2.2.1 VODCA의 구조

대용량 VOD 서비스를 위해, 우리는 클러스터 VOD 서버인 VODCA (Video On Demand on Clustering Architecture)를 구현하였다[9]. VODCA는 front-end 서버인 HS (Head-end Server) 와 back-end 노드인 여러 MMS (Media Management Server)로 구성된다. (그림 1)은 VODCA 서버와 여러 VOD 클라이언트의 구조를 나타낸다. 클라이언트는 HS와 MMS 노드들과 상호 작용하며 HS와 MMS 노드 사이에 내부 네트워크를 통해서 동작 상태와 내부 명령을 전달한다[10].

2.2.2 VODCA에서 비디오 블록의 스트라이핑과 복구

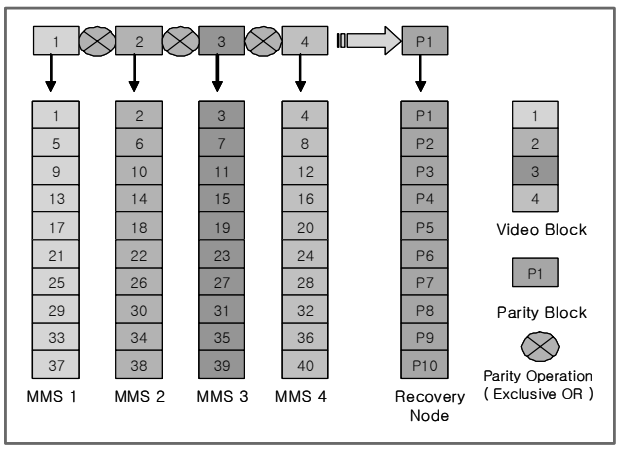
(그림 2)와 같이, 모든 비디오 블록은 각 GOP 크기만큼



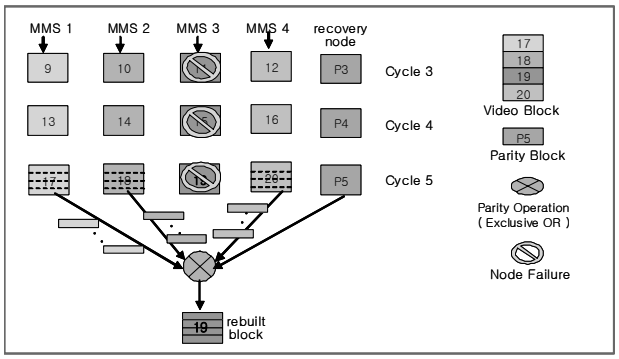
(그림 1) VODCA에서 VOD 서비스의 구조

MPEG 데이터를 가지고 있다. 각 GOP들의 크기는 서로 다르며 보통 수 KBytes 이상이다. GOP 크기가 서로 다르기 때문에 같은 계층(rank)에 포함된 모든 비디오 블록은 다른 크기를 가진다. 같은 계층에 포함된 비디오 블럭의 서로 다른 크기 때문에 패리티 블록은 같은 계층에서 가장 큰 비디오 블록의 크기에 맞추어 생성되야 한다. 복구 노드는 각 MPEG 영화의 패리티 블록뿐만 아니라 비디오 블록 수, 각 비디오 블록의 크기, 저장된 위치 등의 인덱스 정보를 유지해야 한다. 인덱스 정보는 다음 장에서 설명될 복구 과정에서 사용된다.

MMS 노드가 오류 상태가 될 때, 도달할 수 없는 블록의 복구는 패리티 데이터와 다른 MMS 노드에 저장되어 있는 데이터를 배타적 OR 연산을 통해 수행된다. (그림 3)은 MMS 3 노드가 오류 상태일 때의 복구 과정을 나타낸다. 오류가 발생한 MMS 3 노드에 저장된 19번 비디오 블록은 패리티 블록 P5와 17, 18, 20번 블록의 배타적 OR 연산을 통해 생성된다.



