VOD 시스템을 위한 Clustered Proxy Server 설계[1]

배기범°, 김종훈, 이철훈
충남대학교 컴퓨터공학과

A Design of Clustered Proxy Servers for the VOD System

Ki-Beom Bae, Jong-Hoon Kim, Cheol-Hoon Lee
Department of Computer Engineering, Chungnam National University

요 약

기존의 VOD시스템을 확대하여 원거리에 존재하는 서비스제공자로부터에서 사용자에게 더 빠른 응답서비스를 제공하고, 장애가 발생하더라도 사용자에게 중단 없이 제공되도록 하기위하여 프락시 서버를 두었는데, 특히 일정 지역내 존재하는 프락시 서버들간에 클러스터를 형성하여 단일 프락시 서버로 만족할수 없는 다수의 클라이언트로부터의 요구를 받아들이고 cache 사이즈의 감소와, 더 높은 hit rate를 제공하도록 한다. 또한 이 클러스터내의 프로그램관리를 위해 비디오 데이터에 대한 정보를 hint로 저장하여 각 프락시에서 관리함으로써 네트웍을 통해 사용자까지 실시간으로 비디오 데이터를 전송이 가능하고, 직접 데이터의 신경 없이 클러스터를 효율적으로 관리할수 있도록 한다

1. 서 론

네트웍을 통해 사용자로부터 요구를 받아 영화, 비디오게임, 홈쇼핑등의 비디오 관련 서비스를 제공 하는 것을 주문형 비디오(Video On Demand) 서비스라 한다. 현재 VOD 서비스를 위한 표준으로 DAVIC(Digital Audio Visual Council)에서 국제 표준 규격을 정하고 이에따라 주문형 비디오시스템이 개발되고 있다. DAVIC 표준에 의하면 VOD 참조모델은 표준에 따라 다음과 같이 구성된다. 영화 제작자와 같이 내용 자체를 제공하는 내용 제공자 시스템 (content Provider System), Juke box나 Disk array 등과 같은 저장소를 가지고 있으며, 사용자에게 서비스를 제공하는 서비스 제공자 시스템(Service Provider System), 정보 전달의 매체가 되는 네트웍과 같은 전달 시스템 (Delivery System), 그리고 서비스를 최종적으로 사용하는 소비주체로서의 사용자 시스템 (Service Consumer System)이 있다. 이러한 구성을 기본으로 효과적인 서비스를 위한 시스템 구조들이 활발히 연구중에 있다. 망구조를 구성하는 방법으로 중앙 집중형 서버 구조와 분산 서버 구조로 분류할수 있는데 전자의 경우 사용자의 요구가 급증하면 병목 현상이 발생하고, 사용자가 느끼는 응답시간의 증가로 서비스질 감소를 가져오게 되는 반면 요청한 모든 프로그램을 서버에서 가지고 있기때문에 hit rate의 차원에서는 효과적이라 할 수 있다. 이와 달리 분산 서버의 경우 사용자가 원하는 자료가 어느 서버에 있는가를 알수있게 네임 서버가 있어야 하고, 불필요한 데이터의 중복뿐만 아니라 각 서버들간의 로드 균형 유지 문제와 자료의 일관성 유지가 중요한 문제로 나타난다. 하지만 물리적으로 가까운 곳에 위치한 서버로부터 서비스를 받기 때문에 사용자의 요구에 좀더 신속히 대처할수 있고, 병목 현상의 해소와 고장 감내한 서비스까지도 제공할수 있다. 각각의 단점을 보완한 구조로서 계층형 분산구조[2]를 들수 있는데 이는 일종의 분산 VOD시스템으로 시스템 구조는 메타데이타를 가지고 있는 데이터 베이스, 비디오 전체를 가지고 있는 AS (Archive Server), 그리고 캐쉬 역할을 수행하는 하나이상의 VFS(Video File Server)로 구성된다. 앞으로 설명하려는 프락시 서버는 기능면에서 VFS와 동일하다고 볼수 있으나 단일 서버로 만족할수 없는 다수 클라이언트 허용문제가 클러스터링 기법을 도입함으로써 해결할수 있다.

