

클러스터 시스템에서 효과적인 미디어 트랜스코딩 부하 분산 정책

박충명[○], 이자형, 김병길, 최연욱, 정인병, 김 윤

강원대학교컴퓨터정보통신공학과

{cmpark[○], jhlee, bgkim, muchoi}@snslab.kangwon.ac.kr, {ibjung, yooni}@kangwon.ac.kr

Effective Media Transcoding Load Distribution Policy in the Cluster System

Chongmyung Park[○], Joa Hyoung Lee, Byounggil Kim, Myunuk Choi, Inbum Jung, Yoon Kim

Dept. Computer Information & Telecommunication Engineering

요 약

최근 무선통신기술의 발전으로 PC뿐만 아니라 PDA, 휴대폰 등 다양한 장치를 통하여 멀티미디어 서비스를 제공받을 수 있게 되었다. 무선망은 유선망에 비해 대역폭이 낮기 때문에 높은 비트율을 갖는 스트리밍을 낮은 비트율로 변환하기 위해 트랜스코딩을 이용한다. 따라서 이러한 트랜스코딩 시스템에서 효율적으로 무선 스트리밍을 전송하기 위해서는 부하분산에 관한 연구가 필요하다. 본 논문에서 제안한 클러스터 트랜스코딩 시스템은 스트리밍 서비스의 모바일 호스트 등급에 따른 CPU 사용량, 네트워크 대역폭, 메모리의 가중치를 계산하여 부하 배분에 사용할 수 있는 부하 분산 모형을 제시한다.

1. 서 론

최근 멀티미디어와 정보통신망의 발전에 따라 영상 정보 서비스에 대한 요구가 날로 다양해지고 있다. 멀티미디어 서비스의 급속한 발전으로 사용자는 유선뿐만 아니라 무선망을 통하여 무선단말기로 멀티미디어 스트리밍을 전송하고 재생하는 스트리밍 서비스를 받을 수 있게 되었다. 무선망에서는 네트워크 대역폭이 유선망보다 상대적으로 열악한 환경을 가지고 있다. 또한 모바일 호스트의 낮은 컴퓨팅 파워와 시스템 자원은 서버로부터 전송되는 높은 품질과 고용량의 멀티미디어 스트리밍을 적절하게 처리할 수가 없기 때문에 호스트에 따라 스트리밍을 적절한 품질과 크기를 바꾸어 트랜스코딩하는 연구들이 진행 중에 있다[1].

본 논문에서는 트랜스코딩 시스템 부하분산 연구를 위하여 공개 소스 기반인 리눅스를 이용하여 클러스터 트랜스코딩 시스템을 구현하였다. 트랜스코딩 시스템은 트랜스코딩 서버, 부하분배서버, 클라이언트로 구현하였다. 트랜스코딩서버에서는 MPEG 프로파일 등급에 따른 트랜스코딩과 시스템자원 상태정보를 분배서버에 보내주는 기능을 구현하였고 분배서버에서는 본 논문에서 제안한 부하분배 알고리즘을 이용하여 부하를 분배하도록 구현하였다[2].

본 논문에서 실험은 사용자들의 다양한 영화선호도 환경에서 트랜스코딩 시스템의 부하분산을 측정하였다. 트랜스코딩 부하분산 알고리즘은 기존의 클러스터 웹서버의 부하분산 알고리즘인 WRR을 보완하여 동적으로 시스템 가중치가 바뀌는 DWRR 알고리즘과 스트리밍 서비스의 모바일 호스트 등급에 따른 부하 가중치를 계산하여 부하배분에 사용할 수 있는 RWR 알고리즘을 구현하고 측정하였다. 제안된 알고리즘 방법은 기존의 부하분산 알고리즘인 RR과 비교하여 효과적인 부하분산이 된다는 것을 실험을 통하여 확인하였다.

2. 부하분산 정책

이 장에서는 클러스터 트랜스코딩 시스템에서 자원가중치 기반의 부하 분산 방법을 제안한다. N개의 클러스터 트랜스코딩 서버에서 고화질 영상을 각 요청 등급에 맞게 트랜스코딩을 하게 된다.

그림1은 LVS를 이용한 트랜스코딩 시스템구조를 나타내고 있다. 클라이언트 요청은 서버에서 순차적 단계의 처리과정을 거치게 된다. 웹서버에서 사용자의 요구가 서버에 도착하게 되면, 인증 서버에서 서비스 등급을 설정하게 되고 분배서버에 의해 클러스터 트랜스코딩 서버로 연결하게 된다. 요청을 받은 요구를 처리하는 과정에서 트랜스코딩에 따른 서버에서 소모되는 자원소모는 모바일 호스트 등급에 따라 SQCIF, QCIF, CIF, 4CIF 급으로 나누어진다. 트랜스코딩 서버에서는 등급으로 나누어진 기준에 따라 비디오 사이즈, 비트율, 프레임 크기를 트랜스코딩하여 모바일 호스트에 전송하게 된다[3].