(그림 2) 분산된 데이터 블록과 패리티 블록



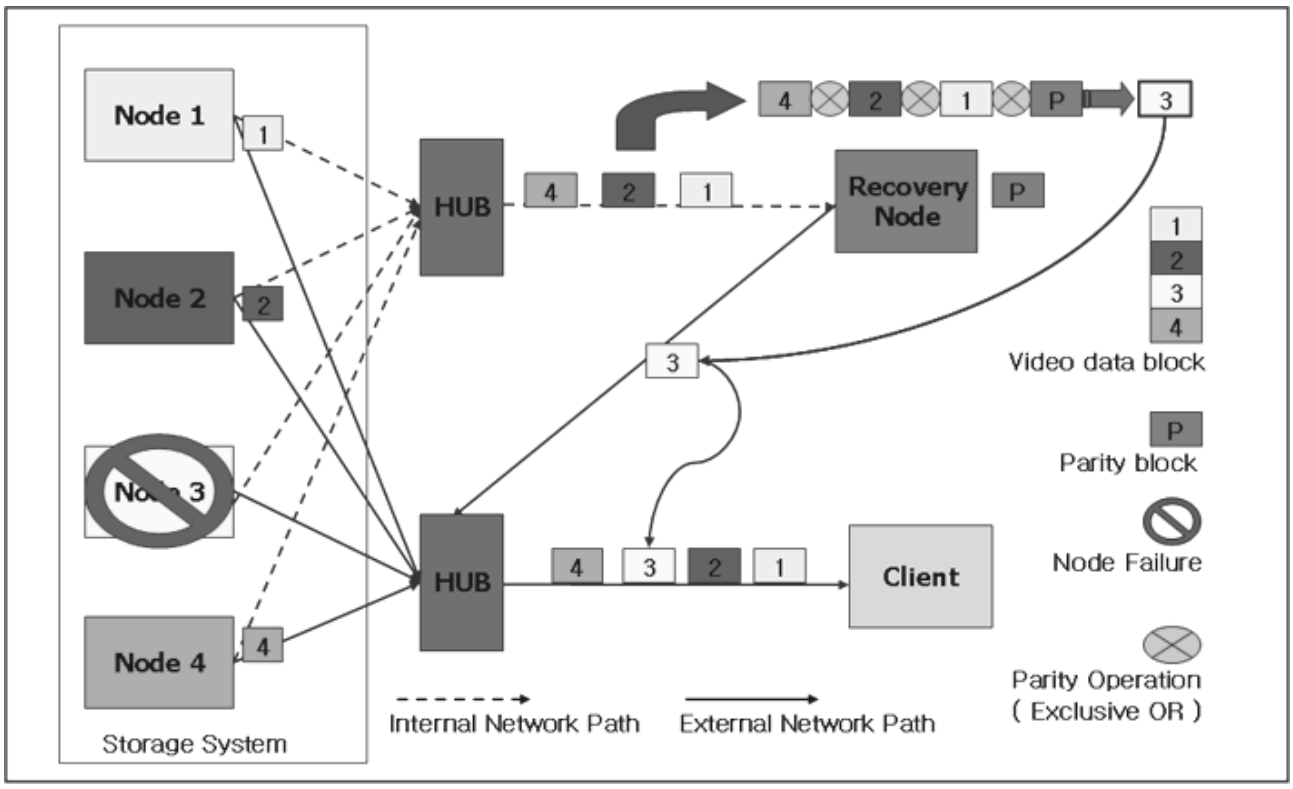
(그림 3) 비디오 블록의 복구

3. 기본적인 복구 시스템

3.1 시스템 구조

(그림 4)는 기본적인 복구 시스템 RSBM (Recover System based on Basic Mechanisms) 구조를 나타낸다. 본 시스템은 2장에서 설명된 VODCA 서버에 구현되었다. (그림 4)와 같이 본 시스템에는 2종류의 네트워크 패스 - MMS 노드와 VOD 클라이언트 사이의 외부 네트워크 패스, MMS 노드들과 복구 노드 사이에 설치된 내부 네트워크의 내부 네트워크 패스 - 가 존재한다. MMS 노드에 장애 발생시, 비디오 블록은 복구 노드로 전송되어야 한다. 이 블록들은 일반 데이터와 분리되어 내부 네트워크를 통해 전송된다. 그러므로 외부 네트워크는 복구 과정에 따른 네트워크 부하 없이 클라이언트의 QoS 를 보장할 수 있다.

모든 MMS 노드가 정상 동작할 때, 모든 MMS 노드는 저장된 비디오 블록을 외부 네트워크를 통해 클라이언트에게 직접 전송한다. 반면에 MMS 노드에 장애 발생시, 정상적인 MMS 노드들은 비디오 블록을 클라이언트와 복구 노드로 동시에 전송한다. 복구 노드는 MMS 노드로부터 받은

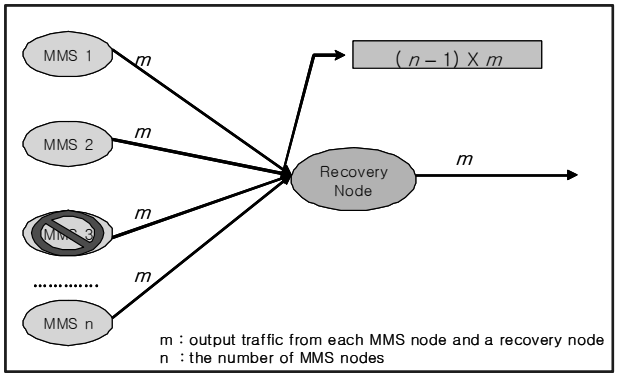


(그림 4) RSBM의 구조와 비디오 블록의 흐름

비디오 블록과 디스크에 저장된 패리티 블록을 이용하여 장애가 발생한 노드의 비디오 블록을 생성한다. MMS 노드들과 복구 노드는 비디오 데이터의 복구를 위해 내부 네트워크를 사용하기 때문에 외부 네트워크를 통해 VOD 클라이언트에게 QoS 스트림을 제공할 수 있다.

3.2 네트워크 부하

(그림5는) MMS 노드 오류시 복구 동작을 수행하기 위한 네트워크 부하를 나타낸다. n - 1 MMS 노드가 살아있고 각 MMS 노드의 출력 트래픽이 m 이라면 (n - 1) x m 만큼의 네트워크 트래픽이 복구 노드의 입력 네트워크로 흘러 들어간다. 복구 노드의 네트워크 인터페이스 능력이 B라면 m 값은 m = B / (n - 1)로 계산할 수 있다. 예를 들어 5개의 MMS 노드가 있고, 네트워크 대역폭이 100Mbps 라면 m 은 25Mbps = 100Mbps / (5 - 1) 이 된다. 이것은 각 MMS 노드가 복구 노드에게 25Mbps 보다 작은 크기로 비디오 데이터를 전송해야 한다는 것을 의미한다. 25Mbps 이상의 비디오 데이터는 QoS 스트림을 제공할 수 없기 때문에 복구 노드 관점에서 필요 없는 데이터를 전송받게 되는 것이다. 이 계산에서 복구 노드 측의 입력 네트워크 양은 MMS 노드 수에 의존적이다.