본고에서는 기본적인 DAVIC VOD시스템에 대해 살펴보고, 캐쉬기능을 가지고 있는 프락시 서버들간에 클러스터를 형성함으로써 개개의 프락시 서버내에 필요한 디스크 사이즈의 감소와 네트웍 부하의 분산효과가 나타남을 알아본다

2. DAVIC VOD 시스템

2.1 서비스 제공자 시스템(Service Provider System)

대용량 비디오데이터를 저장하고 있으면서 고속으로 비디오 스트림을 전송하는 기능으로 다음과 같이 구성된다

Storage manager : 비디오 스트림이 저장된 디스크 어레이와 데잎에 대한 관리를 담당

Application Service : 멀티미디어 정보에 대한 서비스 조작 경

[1] 본 논문은 과기부-한국과학재단 지정 충남대학교 소프트웨어 연구센터의 지원에 의한 것입니다

보를 처리한다

. Meta data : 사용자에 대한 모든 정보를 가지고 있다

. Stream Service : 스트림 계층에서 내용 흐름에 대한 처리를 수행한다

Service Gateway : 시스템에 접속된 모든 서비스를 등록받아 관리하며, 사용자에게 메뉴리스트를 보여주고 서비스 제공자 시스템들의 주소정보를 사용자에게 제공

Session Gateway : 망내 시스템들간 연결을 위한 자원의 추가 및 제거를 담당한다.

Client Profile : 메타 데이터를 참조하여 서비스를 요구한 사용자에 대한 인증 검사를 수행

2.2 서비스 사용자 시스템 (Service Consumer System)

서비스 사용자 시스템은 압축된 비디오 데이터를 재생하는 Set top box와 출력장치로 구성된다

Environment : 서비스 제공자 시스템과 서비스 품질(QoS)협상 및 세션을 설정하는 기능 수행

. Application : Service Gateway로부터 받은 메뉴 선택과 프로그램에 대한 네비게이션을 수행

. Product : 서버로부터 받은 스트림을 디코딩하여 출력장치로 전송한다

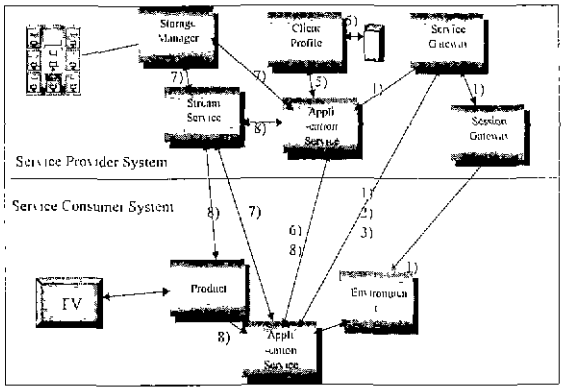

그림 2 DAVIC VOD 시스템 구조

2.3 VOD 서비스 시나리오

1) 서비스 사용자가 출력장치를 통해 Environment 요소에서 서비스제공자의 Session Gateway에게 세션 설정을 요구한다 Session Gateway는 Service Gateway에게 세션을 넘겨주고 Service Gateway에서 서비스 사용자에게 세션 설정에 대한 응답을 한다

2) 응답을 받은 서비스 사용자는 Service Gateway에게 메뉴리스트를 요구한다 Service Gateway는 서비스 사용자에게 메뉴리스트를 넘겨준다. (VOD, Home shopping, Game etc)

3) 서비스 사용자는 메뉴에서 VOD를 선택한다

4) Service Gateway는 VOD 서비스 제공자와 서비스 사용자 사이에 세션을 연결한다

5) 서비스 제공자의 Application Service는 요구받은 서비스에 대해여 서비스 사용자가 정당한 사용자인지 조사하여, 정당한 사용자이면 VOD서비스의 리스트를 사용자에게 전송한다 그렇지 않은 경우 서비스를 거절한다.

6) 프로그램 리스트를 전송받은 서비스 사용자는 프로그램을 선택하여 서비스 제공자에게 보낸다.

7) 서비스 제공자의 Application Service는 Storage Manager에게 프로그램 검색을 요구하고 Stream Service는 서비스 사용자의 application에게 스트림 서비스 준비 통보를 한다.

8) 서비스 사용자로부터 play 명령을 요구받으면, 실제적인 스트림을 전송한다.

