

# 캐시 적중률 향상을 위한 웹 서버 클러스터의

## 협력적 캐싱

김희규\* 최창열\* 박기진\*\* 김성수\*  
아주대학교 정보통신전문대학원 정보통신공학과\* 안양대학교 소프트웨어학과\*\*  
(ikki, clchoi, sskim)@ajou.ac.kr kiejin@aycc.anyang.ac.kr\*\*

### Cooperative Caching of Web Server Cluster for Improving Cache Hit Rate

Heegy Kim\* Changyeol Choi\* Kiejin Park\*\* Sungsoo Kim\*  
Graduate School of Information and Communication, Ajou University\*  
Dept. of Software, Anyang University\*\*

#### 요 약

최근 클러스터에 대한 연구는 내용 기반 클러스터의 부하 분배와 캐시 정책이 집중되고 있다. 본 논문에서는 웹 서비스의 고가용성 및 확장성을 제공하는 클러스터 환경에서 힌트 기반 협력적 캐싱의 캐시 적중률을 향상시키기 위해 기존의 DFR 기법을 개선하였다. 서비스 접근 확률을 이용하여 주 복사본과 종속 복사본을 선택적으로 제거하는 메모리 교체 방법을 제시하였으며, DFR 방식과 성능을 비교, 분석한 결과 DFR 방식보다 적은 디스크 접근률을 얻을 수 있었다.

#### 1. 서 론

급속한 인터넷의 성장으로 인해 인터넷의 과부하가 발생하고 있다. 서버의 처리 속도가 빠르게 증가함에도 불구하고 많은 사용자가 이용하는 웹 서버의 경우 자주 과부하가 걸린다. 이러한 과부하로 인한 웹 서비스의 중단은 서비스 제공자와 사용자에게 큰 손실을 가져올 수 있기 때문에 웹 서버는 지속적으로 서비스가 가능하도록 설계되어야 한다.

웹 서버 클러스터가 충분한 성능을 가지기 위해 두 가지 요건이 필요하다. 하나는 효율적인 부하 분배이며, 다른 하나는 적은 디스크 접근이다. 웹 서버 클러스터의 부하 분배기는 제공하는 서비스에 따라 적절한 스케줄링 알고리즘을 통해 부하를 분산시키는 역할을 하며 여러 노드가 하나의 가상서버로 보이게 하는 역할을 한다. 부하 분배 알고리즘은 서버의 성능에 많은 영향을 미치며 노드 선택을 위한 기준은 간단하고 빨리 계산될 수 있어야 한다.

최근 부하 분배에 대한 연구들이 많이 이루어졌으나 잦은 디스크 접근으로 인해 전체 성능 저하가 일어난다. 이러한 문제를 해결하기 위해 내용 기반(Content-based) 부하 분배 알고리즘과 캐싱 정책이 제시되었다. 내용 기반 부하 분배 알고리즘의 대표적인 것으로 LARD(Locality-Aware Request Distribution) 방식이 있다[1]. 이 방식은 같은 내용의 요구를 같은 서버에 보냄으로써 캐시의 지역성(Locality)을 최대한 이용하여 디스크 접근을 줄이는 방법이다.

이 방법 외에 협력적 캐싱 방법이 있는데 유닉스 네트워크 파일 시스템에 적합하도록 만들어진 것이기 때문에 클러스터의 특성이 고려되지 않았다.

본 논문에서는 웹 서비스의 고가용성 및 확장성을 제공하는 클러스터 환경에서 힌트 기반 협력적 캐싱(Hint-based Cooperative Caching)[2]의 캐시 적중률을 향상시키기 위한 방법을 제안한다. 본문의 구성으로 2장에서는 관련 연구에 관해서 언급하고, 3장에서 서비스 접근 확률을 고려한 협력적 캐싱 방법을 설명한다. 4장에서는 시뮬레이션 결과를 설명하고, 마지막으로 5장에서 결론을 맺는다.

#### 2. 관련 연구

##### 2.1. 힌트 기반 협력적 캐싱

힌트 기반 협력적 캐싱은 N-Chance Forwarding[3]과 같이 중앙에서 캐시를 관리하는 것이 아니라 힌트를 사용한다. 힌트는 디스크에서 읽어 캐시에 저장된 블록 위치를 나타내며, 각 노드는 중앙에서 관리자 없이 힌트를 확인하여 캐시에서 원하는 블록의 위치를 찾아낸다. 이 메커니즘은 중앙에 관리자가 없어 N-Chance Forwarding의 경우 같이 관리자에 부하가 발생하는 것을 막을 수 있다.

