VOD 스트리밍 데이터를 위한 변형된 Scope Consistency 알고리즘 설계

김기환*, 장승주*
*동의대학교 컴퓨터공학과
e-mail:{khkim, sjjang}@deu.ac.kr

Design of the Modified Scope Consistency Algorithm for VOD Streaming Data

Ki-Hwan Kim*, Seung-Ju Jang*
*Dept of Computer Engineering, Dong-Eui University

요 약

본 논문은 VOD 스트리밍 데이터에 대한 변형된 Scope Consistency 알고리즘 설계 방안을 제시한다. Consistency Model 은 주로 전자 계산에서 데이터 동기화에 적용될 수 있는 메커니즘이다. 그 중 Scope Consistency는 Memory Consistency Model의 한 종류로, Scope와 같은 lock을 사용하여 임계영역을 정의한다. Scope Consistency를 지원하는 시스템으로 JIAJIA가 있다. JIAJIA는 lock 기반 캐쉬 일관성을 구현한다. 본 논문에서는 JIAJIA에서 사용하는 동기화 메커니즘 중 barrier 메커니즘을 이용하여 일반적으로 전자 계산에서 사용되는 메커니즘을 VOD 스트리밍 데이터가 처리될 수 있는 변형된 Scope Consistency 알고리즘을 설계한다.

1. 서론

초고속 통신망을 통한 인터넷의 발달은 많은 사회적 변화를 일으켰지만, 그 중에서도 제일 큰 변화를 가져온 것은 영상 산업 분야일 것이다. 인터넷을 통한 여러 매체가 활성화되면서 VOD 서비스는 일반화되고 있다. VOD에 대한 수요가 급증하여 다양한 VOD 서비스가 도입되고 있는 상황에서 데이터 및 사용자의 증가에 따른 문제점을 해결하는 것이 필수적이다. VOD 서버의 문제점을 해결하기 위해 병렬 VOD 서버에 대한 여러 연구가 진행되고 있다. 병렬 VOD 서버는 저가의 일반 시스템을 통해 여러 대 클러스터링으로 연결하여 높은 가용성을 유지하면서 사용자들의 요구를 보다 많이 수용하면서 추후 클라이언트 및 데이터의 사용의 증가에 따라 유용하게 대비할 수 있도록 하고 있다.

병렬 프로그래밍 기법 중 다른 머신간에 추상적인 공유 메모리를 프로그래머에게 제공하는 것은 공유된 메모리의 다중 복사를 초래한다. 이들 복사들의 일관성을 유지해야 하며, 공유 메모리에 접근한 어떠한 프로세서는 올바른 결과를 반환해야 한다. 이 작업을 Memory Consistency Model이 담당한다.

본 논문은 VOD 스트리밍 데이터의 처리 방식을 Memory Consistency Model 중 Scope Consistency[1]의 Barrier 메커니즘 개념을 도입하여 변형된 Scope Consistency 알고리즘을 설계한다.

본 논문의 구성은 다음과 같다. 2장에서는 VOD 스트리밍에 대한 관련연구 및 JIAJIA를 기술한다. 3장에서는 Consistency Model의 종류 및 기능에 대해 살펴본다. 4장에서는 VOD 스트리밍 데이터 처리를 위한 변형된 Scope Consistency 알고리즘을 제시한다. 5장에서는 결론을 맺는다.

2. 관련 연구

2.1 VOD 스트리밍

VOD(Video On Demand)와 AOD(Audio On Demand) 같은 주문형 멀티미디어 서비스를 지원하는 대표적인 프로토콜로는 H.323, SIP, RTSP 등이 있는데, 이들 프로토콜들은 RTP(Real-time

이 논문은 2004년도 BB21 연구비를 지원받았음.