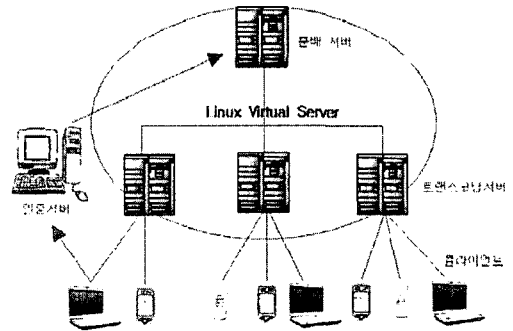


그림1. LVS를 이용한 트랜스코딩 시스템 구조

2.1 Round Robin

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

• 본 연구는 강원대학교 JTRC의 지원을 받아 수행하였음.
• 본 연구는 한국과학재단 목적기초연구(R05-2003-000-12146-0)의 지원으로 수행되었음.

2.2 Dynamic Weighted Round-Robin

DWRR 알고리즘은 WRR 알고리즘처럼 시스템 사양에 따라 고정된 가중치를 가지는 것이 아닌 주기적으로 서버의 상태정보를 파악하여 동적 가중치를 계산하여 분배하는 알고리즘이다. 본 논문에서는 클러스터 웹서버의 WRR방식을 응용하여 서버의 상태정보를 이용하여 동적 가중치를 계산하여 부하를 분배하는 시스템을 구현하였다. 서버의 정보를 측정하는 방식에는 주기적으로 서버의 정보를 보내는 방식과 클라이언트에서 요청이 들어왔을 경우에 서버의 정보를 분배서버에 보내는 두 가지 방식이 있다.

2.3 Resource Weighted Round-Robin

RWRR(Resource Weight based Load Distribution) 방식은 클러스터 서버들의 부하를 측정하지 않고 서버에 요청하는 스트리밍 서비스의 등급에 따른 CPU, 메모리, 네트워크 자원 가중치를 계산하여 가중치 테이블을 작성한다. RWRR은 이 가중치 테이블의 값에 따라 부하를 분산하는 모델이다.

3. Resource Weighted Round-Robin

그림 2는 RWRR 알고리즘의 시퀀스 다이어그램이다. 트랜스코딩 서버와 분배서버에서 소켓연결설정을 한 후 서버에서는 클라이언트의 요청을 받아 들인다. 클라이언트에서는 두 개의 소켓을 생성하고 하나는 분배서버에 클라이언트 정보를 보내고 트랜스코딩 스트리밍을 대기한다. 분배서버에서는 클라이언트의 정보를 분석하여 미리 계산된 가중치 테이블을 이용하여 트랜스코딩 서버를 선택하게 된다.

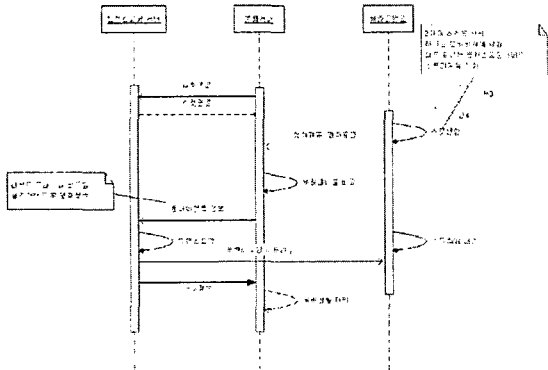


그림2. RWRR 시퀀스 다이어그램

가중치 테이블로만 부하분산을 하는 경우에는 트랜스코딩시 서버에서 부하 측정 방식이 정확하지 못하여 부하 불균형이 일어날 수 있다. 이를 보완하기 위해서 트랜스코딩이 완료된 경우나 클라이언트 접속이 끊어졌을 경우에 서버의 정보를 분배 서버에 보낸다. 분배서버에서는 가중치 테이블과 서버의 상태정보를 이용하여 분배서버의 부하는 줄이면서 좀 더 정확한 부하 측정을 하게 된다.

트랜스코딩 서버는 미디어 데이터를 저장하고 있고 분배서버의 요청에 따라 사용자에게 트랜스코딩 스트림을 전송하는 서버이다. DWRR알고리즘 방식의 경우 1초 간격으로 자신의 시스템 자원 상

의 정보를 분배서버에게 보낸다. 분배서버는 이 정보를 받아서 부하를 파악하지 않고 분배 알고리즘에 따라 부하 분산을 한다.

