

효과적인 트랜스코딩 부하 분산을 위한 자원 가중치 부하분산 정책

(Resource Weighted Load Distribution Policy for Effective Transcoding Load Distribution)

서 동 만 [†] 이 좌 형 [†] 최 면 옥 ^{**} 김 윤 ^{***} 정 인 범 ^{***}
 (Dongmahn Seo) (Joahyoung Lee) (Myunuk Choi) (Yoon Kim) (Inbum Jung)

요 약 최근 무선통신 기술의 발전으로 PC뿐만 아니라 PDA, 휴대폰 등 다양한 이동 단말 장치를 통하여 멀티미디어 서비스를 제공받을 수 있게 되었다. 이동 단말 장치는 제한된 하드웨어 성능을 갖는 것 이외에 무선망에서 동작하므로 낮은 네트워크 대역폭을 사용하게 된다. 이러한 이동 단말기의 특성을 고려하여 스트리밍 미디어 서비스를 받기 위해서는 동작 환경에 적합하게 미디어를 트랜스코딩 하는 기술이 필요하다. 미디어에 대한 트랜스코딩은 트랜스코딩 서버들에서 이동 단말기 등급별로 수행되어 스트리밍 미디어의 실시간 전송 요구사항에 맞추어 사용자에게 보내져야 한다. 대규모의 이동 단말 사용자들에게 각각에 맞는 트랜스코딩 된 스트리밍 미디어를 QoS 하기 위해서는 트랜스코딩 서버들의 부하분배 정책에 미디어의 등급별 특성을 반영하는 것이 필요하다. 본 논문에서는 이동단말들의 등급별 자원가중치 및 트랜스코딩 서버에서의 등급별 최대 사용자 수에 기반 한 자원가중치 기반 부하분산 정책을 제안한다. 제안된 정책을 클러스터 기반 시스템에서 구현하여 트랜스코딩 서버들 사이에 부하배분이 균등하게 이루어짐을 확인하고 트랜스코딩 서버 개수의 증가에 따른 선형적 성능확장성을 평가한다.

키워드 : 트랜스코딩, 부하 분산, QoS, 클러스터, MPEG, 확장성

Abstract Owing to the improved wireless communication technologies, it is possible to provide streaming service of multimedia with PDAs and mobile phones in addition to desktop PCs. Since mobile client devices have low computing power and low network bandwidth due to wireless network, the transcoding technology to adapt media for mobile client devices considering their characteristics is necessary. Transcoding servers transcode the source media to the target media within corresponding grades and provide QoS in real-time. In particular, an effective load balancing policy for transcoding servers is inevitable to support QoS for large scale mobile users. In this paper, the resource weighted load distribution policy is proposed for a fair load balance and a more scalable performance in cluster-based transcoding servers. Our proposed policy is based on the resource weighted table and number of maximum supported users, which are pre-computed for each pre-defined grade. We implement the proposed policy on cluster-based transcoding servers and evaluate its fair load distribution and scalable performance with the number of transcoding servers.

Key words : transcoding, load distribution, QoS, cluster, MPEG profile grade, scalability

1. 서 론

최근 멀티미디어와 정보통신망의 발전에 따라 영상 정보 서비스에 대한 요구가 날로 다양해지고 있다. 멀티미디어 서비스의 급속한 발전으로 사용자는 유선망 이

외에 무선망을 통하여 무선 이동 단말기로 스트리밍 미디어를 전송하고 재생하는 서비스를 받을 수 있게 되었다. 그러나 스트리밍 서비스를 위해서는 영상 정보의 양이 다른 일반 데이터의 정보량에 비하여 매우 크기 때

· 이 논문은 강원대학교 두뇌한국21사업에 의해 지원되었음

[†] 학생회원 : 강원대학교 컴퓨터정보통신공학과

dmseo@snslab.kangwon.ac.kr

jhlee@snslab.kangwon.ac.kr

^{**} 정 회원 : 신한생명 사이버 마케팅팀 주임

dreammak@shinhanlife.co.kr

^{***} 정 회원 : 강원대학교 전기전자정보통신공학부 교수

yooni@snslab.kangwon.ac.kr

ibjung@snslab.kangwon.ac.kr

논문접수 : 2005년 3월 3일

심사완료 : 2005년 8월 2일

문에 광대역 네트워크 대역폭 및 고성능의 컴퓨터를 필요로 하고 있다[1-3].

무선망에서는 네트워크 대역폭이 유선망보다 상대적으로 열악한 환경을 가지고 있으며, 이동 단말기의 낮은 컴퓨팅 파워와 시스템 자원은 서버로부터 전송되는 높은 품질의 스트리밍 미디어를 적절하게 처리할 수가 없다. 이러한 문제점을 해결하기 위하여 최근에 스트리밍 미디어를 이동 단말기에 적합한 품질로 바꾸는 트랜스코딩 기술이 연구되고 있다[4,5]. 트랜스코딩은 멀티미디어 콘텐츠를 최초 인코딩된 형태에서 목적하는 단말에 적합하게 변환시키는 기술이다. 변환 종류로는 단말 사양에 맞게 스트리밍 미디어의 프레임율, 해상도, 디스플레이 크기, 비트율의 조절을 포함하여 MPEG-1, 2 미디어를 MPEG-4 미디어로 변환시키는 요구도 포함한다. 또한 정지 화상의 경우 JPEG을 GIF 형식으로 변용하는 작업을 수행하기도 한다.

트랜스코딩 시스템은 인코딩된 영상 데이터를 가지고 있는 멀티미디어 서버와 영상 데이터를 단말 환경에 맞게 변환하는 작업을 수행하는 트랜스코딩 서버로 구성된다. 이동 단말기에서 스트리밍 미디어 서비스 요청을 하게 되면 멀티미디어 서버에 저장중인 영상 미디어를 사용자 단말 환경에 적합한 형태로 변경하기 위하여 트랜스코딩 서버로 전송한다. 트랜스코딩 서버는 전송된 영상 미디어를 이동 단말 환경에 적합한 형태의 스트리밍 미디어로 바꾸어 스트리밍 서비스를 한다.

최근에 영상 데이터의 압축 및 복원에 탁월한 기능을 제공하는 MPEG 미디어가 스트리밍 미디어 서비스를 위하여 사용되어진다. MPEG 미디어 스트리밍은 사용자의 무선 환경에 따라 여러 등급으로 나누어 정의되어진다. 즉, 각 등급에 따라서 CPU, 네트워크 대역폭, 메모리 등의 요구되는 컴퓨팅 자원의 소모량이 다르다. 트랜스코딩 작업은 미디어 데이터의 형식과 크기 등을 변환하기 때문에 서버의 자원 소모가 많은 특징을 가지고 있다. 특히, 트랜스코딩 하고자 하는 스트리밍 미디어의 등급에 따라 서버의 자원 사용량은 증속되므로 특정 서버에 높은 등급의 트랜스코딩 부하가 집중될 경우 전체 서버들의 활용성이 떨어지며, 또한 성능 확장성도 보장되지 않으므로 효과적인 트랜스코딩 부하 분배 알고리즘 연구가 필요하다.

본 논문에서는 트랜스코딩 시스템의 부하분산 정책을 가격 대 성능비가 뛰어난 리눅스 기반 클러스터 시스템에서 연구하였다. 구성된 트랜스코딩 시스템은 부하분배 서버, 트랜스코딩 서버 그리고 모바일 단말기로 이루어졌다. 부하분배 서버는 채택된 부하 분배 정책에 따라서 트랜스코딩이 필요한 사용자들의 요구를 트랜스코딩 서버들에게 분배한다. 반면에 트랜스코딩 서버는 무선 이

동 단말기에 대한 MPEG 등급에 기준하여 사용자들이 요구한 트랜스코딩 요청을 수행하고, 트랜스코딩 처리된 미디어를 사용자에게 스트리밍 전송하는 역할을 수행한다. 본 논문에서는 트랜스코딩 서버가 초기에 인코딩된 MPEG 영상 데이터를 저장하는 멀티미디어 서버 역할도 같이하는 것으로 간주하였다.

클러스터 시스템을 웹 서버로 사용하는 경우 대표적인 부하 분배 정책으로는 RR(Round Robin), WRR(Weighted Rounded Robin), WLC(Weighted Least Connection)정책들이 있다. 이런 정책들은 클러스터 시스템을 웹 서버로 사용할 때 일반적으로 채택되어 사용되고 있다. 그러나 스트리밍 미디어는 실시간성 서비스를 사용자들에게 보장하여야 하므로 스트리밍 미디어 이외에 다양한 멀티미디어 데이터를 서비스하는 웹서버에서 사용하는 부하분배 정책으로는 좋은 성능 및 효율적 자원 사용 목적을 달성할 수 없다.

트랜스코딩 시스템에서 스트리밍 미디어의 QoS를 위한 실시간적 요구사항은 원하는 품질의 스트리밍 미디어로의 변환 및 사용자로의 전송이 제한된 시간에 정확히 맞추어 공급되는 것을 의미한다[3,6]. 즉, 제한된 시간보다 늦게 수행 완료되어도 문제이지만 빨리 수행 완료하여도 제공되는 스트리밍 미디어의 품질에는 변화가 없으므로 바람직한 현상은 아니다. 이런 현상을 해결하기 위하여 본 논문에서는 클러스터 기반 트랜스코딩 시스템에서 스트리밍 미디어 서비스의 실시간적 요구사항을 만족시키며 최대의 성능 확장성을 제공하는 RWLD(Resource Weighted Load Distribution) 정책을 제시한다. RWLD 정책에서는 각각의 트랜스코딩 서버들에서 사용자의 이동 단말기 등급별로 트랜스코딩에 필요한 실제 자원 소모량을 측정하여 부하 분배를 위한 가중치로 사용한다. 또한 각각 트랜스코딩 서버들에서 등급별 QoS 규정을 만족하며 트랜스코딩 할 수 있는 최대 사용자수를 측정하여 부하 분배 정책에 반영하므로 스트리밍 미디어의 실시간적 특성이 반영되도록 하였다.