Transport Protocol)[4]를 멀티미디어 데이터 전송 프로토콜로 채택하고 있다. RTP에 의해서 제공되는 서비스들은 time reconstruction, loss detection 그리고 content identification을 포함한다. 멀티미디어 컨테츠를 수신하여 재생하기 위해서는 적절한 타이밍을 요구하는데, RTP는 이를 위해서 time stamping, sequence numbering과 같은 메커니즘을 제공한다. 이 메커니즘을 이용하여 RTP는 종단간 실시간 데이터 전송을 하여 어플리케이션 측에서 timestamp, sequence number를 이용하여 서로 다른 스트림을 동기화시킨다. 본 논몬에서는 RTP 프로토콜 전송 패킷을 처리하기 위해 Scope Consistency Model을 제안한다.

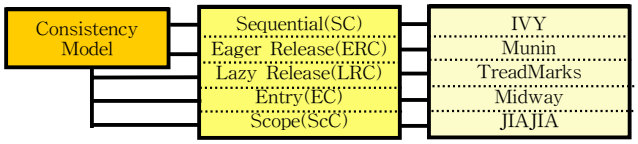
2.2 JIAJIA

JIAJIA[2]는 scope consistency를 지원하는 소프트웨어이다. JIAJIA는 두 가지 주요 특징을 가지고 있다. 첫째로, NUMA와 같은 구조를 가지고 있으며, 더욱 큰 공유 메모리 공간을 형성하기 위하여 다중 컴퓨터의 물리적인 메모리 공간을 조합한다. 둘째로, lock 기반 캐쉬 일관성 규약을 구현하며 lock에서 가지게 되는 write notice에 접근함으로써 일관성을 유지한다.

JIAJIA에서는 세 가지 동기화 기법을 제공하는데, lock, barrier와 조건 변수가 있다. 그 중 barrier는 모든 프로세스들의 처리가 완료될 때까지 어떠한 프로세스로부터의 처리를 금지함으로써 전역적인 동기화 기법을 제공한다.

3. Consistency Model

분산 공유 메모리는 모든 프로세서들이 하나의 유일한 메모리가 사용되어지는 것과 같은 느낌을 준다. 실제로 각 프로세서는 자기 자신이 소유한 메모리를 단지 접근한다. 그러므로 분산 공유 메모리는 요구되어지는 다른 프로세서로부터의 메모리 내용을 가져올 수 있어야 한다. 이것은 다른 물리적인 메모리에서 같이 공유된 메모리의 다중 복사를 초래한다. 다중 복사는 일관성(consistency)을 유지해야 하며, 공유 메모리에 접근하는 어떠한 프로세서는 올바른 결과를 반환해야 한다. 이러한 작업은 *Memory Consistency Model*[3]이 담당한다. 다음 [그림 1]은 여러 종류의 Memory Consistency Model 등을 나타낸다



[그림 1] Memory Consistency Model의 종류

본 연구에서 살펴보고자 하는 것은 Scope Consistency(ScC) Model을 사용하는 JIAJIA를 바탕으로 VOD 스트리밍 데이터에 대한 동기화 기법을 적용하였다.

3.1 Sequential Consistency

Lamport에 의해 처음으로 제안된 모델이다(1979). 공유 메모리 다중 프로세서를 고안된 가장 일반적인 메모리 일관성 모델은 Sequential Consistency이다. Lamport에서는 다음 <표 1>과 같이 정의한다.

<표 1> Sequential Consistency의 정의

정의 : 어떠한 실행 결과가 마치 모든 프로세서의 오퍼레이션이 임의의 순차적인 순서로 실행했던 것과 같고, 각 프로세서의 오퍼레이션들은 프로그램에 의해 명시된 명령의 순서로 나타난다.

Sequential Consistency에서 나타나는 두 가지 경향이 있다. 하나는 개개의 프로세서들로부터 오퍼레이션들간에 프로그램 연속순을 유지하고 있으며, 다른 하나는 모든 프로세서들로부터 오퍼레이션들간에 단일 연속순을 유지하는 것이다.

