

멀티미디어 스트리밍 시스템을 위한 LTS 모델 설계

정규수* · 김형진**

*엔코아컨설팅 · **전북대학교

A Design of LTS Model for Multimedia Streaming System

Kil-dong Hong* · Hyoung-jin Kim**

*Encore Consulting Ltd.

**Chonbuk National University

E-mail : kim@chonbuk.ac.kr

요 약

본 논문에서는 복잡한 구조의 멀티미디어 스트리밍 시스템을 단순하고 기본적인 작업 구조를 설계하고, 이 구조를 바탕으로 확장성과 이식성을 높일 수 있는 시스템 설계에 목적을 두고 있다. 따라서 복잡하고 확장성이 떨어지는 문제점을 해결하기 위하여 시스템의 처리 구조를 선형 작업 집합(LTS : Linear Task Set) 모델로 제안하였다. 단순한 기본 구조를 바탕으로 일관성 있는 처리과정을 정의하고 단순한 시스템 구조에서 서비스의 목적과 요구사항에 따라 자유로운 구조 변경을 허용할 수 있도록 하였다. 또한 이 모델을 바탕으로 멀티미디어 스트리밍 서비스가 가능하도록 컴포넌트 시스템 모델을 제시하였다.

키워드

멀티미디어 스트리밍 시스템, 선형 태스크 집합, 컴포넌트 시스템

1. 서 론

멀티미디어 스트리밍 서비스에는 정의된 시공간 정보를 바탕으로 미디어의 상태, 네트워크 대역폭, 버퍼활용률, 디스크 접근시간, 메모리관리 등 시스템의 종합적인 기술들이 활용되고 있고, 사용자의 환경과 조건에 맞는 서비스 품질을 유지하기 위하여 네트워크 상태, 버퍼 활용률, 패킷 손실률 등 서비스 품질에 직접 및 간접적인 영향을 미치는 파라미터의 종합적인 처리에 대하여 고려해야 한다[1]. 또한, 시스템의 새로운 기능은 확장성이나 이질적인 플랫폼에 대한 이식성, 여러 멀티미디어 데이터 형식에 대하여 적용할 수 있도록 미디어 데이터의 유연성과 적응성에 대하여도 고려해야 한다. 이를 위하여 멀티미디어 시스템에서는 멀티미디어 프레임워크가 개발되어 설계 및 구축이 용이해 지고, 화상회의나 영상메일, 감시시스템 등 멀티미디어 데이터를 사용하는 시스템들의 생산성이 향상되었다.

Infopipes는 분산 멀티미디어 스트리밍 시스템에 대하여 운용되는 태스크(task)를 추상화 하는 단계에서 이해하기 쉽게 컴포넌트로 분류하고 표현하는데 중점을 두었고[2], SMART는 efficient, scalable, robust 한 스트리밍 비디오 시스템으로 MPEG-4 FGS를 이용하여 다중 루프 예측, 손실 감소, 매크로블록의 처리모드 선택 하는 방법을 통하여 시스템의 성능에 중점을 두었다[3]. 기존 연구들은 대부분 영상 압축 기술, 인코딩/디코딩 기술, 실시간전송기술과 같은 멀티미디어 처리 기술에 초점을 두고 있으며, 이러한 시스템은 멀티미디어 데이터와 처리 기술 간의 결합도가 높고 미디어 동기화나 전송 모듈이 데이터 처리 기술과 종속성이 높기 때문에 시스템의 확장이나 유지 보수에 많은 어려움이 발생할 수 있는 문제점을 안고 있다.