본 논문에서는 제안된 RWLD 정책의 성능평가를 위하여 기존의 부하 분산 알고리즘인 RR 이외에 WRR의 정적 가중치로 인한 부하 불균형 문제를 보완한 DWRR(Dynamic Weighted Round Robin) 정책을 구현하여 각각의 부하분산 정책들의 성능 및 부하 배분의 균형성을 평가하였다. 제안된 RWLD 정책은 트랜스코딩 시스템의 동작 전에 트랜스코딩 등급별 자원소모량 및 최대 사용자수를 계산하기 위한 부담을 가지고 있다. 그러나 RWLD 정책은 트랜스코딩 등급별 가중치 값을 부하분산에 이용하므로 트랜스코딩 서버들 사이의 균형적 부하배분을 제공할 수 있으며, 또한 제한된 시간 이내에 해당 트랜스코딩을 완료하기 위하여 트랜스코딩 서버의

최소 CPU 사용률을 배정하므로 트랜스코딩 시스템의 성능 확장성 향상에 기여한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에 관련연구에 대하여 설명한다. 3장에서는 등급별 트랜스코딩에 필요한 자원 소모량을 측정 및 제안된 RWLD 부하분배 정책에 대하여 설명한다. 4장에서는 실험에 사용된 클러스터 시스템의 환경을 설명하고, 5장에서는 구성된 클러스터 시스템에서 제안된 정책에 대한 실험 및 성능 평가를 하며 6장에서는 본 논문의 결론을 맺고 향후 연구 계획을 설명한다.

2. 관련 연구

2.1 무선망 대역폭

유선망에서의 서비스되는 스트리밍 미디어를 무선망의 환경에서 이동 단말기를 통하여 공급하기 위해서는 무선망의 낮은 대역폭 및 낮은 안정성이 고려되어야 한다. 특히 다양한 무선망들을 통하여 기존에 인코딩된 스트리밍 미디어를 트랜스코딩 하여 스트림 전송하기 위해서는 무선망 규격에 대한 조사가 있어야 한다.

무선 통신의 무선 랜 표준규약인 802.11은 IEEE 작업 그룹이 개발한 무선 랜을 위한 규격으로 802.11, 802.11a 802.11b, 802.11g 등 4가지 규격이 이에 속한다 [7]. 802.11은 기초적인 무선 랜 표준안으로 CSMA/CA를 지원하고 최대 속도는 2Mbps이다. 802.11b는 11Mbps의 최대 속도를 제공하는 무선 랜 표준이다. 많은 이동 단말기 제품이 802.11b의 표준을 지원하고 있다. 802.11g와 802.11a는 최대 54Mbps까지 데이터 전송을 지원하는 표준으로 차세대 무선 랜 지원하기 위한 표준이다. 본 논문에서는 802.11과 802.11b의 규격 속도를 기준으로 연구를 한다.

2.2 MPEG profile

이동 단말기에 따른 컴퓨팅 파워, 메모리, 네트워크 대역폭이 각각 틀리기 때문에 스트리밍 미디어 서비스 역시 단말기 환경에 따라 달라져야 한다. 이러한 이동 단말의 등급에 따른 MPEG 미디어의 표준 규격이 이미 제시되어있다[8]. 표 1은 스트리밍 미디어 서비스 환경에 따른 Video size, Frame rate, Bit rate 항목을 기반으로 한 MPEG profile을 보여주고 있다.

표 1을 보면 MPEG 등급에 따른 규격이 이동 단말기

표 1 MPEG 등급에 따른 프로파일

등급	Video size	Frame rate	Bit rate(KB/s)
SQCIF	128×96	15	50
QCIF	176×44	15	70
CIF	352×288	26	100
4CIF	704×576	30	200

환경에 따라 화면크기나 비트율이 다르게 설정됨을 알 수가 있다. MPEG 프로파일을 살펴보면 SQCIF(Sub-Quater Common Intermediate Format), QCIF(Quater Common Intermediate Format), CIF, 4CIF급으로 나눌 수가 있다. SQCIF급은 모바일 핸드폰, QCIF급은 PDA, CIF급은 무선 노트북, 4CIF급은 일반 PC급이다.

2.3 트랜스코딩 시스템

일반적인 트랜스코딩 시스템의 구조는 그림 1과 같다. 사용자는 트랜스코딩에 필요한 정보를 트랜스코딩 서버에 전송을 한다. 트랜스코딩 서버에서는 요구된 미디어 스트림의 원본을 미디어서버에서 읽어 사용자가 요구한 영상크기, 비트율, 프레임율에 따라서 트랜스코딩한 후 사용자에게 전송한다. 예를 들면 트랜스코딩 서버로부터 CIF(352X288)등급의 25프레임/초, 비트율 100kbps의 비디오 스트리밍을 QCIF(176X44)등급의 15프레임/초, 비트율 50kbps의 스트리밍으로 사용자에게 전송할 수가 있다. 트랜스코딩 서버에서는 무선망의 특성에 적합하도록 비트율, 해상도, 프레임율을 가변적으로 줄일 수가 있다.

기존의 트랜스코딩 시스템을 구축하는 방법들로는 소스기반 정적 인코딩 시스템, 정적 트랜스코딩 서버 할당 방식, 부하분산 트랜스코딩 서버 방식이 있다[9]. 그림 2는 소스기반 정적 인코딩 시스템을 보여준다. 이

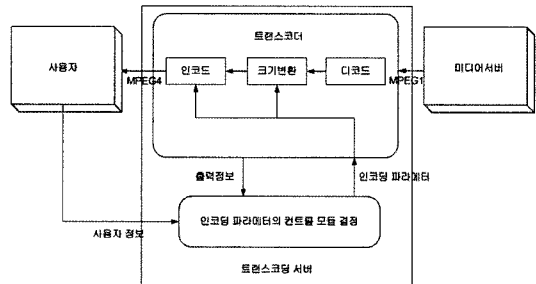


그림 1 트랜스코딩 시스템 구조

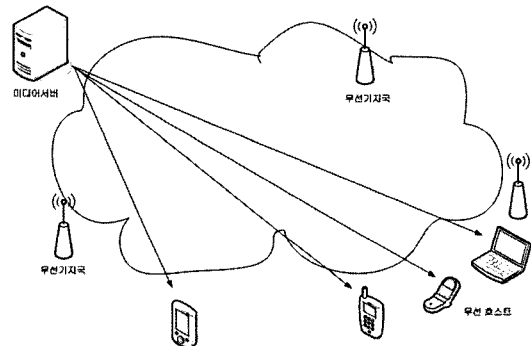


그림 2 소스기반의 정적 인코딩 시스템

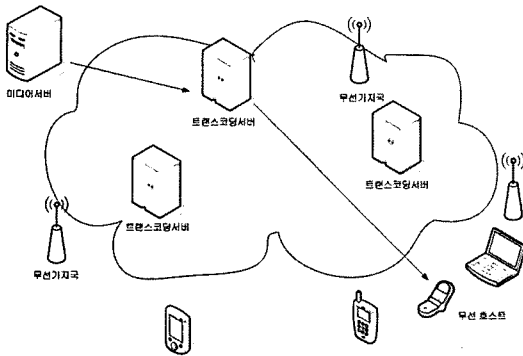


그림 3 정적 트랜스코딩서버 할당

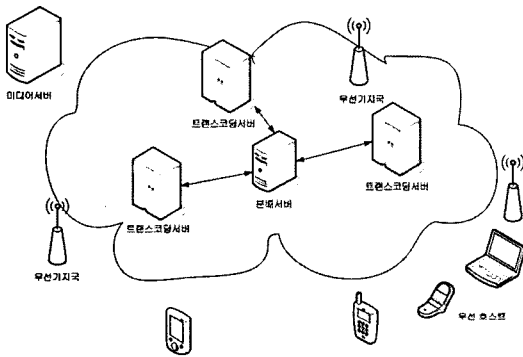


그림 4 부하분산 트랜스코딩 시스템

방식은 사용자의 요구에 대하여 미디어 파일들을 미리 사용자 등급별로 트랜스코딩 하여 서버에 저장하여 사용하는 방식이다. 이러한 시스템은 스트리밍 미디어를 실시간으로 트랜스코딩 하여 전송하는 방식보다 서버의 부하가 심하지 않다는 장점이 있으나 무선 랜 환경에서 사용자의 이동에 의한 네트워크 변화에 적응할 수 없다는 단점이 있다. 또한 모든 미디어 파일들을 각 등급별로 트랜스코딩 하여 각각의 다른 미디어 파일들로 저장해야 한다는 부담이 있다.

그림 3은 정적 트랜스코딩 서버 할당 방식을 보여주고 있다. 이 방식은 이동 단말기를 사용하는 사용자와 연결된 무선기지국에서 가장 가까운 트랜스코딩 서버를 선택하고 스트리밍 서비스를 받게 된다. 이 방법은 트랜스코딩 서버를 선택하는 조건으로 사용자와의 근접도 이외에 서버를 선택하는 기준이 없기 때문에 특정 트랜스코딩 서버에 부하가 집중될 수 있는 단점이 있다.

정적 트랜스코딩 서버 방식의 문제점인 특정 서버에 부하가 집중 되는 부하 불균형을 피하기 위하여 분배 서버를 두고 트랜스코딩 서버들의 부하 상태를 파악하여 트랜스코딩 요구들을 처리하는 부하분산 트랜스코딩 시스템 방식이 있다. 그림 4에 나타나듯이 이 방식에서

는 분배서버는 트랜스코딩 서버들과 연동하여 정해진 부하분배 정책에 따라서 트랜스코딩 서버를 선택하게 된다. 전통적으로 클러스터 시스템에서는 부하 분산 정책으로는 RR, WRR, DWRR, WLC 알고리즘이 사용되고 있다[9,10].

2.4 부하 분산 정책

트랜스코딩 시스템은 최종적으로 실시간적 특성을 가지고 있는 스트리밍 미디어를 사용자에게 전송하는 요구 사항을 만족하여야한다. 반면에 웹서버는 다양한 스트리밍 미디어 이외에 다양한 웹 프로그램들에 대한 서비스를 하므로 실시간적 기능을 만족시키기가 어렵다. 그러나 트랜스코딩 시스템에서 부하 배분을 위한 연구들은 아직 미미하므로 기존의 클러스터 웹서버 시스템에서 사용하는 부하분산 정책을 참조하고자한다. 기존에 사용되는 부하분산 정책들로는 RR(Round Robin), LC(Least Connection), WRR(Weighted Round Robin), DWRR(Dynamic Weighted Round Robin) 등이 있다[11-13].