##### 2.2. Duplicate Copy First Replacement(DFR)

[4]에서는 클러스터에 적합한 협력적 캐싱을 위한 DFR 메커니즘을 제시하였다. DFR은 힌트 기반 협력적 캐싱에 비해 불필요한 블록의 이동과 힌트 수정을 줄이

이 논문은 2003년도 두뇌한국21사업에 의하여 지원되었음.

고 캐시 적중률을 높여 큰 규모의 클러스터에서도 사용 가능하다.

### 3. 개선된 협력적 캐싱 방법

본 논문에서는 힌트 기반 협력적 캐싱과 DFR 메커니즘을 개선한 힌트 기반 협력적 캐싱 정책을 제안한다. 제안된 힌트 기반 협력적 캐싱 정책은 서비스 접근 확률을 이용한 메모리 교체 방식을 사용한다.

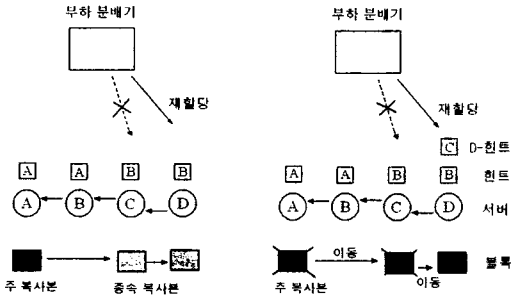


그림 1 힌트 기반 협력적 캐싱      그림 2 DFR

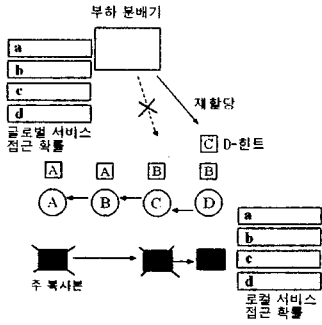


그림 3 서비스 접근 확률을 이용한 메모리 교체 기법

그림 1, 2와 3은 협력적 캐싱 정책들을 나타낸 것으로 모두 힌트를 사용한 협력적 캐싱 정책들이다. 여기서 힌트는 주 복사본(Master Copy)의 위치를 나타낸다. 그리고 주 복사본은 디스크에서 처음 읽어 온 블록을 말한다. 주 복사본은 디스크에서 읽어 왔기 때문에 많은 시간과 비용이 든다. 이외에 힌트 기반 협력적 캐싱에서 사용되는 요소로는 중속 복사본(Duplicate Copy)이 있다. 중속 복사본은 주 복사본에서 복사된 것이며 주 복사본과의 차이는 힌트가 없다는 것이다.

그림 1은 힌트 기반 협력적 캐싱에서 힌트의 사용과 주 복사본과 중속 복사본의 이동을 보여주고 있다. 로컬 캐시에 원하는 블록이 없는 경우 힌트를 보고 다른 노드의 메모리에 있는 주 복사본으로부터 복사하여 중속 복사본을 가져오는 과정을 보여주고 있다[2].

새로 제안한 서비스 접근 확률을 이용한 메모리 교체 방식은 DFR과 마찬가지로 힌트 이외에 D-힌트를 사용한다. D-힌트는 재할당된 서버에게 발급 전에 서비스하

던 노드의 위치이며 부하 분배기로부터 전송된다. 그림 1에서 서버 C의 경우 주 복사본이 서버 B에 있다는 것을 힌트를 통해 알 수 있고 서버 B의 경우 서버 A에 있다고 나타나 있다. 힌트는 주 복사본이 이동할 때마다 이동한 서버만 힌트가 바뀌게 된다. 따라서 서버 C가 주 복사본을 얻기 위해서는 서버 B를 거쳐 서버 A에서 주 복사본을 얻을 수 있다. 이렇듯 주 복사본이 많이 이동하게 되면 원하는 블록을 얻기까지 많은 지연이 일어난다. 이 문제를 해결하기 위해 D-힌트를 사용한다. 그림 2와 3을 보면 D-힌트는 재할당된 서비스를 바로 전에 제공한 노드의 위치이므로 결국 주 복사본의 위치가 되고 바로 그 서버에서 주 복사본을 가져오면 된다. 또 다른 차이점은 그림 1과 같이 힌트 기반 협력적 캐싱은 원하는 블록이 없어 힌트를 사용하여 다른 노드에 있는 주 복사본으로부터 중속 복사본을 가져오는 대신에 그림 2와 3과 같이 DFR과 서비스 접근 확률을 이용한 메모리 교체 방식은 원하는 블록이 없을 때 주 복사본을 복사한 뒤 중속 복사본을 현재 서버에 두고 주 복사본을 요청한 서버로 이동시킨다. 이렇게 함으로써 LRU(Least Recent Used) 정책에 의해 오랫동안 주 복사본을 협력적 캐싱에 가지고 있을 수 있다.