이러한 문제점을 해결하기 위하여 본 논문은 복잡한 구조의 멀티미디어 스트리밍 시스템을 단순하고 기본적인 작업 구조를 설계하고, 이 구조

를 바탕으로 확장성과 이식성을 높일 수 있는 시스템 설계에 목적을 두고 있다. 복잡한 멀티미디어 스트리밍 시스템은 단순한 작업(task) 처리 구조로 구성될 수 있고, 추가적인 기능을 확장할 수 있게 설계하였다. 또한 시스템은 서비스의 목적과 요구사항에 따라 자유로운 구조 변경을 허용하였고, 시스템의 유지보수와 이식성이 높일 수 있는 선형 태스크 집합(LTS, Linear Task Set) 모델을 제안하였다. LTS는 태스크 간의 데이터 흐름을 관리하는 부분과 처리 시간이 다른 태스크 간의 작업 흐름을 관리하는 부분 그리고 데이터 흐름에 따른 버퍼 관리하는 부분으로 구성된다. 따라서 본 논문에서는 이 모델을 바탕으로 멀티미디어 스트리밍 시스템의 기본적인 컴포넌트 모델을 제시하고자 한다. 또한 기본 컴포넌트를 조립하여 확장할 수 있고, 새로운 알고리즘이나 아키텍처를 추가하여 새로운 컴포넌트를 생산해 낼 수 있는 확장성 및 이식성이 강하게 설계하고자 한다[5].

II. 선형 태스크 집합(LTS : Linear Task Set) 모델에 대한 제안

2.1 필터와 파이프 모델

본 논문에서 멀티미디어 스트리밍 시스템에 대하여 확장성과 생산성이 뛰어나고 새로운 기술의 이식성 및 유지보수도 탁월한 시스템 모델을 설명하는 기본적인 모델이다.

이 모델에서는 필터와 파이프로 구성되며, 각 필터(Filter)는 매우 단순한 인터페이스로 다른 필터와 연결되어 있다. 또한, 입력을 받는 파이프(pipe)를 통해서 메시지를 전송 받은 필터는 자신이 처리하고자 하는 범위의 작업을 처리하고, 처리 결과에 대하여 출력하는 파이프를 통하여 다음 작업을 처리할 필터로 보낸다. 파이프는 하나의 필터에서 다음 필터와 연결되고, 하나의 필터로부터 다음 필터로 출력 메시지를 전송하는 것을 의미하는 기본적인 패턴 모델이다.

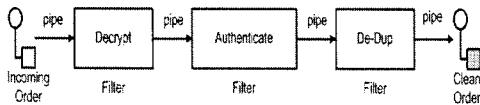


그림 1. Filters and Pipes 패턴

그림 1은 [4]에서 제시한 "Filters and Pipes" 패턴의 구성도로서 암호화된 주문에 대한 메시지가 해독필터(Decrypt)에 전달되면 수신된 메시지에 대하여 암호를 해석하는 기능을 처리하고 다음 필터로 전달하게 된다. 그 다음 인증처리 필터는 파이프를 통하여 전달받은 데이터에 대하여 인증 처리를 한 후 그 결과를 다음 필터로 전송하는 것을 보여 주고 있다. 즉, 필터와 파이프는 하나의 프로세스 순서에 대해 기능별로 분할하여 객체화하였고, 객체 간에 동일하게 파이프(pipe)

로 연결함으로써 각 객체 간의 의존도가 떨어져 고 단순한 구조로 구성되어 객체와 객체 사이에 새로운 기능의 객체를 추가할 수 있다.

2.2 필터의 집합

시스템의 작업 구조는 확장성과 이식성을 높이기 위하여 종속성과 결합도를 낮추어야 한다. 즉, 시스템의 작업들은 작업과 작업 간에는 독립적이어야 하며 임의의 작업이 다른 작업에게 직접적인 영향을 주지 않는 범위에서 태스크 객체는 단순해야 한다. 단순한 태스크 객체는 자유롭게 주어진 작업에만 집중할 수 있으며 서로 다른 기능을 갖는 태스크들 간의 데이터를 전달하는 방법이 정의되어야 한다.

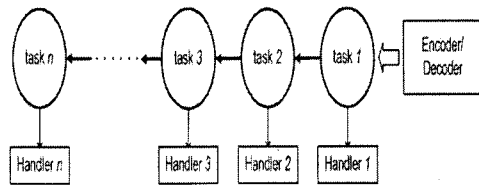


그림 2. 태스크 프로세스의 선형 구조

스트림을 전송만 처리하기 위한 모델이라면 시스템의 작업은 단방향으로 구성되며 그림 2와 같이 task₁에서 task_n으로 데이터가 태스크를 통과하면서 태스크의 기능이 처리된다. 태스크의 처리 구조가 데이터를 전송 뿐 만 아니라 클라이언트나 다른 시스템에서 보내오는 패킷 정보를 수신하여 임의의 태스크가 분석해서 로그를 만드는 모델이라면 양방향 구조로 구성된다.