2.4.1 Round Robin

라운드로빈 방식은 사용자 요청이 들어오면 차례대로 각 서버에게 할당하는 방식을 말한다. 라운드로빈 방식은 우선순위의 개념이 없고 실제서버의 연결 개수나 반응시간 등은 고려를 하지 않는다. 스케줄링의 기초는 사용자 기반이며 실제 서버들의 상태를 고려하지 않기 때문에 효과적인 부하 분산을 기대하기 힘들다. 따라서 서버들 사이에 부하 불균형이 심각해지는 현상이 발생할 수 있다.

2.4.2 Least-Connection Scheduling

최소 접속 스케줄링은 가장 접속이 적은 서버로 요청을 직접 연결하는 방식을 말한다. 각 서버에서 동적으로 실제 접속한 숫자를 누적하여 부하분산에 사용하므로 동적인 스케줄링 알고리즘중의 하나이다. 이 방식은 스트리밍 미디어와 같은 긴 접속을 요구하는 사용자와 비교적 간단한 접속 요구를 하는 사용자가 섞여있는 동작 환경에서는 서버들의 부하 불균형을 가져올 수 있다. 특히 성능이 서로 다른 이종 노드들로 구성된 클러스터 서버 환경에서는 LC 같이 사용자 접속 숫자로 부하 배분을 수행하는 것은 서버들의 비효율적 사용 결과를 초래하기 쉽다.

2.4.3 Weighted Round-Robin Scheduling

가중치 기반 라운드 로빈 스케줄링은 실제 서버에 서로 다른 가중치를 설정하여 부하 분산에 사용하는 방식이다. 각 서버에 가중치를 임의로 부여할 수 있으며, 여기서 지정한 가중치 값을 통해 처리 순서를 정한다. 예를 들어 기본 가중치가 1이고 서버 A, B, C가 각각 4, 3, 2의 가중치를 갖는 경우 스케줄링 순서는 ABCABCABA가 된다.

2.4.4 트랜스코딩 서버의 부하 분산 정책

트랜스코딩 서버의 부하 분산 정책에 대한 연구로 기존의 웹 서버에 사용하던 라운드 로빈 방식과 그 방식을 트랜스코딩 서버에 적용한 스트립 기반의 라운드 로빈 방식, 적응적 부하 공유 정책 등이 있다[14]. 스트립 기반의 라운드 로빈 방식은 나머지 연산을 통하여 클러스터 내의 컴퓨터에게 사용자의 요청을 할당해 주는 방식으로 실제 클러스터 내의 서버들의 가용 자원을 고려하지 않아 효과적인 부하 분배를 하기 어렵다. 적응적 부하 공유 정책은 가중치를 부여 받은 클러스터 내 서버들에게 유사 난수 함수를 이용하여 사용자의 요청을 할당하는 방식이다. 각 서버의 가중치는 최초에는 동일한 값의 가중치를 부여받고, 서비스를 시작한 이후에는 서버의 자원 활용도에 대한 정보를 수집하여 결정된다. 기존의 정책들에 비하여 보다 동적으로 부하 분배를 수행하지만, 서버 자원의 활용도가 실제 서비스 되는 트랜스코딩 서버의 성능 향상의 제약 조건이 되는 자원에 대한 고려를 하지 않았기 때문에 좋은 성능을 기대하기 어렵다.

2.4.5 Dynamic Weighted Round-Robin Scheduling

DWRR은 WRR 정책의 단점을 보강하기 위하여 제안되었다. WRR 정책처럼 서버들이 고정된 가중치를 갖지 않고, 주기적으로 서버들의 상태정보를 파악하여 동적 가중치를 계산하여 부하 분배에 활용한다. 트랜스코딩 서버들의 정보를 측정하는 방식에는 2가지 방법이 제안되고 있다. 하나는 트랜스코딩 서버들의 상태 정보를 분배서버에게 주기적으로 보내는 방식과 다른 하나는 사용자들의 요구가 발생 할 때 분배 서버가 모든 트랜스코딩 서버들에게 상태정보를 요구하여 받는 방식이다. 후자의 경우 트랜스코딩 서버가 다운되었을 경우 그것을 인지하는 지연시간이 발생할 수 있고, 또한 사용자 연결 요청 횟수가 급격하게 늘어나면 분배서버와 트랜스코딩 서버들 사이에 과부하가 발생할 수 있다 [10,13,15].

본 논문에서는 1초마다 트랜스코딩 서버가 자신의 상태를 분배서버에게 전송하는 방식을 채택하였다. 트랜스코딩 작업은 CPU 부하를 크게 발생시키는 특징을 보이므로 분배 서버는 1초마다 트랜스코딩 서버로부터 전송된 CPU 사용량을 측정하여 부하분산 관리 테이블을 유지한다. 분배서버에서는 사용자 요청이 들어왔을 경우 가장 적은 CPU 사용량을 가진 트랜스코딩 서버에 부하를 보내어 트랜스코딩 서버들 간의 부하균형을 유지한다. 표 2는 DWRR을 채택한 분배서버에서의 부하분산 관리 테이블을 관리하는 예를 보여준다. 표 2에서 요청3이 들어왔을 시점에 서버 E의 CPU 사용량이 3.5%로 가장 낮으므로 분배서버는 요청3을 서버 E에 할당한다.

표 2 DWRR 방식에서 부하분산 관리 테이블 (CPU 사용량 기준)

구분	선택된 서버	서버 A	서버 B	서버 C	서버 D	서버 E	서버 F
요청1	A	3.2%	3.4%	3.6%	4.2%	3.5%	4.5%
요청2	B	23.6%	3.4%	3.6%	4.2%	3.5%	4.5%
요청3	E	23.6%	25.4%	3.6%	4.2%	3.5%	4.5%
요청4	C	23.6%	25.4%	3.6%	4.2%	44.7%	4.5%
요청5	F	23.6%	25.4%	23.6%	4.2%	44.7%	4.5%

3. 자원 가중치 기반 부하분배 정책

스트리밍 미디어의 서비스는 특성상 사용자들의 실시간적 요구사항 만족시켜야한다. 본 논문에서는 이런 요구사항을 만족시키며 서버의 성능 확장성을 개선한 자원 가중치 기반 부하 분배(RWLD : Resource Weighted Load Distribution) 정책을 제안한다. RWLD 정책이 적용되기 위해서는 우선적으로 트랜스코딩에 필요한 자원의 실제 소모량을 이동 단말기 등급별로 트랜스코딩 서버에서 측정되어야 한다. 실시간적 특성을 갖는 스트리밍 미디어 서비스 환경에서 제한된 시간 이내에 원하는 품질의 스트리밍 미디어로 변환 및 공급을 정확히 맞추기 위해 필요한 자원소모량을 사전에 알아내는 것이다. 그 후 측정된 등급별 자원소모량을 바탕으로 각각의 트랜스코딩 서버들에서 얻어 낼 수 있는 트랜스코딩 등급별 최대의 트랜스코딩 개수들을 얻어낸다.

분배 서버는 등급별 자원소모량 가중치 및 각각의 트랜스코딩 서버별로 처리할 수 있는 등급별 총 자원가중치 값을 바탕으로 트랜스코딩 서버들의 부하분배를 수행한다. RWLD 방식은 스트리밍 미디어의 실시간적 특성을 만족시키며 각각의 트랜스코딩 서버들의 최대의 트랜스코딩 트랜잭션 개수를 부하분배에 활용하므로 클러스터 시스템의 총체적 성능확장성에 기여한다.

3.1 MPEG 등급에 따른 트랜스코딩 자원 소모량 측정

트랜스코딩 요청 사용자들의 단말 등급에 따라서 소요되는 자원의 양을 조사하기 위하여 표 1에 제시된 MPEG 등급에 기준하여 트랜스코딩 시 소모되는 CPU, 네트워크, 메모리 자원들의 사용량을 측정을 하였다. 이 실험을 위하여 CPU 1.4GHz, 메모리 256M, 네트워크 100Mbps, 운영체제는 리눅스(커널 2.4.22)를 탑재한 PC를 사용하였고 트랜스코딩 프로그램으로는 ffmpeg를 사용하였다[16]. 실험에 사용된 총 10 편의 영화들에 대한 정보는 표 3과 같다. 모든 영화는 MPEG-4에 기준한 avi 파일들이며 실험을 진행하기에 충분한 상영시간을 보유하고 있다. 이들 10개의 4CIF급 영상 미디어를 서버에 저장된 원본 미디어로 간주하여 SQCIF, QCIF, CIF 급으로 각각 트랜스코딩 할 때 표 소요되는 자원의 양

표 3 실험에 사용한 MPEG 미디어

구분	영화제목	크기	초당프레임	상영시간
영화1	나루토	640×480	26	35분
영화2	냉정과열정사이	704×384	26	95분
영화3	토토로	640×480	26	30분
영화4	몽키턴	640×480	26	32분
영화5	미도리의 나날	640×480	26	30분
영화6	오늘부터 마왕	640×480	26	30분
영화7	천상전하	640×480	26	35분
영화8	연풍	640×480	26	35분
영화9	유고	640×480	26	30분
영화10	미루모데	640×480	26	30분

4 MPEG 등급별 자원 소모량

등급	CPU	Memory	Network
SQCIF	8.3%	5.7Mb	50kbps
QCIF	8.5%	5.8Mb	70kbps
CIF	16.3%	6.4Mb	100kbps

을 측정하였다.

실험 결과로 SQCIF, QCIF, CIF 등급으로 트랜스코딩을 할 때 트랜스코딩 서버에서 소모되는 자원의 크기는 원본 영화에 종속되지 않고 일정하다는 것을 관찰할 수 있었다. 표 4는 10개의 4CIF급 MPEG 미디어를 등급별로 트랜스코딩 될 때 CPU, 메모리, 네트워크 자원의 소모량을 보여준다. 표 4에 나타난 것처럼 트랜스코딩이 수행될 때 전체 시스템 자원들 중 CPU 자원 사용량이 상대적으로 큰 비중을 차지함을 알 수 있다. 측정된 값은 10개의 MPEG 미디어 각각에서의 값의 평균이며 표준 편차는 0.5% 이내였으므로 자원 소모량은 같은 트랜스코딩 등급에서는 크게 벗어나는 경우는 없고 거의 고정적인 비율의 자원 소모를 보여주고 있었다. 실험 결과 등급별 트랜스코딩이 진행될 때 자원 소모율이 일정한 것을 기준으로 해서 각 등급별로 필요한 자원 가중치 값을 얻을 수 있게 되었다. 단, 4CIF급의 요청이 들어왔을 경우 트랜스코딩을 거치지 않고 바로 스트리밍 서비스되기 때문에 4CIF급은 측정에서 제외하였다.