(그림 5) RSBM 에서 네트워크 트래픽

3.3 실험 환경

3.3.1 복구 노드를 가진 VODCA 설정

실험을 위한 VODCA 서버는 HS 노드와 4개의 MMS 노드 그리고 복구 노드로 구성되며, 각 노드는 Linux 운영체제로 동작한다. MMS 노드, HS 노드, 클라이언트는 100Mbps 이더넷 스위치를 통해 연결되어 있다. 모든 MMS 노드와 복구 노드 또한 100Mbps 이더넷 스위치를 통한 내부 네트워크로 연결되어 있다.

HS 노드의 시스템 관리 툴을 포함한 모든 어플리케이션은 QT, C, C++ 라이브러리를 이용해 개발 되었다. <표 1>은 VODCA 시스템에서 각 MMS 노드의 하드웨어 컴포넌트를 나타낸다. <표 2>는 실험에 사용된 영화에 관한 세부 정보를 나타낸다. MPEG-2 영화를 사용하였으며 시스템의 성능을 측정하기에 충분한 상영시간을 가지고 있다. 또한, <표 2>에서처럼 영화의 GOP들과 I 프레임의 크기를 측정

〈표 1〉 MMS 노드와 복구 노드 사양

CPU	Intel Pentium 4, 1.6 GHz
Memory	256 Mbyte DDR
Disk	Segate Baracuda ATA IV 40GB 7200RPM x 2
OS	RedHat 7.3 (Kernel 2.4.18)
Network	100 Mbps Fast Ethernet, 100Mbps Ethernet Switch with 24 ports

〈표 2〉 실험에 사용된 영화 정보

Movie name	John Q	Ice Age
Frame size(H x V)	352 x 288	352 x 288
Frame rates(number/sec)	25	25
Bit rates(Kbps)	1,437.6	1,437.6
Running time(Minutes)	110	85

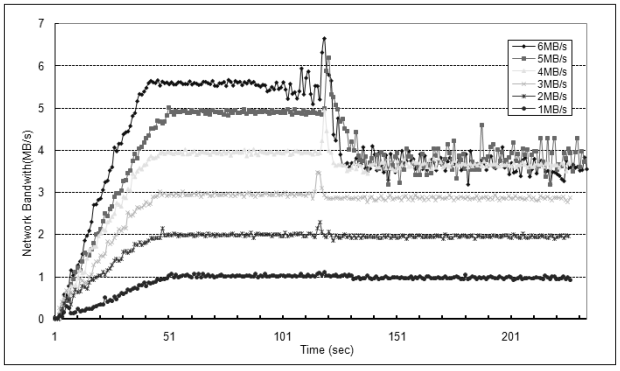
하였다. 실제로 이 정보는 다음 장에서 설명될 성능 병목현상의 원인을 알아내기 위해 사용된다.

클러스터 기반의 VOD 서버의 성능을 측정하기 위해 yardstick program 을 사용하였다[13]. Yardstick 프로그램은 가상 부하 생성기와 가상 클라이언트 데몬으로 구성된다. 가상 부하 생성기는 HS 노드에 위치하고 λ = 0.25의 포아송 분포에 따라서 클라이언트 요청을 생성한다[14, 15]. 생성된 요청은 각 MMS 노드에 보내지게 되고 모든 MMS 노드는 클라이언트가 만족할 수 있도록 동시에 미디어 스트리밍 서비스를 시작한다.

3.4 RSBM 의 성능

(그림 6)은 MMS 노드에서 모든 클라이언트로 전송되는 네트워크 트래픽의 양을 나타내며, 결과는 각 MMS 노드의 네트워크 트래픽의 평균이다. (그림 6) 에서 볼 수 있듯이, 부하 생성기는 여섯가지 부하를 각각 생성한다. VODCA 서버는 각 QoS 스트림에 1.5Mbps 의 전송률을 보장한다. 4대의 MMS노드가 1.5Mbps의 트래픽을 분담(1.5/4)하여 전송하므로 하나의 MMS노드에서 전송하는 트래픽은 0.357Mbps가 된다. 6MB/sec 의 출력 네트워크 트래픽은 4대의 MMS 노드가 128((6MBps*8bits/byte)/0.357Mbps)개의 클라이언트에게 QoS 스트림을 제공함을 의미한다. 반면에 1MB/sec의 트래픽은 최소 21개의 클라이언트를 지원함을 나타낸다.

MMS 노드의 오류는 시간축의 120초 에서 발생하였다. 오류 발생 이후 1MB/sec, 2MB/sec, 3MB/sec 의 네트워크 트래픽 변화는 최소이다. 하지만 6MB/sec, 5MB/sec, 4MB/sec