선형이 아닌 비선형 구조로 구성 된다면 비선형 그래프 구조는 독립된 태스크에 대하여 스케줄링과 버퍼 관리, 이벤트 관리에서 그래프, 트리 구조의 장점에 비하여 처리시간의 비용이 훨씬 낮게 된다. 실시간에 전송되어야 하는 멀티미디어 스트리밍 시스템에서 처리시간의 단축은 서비스의 질을 향상시킬 수 있기 때문에 선형 리스트 구조가 그래프보다 연산 시간이 더 효율적이다.

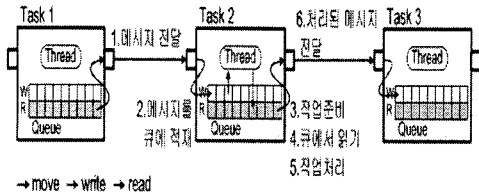
다음은 선형 태스크 집합의 구성에 대한 시스템 설계 시 고려해야 할 사항들을 나열하였다.

- ① 작업 구조 간단하고 명확성
- ② 독립적인 비종속적인 태스크 설계
- ③ 확장성 가능한 컴포넌트 정의
- ④ 단순화 작업 구조로 확장성 고려
- ⑤ 컴포넌트 간의 데이터 전송
- ⑥ 컴포넌트 간의 동기화
- ⑦ 컴포넌트 재배치

2.3 메시지 블록 전송 흐름

LTS는 태스크들 간의 독립성을 보장하며 데이터를 처리하기 위하여 태스크와 태스크의 데이터 전송을 일관성 있게 처리해야 한다. 태스크와 태

스크 사이의 메시지 블록을 전송하는 흐름은 그림 3과 같이 메시지를 전달하고 작업한 후 다음 태스크로 보내진다. 이 전송 흐름은 그림 3에서 보여지는 것과 같이 순차적인 작업을 표현한 것이 아니라 3개의 태스크 객체가 스레드를 상속 받고 있기 때문에 task1에서 task3로의 흐름은 동시에 발생하게 된다. 작업은 다음과 같이 정의되며 순차적으로 발생하는 것이 아니라 스레드에 의하여 6가지의 작업이 거의 동시(concurrent)에 처리된다.



→move → write → read

그림 3. 태스크 간의 메시지 전송 흐름

- ① 이전 컴포넌트에서 메시지를 전달한다.
- ② 전달받은 메시지를 자신의 메시지 큐에 저장한다.
- ③ 자신이 처리할 작업을 준비한다.
- ④ 자신의 작업을 처리하기 위하여 큐에서 메시지가 존재하면 읽는다.
- ⑤ 주어진 작업을 처리한다.
- ⑥ 자신의 작업이 끝나면 처리된 메시지를 다음 태스크에 전달한다.

III. 선형 태스크 집합(LTS : Linear Task Set)을 기반으로 한 컴포넌트 모델

3.1 컴포넌트 등록

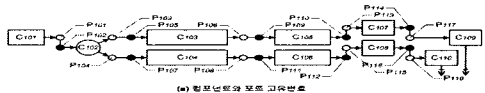
멀티미디어 스트리밍 시스템을 구성하는 컴포넌트들은 모두 시스템에 등록되어야 한다. 2장에서 제안한 선형 태스크 집합의 구조는 작업을 선형으로 처리하는 것을 목적으로 하였지만, 시스템을 구성하는 컴포넌트의 관계를 복잡하기 때문에 명확하게 하기 위하여 컴포넌트 정보를 저장하고 컴포넌트를 연결하는 포트들에 대한 정보를 관리한다. 포트들의 정보는 컴포넌트의 연결 정보로서 활용될 뿐만 아니라 컴포넌트들의 입출포트의 동기화를 위한 명확한 지점을 명시할 수 있다. 또한 컴포넌트를 등록하고 컴포넌트를 연결할 때 포트의 무결성을 지킬 수 있는 정보를 관리해야 한다.

컴포넌트 테이블은 컴포넌트 고유번호(C_{ID} , Component ID)와 컴포넌트 이름(C_{NAME} , Component Name)으로 구성되며, 포트 테이블은 포트고유번호(P_{ID} , Port ID), 소유자번호(O_{ID} , Owner ID), I/O 구분, 활성화 구분, 참조번호(R_{ID} , Reference ID)로 구성된다. 참조 번호는 컴

포넌트의 포트와 연결된 다른 컴포넌트의 포트 고유번호를 나타낸다.