3.2 자원가중치 테이블

QoS가 보장되는 스트리밍 미디어 서비스를 위해서는 등급에 맞는 미디어를 제한된 시간내에 트랜스코딩 하여 끊김이나 흔들림 없이 사용자에게 전송하여야한다. 그러나 트랜스코딩을 요청하는 사용자의 수가 증가할 경우 제한된 트랜스코딩 노드내의 자원들의 고갈로 인하여 등급별 QoS 기준을 만족시키지 못하는 현상이 발생한다.

클러스터 구조로 구성된 트랜스코딩 서버 환경에서 각각의 트랜스코딩 노드들은 같은 하드웨어 자원들로 구성될 수도 있지만 서로 다른 규격, 즉 서로 다른

CPU 처리량, 메모리 크기, 네트워크 대역폭 을 갖는 노드들로 구성될 수도 있다. 각각의 노드들이 서로 다른 규격을 가지고 있을 경우 트랜스코딩 작업을 수행하기 위하여 소모되는 자원들의 양은 각각의 노드들마다 다를 수밖에 없다. RWLD는 트랜스코딩을 수행하기 위하여 각각의 서버들이 사용자 등급별 트랜스코딩을 진행할 때 소모되는 자원의 양을 부하배분의 가중치로 하여 노드별 부하배분을 수행하는 방식이다. 이를 위하여 트랜스코딩 등급별, 트랜스코딩 서버의 규격별 자원가중치 값이 필요하다.

표 5 등급별 자원가중치 계산 알고리즘

<pre> i. 서버 가중치 측정 */ if (M / Qm > B / Qn > C / Qc) { Wn = $\frac{Qm_n \times 100}{\sum_{k=1}^n Qm_k}$ (n = 1, 2, ..., i) } else if (B / Qn > C / Qc > M / Qm) { Wn = $\frac{Qn_n \times 100}{\sum_{k=1}^n Qn_k}$ (n = 1, 2, ..., i) } else if (C / Qc > M / Qm > B / Qn) { Wn = $\frac{Qc_n \times 100}{\sum_{k=1}^n Qc_k}$ (n = 1, 2, ..., i) } </pre>	<p>부하측정 용계</p> <p>B : 가용 네트워크 대역폭</p> <p>M : 가용 메모리</p> <p>C : 가용 CPU 역량</p> <p>i : 트랜스코딩 등급</p> <p>Qc : 등급 i 로 트랜스코딩 할 때 cpu 사용량</p> <p>Qn : 등급 i 로 트랜스코딩 할 때 네트워크 사용량</p> <p>Qm : 등급 i 로 트랜스코딩 할 때 메모리 사용량</p> <p>Wn : 등급별 자원 가중치</p>
---	---

표 5는 사용자 등급별로 트랜스코딩 서버들에서 자원 가중치 값을 계산하는 것을 보여주고 있다. M은 서버의 가용 메모리, B는 네트워크 대역폭, C는 가용한 CPU 처리량을 나타낸다. 스트리밍 미디어 등급을 i개로 나누었을 경우 Qci, Qni, Qmi는 트랜스코딩이 진행될 때 등급별로 소모되는 CPU, 네트워크, 메모리 사용량을 의미한다. 즉 요구되는 트랜스코딩을 SQCIF, QCIF, CIF, 4CIF 의 4등급으로 나누면 Qc1, Qc2 Qc3 Qc4는 각 등급으로 트랜스코딩 될 때 요구되는 CPU 사용량을 나타낸다. 트랜스코딩 수행시 이들 자원들 중 우선적으로 고갈되는 자원은 처리 가능한 총 트랜스코딩 수를 제한하

게 된다. RWLD 정책에서는 우선적으로 고갈되는 자원을 기준으로 각각의 트랜스코딩 서버에서 사용자 등급별 자원가중치 값 W_n 을 구한다. 즉, 트랜스코딩 등급 i 를 수행할 때 임의의 트랜스코딩 서버의 가용메모리를 기준으로 하여 허용 가능한 트랜스코딩 개수는 M/Q_{mi} , 가용 CPU 역량을 기준으로 할 때 C/Q_{ci} , 네트워크 대역폭을 기준으로 할 때 B/Q_{ni} 로 계산된다. 이중 가장 적은 값이 해당 트랜스코딩 서버의 해당 사용자 등급의 자원 가중치로 설정된다.

다음 식 (1)은 CPU 자원이 우선적으로 고갈되는 자원인 경우 CPU 자원을 기준으로 트랜스코딩 등급별 자원 가중치 값을 계산하는 것을 보여준다.

$$W_n = \frac{Q_{cn} \times 100}{\sum_{k=1}^i Q_{ck}} \quad (n=1,2,\dots,i) \quad (1)$$

현재는 4개의 등급으로 스트리밍 미디어를 구분하였으므로 i 의 값은 4이다. 식 (1)에서 해당 등급의 CPU 점유율 Q_{cn} 을 모든 등급의 CPU의 점유율 $\sum_{k=1}^i Q_{ck}$ 로 나누면 해당 등급의 CPU 가중치가 계산된다. 예를 들어 SQCIF 급의 CPU 점유율이 12%, QCIF 급의 CPU 점유율이 20%, CIF 급의 CPU 점유율이 32%라고 하면 SQCIF 급의 가중치는 식 (1)에 의하여 18.75, QCIF 급의 가중치는 31.25 CIF급의 가중치는 50이 된다.

분배서버는 표 5의 알고리즘에 의거하여 트랜스코딩 서버별로 트랜스코딩 등급들에 대한 자원 가중치 값들을 계산한 후 이들을 자원 가중치 테이블에 등록한다. 또한 트랜스코딩 서버별로 스트리밍 미디어의 QoS를 만족시키며 얻을 수 있는 최대 사용자수를 사용자 등급별로 측정한다. 이렇게 얻어진 최대 사용자 숫자와 이미 얻어진 등급별 자원 가중치 값을 곱하여 트랜스코딩 서버 각각의 트랜스코딩 등급별 총 가중치 값들을 계산하여 이들을 자원 가중치 테이블에 등록한다. 이들 값들 이외에 자원가중치 테이블에는 현재 처리 중인 트랜스코딩 요구들에 의하여 누적된 자원가중치 값을 트랜스코딩 노드별로 가지고 있다.

분배서버는 자원가중치 테이블에 등록된 각 트랜스코딩 노드에 대한 사용자 등급별 자원 가중치 값 및 각각의 노드별 누적된 자원 가중치 값을 가지고 부하분배를 수행한다. 또한 가중치 테이블의 등급별 총 가중치 값을

가지고 QoS를 위한 진입제어 결정을 수행한다. 표 6은 트랜스코딩 서버가 사용자들의 요청을 받기 전 초기화 되었을 때 2개의 트랜스코딩 노드들에 대한 자원 가중치 테이블의 값들의 초기 값을 예로서 보여주고 있다. 누적 가중치 항목은 초기 값인 0으로 설정되고, 자원 가중치 값 및 최대 사용자 값은 미리 측정된 실험 결과 값으로 설정된다.

3.3 자원 가중치 기반 부하분배

표 7은 자원 가중치 기반 부하분배 알고리즘이 자원 가중치 테이블의 정보를 이용하여 트랜스코딩 서버를 선택하는 것을 보여준다. 사용자의 요청이 들어오게 되면 분배서버에서 사용자 정보를 얻게 되고 요청에 대한 처리를 하기 위해 `thread_function()` 함수를 생성한다. `thread_function()` 함수는 누적 가중치가 가장 적은 서버를 선택하기 위하여 `chkResource()` 함수를 호출한다. `chkResource()` 함수는 현재 동작중인 트랜스코딩 서버들의 누적된 자원 가중치 값들을 비교하여 가장 적은 값을 갖는 트랜잭션 서버를 찾아낸다. `accumulated_weight[BACK_END]`은 각각의 트랜스코딩 서버의 누적된 자원 가중치 값을 저장하고 있으며, 변수 `BACK_END`는 트랜스코딩 서버들의 수를 의미한다. 표 6에 `chkResource()` 함수는 선택된 트랜스코딩 서버를 변수 `idx`를 사용하여 복귀한다. 복귀 후 `thread_function()` 함수에서는 요청된 사용자에 대한 진입제어 함수인 `admission_control()` 함수를 호출한다. 진입제어가 성공할 경우 선택된 트랜스코딩 서버로 사용자의 트랜스코딩 요구를 전달한다. 그 후 자원 가중치 테이블에서 해당 트랜스코딩 서버의 항목의 누적 가중치 값을 변경된 `accumulated_weight[target]` 값으로 갱신한다.

3.4 진입제어(Admission Control)

스트리밍 미디어 서비스는 그 특성상 사용자들에게 끊어짐이 없는 영상을 제공하여야한다. 이미 다수의 사용자가 만족스러운 QoS 스트림을 제공받고 있는 상태에서 추가적인 사용자로 인하여 모두의 QoS가 보장되지 못하는 현상을 방지하기 위하여 진입제어 프로그램이 필요하다.

RWLD 정책에서는 진입제어를 위하여 분배서버는 트랜스코딩 서버별로 각각의 트랜스코딩 등급들에 대한 가중치 값을 계산하여 가중치 테이블에 등록할 뿐만 아니라, 트랜스코딩 서버별로 스트리밍 미디어의 QoS를 만족시키며 공급할 수 있는 등급별 최대 사용자 수를 측정한다. 측정된 등급별 최대 사용자수를 이미 얻어진 등급별 가중치 값과 곱하여 트랜스코딩 등급별 총 가중치 값을 각각의 트랜스코딩 서버별로 얻는다. 등급별 총 가중치는 부하분배 및 진입제어에 사용된다.

