# 이동 컴퓨팅 환경에서 연결 상태 전이에 적응하는 데이터베이스 캐싱 기법

박 경 두°, 김 유 성
인하대학교 전자계산공학과

# Adaptive Database Caching Method to Connection States in Mobile Computing Environment

Kyoung-Doo Park°, Yoo-Sung Kim
Dept. of Computer Science & Engineering, INHA UNIV.

## 요 약

고정 네트워크를 기반으로 하는 클라이언트/서버 환경에서 클라이언트가 제기하는 요청에 대한 응답시간을 최소화하기 위해 사용되는 캐싱 기법은 이동 컴퓨팅 환경에서도 이동 클라이언트가 제기하는 질의에 대한 응답시간을 최소화하기 위해 적용될 수 있다. 기존의 분산 환경과 비교해 보면, 이동 컴퓨팅 환경에서는 연결 상태의 변화가 빈번하게 발생한다. 따라서 일반적인 분산 환경에서의 캐싱 기법이 이동 컴퓨팅 환경에 적용되기 위해서는 이동 컴퓨팅 환경에 맞도록 수정이 되어야만 한다. 즉, 이동 호스트의 이동성으로 인하여 연결 상태가 변화 할 수 있기 때문에 이동 컴퓨팅 환경을 위한 캐싱 기법은 이동 호스트의 연결 상태 전이에 적응성을 갖도록 해야 한다. 연결 상태 전이에 적응성을 갖도록 하기 위해서 본 논문에서는 연결 상태 전이 예측 윈도우를 사용하고 연결 상태 전이에 미리 대비하여 캐싱을 수행함으로써 이동 호스트의 캐쉬는 최신의 유효한 데이터의 일관성을 보장하며, 사용자에게는 신속한 응답을 수행할 수 있다.

## 1. 서론

무선 네트워크(Wireless Network) 기술의 발달과 하드웨어 기술의 발달은 클라이언트/서버 환경과 같은 분산 컴퓨팅 패러다임의 범위를 더욱 확장하는데 커다란 역할을 하였다. 사용자들은 휴대가 용이한 소형의 이동 장치들을 통해 무선망에 방송되는 정보들을 쉽게 이용할 수 있게 되었다. 그러나, 이동 컴퓨팅 환경에서는 네트워크의 낮은 신뢰성, 제한된 대역폭과 배터리 용량과 같은 MH(Mobile Host)의 이동성(Mobility)때문에 발생하는 문제가 여전히 존재한다[1]. 이러한 문제때문에 MH와 MSS(Mobile Support System)는 제한된 무선망을 효율적으로 사용하여 사용자 질의에 대해 신속한 응답을 보장하기 위한 방안으로 MSS와 MH간의 메시지 교환을 최소화 시켜야만 한다. 따라서 이동 컴퓨팅 환경에서의 캐싱은 MH의 메모리에 데이터를 저장하고, 저장된 데이터를 참조하고자하는 요청이 있을 경우, 네트워크를 액세스하지 않고 캐쉬 메모리를 참조하게 함으로써 응답시간을 크게 감소시키기 위해 널리 쓰인다 일반적인 시스템에서는 캐싱을 위해 정해진 형태의 캐싱 정책을 채택한다[2][3][4][5] 이동 컴퓨팅 환경에서는 MH의 이동으로 인한 연결 상태 전이가 빈번하게 발생한다. 연결 상태 전이가 발생하는 경우를 위해 연결 상태에 따라 캐싱 정책을 동적으로 변경시키는 적응성을 갖는 캐싱 기법이 제시되어야 한다[4]. 본 논문에서는 캐싱 기법이 연결 상태가 전이될 수 있는 경우를 예측하고 예측된 결과에 따라 캐싱 기법을 위한 자원 분배를 동적으로 수행하는 이동 컴퓨팅 환경에서 연결 상태 전이에 적응하는 데이터베이스 캐싱 기법을 제안한다. 본 논문에서 제안된 캐싱 기법은 연결 상태 전이에 적응하기 때문에 이동 컴퓨팅 환경과 같이 네트워크 연결 상태가 일정하지 않은 경우에서도 사용자 질의에 대한 응답시간을 일정하게 유지할 수 있다

본 논문의 구성은 다음과 같다. 2장에서는 분산 컴퓨팅 환경과 이동 컴퓨팅 환경에서의 캐싱 기법의 차이점을 살펴본다. 3장에서는 본 논문에서 제안된 이동 환경에서의 연결 상태 전이에 적응하는 효과적인 데이터 캐싱 기법을 설명하고, 마지막으로 4장에서는 결론과 향후 연구방향에 대하여 기술한다