그림 4의 (a)에서는 구성된 컴포넌트들에 한 컴포넌트 고유번호와 포트 고유번호를 그림 4의 서버 모델 부분과 대응하여 보여주고 있다. 컴포넌트를 등록하게 되면 그림 4의 (b)와 같이 컴포넌트들은 고유번호와 고유이름을 갖게 되며, 각 컴포넌트들이 소유하고 있는 포트들에 대해서는 그림 4의 (c)의 포트 테이블에 등록이 된다. (b)에서 C_{105} 의 컴포넌트가 (c)의 포트테이블에 입력포트 P_{109} 와 출력 포트 P_{110} 를 소유주로 등록되어 있고, 두 포트에 대해서 입출력 여부와 활성화 포트 여부를 확인할 수 있다. C_{105} 컴포넌트의 활성화 및 입력포트 P_{109} 는 컴포넌트 C_{103} 의 비활성화 출력 포트 P_{106} 와 연결되어 있고, 활성화 출력포트 P_{110} 은 컴포넌트 C_{107} 의 비활성화 포트 P_{113} 과 연결되어 있음을 알 수 있다.

컴포넌트를 등록하고 컴포넌트들의 포트를 연결하는 시점에서 포트들이 연결이 가능한 상태인지를 검사해야 한다.



(a) 컴포넌트와 포트 고유번호

Comp. Table		PortTable			
CID	CNAME	PID	CID	IO	ACT
C101	Queue	P101	C102	I	A
C102	SRC	P102	C103	I	A
C103	TS	P103	C104	I	A
C104	Video Decoder	P104	C105	I	A
C105	Audio Decoder	P105	C106	I	A
C106	Video Encoder	P106	C107	I	A
C107	Audio Encoder	P107	C108	I	A
C108	BUF1	P108	C109	I	A
C109	NET1	P109	C110	I	A
C110	SyncrGizmo	P110	C105	O	A
C110	SyncrGizmo	P111	C107	O	A
C110	SyncrGizmo	P112	C108	O	A
C110	SyncrGizmo	P113	C109	O	A
C110	SyncrGizmo	P114	C110	O	A

(b) 컴포넌트 등록 테이블

그림 4. 컴포넌트 등록 테이블과 포트 등록 테이블 예

3.2 컴포넌트 시스템 모델

다양한 멀티미디어 스트림을 지원하고자 할 경우, 다양한 포맷에 대한 모든 처리를 하는 것보다 하나의 목표 스트림으로 변환하고 그 스트림을 전송한 후 다시 스트림을 원하는 포맷으로 변환하는 트랜스코딩 방식과 MPEG-2에서 지원하는 공간해상도와 시간해상도의 제어 기법 두 가지를 적용할 수 있다.

스트리밍을 전송하는 서버에서 오디오와 오디오 소스에 대하여 다양한 포맷을 수용할 수 있도록 하기 위해 소스에 대한 포맷을 변환하는 트랜스코더(transcoder)는 MPEG-2 스트림으로 변환한다. 변환된 스트림은 디코더에 의해 DCT 양자화 계층을 통하여 공간 스케일러빌리티(Spatial Scalability)를 지원한다.

QoS 맵퍼(QoS mapper)는 Combiner로부터 프레임을 뽑아내고 패킷의 중요도와 사용자의 요구사항을 바탕으로 시간적 공간적 최적화된 QoS 파라미터 정보와 우선순위 정보를 설정한다. 맵퍼

에서 사용자가 프레임 재생률이나 낮은 에러율을 선택할 수도 있으며, 이보다는 해상도에 더 중점을 줄 수도 있다. 이렇게 사용자의 요구사항과 패킷의 중요도에 따라 우선순위를 설정하고 재정렬 후 버퍼에 우선순위대로 저장한다. 또한, QoS 맵퍼는 디코더에 공간해상도와 시간해상도를 조정할 수 있는 파라미터를 제공하여 시공간 해상도를 조절하는 파라미터로 사용할 수 있게 한다.