서비스 접근 확률을 이용한 메모리 교체 방식과 힌트 기반 협력적 캐싱, DFR과의 차이는 메모리 교체 정책에 있다. 세가지 방식 모두 LRU 정책을 사용하지만 힌트 기반 협력적 캐싱은 주 복사본을 메모리에서 제거하며, DFR은 중속 복사본을 메모리에서 제거한다. 힌트 기반 협력적 캐싱의 경우 메모리를 교체하려는 노드에서 LRU 방식에 의해 가장 오래된 주 복사본을 선택하게 된다. 그런 다음 글로벌 메모리에서 제거될 주 복사본을 선택하고 로컬 메모리에서 선택된 주 복사본이 글로벌 메모리에서 제거될 주 복사본이 있던 곳으로 이동되며 글로벌 메모리에서 주 복사본은 제거된다. 주 복사본은 힌트를 가지고 있고 디스크로부터 읽어 온 블록이므로 더 많은 비용이 들고 힌트를 자주 수정하게 된다. 따라서 주 복사본을 제거하는 것은 중속 복사본을 제거하는 DFR보다 성능이 떨어진다. 그러나 중속 복사본을 제거하는 방법은 주 복사본을 글로벌 캐시에 최대한 많이 오래 가지고 있게 되기 때문에 글로벌 캐시 적중률은 좋지만 로컬 캐시 적중률은 떨어지게 된다. 글로벌, 로컬 캐시 모두 적중률을 높이기 위해 서비스 접근 확률을 이용한 메모리 교체 방식은 주 복사본과 중속 복사본 중 가장 접근이 적은 블록을 제거한다.

모든 서비스는 같은 확률로 요구되지 않으며 서비스가 요구되는 확률은 전부 다르다. 또한 서비스 요구의 90%가 전체 서비스의 5%에 해당된다는 연구가 있다[5]. 만약 시스템 전체적으로 서비스 요구 확률이 일정하다고 하더라도 내용 기반 클러스터 시스템에서는 각 노드에서 서비스하는 내용이 다르기 때문에 각 노드에서 서비스하는 서비스의 접근 확률은 다르게 나타난다. 따라서 어떤 주 복사본은 자주 요구될 것이며 또 다른 주 복사본은 요구가 거의 없을 수도 있다. 요구 확률이 적은 주 복사본이 임의의 서버에 있다고 할 때 이것이 계속 글로벌 캐시에 있는 것은 캐시 적중률을 낮추는 요인이 될 수 있다. 이런 경우 메모리를 교체할 때 중속 복사본을 제

거하는 것보다 요구 확률이 적은 주 복사본을 제거하는 편이 효율적이라 할 수 있다.

서비스 접근 확률을 이용한 메모리 교체 방식은 부하 분배기에 글로벌 서비스 접근 확률을 가지고 있다. 이것은 주 복사본의 접근 확률이다. 반면에 각 노드는 로컬 서비스 접근 확률을 가지고 있으며, 이는 중속 복사본의 접근 확률이다. 글로벌 서비스 접근 확률은 클러스터 시스템으로 들어오는 총 요청 수로 각 서비스 요청 수를 나눈 값이며 로컬 서비스 접근 확률은 노드에 들어오는 총 요청 수로 각 서비스 요청 수를 나눈 값이다. 한 노드에서 메모리 교체가 이루어 질 때 LRU 정책에 의해 오래된 주 복사본과 중속 복사본의 접근 확률을 비교하게 된다. 그 중 가장 접근 확률이 적은 블록을 제거하게 된다.

DFR은 글로벌 캐시의 적중률을 높여 힌트 기반 협력적 캐시에 비해 좋은 성능을 얻었지만 로컬 캐시의 적중률은 떨어졌다. 하지만 서비스 접근 확률을 이용한 메모리 교체 방식은 로컬 캐시와 글로벌 캐시의 적중률을 적절히 조화시켜 성능을 향상시킬 수 있다. 하지만 단점은 읽기 작업이 주로 일어나는 웹 서비스의 경우에 좋은 성능을 얻을 수 있지만 많은 쓰기 작업이 일어나는 경우 성능 저하가 일어난다. 그러나 이것은 힌트 기반의 협력적 캐싱 방법을 쓰는 클러스터의 경우 모두에게 해당되는 문제점이라 할 수 있다.