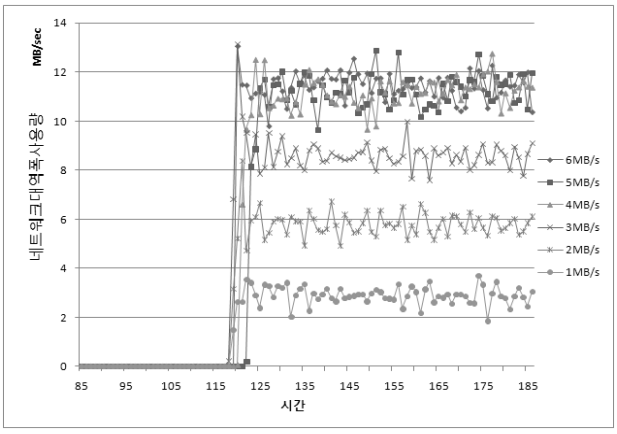


(그림 6) 노드에서 모든 클라이언트로의 네트워크 트래픽

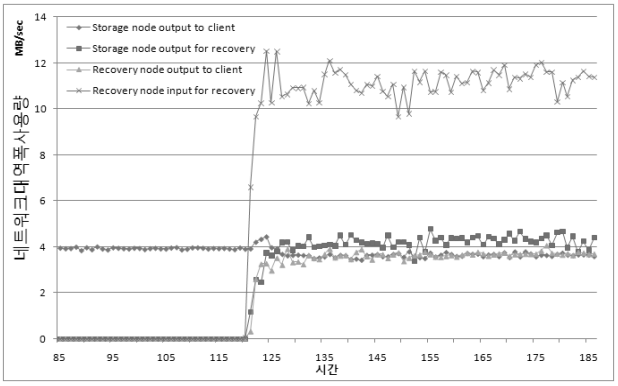
에서 120초 이후에도 네트워크 트래픽 변동이 계속하여 발생한다. 특히 6MB/sec와 5MB/sec의 경우 네트워크 트래픽이 급격히 줄어드는 현상이 발생하였다. 네트워크 병목현상으로 복구 노드가 더 이상 비디오 블록을 받을 수 없기 때문에 MMS 노드가 클라이언트에게 비디오 블록을 완전히 전송할 수 없다. 5MB/sec와 6MB/sec 의 부하 하에서 3 대 MMS 노드로부터 복구 노드의 네트워크 트래픽이 각각 15MB/sec와 18MB/sec에 이른다. 복구 노드의 입력 네트워크 성능이 12MB/sec로 제한되기 때문에 복구 노드는 네트워크의 입력단에서 병목현상을 일으킨다.

(그림 7)은 각 6개의 부하 환경에서 복구 노드의 입력 네트워크 트래픽을 나타낸다. MMS 노드가 오류 발생시, 모든 정상 노드들은 오류가 발생한 MMS 노드에 저장된 비디오 블록을 복구하기 위해 비디오 블록을 클라이언트와 복구 노드에 동시에 전송한다. 4MB/sec 트래픽 보다 적은 부하에서 복구 노드의 입력 트래픽은 3대의 MMS 노드의 출력 트래픽의 합과 같다. 하지만 5MB/sec와 6MB/sec 에서는 입력 트래픽의 양이 부하의 증가비율과 비례하여 증가하지 않는다. 그 이유는 입력 네트워크 트래픽이 12MB/sec로 제한되기 때문이다. 이런 경우에 VODCA 서버는 모든 동시 접속자에게 QoS 스트림을 제공하기 어렵게 된다. 그러므로 서비스 받는 클라이언트 수를 줄이는 것이 불가피 하다.

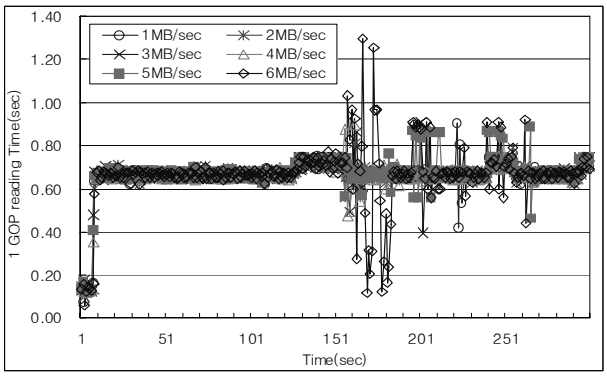
MMS 노드와 복구 노드 사이의 상호작용을 세밀하게 관



(그림 7) 복구 노드의 입력 네트워크 트래픽



(그림 8) 4MB/sec 에서 MMS 와 복구 노드의 네트워크 트래픽



(그림 9) RSBM에서 클라이언트의 GOP reading time

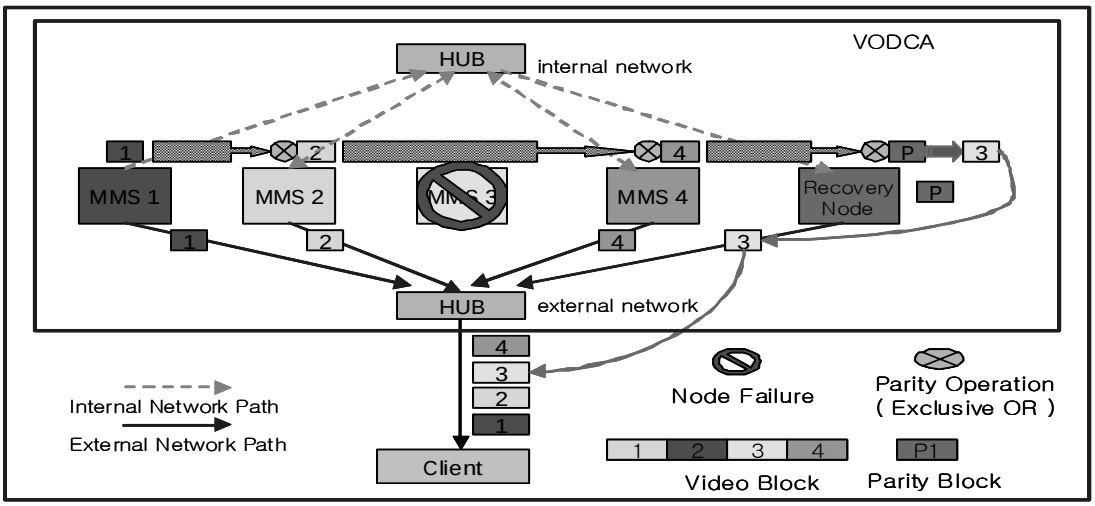
찰하기 위하여, (그림 8)은 4MB/sec의 부하 발생시 MMS 노드와 복구 노드에서의 네트워크 트래픽을 나타낸다. 시간축의 120초에서 MMS 노드의 오류가 발생된 후에 MMS 노드는 계속하여 모든 클라이언트에게 스트리밍 서비스를 제공하고 또한 즉시 복구 노드로 비디오 블록을 전송한다. 복구 노드는 MMS 노드로부터 비디오 블록을 수신한 후, 오류가 발생한 MMS 노드에 저장된 비디오 블록을 복원한다. 그 후, 복구 노드는 복원된 비디오 블록을 모든 클라이언트에게 전송한다. (그림 8)에서 복구 노드의 입력 트래픽의 양은 약 12MB/sec에 이른다. 이것은 3대의 MMS 노드의 출력 트래픽의 합과 같다.