트랜스코딩 프로그램은 오픈소스 프로젝트의 ffmpeg를 수정하여 사용하였다[4]. ffmpeg는 트랜스코딩시 CPU를 최대로 사용하여 최대한 빨리 트랜스코딩 하기 때문에 실시간으로 스트리밍으로 보내기에는 부적합하다. 따라서 재생시간에 맞추어 1초마다 필요한 프레임만을 트랜스코딩 하도록 소스를 수정하였고 트랜스코딩된 데이터를 디스크에 저장하는 것이 아니라 스트리밍 서비스를 하도록 소스를 수정하였다.

트랜스코딩된 미디어 스트림은 미리 정의 되어 있는 TCP연결 소켓으로 스트림을 전송하게 되고 DWRR 알고리즘 방식 같은 경우에는 영화의 트랜스코딩 과정이 끝나거나 중간에 클라이언트 접속이 끊길 경우 트랜스코딩 서버의 상태 정보를 분배서버에 전송하게 된다.

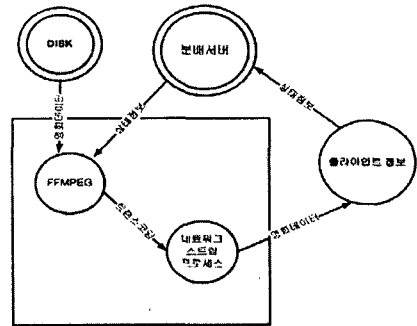


그림3. 트랜스코딩 서버 구성

그림 3은 트랜스코딩 시스템에 구현된 트랜스코딩 서버의 구성을 보여준다. 클라이언트에서 보내진 상태정보를 분배서버에서 트랜스코딩서버로 전송하게 되면 ffmpeg에서는 디스크에서 해당 영화를 읽어 들인 후 클라이언트에서 요청한 영화제목, 프레임율, 영상크기, 비트율의 정보대로 트랜스 코딩을 하게 된다. 트랜스코딩된 데이터는 클라이언트 IP에 스트리밍 전송을 하게 된다.

4. 실험 및 분석

서버의 구성 형태가 다를 경우 RR, DWRR, RWRR 알고리즘 방식을 측정하였다. 클러스터 1,2,3호기를 이용하여 측정하였고 24대의 트랜스코딩 서버를 사용하였다.

그림 4, 5, 6의 실험 결과는 클러스터 1, 2, 3호기 중에서 각각 2대의 서버 자원현황을 그래프로 나타낸 것이다. 표 1은 각 클러스터 시스템에서의 최대 클라이언트 수를 측정한 결과이다. 표 1의 한 칸의 항목은 1개의 서버에서 각 등급별로 트랜스코딩한 개수이다.

등급	클러스터1 (노드×8)	클러스터2 (노드×8)	클러스터3 (노드×7)
SQCIF	8	9	14
QCIF	7	8	12
CIF	5	6	9

표1. 등급별 가능한 최대 클라이언트 수

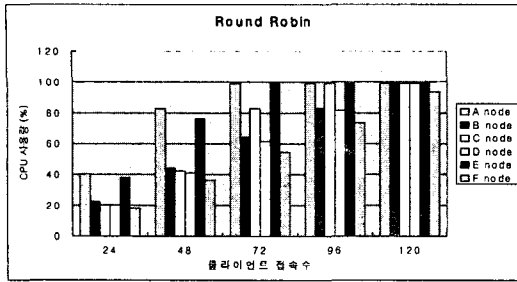


그림 4. Round Robin 방식

그림 4는 이중 서버환경의 Round Robin 방식의 그래프를 나타낸 것이다. A, B 는 클러스터1, C,D는 클러스터2, E, F는 클러스터3의 서버들이다. 이중 환경간의 CPU사용량 분포를 살펴보면 A서버와 E서버에 부하분산이 크게 불균형하다는 것을 알 수가 있다. 이는 이중서버 환경일 경우 서버의 가중치 계산이 올바르게 이루어지지 않기 때문에 부하분산의 불균형을 가져올 수 있다는 것을 보여준다.

그림 5의 경우는 이중 서버간의 DWRR 가중치 알고리즘 측정 결과이다. 이중 서버환경에서는 DWRR 알고리즘이 최고의 부하 분산 성능을 나타내 주고 있다. 하지만 주기적으로 분배서버에 서버의 상태 정보를 보내주어야 하기 때문에 서버가 크게 늘어나게 되면 분배서버의 네트워크 대역폭이 좁아진다는 단점이 있어 확장성에 문제가 생기게 된다.

그림 6은 이중 서버간의 RWRR 알고리즘 방식의 그래프를 나타낸 것이다. 클라이언트 48번째에서 A서버의 부하 불균형이 일어나는 것을 볼 수가 있는데 이는 이중 서버 환경에서 자원가중치 값이 서버의 성능에 따라 각각 틀리게 때문에 CPU 변화량을 정확히 반영하지 못한다는 것을 보여주고 있다. CPU 변화량을 보다 정확히 반영하기 위해서 본 논문에서는 클라이언트의 접속이 끊기거나 트랜스코딩이 완료된 경우 서버의 상태정보를 분배서버에 보내주어서 이러한 부하 불균형 문제를 보완하였다.