3. Proxy server Clustering

비디오 데이터는 저장되어야할 데이터 양이 방대하고, 이를 처리하는 네트워크에서도 높은 전송율이 요구된다. 또한 비디오의 경우 초당 30프레임 정도의 출력을 요하므로 이러한 처리속도를 만족시키지 못하는 경우 매체로서의 의미를 상실하게 된다 따라서 사용자입장에서 위치적으로 근접한 곳에 서버를 설치하여 사용자가 느끼는 비디오 억세스 시간은 단축하고, 네트웍의 효율은 증가되는 효과를 얻는 방법이 요구되었다 이를 위해 분산 서버들간의 클러스터링 기법을 도입한 시례가 있지만 서버들간의 프로그램 중복을 피하기 위한 트래픽 발생이 상당한 오버헤드를 유발하고, 각 노드간의 load balancing 문제 또한 난항으로 나타난다. 따라서 여기서는 모든 프로그램을 가지고 있는 메인서버와 더 작은 용량으로도 클라이언트 요구를 만족할수 있는 프락시 서버의 개념을 도입하게 되었다. 그러나 단일 프락시 서버만으론 그 캐쉬 용량이 작을뿐더러 네트웍 트래픽이 집중될 위험이 존재하므로, 각 프락시 서버들이 가지고 있는 자료에 대해 어느 일정 지역내에서 자료를 공유함으로써 기시적인 캐쉬 사이즈의 증가를 가져오는 프락시 서버간에 클러스터를 이루는 방법을 제안한다 서버에게로 모든 요청이 집중됨으로써 네트웍의 과부하를 불러일으킬수 있는 경우를 개선하기위해 클라이언트가 프락시 서버에게 요청을 하면 프락시 서버는 자신에게 존재하는지, 동시에 힌트테이블을 통해 클러스터링 내에 속하는 프락시들에 존재하는 프로그램인지 확인한후 메인 서버에게 요청해야 할지를 판단 한다 좀더 빠른 응답시간을 기대하기 위해선 프로그램이 프락시에 존재하거나 다음으로 클러스터 내에 존재할 확률이 높을수록 성능이 향상된다는걸 알수 있다. 따라서 클러스터링으로 연결된 프락시 서버들은 정보교환을 통해 서로 중복된 프로그램을 가지고 있지 않도록 계속 적으로 유지를 함으로써 프로그램 hit rate를 개선시키도록 한다

3.1 프락시 서버의 기능

구체적으로 프락시 서버의 기능을 알아보면 다음과 같다

. VPS(Video Proxy Server)는 사용자가 요청한 프로그램이 자신의 디스크에 없을 경우 힌트 테이블을 통해 자신이 속한 클러스터 내에 프로그램이 있는지 조사하여 있을 경우 실시간으로 사용자에게 제공하고, 없을때는 메인 서버에게 요청하여 자신의 캐쉬에 저장을 하면서 사용자에게 서비스 해준다.

VPS에 프로그램을 저장할 때 캐쉬 대치 알고리즘을 수행하여 저장공간을 확보한다.

사용자의 요청빈도가 높은 자료에 대해서는 클러스터내에서 중복을 허용하도록 유지한다

힌트 테이블은 항상 최신의 자료를 가지도록 유지한다

3.2 프락시 서버 구성요소

Proxy Storage Manager 비디오 Stream이 저장된 디스크

array 관리

. Proxy Application Service : 멀티미디어 정보의 액세스 제어 및 서비스 조작정보처리

. Proxy Session Manager : 망 내 시스템 간의 연결을 위한 자원의 추가 및 제거와 사용자간의 RPC와 메시지 변환 기능을 수행

Proxy Stream Manager : 내용 흐름에 대한 처리수행 일관성 있는 스트림의 서비스

Proxy Client Profile : 사용자에 대한 인증검사 수행 사용자의 서비스이용프로그램등에관한 정보저장.

Proxy Service Manager : 시스템 내에 집속된 모든 서비스를 등록받아 메뉴 리스트등을 보내준다

Hint Manager : 프락시 서버가 이루는 클러스터링에 대한 자료의 관리 및 프락시간의 RPC를 담당

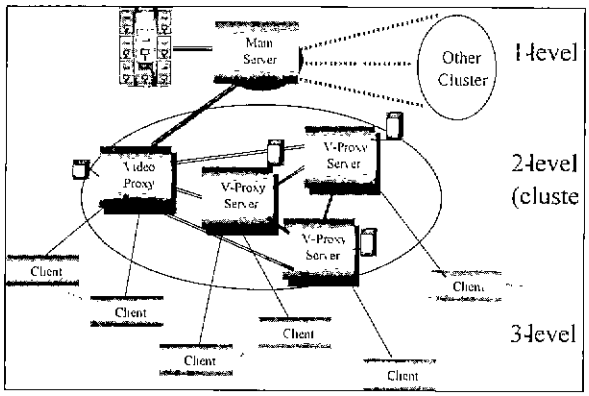

그림 3 프락시 서버의 클러스터 구조

3.3 Service 시나리오

. 요청 프로그램이 프락시 서버에 존재할 때

프락시 서버는 사용자 사이와 세션을 설정하고, 사용자의 제생 명령으로 스트림 진송을 수행한다

요청 프로그램이 클러스터내에 존재할 때

hint table을 통해 어느 프락시에 존재하는 프로그램인지 알아내고 해당 프락시 서버로 제어권을 넘겨준다.