표 8은 부하분산 서버에서 사용자가 트랜스코딩 요구

표 6 분배서버의 자원 가중치 테이블의 구조

	트랜스코딩 서버 A				트랜스코딩 서버 B			
	자원 가중치	최대 사용자	총 가중치	누적 가중치	자원 가중치	최대 사용자	총 가중치	누적 가중치
SQCIF	25	8	200	0	24	14	336	0
QCIF	35	7	245	0	26	12	312	0
CIF	40	6	240	0	50	9	450	0

표 7 자원 가중치 기반 부하분배 알고리즘

```

struct RWT {
    int weight_data[4];
    // RESOURCE WEIGHT, MAX CLIENTS, TOTAL WEIGHT, ACCUMULATED WEIGHT
}Server[BACKEND_NUM][GRADE]; // 트랜스코딩 서버의 자원가중치 테이블
// BACK_END : 트랜스코딩 서버수
// GRADE : 트랜스코딩 등급

void *thread_function(void *arg){

    get TRANSCODING_GRADE from client information;
    ...
    target = chkResource(TRANSCODING_GRADE );
    ...
    if(!admission_control(transcoding_grade, target)
        return -1; // reject a new request
    // 선택된 트랜스코딩 서버의 가중치 증가
    resource_weight = Server[target][TRANSCODING_GRADE].weight_data[RESOURCE_WEIGHT]
    Server[target][TRANSCODING_GRADE].weight_data[ACCUMULATED_WEIGHT] += resource_weight;
    request_transcoding(target, &client_inform()); // 해당 트랜스코딩 서버로 사용자 요구 전달
    ...
}

int chkResource(TRANSCODING_GRADE )
{
    int i, idx, weight; // BACKEND_NUM : 트랜스코딩 서버 수
    int min = MAX_NUM // MAX_NUM : 최대 정수 값
    for (i=0; i < BACKEND_NUM; i++)
    {
        weight = Server[i][TRANSCODING_GRADE].weight_data[ACCUMULATED_WEIGHT];
        if (weight < min)
        {
            // 누적 가중치가 최소인 트랜스코딩 서버 구함
            min = weight;
            idx = i;
        }
    }
    return idx;
}

```

표 8 최대 가중치 기반 진입제어 알고리즘

```

admission_control(transcoding_grade, id)
{
    // id: selected server number
    // transcoding_grade : transcoding grade requested by clients

    int weight;

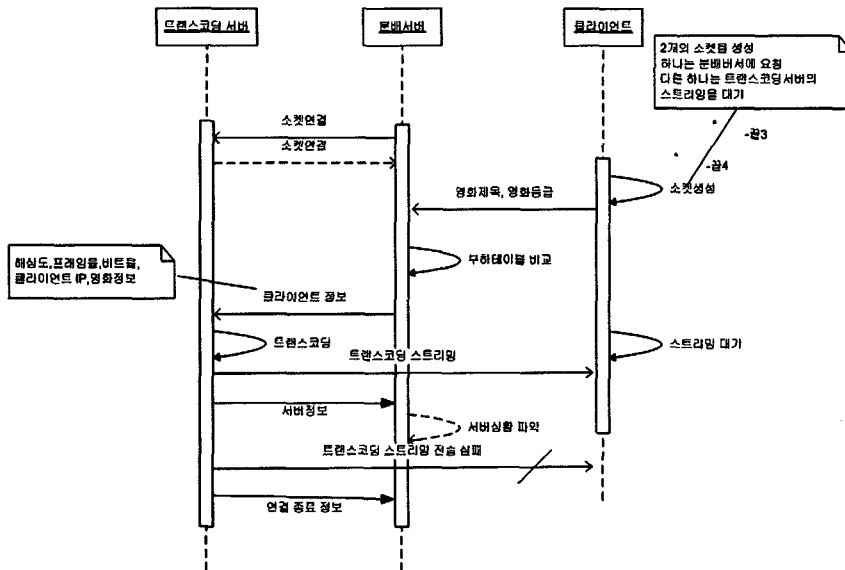
    // MAX_SQCIF, MAX_QCIF, MAX_CIF : 총 가중치 값

    if(transcoding_grade == SQCIF)
    {
        weight = Server[id][transcoding_grade].weight_data[RESOURCE_WEIGHT];
        if((Server[id][transcoding_grade].weight_data[ACCUMULATED_WEIGHT]+weight)> MAX_SQCIF)
            return 0; // reject a new client request
    }
    if(transcoding_grade == QCIF)
    {
        weight = Server[id][transcoding_grade].weight_data[RESOURCE_WEIGHT];
        if((Server[id][transcoding_grade].weight_data[ACCUMULATED_WEIGHT]+weight)> MAX_QCIF)
            return 0; // reject a new client request
    }
    if(transcoding_grade == CIF)
    {
        weight = Server[id][transcoding_grade].weight_data[RESOURCE_WEIGHT];
        if((Server[id][transcoding_grade].weight_data[ACCUMULATED_WEIGHT]+weight)> MAX_CIF)
            return 0; // reject a new client request
    }
    return 1; // accept a new client request
}

```

를 선정된 트랜스코딩 서버에서 QoS를 만족시키면 처리할 수 있는지를 판단하는 진입제어 알고리즘을 나타낸다. 변수 id는 선정된 서버를 나타내며, 변수 transcoding_

grade는 사용자가 요구한 트랜스코딩 등급을 나타낸다. MAX_SQCIF, MAX_QCIF, MAX_CIF 는 각각의 등급별 총 가중치 값을 나타낸다. 새로운 사용자가 추가될



때 부하분산 서버는 동작중인 모든 트랜스코딩 서버들의 누적된 자원가중치 값들을 비교하여 가장 적은 값을 갖는 트랜스코딩 서버를 선정한 후 이 함수를 호출하여 새로운 사용자에 대한 진입제어의 가부를 결정한다.

3.5 RWLD 정책의 동작 순서

그림 5는 자원 가중치 기반 정책을 사용한 트랜스코딩 시스템에서의 트랜스코딩 및 스트리밍 처리를 위한 시퀀스 다이어그램을 보여주고 있다. 다이어그램에서 제시된 것처럼 우선 트랜스코딩 서버와 분배서버 사이에 소켓 연결을 설정 후 분배서버는 사용자의 요청을 받아들인다. 사용자는 두 개의 소켓을 생성하고 하나는 분배서버로 사용자의 정보를 보내는 것으로 사용하고 나머지 소켓은 트랜스코딩 서버로부터 전송되는 스트리밍 미디어를 대기한다. 분배서버에서는 사용자의 요구사항을 분석한 후 미리 계산된 자원 가중치 테이블 및 자원 가중치 기반 부하분배 정책을 이용하여 트랜스코딩 서버를 선택한다.

제안된 RWLD방식은 미리 계산된 자원 가중치를 사용하여 사용자들이 요구하는 트랜스코딩 부하를 분산하기 때문에 트랜스코딩 시스템의 성능 확장성을 위하여 트랜스코딩 서버들을 증가 시켜도 분배서버에 과부하가 발생하지 않는다.

자원 가중치 테이블로만 부하분산을 하는 경우에는 수행중인 트랜스코딩의 종료, 중단 또는 사용자들의 임의적 서비스 중단과 같은 사건들로 인하여 트랜스코딩 서버들 사이에 부하 불균형이 일어날 수 있다. 이를 보완하기 위해서 트랜스코딩이 완료, 중단된 경우나 사용자 접속이 끊어졌을 경우에 트랜스코딩 서버의 정보를

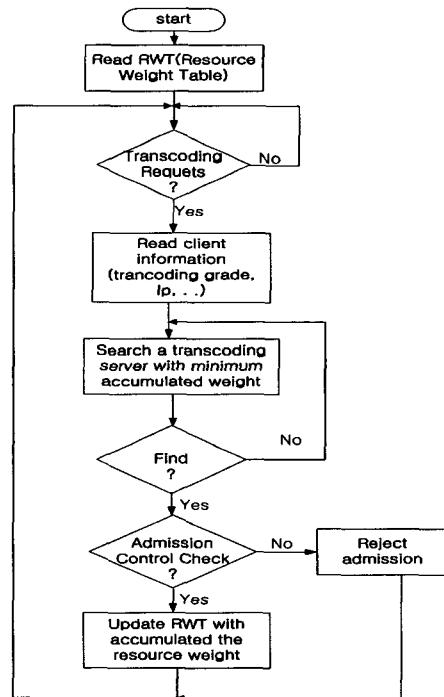


그림 6 RWLD의 제어흐름도

분배 서버에 보낸다. 분배서버에서는 보내온 트랜스코딩 서버들의 상태 정보와 미리 계산된 자원 가중치 테이블을 사용하여 부하 배분을 수행한다.

그림 6은 RWLD 정책을 사용하여 부하분배 및 진입제어를 수행하는 분배서버의 동작에 대한 제어 흐름도

를 보여준다. 그림 6에서 보여주듯이 분배서버는 각각의 트랜스코딩 서버들의 자원가중치 테이블을 초기화한 후 사용자들의 트랜스코딩 요구를 기다리며, 요청이 발생할 때 마다 자원가중치 테이블의 해당 정보를 읽어 해당 등급의 누적 가중치가 가장 적은 트랜스코딩 서버를 선택하므로 균형적인 부하배분을 진행하게 된다. 또한 선택된 트랜스코딩 서버에 대한 진입제어 판단을 하여 진입가부를 결정하므로 현재 서비스를 받고 있는 사용자들에게 안정된 스트리밍 미디어의 QoS를 보장하게 된다.

4. 실험환경

대규모 사용자들의 트랜스코딩 요구에 대한 부하배분 정책들의 성능 평가를 위하여 본 연구에서는 이종 클러스터 시스템 환경에서 트랜스코딩 서버, 분배서버를 구축하였다. 또한 성능 측정을 위한 부하발생기를 구현했으며 기존의 트랜스코딩 프로그램을 스트리밍 미디어의 특성에 맞추어 수정하였다.

4.1 클러스터 기반 트랜스코딩 시스템

트랜스코딩 시스템은 분배 서버와 트랜스코딩 서버로 구성된다. 본 실험에서는 분배 서버는 하나의 SMP 서버를 사용하였고, 트랜스코딩 서버는 3대의 클러스터 시스템을 사용하였다. 이들 클러스터 시스템들은 서로 다른 하드웨어 환경을 가지고 있다. 총 23개의 노드들 구성된 이종 환경의 클러스터 노드들이 트랜스코딩 서버들로 사용되었다. 각 트랜스코딩 서버들은 100Mbps 스위칭 허브에 연결되고 각 클러스터 시스템의 하드웨어 사양은 표 9, 10, 11과 같다.