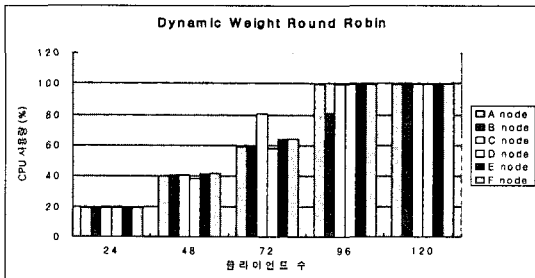


그림 5. DWRR 방식

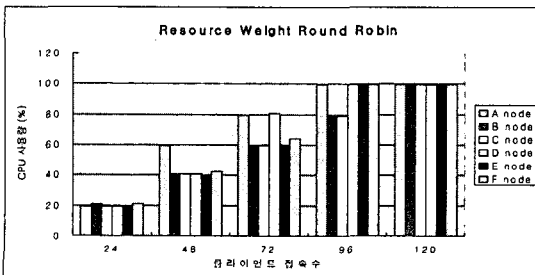


그림 6. RWRR 방식

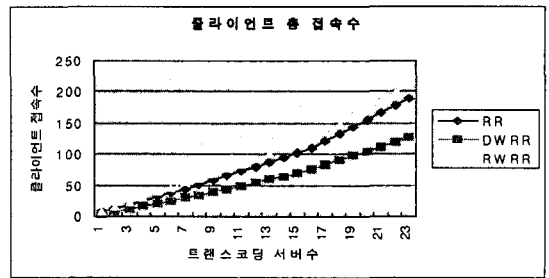


그림 7. 최대 클라이언트 수

그림 7을 보면 트랜스코딩 서버가 증가할수록 클라이언트를 받아들일 수 있는 총 사용자 수는 늘어나고 있다는 것을 알 수 있다. 17~23번의 서버에서 그래프의 기울기가 크게 증가하는 것을 볼 수 있는데 클러스터 3호기의 사양이 높아 클라이언트 처리량이 높아졌기 때문이다. RWRR이 가장 좋은 성능을 보여주고 있으며 DWRR이 가장 낮은 성능을 보여주고 있다. DWRR 성능이 가장 떨어지는 이유는 트랜스코딩에 사용되는 CPU 사용량과 사용자의 미디어 트랜스코딩 요구를 만족시킬 수 있는 최소한의 CPU 사용량과는 차이가 있기 때문이다. RWRR은 트랜스코딩에 따른 필요한 CPU 사용량을 측정하여 가중치 테이블을 계산하기 때문에 이 단 이용해 효율적으로 부하를 분산한다는 것을 그래프를 통해 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 리눅스 기반의 클러스터 시스템에서 미디어 트랜스코딩에 대한 연구를 진행하였다. 부하분산 실험 결과 RWRR방식과 RR방식의 클러스터 트랜스코딩 시스템이 부하분산에 효율적이라는 것을 알 수 있었으며 DWRR 방식은 CPU 부하 분산에 있어서 실제 서버정보를 이용하여 분배된 하기 때문에 가장 좋은 부하분산을 나타내었다.

그러나 실험을 통한 결과 CPU의 사용률이 99%가 넘어서도 정상적인 트랜스코딩이 가능하기 때문에 CPU 사용률을 이용한 DWRR 방식에서는 성능의 확장성에 있어서 커다란 제약이 따른다. 따라서 이를 보완한 자원가중치 기반의 RWRR 알고리즘 방식은 트랜스코딩시 필요한 최소 CPU 사용률을 이용한 자원 가중치 테이블로 부하를 분산하여 트랜스코딩시스템 성능의 확장성을 크게 높였다.

RWRR 방식에서 자원 가중치 테이블만으로 부하를 분산하면 스트리밍 특성에 따라 트랜스코딩시 자원으로움이 조금씩 달라지기 때문에 최적의 부하 분산을 하는데 어려움이 있었다. 이를 보완하여 트랜스코딩 작업이 끝나거나 클라이언트 접속이 끊겼을 경우 서버의 상태정보를 분배 서버에 보냄으로서 단점을 보완하였고 부하테이블과 실제 서버의 상태정보를 이용하여 서버 부하측정의 정확성을 높임으로서 효율적으로 부하가 분산되는 것을 실험을 통하여 확인했다.

6. 참고문헌

- [1] 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.
- [2] <http://www.mpeg.org>.
- [3] <http://www.linuxvirtualserver.org>.
- [4] <http://ffmpeg.sourceforge.net>.