. 요청 프로그램이 클러스터내에 존재하지 않을 때

hint table에 요청한 프로그램이 존재하지 않을때 요청받은 프락시 서버는 메인 서버에게 프로그램을 요청한다 이때 메인서버와 프락시 서버사이에 세션이 설정되고, 프락시 서버와 사용자 사이에 세션이 설정되어 프로그램이 프락시에 저장되는 동시에 사용자에게 실시간으로 전달되도록 한다. 저장이 끝난후 해당 프락시 서버는 자신의 프로그램 데이터가 변경되었음을 클러스터 내의 프락시들에게 알려야 한다 즉 hint 정보를 broadcast한다 hint 정보의 경우 그 size가 작아야 하므로 망내 트래픽 증가는 고려하지 않겠다

3.4 프락시 서버내에서의 캐시 대치

각각의 프락시 서버들은 자신의 프로그램뿐만 아니라 클러스터 내에 있는 모든 프락시들이 가지고 있는 비디오 데이터에 대한 hint table을 가지고 있다. 따라서 사용자로부터 요구받은 프로그램이 자신의 디스크와 클러스터내에 존재하는지 여부를 hint table을 통해 조사하고, 없다면 빈 디스크 공간을 찾아서 서버로부터 저장을 요구해야겠지만 빈 디스크 공간이 존재하지 않을 때 캐시 대치를 수행해야 한다 프로그램 캐쉬의 경우 접근 시간이 가장 오래된 프로그램을 대치하는 LRU(Least Recently Used) Algorithm, 프로그램의 접근 빈도를 비교하여 가장 작은 접근수를 가지는 프로그램을 대치하는 LFU(Least Frequently Used) Algorithm, 가장 먼저 재생된 프로그램을 대치하는 FIFO(First In First Out)의 기법이 적용된다 이 뿐만 아니라 최근 사용 통계치에 따라 대치하거나, 사용자가 사용하는 재생프로그램, 또는 사용자의 요청 내역을 기준으로 대치하는 방법등도 있다.

4 결론 및 향후 연구과제

현재 Sim++시뮬레이션 package를 이용하여 SunOS 5.5.1 Unix에서 모의 실험을 수행중이다.

본 논문에서는 VOD서비와 사용자 사이에 존재하는 Proxy서버들을 클러스터 구조로 연동시키는 방법을 제안하였다. 사용자가 느끼는 응답시간이 빠르기 위해선 요청한 프로그램이 사용자에 가장 근접한 프락시 서버내에 존재하거나 적어도 그 프락시 서버가 속한 클러스터 내에 존재 할 경우 시간을 단축할수 있다 따라서 프로그램의 hit rate가 성능에 가장 중요하게 평가되리라 예상된다 본 클러스터 구조는 여기에 속한 프락시 서버들의 자료를 모두 공유할수 있다는 점에서 묵시적인 cache size의 증가라는 효과가 따를 것이다. 따라서 향후에는 정립된 모델을 기준으로 프락시 서버에서 수용할 수 있는 사용자의 수에대한 고찰과 사용자의 요구를 좀더 민감하게 반응할수 있는 캐쉬 대치 알고리즘에 대한 연구가 진행 되어야 할 것이다.

참고문헌

[1] D James Gemmell Simmon, Harrick M Vin, Dilip D Kandlur, P Uenkat Rangan, "Multimedia Storage Servers : A Tutorial" Computer, vol 18, No1 May, 1995, pp 40-49

[2] Craig Federighi and Lawrence A. Rowe " A Distributed Hierarchical Storage Manager for a Video-on-Demand System" Storage and Retrieval for Image and Video Database 2, IS & T/Spie Symp on Elec Imaging Sci & Tec, San Jose, CA, Feb 1994

[3] Divyesh Jadav and Alok Choudhary "Designing and Implementing High-Performance Media-on-demand servers" IEEE Parallel & Distributed Technology, Summer 1995, pp29-39

[4] Renu Tewari, Harrick M Vin, Asit Dan, and Dinkar Sitaram " Caching in Bandwidth and Space Constrained Hierarchical Hyper-media Servers" Technical Report TR-96-30, Department of Computer Sciences, Univ of Texas at Austin, Dec. 1996

[5] M. Blaze and R. Alfonso, "Dynamic Hierarchical Caching in Large-scale Distributed File Systems" In Proceedings of International Conference on Distributed Computing Systems Jun 1992

[6] P Sarkar and J Hartman, "Efficient Cooperative Caching using Hints" In Proceedings of Operating Systems Design and Implementation Conference, Oct 1996