(그림 9)는 스트리밍 서비스 중에 클라이언트에서 한 개의 GOP를 읽는데 걸린 시간을 나타낸다. 그림에서 모든 MMS 노드가 정상적으로 동작할 때 평균 reading 시간은 약 0.65 초 이고 안정된 상태를 유지한다. 하지만, MMS 노드 오류 발생시, reading 시간은 변한다. 초기 복구 노드의 설정 시간, 복구 노드의 데이터 혼잡 현상과 패킷 데이터 손실로 인해 이러한 변화가 발생한다. 특히, 5MB/sec 와 6MB/sec의 부하 발생시 높은 변화율을 나타냈다. 이러한 작업 부하에서 reading 시간의 불안정한 상태는 156초와 266초사이에 나타난다. 최대 reading 시간과 최소 reading 시간의 차이는 약 1.18초이다. 변동기간이 지난 후에 복구 노드는 정상 동작을 하고, reading 시간은 다시 0.65초 수준으로 바뀐다. MTTR 값은 110 초이다[3, 4]. 이 시간은 VOD 클라이언트에게 긴 시간으로 여겨질 수 있다.

4. 파이프 라인 기반의 복구 시스템

4.1 시스템 구조

이전 장에서 언급했듯이, RSBM의 성능은 복구 노드의 입력 네트워크에 병목현상이 발생한다. 이는 MMS 노드의 수에 의해 제한된다. 이 문제를 해결하기 위해 파이프라인 컴퓨팅 기반의 새로운 복구 시스템을 제안한다. 제안하는 복구 시스템을 RS-LPM(Recovery System based on Linear Pipeline Mechanism) 이라고 나타내도록 한다. 제안된 방법은 복구 과정에 필요한 네트워크 트래픽을 모든 정상 MMS 노드들에 분산하고 MMS 노드의 가용한 CPU 자원을 활용한다. (그림 10)은 RS-LPM의 구조와 VODCA 서버의 비디오 블록의 흐름을 나타낸다. (그림 10)에서 RS-LPM은 복구 과정에 필요한 네트워크 트래픽 뿐만 아니라 모든 MMS 노



(그림 10) RS-LPM 의 구조와 비디오 블록의 흐름

드에 배타적 OR 연산을 분산한다.

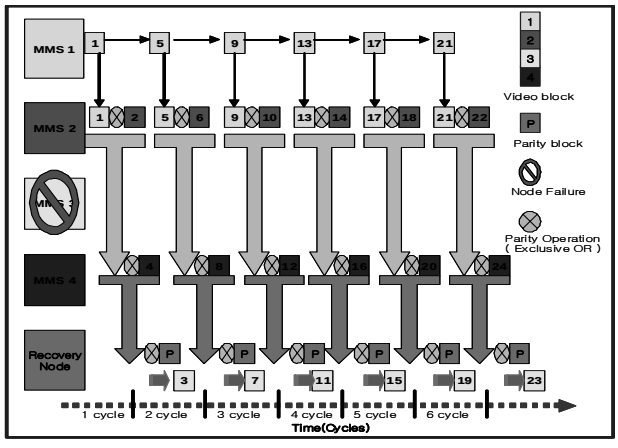
MMS 노드 오류 발생시 모든 정상 MMS 노드는 자신의 비디오 블록을 복구 노드에 직접 전송하지 않고 저장하고 있던 비디오 블록이나 배타적 OR 연산의 결과 블록을 이웃 MMS 노드에 전송한다. 각 MMS 노드는 로컬 디스크로부터 인출한 비디오 블록과 이웃 MMS 노드로부터 수신한 블록으로 배타적 OR 연산을 수행한다. 이웃 MMS 노드로부터 수신된 블록은 디스크에 저장되어 있는 원본 비디오 블록이거나 다른 이웃 MMS 노드에서 배타적 OR 연산에 의해 생성된 결과이다. 계산된 결과는 이웃 MMS 노드에게 보내진다[16].

마지막으로, 복구 노드는 모든 MMS 노드에서 계산된 결과와 패리티 블록으로 마지막 배타적 OR 연산을 수행하여 오류가 발생한 MMS 노드의 비디오 블록을 복원하여 외부 네트워크를 통해 클라이언트로 전송한다. 예를 들면 (그림 10) 과 같이 MMS 3노드가 오류 발생시, MMS 1노드는 비디오 블록 1을 MMS 2로 전송한다. MMS 2 노드는 비디오 블록 1과 2를 가지고 배타적 OR 연산을 수행한 후 비디오 블록 4와 배타적 OR 연산을 수행하기 위해 MMS 4로 전송된다. 마지막으로 모든 정상 비디오 블록을 위한 배타적 OR 연산이 끝난 후, 그 결과는 복구 노드로 전송된다. 복구 노드는 패리티 블록으로 배타적 OR 연산을 수행하여 비디오 블록 3을 생성한다.