#### 4. 성능 평가

이 장에서는 앞에서 제안한 협력 캐싱 정책이 디스크 접근을 줄이는지에 대해 알아보기 위해 시뮬레이션 실험을 수행하였으며 그 결과를 보여준다.

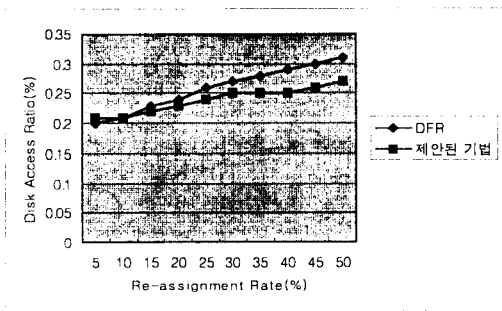


그림 4 디스크 접근 확률 비교

시뮬레이션의 정확성을 위하여 ClarkNet Trace 데이터 로그를 사용하였다[6]. ClarkNet Trace 데이터 로그는 11일간 것이며, 크기는 163MB이다. 또한 중 서비스 요청 수는 1,654,882개이다.

각 서버의 캐시의 크기는 16MB였으며, 캐시 블록 크기는 8KB로 하였다. 캐시 교체 정책은 LRU 방식을 사용하였고, 부하 분배기의 부하 분배 정책은 LARD 방식을 사용하고 있다. 서버(Back-end Server)의 수는 5대로 하였다.

그림 4는 재할당률에 따른 디스크 접근률을 나타낸다.

가로축은 재할당률로 전체 들어온 요청 수에서 재할당된 요청의 수의 비율이며 세로축은 디스크 접근률이다. DFR과 서비스 접근 확률을 이용한 교체 방식을 비교하였으며 서비스 접근 확률을 이용한 교체 방식이 DFR에 비해 적은 디스크 접근률을 보이고 있다.

#### 5. 결론

본 논문에서는 내용 기반 클러스터 시스템에서 부하 분배와 적은 디스크 접근을 위해 힌트 기반 협력적 캐싱 방법을 웹 서버에 적용하였다. 협력적 캐싱 방법 중 DFR과 D-힌트 메카니즘을 기반으로 하여 더 높은 캐시 적중률을 얻기 위해 각 서비스의 접근 확률을 이용하여 메모리 교체를 하였다. 성능 평가를 위해 Trace 데이터에 의한 시뮬레이션을 수행하여 DFR 방식 보다 더 적은 디스크 접근률을 가지는 것을 증명하였다. 향후 연구 과제로 동적 요청을 효율적으로 처리하는 힌트 기반 협력적 캐싱 클러스터에 대한 연구가 필요하다.

#### 참고 문헌

- [1] V. S. Pai, M. Aron, G. Banga, M. Svendsen, P. Druschel, W. Zwaenepoel, and E. Nahum, "Locality-Aware Request Distribution in Cluster-based Network Servers," 8th ACM Conference on Architectural Support for Programming Languages and Operating Systems, Vol. 33, No. 11, pp. 205-216, 1998.
- [2] P. Sarkar and J. Hartman, "Efficient Cooperative Caching Using Hints," 2nd Symposium on Operating Systems Design and Implementation, pp. 35-46, Oct. 1996.
- [3] M. D. Dahlin, R. Y. Wang, T. E. Anderson, and D. A. Patterson, "Cooperative Caching: Using Remote Client Memory to Improve File System Performance," 1st Symposium on Operating Systems Design and Implementation, pp. 267-280, Nov. 1994.
- [4] W. H. Kim, S. H. Park, and D. Park, "Efficient Cooperative Caching for File Systems in Cluster-based Web Servers," 2nd IEEE International Conference on Cluster Computing (CLUSTER 2000), Chemnitz, Germany, Nov. 2000.
- [5] L. Cherkasova and M. Karlsson, "Scalable Web Server Cluster Design with Workload-Aware Request Distribution Strategy WARD," <http://www.hpl.hp.com/techreports/2001/HPL-2001-2R1.html>
- [6] M. Arlitt and C. Williamson, "Web Server Workload Characterization: The Search for Invariants," ACM SIGMETRICS Conference on the Measurement and Modeling of Computer Systems, Philadelphia, pp. 126-137, May 1996.