3.2 Release Consistency

Release Consistency는 원래 메모리 접근과 acquire나 release로 정의된 동기화 접근으로 구별된다. 이러한 가능한 프로그램을 만들기 위해서는 "*properly labeled*"이어야 한다. Release Consistency는 두 가지 모델을 제시하는데, Eager Release Consistency(ERC)과 Lazy Release Consistency(LRC)가 있다.

P1	P2

lock L1;
ptr=non_null_ptr;
unlock L1;
	while(ptr==null){};
	lock L1;
	a=ptr;
	unlock L1;

[그림 2] LRC와 ERC의 차이점에 대한 예제

[그림 2]는 ERC와 RC에서 볼 수 있는 동작으로 새로운 non-null pointer가 P1에 의해 unlock(release)전에 P2로 전달되는 것을 보장한다.

LRC일 경우, P2가 lock을 실행할 때까지 P1은 쓰기를 하지 않는다. while loop에서 read가 있기 전에나 synchronization으로 분류하기 전에, 적절한 acquire synchronization이 위치해야한다.

3.3 Entry Consistency

Entry Consistency 모델은 Midway 소프트웨어 공유 메모리 시스템을 위해 고안되었다. Midway는 페이지-기반 시스템이라기보다는 변수 또는 지역-기반 시스템이다.

P0	P1
Lock_Bind(L0,X0) Acquire_Exclusive(L0) X0=1 Release(L0)	Lock_Bind(L0,X0) Acquire_Non_Exclusive(L0) a=X0 Release(L0)

[그림 3] Entry Consistency에서 동기화 기법

[그림 3]은 entry consistency의 일관성 규칙들을 나타내는 <표 2>로 설명한다.

<표 2> Entry Consistency의 일관성 규칙

(1) acquire가 실행되는 것을 허용하기 이전에, 보호된 공유 데이터에 대한 모든 업데이트는 프로세스에 대하여 실행되어져야 한다. (2) lock이 exclusive mode에서 취득될 때, 다른 프로세스는 lock을 취득할 수 있는 것은 없고 non-exclusive 모드에서도 마찬가지다. (3) lock이 exclusive 모드에서 취득된 후, 다른 프로세스에 의해 실행된 lock의 다음 non-exclusive acquire는 lock의 소유자에 대하여 실행되고 난 후에만 실행되는 것을 허용한다.

Entry Consistency는 다른 relaxed consistency model이다. release consistency에서, consistency 동작들은 synchronization operations과 함께 결합하여 가져진다.

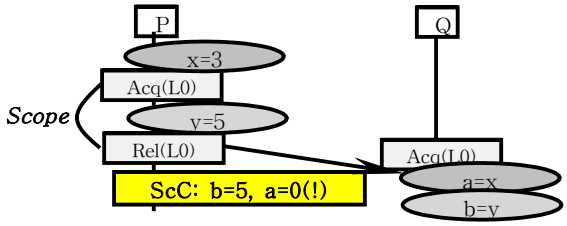
3.4 Scope Consistency

ScC에서, scope 개념을 같은 lock을 사용하는 모든 임계영역들로써 정의한다. 이것은 lock이 scope를 확실히 정의한다라는 의미로 이해를 한다면 쉬운 개념이다. scope는 **acquire** 에서 ***opened***되고 release에서 ***closed***된다. ScC는 다음<표 3>과 같이 정의한다.

<표 3> Scope Consistency의 정의

프로세서 Q는 다른 프로세서 P에 의해 closed되기 전에 scope를 열려고 할 때, P는 Q에서 같은 scope 내에 만들어진 업데이트를 전달한다.

<표 3>에 대한 내용은 [그림 4]를 통해 쉽게 이해할 수 있다.