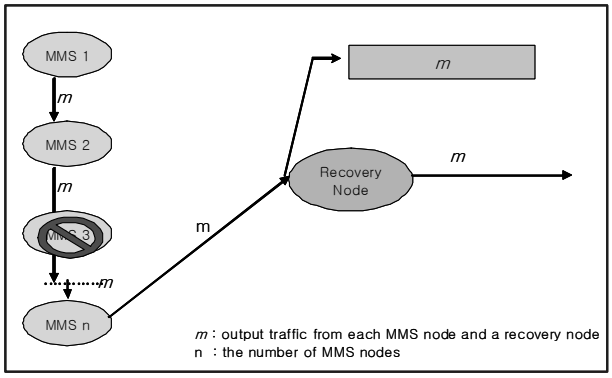
4.2 RS-LPM 의 특징

(그림 11)은 RS-LPM의 파이프라인 개념에 따른 복구 과정을 보여준다. (그림 11)에서, 매 사이클 최소 한번 배타적 OR 연산을 수행, 인출, 전송 과정이 발생하는 것은 명령어의 병렬처리에서 파이프라인 기술과 흡사하다[16]. 오류 블록을 복원하는 이러한 병렬 처리는 제안된 RS-LPM에서 성능을 향상된다. 이 그림에서 오류가 발생한 MMS 3 노드는 3, 7, 11, 15, 19, 23의 비디오 블록을 가진다. 이들 블록은 파이프라인 컴퓨팅에 따라서 복구 노드에서 매 사이클 마다 복원된다.

RS-LPM에서 복구 노드는 각 사이클 동안 한번 배타적



(그림 11) RS-LPM에서 파이프라인 개념 기반의 복원 단계



(그림 12) RS-LPM 의 네트워크 트래픽 모델

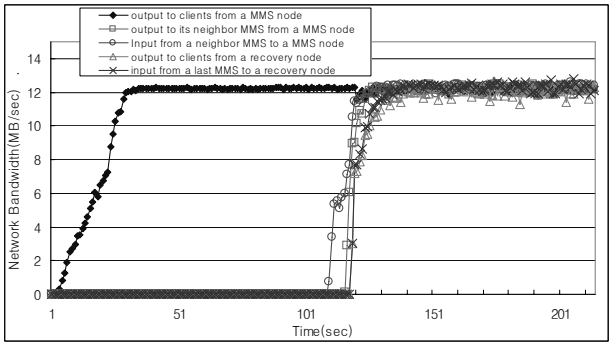
OR 연산을 실행한 후 결과를 클라이언트로 전송한다. 제안된 시스템은 배타적 OR 연산을 위한 컴퓨팅 부하 뿐만 아니라 네트워크 트래픽을 모든 MMS 노드로 분산한다. 복구 노드의 입력 네트워크 트래픽은 MMS 노드 하나의 출력 트래픽과 같다. (그림 12)는 RS-LPM에서의 네트워크 트래픽을 보여준다. 각 MMS 노드의 출력 네트워크 트래픽은 m으로 같다. 복구 노드와 MMS 노드는 내부 네트워크의 성능을 최대로 사용한다. 그림에서와 같이 n-1 MMS 노드가 살아있고 각 출력 트래픽이 m 이라도 복구 노드의 입력 네트워크 트래픽은 m 이 된다. RSBM 과 달리, 복구 노드의 입력 네트워크 트래픽은 MMS 노드의 수에 의존적이지 않아서 RS-LPM은 복구 노드 네트워크의 입력 포트에서 병목현상이 일어나지 않는다.

4.3 RS-LPM의 성능

3.3절에서 설명된 실험과 같은 환경으로 RS-LPM의 성능을 측정하였다. RS-LPM의 복구 노드에서는 네트워크 병목현상이 발생하지 않기 때문에 12MB/sec의 네트워크 부하 환경에서 실험을 수행하였다. 12MB/sec의 출력 네트워크 트래픽은 4 MMS 노드가 256개의 클라이언트에게 QoS 스트림을 제공함을 의미한다.

(그림 13)은 12MB/sec의 네트워크 부하가 발생할 때, MMS 노드와 복구 노드에서의 네트워크 트래픽을 나타낸다. MMS 노드의 오류는 시간축의 120초 에서 발생한다. 위 그림에서 이웃 MMS 노드로의 출력 트래픽은 사각형으로 나타내었다. RS-LPM에서 현재 MMS 노드가 마지막 MMS 노드가 아니라면 자신의 비디오 블록이나 배타적 OR 연산으로 계산된 결과 블록을 이웃 MMS 노드로 전송한다. 원으로 표시된 부분에서 이웃 MMS 노드로부터의 입력 트래픽이 출력 트래픽과 거의 같음을 알 수 있다. 마지막 MMS 노드로부터의 입력 트래픽은 12MB/sec 에 가까워 지며 복구 노드 또한 12MB/sec 의 비율로 비디오 블록을 복구할 수 있다. 그 후 복구 노드는 복구된 비디오 블록을 클라이언트로 전송한다. 그림에서 삼각형으로 표시된 부분에서 복구 노드의 복구된 블록의 출력 트래픽은 12MB/sec 가 된다.

RSBM과 비교했을 때, RS-LPM은 같은 동작 환경에서



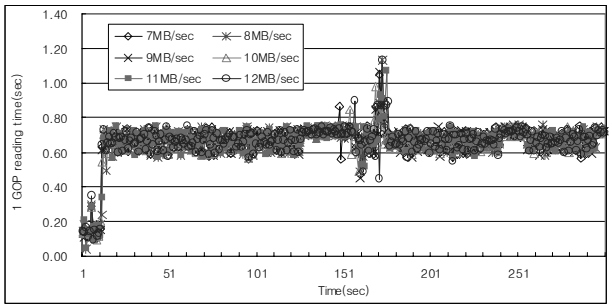
(그림 13) 12MB/s 부하에서 MMS노드와 복구노드의 네트워크 트래픽

두배의 스트림을 제공하는 좀 더 높은 성능을 나타내었다. RS-LPM에서의 메모리 스와핑 문제는 메모리를 추가함으로써 간단히 해결할 수 있다. 메모리를 확보한 추가적인 실험으로, MMS 노드로부터 출력 네트워크 트래픽이 최대 12MB/sec에 도달하는 것을 확인하였다. 하지만 메모리의 크기를 증가시켜도, RSBM에서 내부 네트워크의 병목 현상은 피할 수 없었다. RS-LPM 이 MMS 노드의 가용한 CPU 자원을 사용하고 모든 MMS 노드들이 복구 과정에 참여하기 때문에 클러스터 기반의 VOD 서버에서 MMS 노드의 오류 발생시 좀 더 많은 클라이언트에게 끊기지 않는 스트리밍 서비스를 제공하는 개선된 성능을 제공한다.

