

효율적인 공동저작 지원을 위한 2-레벨 캐싱 기법

김영진, 민승정, 오삼권
호서대학교 컴퓨터공학과

yjkim@osk.hoseo.ac.kr, sjmin@osk.hoseo.ac.kr, ohsk@dogssuri.hoseo.ac.kr

The Two-Level Caching Mechanism for Supporting Efficient Collaborative Authoring

Youngjin Kim, Seungjung Min, Sam K. Oh
School of Computing, Hoseo University

요 약

분산환경에서의 멀티미디어 공동저작은 분산되어 있는 정보와 데이터를 사용하여 작업이 이루어진다. 그러므로 사용자들이 원격지 시스템에 있는 공유객체를 망을 통하여 액세스할 때 데이터 전송으로 인한 작업 지연을 최소화하기 위해 원격지의 데이터중 필요 내용들을 지역 시스템에 효율적으로 캐싱하는 기능이 요구된다. 그러나 공동저작 시 발생하는 전송지연을 최소화하기 위한 캐싱 기법에 대한 연구는 아직 미흡한 실정이다. 본 논문에서는 공동저작에서의 효율적인 캐싱을 위하여 저작 데이터의 특성과 시간, 공간지역성을 고려한 2-레벨 캐싱 기법을 제시한다.

1. 서론

분산환경에서의 멀티미디어 공동저작 관리 시스템은 저작자들이 분산환경에서 멀티미디어 자원을 공유하고 시칭작적인 의사소통을 통해 효율적인 저작이 가능하도록 관리 및 조정을 담당하는 시스템이다[1]. 공동저작은 분산되어 있는 정보와 데이터를 사용하여 작업이 이루어진다. 그러므로 사용자들이 원격지 시스템에 있는 공유객체들을 망을 통하여 액세스할 때 데이터 전송으로 인한 작업 지연을 최소화하기 위해 원격지의 데이터 중 필요 내용들을 지역 시스템에 효율적으로 캐싱하는 기능이 요구된다. 특히 공동저작의 동기적 표현 시 데이터 전송지연은 시스템의 성능에 커다란 영향을 끼친다. 그러나 공동작업 시 발생하는 전송지연을 최소화하기 위한 캐싱 기법에 대한 연구는 아직 미흡한 실정이다. 멀티미디어 공동저작에서의 본 연구는 '98년도 정보통신부 대학기초연구지원을 받아 행해졌음

저작문서는 단일미디어 공동저작과는 달리 하이퍼텍스트(hypertext)기반이다[2]. <그림 1>은 멀티미디어 문서구조의 한 예이다.

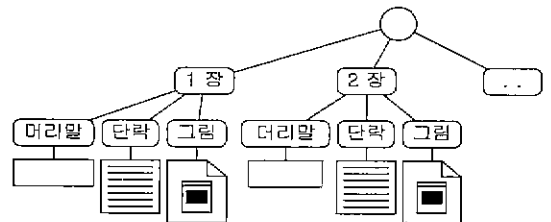


그림 1 문서구조의 예

기존의 캐싱 기법 중 프리페칭(prefetching)은 클라이언트

가 특정 객체를 액세스할 때 객체의 전송으로 인한 지연을 사용자가 느낄 수 없도록 예측된 객체들을 미리 캐싱하는 기법이다[3]. 기존의 프리페칭 기법에서는 사용자의 객체 사용 기록에 근거한 히스토리(history)를 기반으로 캐싱 대상을 결정하였다. 그러나 히스토리에 없는 파일을 빈번히 참조하게 될 경우 히트율(hit ratio)이 현저히 떨어지는 문제가 발생 할 수 있다. 본 논문에서는 이러한 문제점을 보완하기 위해 공동저작 시 하이퍼링크 되어 있는 객체들간의 연관성을 고려하여 프리페칭하고 이를 시간 지역성 기반 캐싱과 공간지역성 기반 캐싱 구조로 나누어 관리하는 2-레벨 캐싱 기법을 제시한다.

본 논문의 구성은 다음과 같다 2장에서 본 논문과 관련된 기존 연구들에 대해 살펴보고 3장에서 본 논문에서 제시하는 2-레벨 캐싱 기법에 대해 소개한다. 마지막으로 4장에서 결론 및 향후 계획을 기술한다.