표 9 클러스터 1호(트랜스코딩 서버 × 8 노드)

CPU	Intel(R) Celeron(R) CPU 1.60GHz
메모리	256M SDRAM
운영체제	RedHat 7.3 (Kernel 2.4.18)
네트워크	100Mbps fast-ethernet

표 10 클러스터 2호(트랜스코딩 서버 × 8 노드)

CPU	Intel(R) Celeron(R) CPU 2.00GHz
메모리	1GB SDRAM
운영체제	RedHat 7.3 (Kernel 2.4.18)
네트워크	100Mbps fast-ethernet

표 11 클러스터 3호(트랜스코딩 서버 × 7 노드)

CPU	CPU : AMD 애슬론 MP 쉐러브레드 2200+ Dual
메모리	1GB DDR
운영체제	RedHat 7.3 (Kernel 2.4.18)
네트워크	100Mbps fast-ethernet

4.2 트랜스코딩 프로그램

트랜스코딩 프로그램은 오픈소스 프로젝트의 ffmpeg를 수정하여 사용하였다[16]. ffmpeg는 서버의 CPU를 최대한 사용하여 최대한 빨리 트랜스코딩 하기 때문에 제한된 시간에 정확히 맞추어 미디어 스트림을 전송하면 되는 실시간 처리 기능에는 부적합하다. 따라서 본 논문에서는 그림 7의 우측 다이어그램에서 나타나듯이 재생시간에 맞추어 1초마다 필요한 프레임들 트랜스코딩 하도록 소스를 수정하였다. 또한 트랜스코딩 된 데이터를 디스크에 저장하는 것이 아니라 스트리밍 서비스를 하도록 수정하였다.

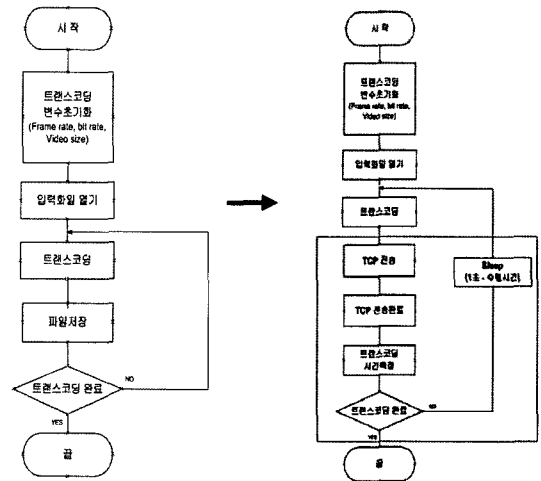


그림 7 수정된 ffmpeg 구조

4.3 트랜스코딩 등급별 최대 사용자 실측

본 논문에서 제안된 RWLD 방식에서는 부하분산을 수행하기 위하여 이전 장에서 설명된 등급별 부하 가중치이외에, 각각의 트랜스코딩 서버에서 표 1에서 제시된 트랜스코딩 등급을 만족시키며 QoS를 보장되는 최대 사용자수에 대한 실측이 필요하다. 측정된 값은 다양한 사용자 등급 분포가 존재하는 동작 환경에서 트랜스코딩 서버들의 성능 확장성 향상에 기여한다. 표 12는 클러스터 1호기, 2호기, 3호기 각각에서 트랜스코딩 등급별 최대 사용자 수를 측정한 결과이다. 표 12의 한 칸의 항목은 1개의 트랜스코딩 서버에서 각각의 등급별로 트

표 12 등급별 트랜스코딩 최대 사용자 수

등급	클러스터 1 (8 트랜스코딩 노드)	클러스터 2 (8 트랜스코딩 노드)	클러스터 3 (7 트랜스코딩 노드)
SQCIF	8	9	14
QCIF	7	8	12
CIF	5	6	9

랜스코딩을 수행하며 규정된 QoS를 만족시킬 수 있는 총 사용자의 수이다. 예를 들어 클러스터 1, 2, 3 시스템 모두에서 모든 사용자의 요구가 CIF 등급을 SQCIF로 트랜스코딩 하는 것일 경우 총 234(8 × 8 + 8 × 9 + 7 × 14)개의 스트림의 QoS를 만족시키며 트랜스코딩 할 수 있다.

4.4 부하 발생기

성능 측정을 위하여 부하 분산 프로그램을 만들었다. 부하 분산 프로그램은 영화정보, 영화선호도, 등급별 사용자 분포도를 기준으로 하여 부하를 발생하게 된다. 표 13은 부하 발생 알고리즘이다. 사용자 요청의 총 개수를 정한다음 부하발생 분포형태에 따라서 등급별로 개수를 정하게 된다. 예를 들어 총 사용자 요청수가 100이고 부하발생이 Zipf 분포 형태로 4:3:2:1의 비율을 따른다면 sqcif = 40, qcif = 30, cif = 20, 4cif = 10으로 된다. sqcif, qcif, cif, 4cif의 개수가 정해지게 되면 랜덤하게 부하를 4가지 중에 하나를 발생시킨다. 부하분포 형태대로 부하를 발생시키며 각 등급별 할당된 부하 개수를 초과하지 않도록 하였다. 즉 총 사용자 요청 개수에서 등급별로 발생빈도를 계산하여 각각의 부하들이 발생되게 하였다.

부하발생기는 분배 서버에 초당 1개의 부하를 발생시키도록 하였고 분배서버에 영화를 요구할 때 영화제목, 해상도, 비트율, 영화등급을 전송한다. 요청한 영화의 선호도는 Zipf분포를 따르고, 그래프의 기울기의 영향을 미치는 skew factor는 일반 영화 선호도의 분포를 따르는 값인 0.271을 사용하였다[17]. Zipf 분포형태로 부하를 발생시키기 위해 사용자 총 요청 수에 따른 등급별 부하 분포를 먼저 계산한 후 계산된 분포도에 따라

표 13 부하 발생 알고리즘

```

/* Work Load */
while(sqcif>0 || qcif>0 || cif>0 || 4cif>0)
{
    val=rand()%4+1;
    if(val==1 && sqcif>0){
        sqcif=sqcif-1;
        Workload(val);
    }
    else if(val==2 && qcif > 0){
        qcif=qcif-1;
        Workload(val);
    }
    else if(val==3 && cif > 0){
        cif=cif-1;
        Workload(val);
    }
    else if(val==4 && cif4 > 0){
        cif4=cif4-1;
        Workload(val);
    }
} // while end
    
```

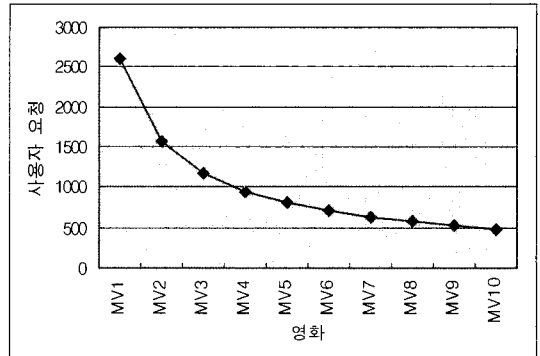


그림 8 영화 선호도 (Zipf 분포 : skew factor = 0.271)

부하를 랜덤으로 발생되도록 구성하였다. 그림 8은 사용자 요청을 총 10,000명으로 했을 때의 영화 선호도 분포를 보여준다.

사용자는 모바일 휴대폰 기종의 SQCIF급, PDA기종의 QCIF급, 무선노트북 기종의 CIF급, 일반 PC기종의 4CIF급으로 나누었다. 사용자 단말기는 SQCIF급의 사용자가 가장 많고 단말기의 분포형태는 Zipf 분포를 따른다고 가정하였고, skew factor는 0.271을 사용하였다.

4.5 성능 측정 프로그램

부하발생기는 성능 측정을 하기위한 사용자수 만큼의 부하를 발생시켜 분배서버에 요청하면, 분배 서버는 채택된 부하 분산 정책에 따라서 해당 트랜스코딩 서버들에게 트랜스코딩 작업을 시작시킨다. 본 논문에서 성능 측정 방법은 기존의 yardstick 모니터링 프로그램 방식을 트랜스코딩 시스템 환경에 맞추어 재구성하여 진행하였다[18]. 이 방법에서는 총 사용자 중 하나의 사용자는 트랜스코딩 되어 전송된 영상 미디어를 이동 단말기에서 상영하며, 나머지 사용자들은 가상 사용자 프로그램을 독립적인 사용자 PC에서 구동시켜 전송되어온 미디어 스트림을 처리하도록 구성된다.

가상 사용자 프로그램은 각각의 사용자별로 요청되어 트랜스코딩 처리 후 전송되어온 미디어 스트리밍을 각각의 빈 버퍼로 읽어 들여 미디어 스트림을 소모시키므로 실제 사용자들이 연결된 환경처럼 동작한다.

본 논문의 성능 측정 프로그램에서는 전송되어온 미디어 스트림이 표 1에 제시된 해당된 MPEG 등급에 만족하는지 시험하였다. 가상 사용자 프로그램은 전송되어온 미디어 데이터의 프레임율과 비트율을 측정하여 규정된 등급이하로 데이터가 수신시 해당 트랜스코딩 서버가 과부하 상태임을 인식한다. 규정된 등급별에 해당된 비트율, 프레임율 이하로 전송되는 사용자별 스트리밍 미디어는 사용자 단말에서 QoS(Quality of Service)

를 제공할 수 없으므로 제안된 트랜스코딩 서버들의 총 사용자 숫자에서 제외되어야 한다.

5. 성능

구축된 클러스터 기반 트랜스코딩 시스템에서 RR, DWRR, RWLD 정책들의 성능을 측정하였다. 성능 측정의 척도로는 트랜스코딩을 진행할 때 CPU 자원이 가장 먼저 소모되므로 사용자의 증가에 따른 각각의 트랜스코딩 서버들의 CPU 사용률 측정하였다. 또한, 제한된 시간 이내에 사용자에게 끊기거나 깨짐이 없는 스트리밍 미디어를 공급해야하는 스트리밍 미디어 서비스의 특징을 만족시키기는 정도를 측정하기 위하여 각각의 사용자 등급별 QoS 기준을 만족하는 최대 사용자수를 측정하였다.