(그림 14)는 클라이언트에서 1 GOP를 읽는데 걸리는 시간을 나타낸다. 본 실험은 7MB/sec와 12MB/sec 사이의 부하 환경에서 수행되었다. RS-LPM은 이러한 네트워크 트래픽 부하를 지원할 수 있다. MMS 노드의 오류는 120초 에서 발생하였다. 서버와 클라이언트 사이에 네트워크 지연이 있기 때문에 클라이언트에서 읽는 시간의 변화는 148초와 176초 사이에 발생하였다. 불안정 상태 후에 읽기 시간은 1.5초 에서 안정 상태로 변한다. 변화는 28초간 지속되었다.

RSBM과 비교 하면, 변동 기간은 매우 짧다. 복구 과정이 모든 MMS 노드로 분산되었기 때문에 복구 노드는 상대적으로 짧은 시간 안에 복구된 비디오 블록을 전송할 수 있다. (그림 9)에서와 같이 MMS 오류가 발생한 후에 RSBM은 안정 상태로 돌아오기 위해 110초의 시간이 필요하다. RS-LPM의 변동 시간이 RSBM보다 4배 짧다.

게다가 (그림 14)에서와 같이 최대 읽기 시간과 최소시간



(그림 14) RS-LPM에서 클라이언트의 GOP reading time

의 차이는 1.68초 이다. 이러한 결과는 RSBM의 절반 수준이다. RS-LPM에서 변동 시간과 진폭은 RSBM보다 작다. 본 실험의 결과로 RS-LPM이 RSBM보다 MTTR 값이 더 작음을 알 수 있다[3, 4].

5. 결론 및 향후 연구

VOD (Video-On-Demand)는 미디어 스트리밍 서비스를 위한 대표적인 기술이며 많은 분야에서 연구되어 왔다. 하지만 성공적인 VOD 서비스를 위해 VOD 서버에서 부분적인 오류가 발생하더라도 사용자가 허용하는 MTTR값 안에서 모든 클라이언트에게 미디어 스트리밍 서비스를 보장해야 한다. VOD 서비스는 주어진 시간 안에 미디어를 인출, 전송, 디스플레이를 끝내야만 한다. 오류 상태에서 VOD 서비스를 제공하기 위해, 미디어 스트리밍의 특징이 복구 방법에 반영되어야 한다. 본 논문에서는 클러스터 기반의 VOD 서버에서 MMS 노드가 오류 상태가 될 때 QoS 스트림을 제공하기 위한 복구 방법에 관하여 연구 하였다.

실제 VOD 서비스에서 복구 시스템에 관해 연구하기 위해, 일반 PC와 내부 네트워크로 구성된 클러스터 기반의 VOD 서버를 구현하였다. 구현된 VOD 서버에서 패리티 연산 기반의 RSBM을 설계하였다. 하지만 RSBM에서 복구 노드의 입력 네트워크는 정상적인 MMS 노드들로부터 전송되는 비디오 블록으로 가득 차게 된다. 복구 노드로 전송되는 비디오 블록의 지연은 서비스 받는 클라이언트의 수를 감소와 비디오 스트림 품질의 악화를 야기시키게 된다. 게다가 RSBM은 MMS 노드의 비효율적인 CPU 사용을 나타낸다. 이러한 방법에서 MMS 노드는 단순히 비디오 블록의 인출 및 전송을 수행하기 때문에 평균 CPU 사용은 10% 이하로 측정된다.

이러한 문제를 해결하기 위해 MMS 노드들과 복구 노드 간에 파이프라인 기반의 RS-LPM 을 제안하였다. RS-LPM에서, 복구 노드는 복구된 비디오 블록을 생성하여 각 사이클에 한번 클라이언트로 전송한다. 이 방법은 명령어 실행 단계의 파이프라인 과정과 비슷하다. RS-LPM은 MMS 노드의 가용한 CPU 자원을 효율적으로 사용하기 위하여 모든 정상적인 노드가 손상된 비디오 블록을 복구하기 위한 복구 과정에 참여하게 한다. 이 파이프라인 컴퓨팅을 기반으로 RS-LPM은 배타적 OR 연산을 위한 컴퓨팅 부하 뿐만 아니라 모든 MMS 노드의 네트워크 트래픽을 분산시킨다. 본 실험으로부터 복구 노드의 입력 네트워크 트래픽이 마지막 MMS 노드에 의해 야기되는 출력 트래픽과 같다는 것을 알 수 있다. MMS 노드의 오류 상태 에서도 RS-LPM은 RSBM과 비교해 최소 두배의 QoS 스트림을 제공하는 개선된 성능을 나타낸다.

VOD 서비스의 중요한 특징 중 하나는 끊김과 지터, 순서가 맞지 않는 프레임의 미디어 스트리밍은 의미가 없다는 것이다. 이러한 요구사항은 VOD 서버의 부분적인 오류 상태에서도 마찬가지이다. 이 특성을 만족시키기 위해 오류

발생 후의 오류 복구 시간은 짧아야 한다. 클라이언트의 GOP 읽기 시간에서도 RS-LPM은 RSBM에 비해 4배 좋은 성능을 나타내었다. 상대적으로 짧은 복구 시간으로 인해, 미디어 스트리밍 서비스는 빠르게 정상 상태로 변하게 된다. 결과적으로, RS-LPM은 좀 더 나은 MTTR 값을 가지게 된다.