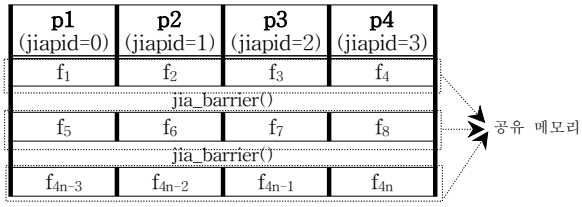
[그림 4] Scope Consistency에 대한 동작 예

[그림 4]는 <표 3>에 대하여 묘사된 그림이다. 프로세서 Q가 L0를 취득함에 따라 scope를 열 때, P에서 같은 scopes 내에 업데이트는 Q로 전달된다. 그러므로, b는 5를 읽는 것이 보장되고, P에 의해 업데이트된 y값을 읽을 수 있다. 그러나, a는 3을 읽는 것을 보장하지 못한다. 왜냐하면, P는 L0에 의해 보호되는 scope 밖에서 x를 업데이트하기 때문이다. ScC는 적용 데이터 양을 전달하기 때문에, 효율성이 증가한다.

4. 변형된 Scope Consistency 알고리즘

4.1 Barrier 메커니즘 활용 방안

JIAJIA라이브러리를 사용한 LU factorization에서 barrier 메커니즘을 활용하여 설계한다. JIAJIA 환경에서의 스트리밍 데이터 읽기는 다음 [그림 5]와 같다.



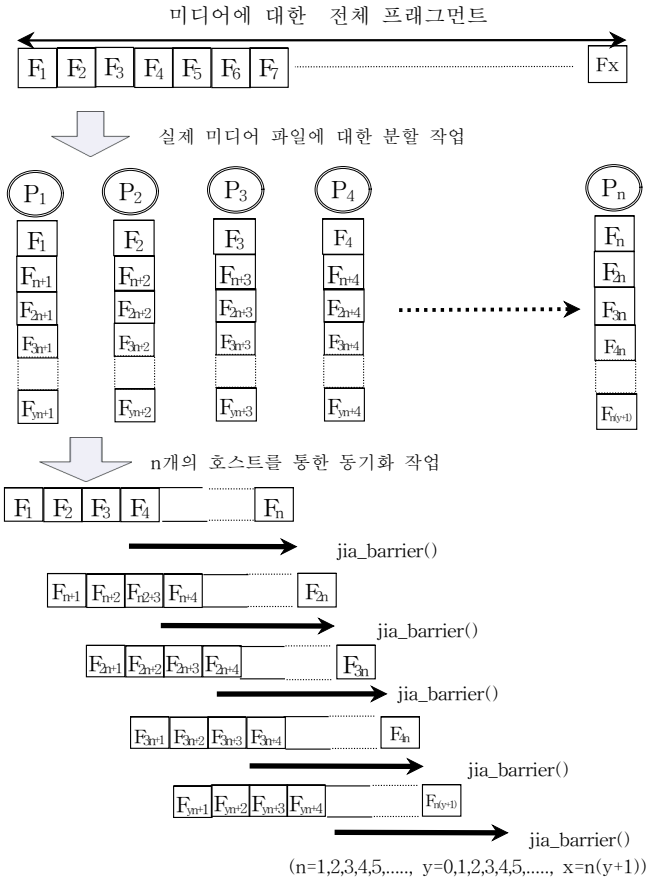
[그림 5] 미디어 파일의 프래그먼트 읽기

[그림 5]에서는 barrier가 적용되어야 하는 지점을 말하는 것이다. p1부터 p4까지 4대의 호스트가 있다고 가정하였을 때 프래그먼트 단위를 구성하는 f_x은 4개의 프래그먼트가 합쳐지면서 공유메모리의 크기를 만족할 때까지 barrier 메커니즘을 통한 동기화를 유지한다.

4.2 n대의 호스트에 대한 VOD 스트리밍 데이터 처리 방안

공유 메모리의 크기는 (n× 프래그먼트의 크기)으로

설정한다. 하나의 호스트는 한 프래그먼트의 크기에 대해서만 동작할 수 있는 공유메모리를 할당한다.