2. 관련 연구

파일 참조 패턴을 트리구조로 표현하고 현재 참조된 파일을 저장된 패턴과 비교하여 해당 패턴에 명시된 파일들을 미리 캐싱하는 파일 단위의 프리페칭 기법[4]과 파일 참조의 기준을 파일의 개방과 폐쇄로 보고 관련된 여러 파일들을 같은 그룹으로 묶어 캐싱하는 정책을 사용하고 있는 Seer 예측 캐싱 기법[5]은 프리페칭 대상의 결정을 파일참조의 히스토리에 초점을 맞추고 있을 뿐 구체적인 캐쉬 구조에 관해서는 언급하고 있지 않다. 웹 브라우저 환경에서 사용자 시스템의 부하, 용량, 자원을 고려하여 프리페칭 대상을 동적으로 선택하는 네트워크 프리페칭 기법[6]에서는 사용자 참조 히스토리와 하이퍼링크된 객체들의 참조 가능성에 의한 프리페칭 기법을 사용한다는 점에서 본 논문에서 제시하는 프리페칭 방식과 유사하지만 구체적인 캐쉬 구조와 캐싱 방법에 대해서는 언급이 되지 않는다.

3. 2-레벨 캐싱 기법

본 장에서는 문서구조 기반 프리페칭 기법을 설명하고, 시간지역성과 공간지역성을 모두 반영하여 운용하는 2-레벨 캐싱 기법을 제시한다.

3.1 문서구조 기반 프리페칭 기법

<그림 2>는 문서구조 기반 프리페칭 방식의 한 예이다.

이러한 문서구조 기반의 프리페칭은 히스토리 기반의 프리페칭 기법에서의 문제점을 보완할 수 있다 그러나 사용자가 저작위치를 변경하면 캐싱되어 있던 내용을 모두 무시 하고 변경된 위치를 기준으로 다시 캐싱해야 하므로

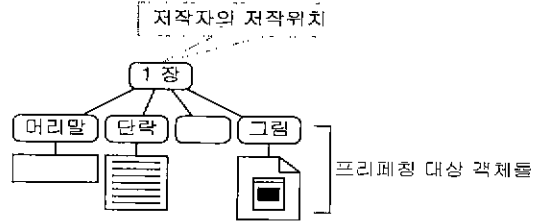


그림 2 문서구조 방식의 프리페칭

전송 지연이 발생할 수 있다는 문제점이 있다.

본 논문에서 제시하는 2-레벨 캐싱 기법은 문서구조에 기반하여 프리페칭된 데이터를 공간지역성 기반의 캐쉬와 시간지역성 기반의 캐쉬에 적용력 있게 적재함으로써 앞서 다른 히스토리 기반의 프리페칭과 문서구조 방식의 프리페칭 기법의 문제점을 보완할 수 있다.

3.1 캐쉬 구조

<그림 3>은 2-레벨 캐쉬의 논리적 구조를 나타내고 있다.