전송컨트롤러는 전송할 패킷을 버퍼로부터 꺼내어 QoS 파라미터를 체크하고 요구 사항에 맞게 최적화된 전송한다. QoS에서 설정한 패킷에서 타임스탬프와 우선순위 정보를 바탕으로 우선순위 스트림 전송을 한다. 네트워크 상태가 지정된 시간 내에 패킷이 도착하지 못할 상태라고 판단될 때 우선순위가 낮은 패킷에 대해 폐기한다.

클라이언트에서는 네트워크 컴포넌트에서 수신되는 이벤트를 모니터링 하기 위한 출력포트가 활성화 속성을 갖고 있다. 수신되는 패킷들은 타임스탬프 순서대로 패킷들을 정렬한 후 버퍼에 저장한다. 버퍼에 데이터가 시스템에서 정해진 상한과 하한치의 버퍼량에 도달하면 트랜스코딩을 시작하도록 이벤트를 통보한다. 버퍼는 상한치와 하한치를 모니터링하면서 지속적으로 이벤트를 통보하여 다른 컴포넌트들이 참고하도록 한다. 트랜스코더와 디코더는 정렬된 패킷을 재생 버퍼로 푸시(push)하여 MPEG이나 이미지 포맷으로 패킷 데이터를 변경한다. 변경된 데이터는 복합기를 통하여 동기화기(synchronizer)에 전달한다. 동기화기는 오디오와 비디오 스트림의 타임스탬프를 이용하여 동기화 작업을 하며, 내부 버퍼가 비워지면 입력포트는 복합기에서 스트림이 존재하는지 확인하고 없으면 버퍼가 비었음을 시스템에 통보한다. 그림 5에서는 전체 시스템에 대한 구성 모델을 보여준다.

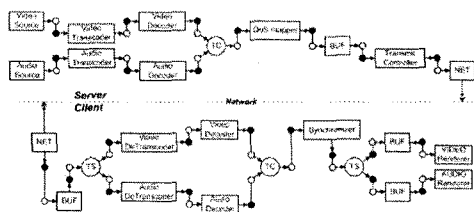


그림 5. 컴포넌트 시스템 구성 모델

V. 결론

본 논문에서는 멀티미디어 스트리밍 서비스를 위한 시스템의 복잡한 작업흐름을 단순한 구조로 모델링하여 확장성이 뛰어난 LTS 모델을 설계하였다. 또한 태스크와 태스크 간의 일관성을 데이터 송수신을 유지하도록 하였다. LTS에서는 태스크 간의 느슨한 제어 메시지를 주고받았다. 하지만, 독립적인 태스크들 사이에서 강력한 제어처리

는 태스크들의 결합도만 높이는 효과가 있다. 이를 태스크에 제어와 동기화를 위한 인터페이스와 각 태스크의 작업을 구성하기 위한 정보를 확장하여 컴포넌트를 구성하였다. 컴포넌트는 서로 연결 가능하고 정보 교환 뿐 만 아니라 서로의 처리를 조절하기 위한 포트의 속성을 갖고 있어 협업이 가능하고 그림 5와 같이 멀티미디어 시스템에서의 관리해야하는 위치를 좀 더 명확하게 분석할 수 있었다.

참고문헌

- [1] Steven Bogaerts, David Leake, "IUCBRF : A Framework For Rapid and Modular Case-Based Reasoning System Development", IUCBRF, TECHNICAL REPORT 617, Aug. 2004
- [2] A. P. Black, J. Huang, R. Koster, J. Walpole, C. Pu, "Infopipes : An Abstraction for Multimedia Streaming", Multimedia System Vol. 8, pp. 406-419, Jan. 2002.
- [3] Feng Wu, Honghui Sun, Guobin Shen, Shipeng Li, Ya-Qin Zhang, Bruce Lin, Ming-Chieh Li, "SMART: An Efficient, Scalable and Robust Streaming Video System", EURASIP Journal on Applied Signal Processing, Special issue on Multimedia over IP and Wireless Networks, Vol. 2, pp 192-206, Jan. 2004.
- [4] T.D.C Little, A. Ghofoor, "Distributed Multimedia Objects Composition and Communication", IEEE Network Magazine, pp. 72-84, Nov. 1990.
- [5] 정규수, "DirectShow Framework을 기반으로 한 분산 멀티미디어 스트리밍 시스템 설계 및 구현", 군산대학교 박사학위논문, 2007