[그림 7] Barrier 메커니즘을 통한 미디어 파일의 동기화

[그림 7]은 형성된 공유메모리를 통하여 Barrier 메커니즘을 적용한 그림이다. 미디어 파일을 구성하는 프래그먼트들은 F_1부터 F_x까지 구성하였다. 이들 프래그먼트들은 라운드로빈방식으로 각 호스트의 스토리지에 저장되도록 한다. 실제 미디어 파일에 대한 분할 작업에서 호스트 P_n은 $F_{n(y+1)}(y=0,1,2,3,4,\cdots,$ n=호스트개수) 프래그먼트들이 배치될 수 있도록 하였다. 공유메모리로 불려들이는 각 호스트의 프래그먼트들은 barrier 메커니즘을 통하여 제어한다. barrier 메커니즘은 공유 메모리에 대한 사용이 모든 호스트의 프로세서에서 작업이 완료될 때까지 반드시 대기하고 있어야 한다.

4.3 미디어 파일 처리를 위한 동기화 알고리즘

JIAJIA 라이브러리를 사용하여 프래그먼트 단위를 처리하기 위한 barrier 동기화 메커니즘을 다음 [알고리즘 1]과 같이 설계한다.

```
미디어 재생 어플리케이션 각 호스트로 복사
각 호스트의 프래그먼트 처리를 위한 공유 메모리 설정
while(미디어에 대한 프래그먼트를 다 읽을 때까지)
{
   if(jiapid에 따른 n이면)
   {
      n번째 호스트의 스토리지에 위치한 프래그먼트 읽기
      공유 메모리에 읽어 온 프래그먼트를 저장
      if(공유 메모리의 프래그먼트 크기가 n이면)
      {
         RTP패킷 전송 단위로 결정
      }
   }
   공유 메모리의 프래그먼트 저장을 위한 동기화
}
```

[알고리즘 1] 미디어 파일의 동기화 알고리즘

[알고리즘 1]은 제안한 알고리즘에 대해서 의사코드로 나타내었다. 미디어 파일 재생 프로그램을 각 호스트로 복사함으로써 미디어에 따라 각 호스트의 프래그먼트 처리를 분리시켜 공유메모리가 형성된다. 그리고 전체 미디어 파일의 프래그먼트에 대한 읽기 작업이 완료될 때까지 n번째 위치하는 프래그먼트를 읽으면서 공유메모리를 형성한다. 형성된 공유메모리의 데이터는 동기화를 유지하며, 다음 프래그먼트 처리를 대기하고 RTP 패킷 전송 단위를 결정한다.

5. 결론

본 논문에서는 VOD 스트리밍 데이터에 대한 Consistency Model로 Scope Consistency를 선택하여 미디어 파일에 대한 동기화 모델을 설계하였으며, 실제 RTP 패킷의 전송하게 되는 단위를 결정하기 위해 barrier 메커니즘을 적용하였다.

실제 미디어 파일에 대한 분할 작업을 통해 이루어지는 동기화로부터 제안한 본 논문의 알고리즘은 VOD 서버의 미디어 파일 접근 방식을 분산 처리하여 성능을 향상시킬 수 있는 방안이라 고찰된다.

참고문헌

[1] L.lftode, J.P. Singh and K.Li. "Scope Consistency: A Bridge between Release Consistency and Entry Consistency". In Proc. of the ACM Annual SPAA'96, p.277-287, June 1996.

[2] W. Hu, W. Shi, Z. Tang, M.Rasit. Eskicioglu "JIAJIA User's Manual", June 3, 1998.

[3] The JUMP Software DSM Software, http://www.srg.csis.hku.hk/srg/html/jump.htm

[4] H. Schulzrinne, et al., "RTP: A Transport Protocol for Real-Time Application", RFC 1889, Jan. 1996