참 고 문 헌

[1] Dinkar Sitaram, Asit Dan, "Multimedia Servers: Applications, Environments, and Design," Morgan Kaufmann Publishers, 2000.

[2] http://www.mpeg.org

[3] Armando Fox, David Patterson, "Approaches to Recovery Oriented Computing," IEEE Internet Computing, Vol.9, No.2, pp.14-16, 2005.

[4] Dong Tang, Ji Zhu, Roy Andrada, "Automatic Generation of Availability Models in RAScard," IEEE International Conference of Dependable Systems and Networks, June, 23-26, pp.488-494, 2002.

[5] T. Chang, S. Shim, and D. Du, "The Designs of RAID with XOR Engines on Disks for Mass Storage Systems," IEEE Mass Storage Conference, March, 23-26, pp.181-186, 1998.

[6] Prashant J. Shenoy, Harrick M. Vin, "Failure recovery algorithms for multimedia servers,"Multimedia Systems, 8: pp.1-19, Springer-Verlag, 2000.

[7] Jack Y.B. Lee, "Supporting Server-Level Fault Tolerance in Concurrent-Push-Based Parallel Video Servers,"IEEE transactions on Circuits and Systems for Video Technology, Vol.11, No.1, pp.25-39, January, 2001.

[8] Jamel Gafsi, Ernst W. Biersack,"Modeling and Performance Comparison of Reliability Strategies for Distributed Video Servers,"IEEE Transactions on Parallel and Distributed Systems, Vol.11, No.4, pp.412-430, 2000.

[9] 서동만, 방철석, 이좌형, 김병길, 정인범, "리눅스 기반의 클러스터 VOD 서버와 내장형에 클라이언트의 구현", 정보과학회 논문지 제10권 제6호 pp.435-447, 2004

[10] Jung-Min Choi, Seung-Won Lee, Ki-Dong Chung, "A Muticast Delivery Scheme for VCR Operations in a Large VOD System," 8th IEEE International Conference on Parallel and Distributed Systems, pp.555-561, June, 26-29, 2001.

[11] D.A. Patterson, G. Gibson, and R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks(RAID),"Proceedings of the 1988 ACM Conferences on Management of Data, pp.109-116, June, 1988.

[12] M. Holland, G.Gibson, and D. Siewiorek, "Architectures and algorithms for on-line failure recovery in redundant disk arrays," Journal of Distributed and Parallel Databases, Vol.2, pp.295-335, 1994.

[13] Brian K. Schmidt, Monica S. Lam, J. Duane Northcutt, "The interactive performance of SLIM: a stateless, thin-client architecture," ACM SOSP'99, pp.31-47, 1999.

[14] W.C. Feng and M. Lie, "Critical Bandwidth Allocation Techniques for Stored Video Delivery Across Best-Effort Networks,"20th International Conference on Distributed Computing Systems, pp.201-207, April, 2000.

[15] Jung-Min Choi, Seung-Won Lee, Ki-Dong Chung, "A Muticast Delivery Scheme for VCR Operations in a Large VOD System," 8th IEEE International Conference on Parallel and Distributed Systems, pp.555-561, June, 26-29, 2001.

[16] David A. Patterson and John L. Hennessy, "Computer Organization & Design," pp.392-490, Morgan Kaufmann, 1998.

[17] Nabil J. Sarhan, Chita R. Das, "Caching and Scheduling in NAD-Based Multimedia Servers," IEEE Transactions on PARALLEL AND DISTRIBUTED SYSTEMS, Vol.15, No.10, pp.921-933, 2004.

[18] Sang-Ho Lee, Kyu-Young Whang, Yang-Sae Moon, Wook-Shin Han, "Dynamic Buffer Allocation in Video-on-Demand Systems," IEEE Transactions on PARALLEL AND DISTRIBUTED SYSTEMS, Vol.15, No.6, pp.1535-1551, 2003.

[19] Sooyong Kang, Heon Y. Yeom, "Modeling the Caching Effect in Continuous Media Servers," Multimedia Tools and Applications, 23(3), pp 203-224, 2003.

[20] J. Gafsi and E.W. Biersack, "Data Striping and Reliablity Aspects in Distributed Video Servers," In Cluster Computing: Networks, Software Tools, and Applications, 2 (1): pp.75-91, February, 1999.

[21] W.J. Bolosky, R.P. Pitzgerald, J.H. Draves,"Distributed schedule management in the Tiger video fileserver," Proceedings of the sixteenth ACM symposium on Operating systems principles, Saint Malo France, October, 05-08, pp.212-223, 1997.

이 좌 형

e-mail : jinnie4u@kangwon.ac.kr

2003년 강원대학교 정보통신공학과(공학사)

2005년 강원대학교 컴퓨터정보통신공학과 (공학석사)

2005년~현 재 강원대학교 컴퓨터정보통신공학과(박사과정)

관심분야 : 멀티미디어 시스템, 센서 네트워크

정 인 범

e-mail : ibjung@kangwon.ac.kr

1985년 고려대학교 전자공학과(학사)

1985년~1995년 ㈜삼성전자 컴퓨터 시스템 사업부 선임 연구원

1992년~1994년 한국과학기술원 정보통신공학과(석사)

1995년~2000년 한국과학기술원 전산학과(박사)

2001년~현 재 강원대학교 컴퓨터정보통신공학전공 교수

관심분야 : 운영체제, 소프트웨어 공학, 멀티미디어 시스템, 센서 네트워크