5.1 CPU 사용률

그림 9, 10, 11은 RR, DWRR, RWLD 정책들에서 트랜스코딩 서버들의 CPU 사용률을 보여주고 있다. 모든 트랜스코딩 서버들을 그래프에 나타낼 수 없으므로 3개의 클러스터 기중에서 각각 2개의 트랜스코딩 서버들을 선택하여 CPU 사용량을 그래프에 나타내었다. A, B는 클러스터 1, C, D는 클러스터 2, E, F는 클러스터 3안에 포함된 트랜스코딩 서버들이다.

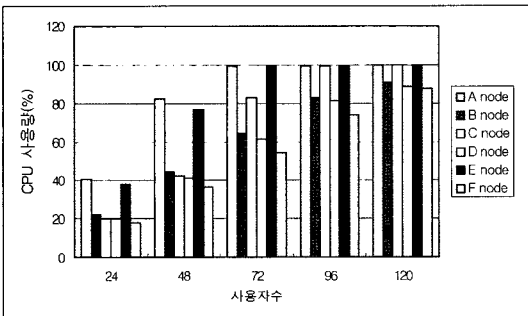


그림 9 Round Robin 방식

그림 9는 부하분산 정책으로 RR(Round Robin) 방식을 사용한 경우의 결과이다. RR 정책은 사용자의 요청 순서대로 서버를 분배하는 방식이다. 그림 9에서 나타나듯이 서버들 사이의 CPU 사용량이 차이가 전 구간에서 발생하고 있다. 이런 결과는 RR 정책에서는 CPU 사용량이 많이 요구되는 트랜스코딩 등급들을 분배받은 서버들과 가벼운 처리 등급들을 받은 서버들이 공존하기 때문이다. 특히 사용자수가 120명이 되는 시점에서 서버 A, C, E의 CPU 사용량은 100%에 이르고 있고 이들 서버에서 진행되는 트랜스코딩 작업들이 사용자 등급에 맞는 QoS를 제공하지 못하는 경우가 측정되었다. RR 정책에서는 스트리밍 미디어의 QoS를 만족시키기 위한

진입제어(admission control)가 사용될 수 없다. 따라서 특정 서버로 요청들이 폭주될 경우 해당 서버에서 처리 중인 모든 스트리밍 서비스의 QoS가 망가지는 현상이 발생한다. 반면에 서버 B, D, F는 상대적으로 처리량이 적은 등급들을 분배받으므로 CPU의 사용률이 적으며 이것은 자원의 활용 측면에서 단점이 된다.

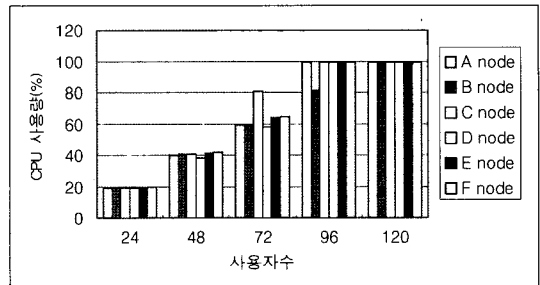


그림 10 DWRR 방식

그림 10은 부하분산 정책으로 DWRR(Dynamic Weighted Round Robin) 방식을 사용한 경우의 결과이다. 논문에서 구현한 DWRR 정책은 트랜스코딩 서버들에서 주기적으로 보내주는 CPU 사용량에 근거하여 분배서버는 부하분배 관리 테이블을 유지하고 가장 적은 CPU 사용량을 나타내는 트랜스코딩 서버로 새로운 사용자 요구를 배분하는 방식이다. 그림 10에 나타난 것처럼 DWRR 방식은 RR 보다 트랜스코딩 서버들의 CPU 사용량에 있어서 보다 균형적 사용을 보여주고 있다. DWRR에서는 CPU 사용량이 100%에 도달한 트랜스코딩 서버에게는 새로운 사용자의 트랜스코딩 요청을 분배하지 않으므로 사용자들의 요구한 등급에 맞는 QoS를 만족하지 못하는 경우는 발생하지 않는다. 하지만 트랜스코딩 서버가 주기적으로 분배서버에 자신의 상태 정보를 보내주기 위하여 CPU를 사용하는 부담이 있고, 트랜스코딩 서버수가 크게 늘어날 경우 분배서버와 트랜스코딩 서버들사이의 네트워크 대역폭의 소모가 많아져 트랜스코딩 시스템의 전체적인 성능 확장성에 부정적 영향을 미치게 된다. 또한 DWRR 방식은 단지 트랜스코딩 서버들의 CPU 사용량만을 부하분산의 기준으로 삼았기 때문에 제한된 시간 이내에 트랜스코딩 후 스트리밍 전송을 만족시키면 되는 스트리밍 미디어 서비스의 특징을 반영하지 못하고 있다. 그림 10에서 보듯 사용자 120명 수준에서 트랜스코딩 서버들은 CPU 사용량이 99% 이상에 도달하므로 분배서버는 이들 트랜스코딩 서버들에 추가적 트랜스코딩 요청을 할 수 없다.

그림 11은 부하분산 정책으로 RWLD 방식을 사용한

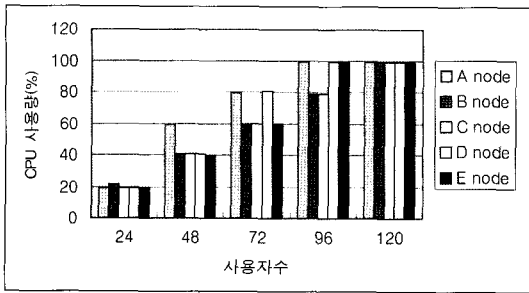


그림 11 RWLD 방식

경우 트랜스코딩 서버들의 CPU 사용량을 측정할 결과이다. 그림 11에 나타난 것처럼 RWLD 정책은 DWRR처럼 CPU 사용량에 있어서 트랜스코딩 서버들 사이에 균형적 사용을 보여주고 있다. 또한 RWLD 정책은 사용자 수가 120명에 도달하여 CPU 사용량이 100%가 될 지라도 추가적인 트랜스코딩 요구들을 처리할 수 있다. 이런 추가적 요구가 가능한 것은 RWLD를 사용하는 분배서버는 미리 계산된 트랜스코딩 등급별 자원 가중치 테이블 및 표 9에 제시된 것처럼 각각의 트랜스코딩 서버에서 처리할 수 있는 등급별 최대 사용자수 정보를 사용하여 증가되는 사용자들의 진입제어(admission control)를 판단하기 때문이다. RWLD 정책은 분배서버에서 미리 계산된 자원 가중치 테이블 및 최대 사용자수 정보를 사용하므로 DWRR 정책처럼 트랜스코딩 서버들이 주기적으로 자신의 정보를 보내주기 위하여 CPU 및 네트워크 대역폭을 사용하는 부담이 없다.

5.2 성능 확장성

그림 12는 RR, DWRR, RWLD 정책 각각에서 표 1에서 등급별로 규정된 QoS를 만족시키며 스트리밍 미디어 서비스를 공급받는 총 사용자수를 보여주고 있다. 스트리밍 미디어 서비스의 필수조건은 등급별 QoS를 만족 시켜야하는 것이다. 사용자의 등급별 QoS를 만족시키지 못하는 트랜스코딩 처리는 총 사용자 누적에서 제외된다.

부하 발생기에서 부하는 총 294개의 부하를 발생시켰고 Zipf 분포를 따라 등급 분포는 SQCIF 143개, QCIF 86개, CIF 64로 발생되었다. 그림 12를 보면 트랜스코딩 서버의 수가 증가할수록 받아들일 수 있는 총 사용자 수는 늘어나고 있다는 것을 알 수 있다. 17~23번의 서버에서 그래프의 기울기가 조금 더 증가하는 것을 볼 수 있는데 클러스터3호기의 사양이 높아 사용자 처리량이 높아졌기 때문이다. 그림 12에 나타난 것처럼 RWLD 방식은 RR, DWRR 방식보다 선형적 성능 확장성을 보이고 있다.

RR 방식은 과부하가 걸린 트랜스코딩 서버들에서 처

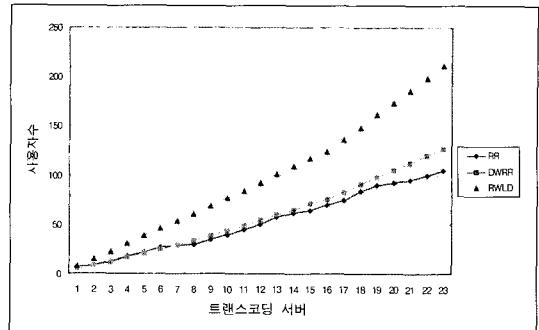


그림 12 성능 확장성

리되는 트랜스코딩 작업은 스트리밍 미디어서비스의 필수 조건인 등급별 QoS를 만족시키기 못한다. 특히 QoS를 만족하며 동작중인 트랜스코딩 서버들도 추가되는 사용자 요구에 의하여 서비스되는 전체 사용자들 모두의 QoS를 만족 시키지 못하는 경우가 발생하였다. 이런 현상들로 인하여 RR 정책은 스트리밍 미디어 서비스에 적합한 부하분산 정책이 아님이 판명되었다.

DWRR 방식이 RWLD 정책 보다 성능이 떨어지는 이유는 트랜스코딩 서버들의 CPU 사용량만을 단순히 부하분산의 기준으로 사용했기 때문이다. 원하는 등급의 미디어로 트랜스코딩을 진행하며 스트리밍 처리를 위해 필요한 최소한의 CPU 사용량에 대한 고려 없이 일반적으로 측정되는 트랜스코딩 서버의 CPU 사용량만을 부하분산의 기준으로 삼았기 때문이다. 트랜스코딩 서버들의 CPU 사용량이 100%가 된 후에도 추가적으로 증가하는 사용자들에 대한 트랜스코딩 및 스트리밍 전송이 등급별 QoS 범위 내에서 가능하다. 또한, DWRR 정책은 분배서버와 트랜스코딩 서버들 사이에 상태정보를 주고 받기위한 일련의 과정을 수행하므로 CPU 사용량이 큰 부분을 차지하는 트랜스코딩 작업에 부담이 되고 그 결과 성능 확장성에 부정적 영향을 미치게 되었다.