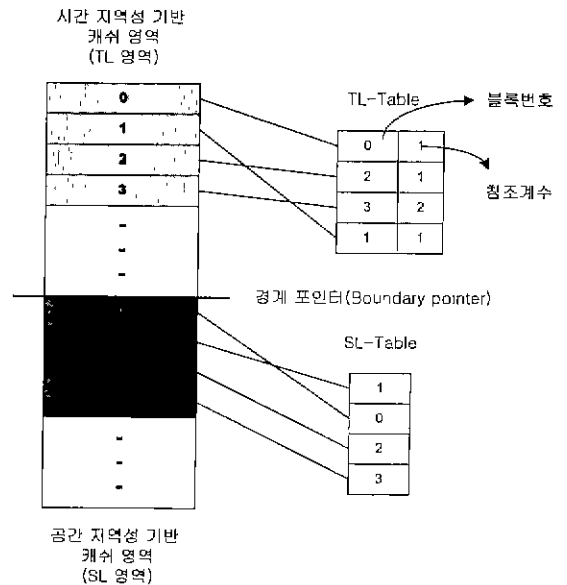


그림 3 두 개의 지역성을 갖는 2-레벨 캐쉬의 논리적 구조

공간지역성과 시간지역성을 기반으로 하는 각각의 캐쉬 영역은 고정된 캐쉬 크기를 갖는 것이 아니라 캐싱 상황

에 따라 적응력 있게 변할 수 있다. 여기서 캐싱 상황이란 사용자의 객체 사용 패턴에 따라 변경될 수 있는 각 캐쉬 영역의 캐쉬 히트율을 말한다. 캐쉬 크기의 적응력 있는 변화는 캐쉬의 가용공간을 충분히 활용하여 캐쉬 영역의 낭비를 줄일 수 있다. TL-Table(Temporal Locality Based Cache Table)은 시간지역성을 적용하여 캐싱한 데이터들의 정보, 즉 블록번호와 참조계수를 유지하기 위한 테이블로 이 테이블에 의해 참조된 캐쉬 영역을 시간지역성 기반 캐쉬 영역(이하 TL영역)이라 한다. SL-Table(Spatial Locality Based Cache Table)은 공간지역성을 적용하여 캐싱한 데이터들의 정보, 즉 블록번호를 유지하기 위한 테이블로 이 테이블에 의해 참조된 캐쉬 영역을 공간 지역성 기반 캐쉬 영역(이하 SL영역)이라 한다.

4. 결론 및 향후 연구

분산환경에서의 분산멀티미디어 공동저작환경에서는 원격지 시스템에 있는 공유객체 데이터의 전송으로 인하여 발생할 수 있는 작업 지연을 최소화하기 위해 필요 데이터들을 지역 시스템에 효율적으로 캐싱하는 기능이 필요하다. 그러나 공동저작 시 발생하는 전송지연을 최소화하기 위한 캐싱 기법에 대한 연구는 아직 미흡한 실정이다. 본 논문은 분산환경에서의 멀티미디어 공동저작을 효율적으로 지원하기 위한 2-레벨 캐싱 기법을 제시하였다. 2-레벨 캐싱 기법은 문서구조에 기반하여 프리페칭된 데이터를 공간지역성 기반의 캐쉬와 시간지역성 기반의 캐쉬에 적응력 있게 적재함으로써 앞서 다룬 히스토리 기반의 프리페칭과 문서구조 방식의 프리페칭 기법의 문제점을 보완할 수 있다. 그러나 앞에서 언급한 2-레벨 캐쉬 구조에 따른 효율적인 캐싱이 이루어지려면 다음과 같은 사항들이 충분히 고려되어야 하며 현재 연구진행 중이다. 첫째, 경계 포인터의 위치 설정 방법이 고려되어야 한다. 사용자의 객체 사용 패턴에 따라 각 영역의 크기를 적응력있게 변경할 수 있어야 한다. 둘째, 각 캐쉬영역의 특성에 맞는 효율적인 캐쉬 교체 정책이 고려되어야 한다. 두 개의 캐쉬영역에 더 이상의 가용공간이 없을 경우 각 캐쉬 영역의 특성에 따라 캐쉬 교체가 이루어져야 한다 셋째, 저작 시스템의 부하, 자원, 네트워크 상황을 충분히 고려한 프리페칭에 대한 연구가 필요하다.

6. 참고문헌

[1]김동성, 이광행, 오삼권 "분산 멀티미디어 시스템에서의

공동저작 관리" 한국 정보처리학회 추계 학술발표논문집, 제4권, 2호 1997.
 [2]김명호, 이윤준 "멀티미디어 개념 및 응용", 홍릉과학 출판사 1997.08
 [3]박재원, 김문경, 엄영익 "이동 컴퓨팅 환경의 분산 파일 시스템" 정보과학회지 16권, 1호, 1998.1.
 [4]Lei, H., Duchamp. D., "Transparent File Prefetching," Technical Report, Dept. of Computer Science, Columbia Univ., New York, NY, 1995.
 [5]G. H. Kuenning, "The Design of the SEER Predictive Caching System", Proceedings of Mobile Computing Systems and Applications 1994, Santa Cruz, CA. December 1994.
 [6]Zhimei Jiang and Leonard Kelenrock, "An Adaptive Network Prefetch Scheme" IEEE Journal on selected in Communications, Vol. 16, No. 3, April 1998.
 [7]이동희, 노삼혁, 민상렬, 조유근 "LRFU : 무한 참조 정보를 이용하는 블록 교체 기법" 정보과학회논문지(A) 24 권, 7호, 1997.7.
 [8]Jonnl Hennessy & David A Patterson "Computer Architecture a Quantitative Approach", Morgan Kaufmann Publishers, Inc., 1990.