그림 12에 나타났듯이 RWLD은 트랜스코딩 서버의 수가 증가함에 따라 추가되는 사용자들의 QoS 수준을 만족시키며 다른 정책들보다 선형적 성능 확장성을 보여주고 있다. RWLD 정책에서는 각각의 등급별 트랜스코딩을 위한 필요한 CPU사용량을 미리 측정하여 가중치 테이블을 계산하였고, 또한 각각의 트랜스코딩 서버들마다 QoS를 보장할 수 있는 사용자수를 미리 계산하여 부하분배에 사용하였다. 이런 사전 정보들을 바탕으로 부하 분산 정책을 수행하므로 트랜스코딩 서버내의 자원을 최대한 활용할 수 있었고 그 결과 양호한 성능 확장성을 나타내었다.

6. 결론 및 향후 연구

최근 다양한 형태의 이동 단말기들에서 스트리밍 미디어와 같은 고품질의 멀티미디어 데이터를 전송하고 처리하는 서비스를 제공하고 있다. 스트리밍 미디어는 그 특성상 사용자들에게 끊음이 없고 흔들림이 없는 영상 화면을 제공해야한다. 특히 이동 단말기는 기존 유선망과 다른 무선 네트워크 및 제한된 하드웨어 사양에서 동작하므로 기존에 사용되던 스트리밍 미디어는 이동 단말기 환경에 맞추어 트랜스코딩 한 후 스트리밍 전송을 수행하여야 한다.

트랜스코딩 된 미디어를 사용자들에게 스트리밍 형태로 전송 서비스하기 위해서는 이동 단말기 등급에 맞는 QoS 규격을 만족시켜야한다. 특히, 상업적 가치가 보장되며 대규모의 이동 단말기 사용자들에게 QoS가 보장되는 스트리밍 미디어 서비스를 제공 하기위해서 클러스터 서버와 같은 가격대 성능비가 우수한 서버의 도입이 필수적이다. 클러스터 기반 서버 구조에서 트랜스코딩 및 스트리밍 미디어 서비스를 받는 사용자들의 최대수를 도출시키기 위해서는 스트리밍 미디어에 대한 이해와 서버내부 노드들의 자원 특성을 이용한 부하분산 정책이 있어야한다.

본 논문에서는 클러스터 서버를 트랜스코딩 시스템으로 구성하기위하여 클러스터의 노드들을 부하 분배 서버, 트랜스코딩 서버로 구분하였고 트랜스코딩 요청 사용자들이 대규모로 발생할 때 부하 배분 정책을 연구하였다. 부하배분 정책은 트랜스코딩 서버의 자원을 효과적으로 사용하여야하며 또한 사용자들의 등급별 QoS를 만족시키는 스트리밍 서비스를 제공해야한다. 본 논문에서는 부하 배분 정책으로 기존의 클러스터 서버에서 사용되던 RR, DWRR 정책들 이외에 새로운 부하배분 정책인 RWLD 정책을 제안하였다.

구현된 이중 클러스터 서버 구조에서 3개의 부하배분 정책들에 대한 실험 결과, RR 정책은 트랜스코딩 서버들의 CPU 사용량에 있어서 불균형을 발생하였다. RR 정책은 트랜스코딩 등급별 부하 차이를 배려하지 않고 입력되는 사용자 순서로 부하 배분을 하였기 때문이다. 또한 RR 정책은 과부하가 발생한 트랜스코딩 서버들에 추가적인 사용자들의 진입을 통제할 수 있는 진입제어 기능이 없으므로 QoS를 보장해야하는 스트리밍 미디어 서비스를 위한 부하배분 정책으로는 적합하지 않았다. 반면에 DWRR 정책은 트랜스코딩 서버들의 CPU 사용량에 있어서는 균형을 나타내었다. 그러나 DWRR 정책은 주기적으로 분배서버에 트랜스코딩 서버들이 자신의 상태 정보를 보내주기 위하여 CPU를 사용하는 부담이 외에 분배서버와 트랜스코딩 서버들사이의 네트워크 대역폭을 소모하는 문제점을 가지고 있었다. 또한, DWRR 정책은 트랜스코딩 서버들의 CPU 사용량만을 부하분산

의 기준으로 사용했기 때문에 제한된 시간 안에 트랜스코딩 후 스트리밍 전송을 만족시키면 되는 스트리밍 미디어 서비스의 특징을 반영하지 못하였다.

본 논문에서 제안한 RWLD 정책은 트랜스코딩 서버들의 자원가중치 테이블 및 트랜스코딩 등급별 최대 사용자 정보를 사용한다. 실험 결과 RWLD 정책은 트랜스코딩 서버들의 CPU 사용량에 있어서 균형적 분포를 나타내었고, 성능 확장성 측면에서는 트랜스코딩 서버의 개수가 증가됨에 따라 다른 정책들보다 선형적 성능 확장성을 나타내었다. 이런 성능 확장성은 스트리밍 미디어의 실시간적 특성을 반영한 RWLD의 부하 분배 정책의 특성 및 분배서버와 트랜스코딩 서버들 사이의 상태 정보를 유지하기위한 과부하를 제거하였기 때문에 가능하였다.

본 논문에서는 트랜스코딩 시스템에서 부하배분 정책을 하나의 영화 스트림을 기준으로 하였다. 향후에는 영화 스트림을 독립적인 처리를 할 수 있는 작은 성검도 단위에서 병렬 트랜스코딩 하여 트랜스코딩 서버들의 자원 활용률을 더욱 향상시키는 방안에 대하여 연구하고자 한다.

참고 문헌

- [1] Dinkar Sitaram, Asit Dan, "Multimedia Servers: Applications, Environments, and Design," Morgan Kaufmann Publishers, 2000.
- [2] W.C. Feng and M. Lie, "Critical Bandwidth Allocation Techniques for Stored Video Delivery Across Best-Effort Networks," The 20th International Conference on Distributed Computing Systems, pp. 201-207, April 2000.
- [3] D.H.C. Du and Y. J. Lee, "Scalable Server and Storage Architectures for Video Streaming," IEEE International Conference on Multimedia Computing and Systems, pp. 191-206, June 1999.
- [4] H.Bhradvaj, A. Joshi and S. Auephanwiriyakul. "An active transcoding proxy to support mobile web access." In Proceedings of International Conference on Reliable Distributed System, pp 118-123, 1998.
- [5] Vetro. A.; Sun, H., "Media Conversions to Support Mobile Users", IEEE Canadian Conference on Electrical and Computer Engineering (CECE), May 2001.
- [6] Florin Lahan, Irek Defee, Marius Vlad, Aurelian Pop, Prakash Sastry, "Integrated system for multimedia delivery over broadband ip networks," IEEE Transactions on Consumer Electronics, Vol. 48, No. 3, pp. 564-565, 2002.
- [7] <http://www.ieee802.org>
- [8] <http://www.mpeg.org>
- [9] Sumit Roy, Michele Covell, John Ankcorn, and Susie Wee, "A System Architecture for Managing

Mobile Streaming Media Services", Takeshi Yoshimura Streaming Media Systems Group, Hewlett-Packard Laboratories, Palo Alto, CA 94304.

- [10] J. Song, E. Levy, A. Iyengar, and D. Dias, Design alternatives for scalable web server accelerators, Proceedings of the 2000 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), 2000.
- [11] A. Iyengar, E. MacNair, and T. Nguyen, "An analysis of web server performance," Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 347-357, 1998.
- [12] W. Zhang, "Linux virtual servers for scalable network services," Proc. Ottawa Linux Symposium, 2000.
- [13] J.P. Chew, A.K. Gupta, "Using Dynamic Weights for Improving Fairness in the ATM ABR Service," 5th IEEE Symposium on Computers and Communications, pp. 372-377, July, 2000.
- [14] Jiani Guo, Fang Chen, Laxmi Bhuyan, Raj Kumar, "A Cluster-based Active Router Architecture Supporting Video/Audio Stream Transcoding Service", IEEE, 2003.
- [15] Surendar Chandra, Carla Schlatter Ellis and Amin Vahdat, "Differentiated Multimedia Web Services Using Quality Aware Transcoding", INFOCOMM 2000.
- [16] <http://ffmpeg.sourceforge.net>
- [17] C.C. Aggarwal, J.L. Wolf, and P.S. Yu, "On optimal batching policies for video-on-demand storage servers," Proc. of IEEE ICMCS'96, pp. 253-258, Hiroshima, Japan, June 1996.
- [18] 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.
- [19] Susie Wee, John Apostolopoulos, Wai-tian Tan, Sumit Roy. "Research and Design of a Mobile Streaming Media Content Delivery Network", IEEE ICME, July 2003.
- [20] Kai Shen, Hong Tang, Tao Yang, Lingkun Chu, "Integrated resource management for cluster-based Internet services", ACM SIGOPS, Vol. 36, pp. 225-238, 2002.



이좌형

2003년 2월 강원대학교 정보통신공학과 학사. 2005년 2월 강원대학교 컴퓨터정보통신공학과 석사. 2005년 9월 강원대학교 컴퓨터정보통신공학과 박사 과정. 관심분야는 운영체제, 병렬처리, 파일시스템, 멀티미디어시스템, 센서네트워크



최면욱

2002년 2월 강원대학교 정보통신공학과 학사. 2004년 8월 강원대학교 컴퓨터정보통신공학과 석사. 2004년 9월~현재 신한생명 사이버마케팅팀. 관심분야는 멀티미디어 시스템, 운영체제, 병렬처리, 멀티미디어 시스템, 분산시스템



김윤

1993년 8월 고려대학교 전자공학과 학사. 1995년 8월 고려대학교 전자공학과 석사. 1995년~1999년 LG전자 안양연구소 주임연구원. 2003년 8월 고려대학교 전자공학과 박사. 2004년 3월~현재 강원대학교 교수. 관심분야는 디지털 영상 신호 처리, 멀티미디어 통신, 센서 네트워크



정인범

1985년 고려대학교 전자공학과 졸업(학사). 1985년~1995년 (주) 삼성전자 컴퓨터 시스템사업부 선임 연구원. 1992년~1994년 한국과학기술원 정보통신공학과 졸업(컴퓨터공학 석사). 1995년~2000년 8월 한국과학기술원 전산학과 졸업(박사). 2001년~현재 강원대학교 전기전자정보통신공학부 컴퓨터전공 교수. 관심분야는 운영체제



서동만

2002년 2월 강원대학교 정보통신공학과 학사. 2004년 2월 강원대학교 컴퓨터정보통신공학과 석사. 2004년 3월~현재 강원대학교 컴퓨터정보통신공학과 박사 과정. 관심분야는 병렬처리, 멀티미디어 시스템, 운영체제, 센서네트워크