## 2. 관련 연구

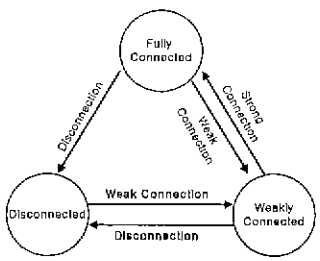
이질적인 전송 매체로 인해 발생하게 되는 전송 지연현상을 해결하기 위해 전송 매체 사이에 위치하게 되는 캐쉬는 낮은 처리 속도를 갖는 시스템이 전송매체를 액세스하는 빈도를 낮춤으로써 시스템 전체 성능을 향상시키는 역할을 한다 일반적인 캐싱 기법으로는 클라이언트의 연산에 사용하게될 데이터를 사전에 서버로부터 Fetch 하는 Prefetching 기법과 클라이언트의 사용패턴을 저장하고 이를 분석하여 네트워크 오류가 발생하는 경우를 대비해 데이터를 미리 전체적으로 저장하는 Hoarding 기법이 있다 분산 컴퓨팅 환경에서는 네트워크의 데이터 전송에 대한 신뢰성이 높기 때문에 연결 상태 전이에 대한 고려를 하지 않고, 일반적인 서버의 디스크 검색 속도와 네트워크의 전송 대역폭 차이 때문에 발생할 수 있는 응답시간을 감소시키기 위해 캐싱 기법을 이용한다[2][3]. 이동 컴퓨팅 환경에서는 MH와 MSS가 무선 네트워크를 통해 연결된 상태에서 MH가 이동을 하기 때문에 연결 상태가 변할 수 있다. 따라서, 연결 상태가 일

정하지 않은 이동 컴퓨팅 환경에서는 MH의 이동성 때문에 연결 상태를 구분하고 각 상태에 맞는 캐싱 기법을 적용해야 한다. 즉, 연결 상태를 구분하고 구분된 상태에 따라 캐싱 기법을 동적으로 다르게 할 수 있는 캐싱 기법이 필요하다 또한, 일반적인 이동 컴퓨팅 환경에서 캐싱 기법은 현재 상태를 기준으로 데이터를 캐쉬하기 때문에 불필요한 자원의 낭비를 가져올 수 있다 [4]. 따라서 발생 가능한 상태 전이에 대해서 예측 기법을 통해 캐싱 기법을 동적으로 전환하여 캐쉬 참조율을 높이고 상태 전이가 발생할 때 캐싱 기법 전환을 위한 부하를 감소시킬 수 있는 방법이 필요하다.

## 3. 연결 상태 전이에 적응하는 동적 캐싱 기법

### 3.1 이동 컴퓨팅 환경의 연결 상태 구분

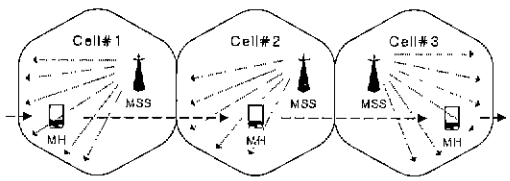
이동 컴퓨팅 환경은 MSS(Mobile Support Station)가 무선 네트워크를 통해 MH(Mobile Host)와 통신 가능한 셀의 집합으로 구성된다. 셀의 범위는 MSS의 성능에 따라 결정되며, MH는 사용자가 직접 휴대하고, 이동하면서 무선 네트워크를 통해 MH가 위치한 영역을 관리하는 MSS와 통신을 하게된다. MH가 셀을 이동하면서 연결 상태는 [그림 1]과 같이 세가지 상태로 구분할 수 있다. 특히, 고품위 서비스를 제공하기 위해서 셀의 크기는 점점 작아지고, MSS가 관리하는 영역 내에 존재하는 MH의 수는 증가하기 때문에 셀들을 이용하는 MH의 이동 때문에 연결 상태는 변화가 증가하기 때문에 연결 상태를 구분해야만 한다.




[그림 1] 이동 컴퓨팅 환경의 상태 전이 구분

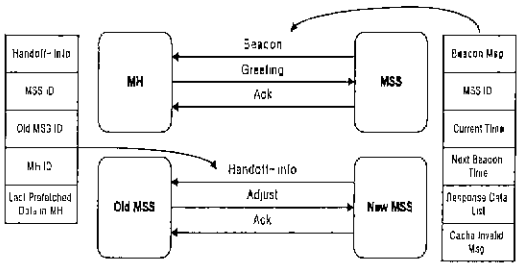
### 3.2 핸드오프(Hand-Off)와 접속 상태 전이

이동 컴퓨팅 환경에서 MH는 [그림 2]에서 보여지는 것과 같이 여러 셀들을 거치면서 이동할 수 있다. MH는 각 셀을 담당하는 MSS로부터 데이터를 수신하면서 이동하는데, 셀과 셀이 접하는 부분을 이동하게 되는 경우 핸드오프(Hand-Off) 현상이 발생하게 된다. 즉, 일시적인 접속 상태 전이가 일어나게 된다.



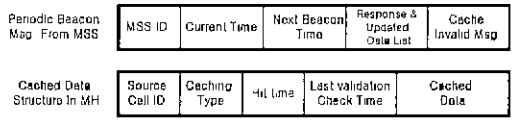

[그림 2] MH(Mobile Host)의 이동 패턴 예

MH가 연결이 설정된 상태에서 한 셀에서 다른 셀로 이동할 때 세션정보가 전달되도록 MSS에서는 핸드오버(Hand-Over)를 실행하게 된다. 그러나 핸드오버가 늦게 일어나게 되면, 설정된 세션에 대한 신뢰도가 낮아지게 되어 재접속을 해야하는 경우가 발생할 수 있다. [그림 3]에서는 핸드오프시 MH와 현재 MH가 속한 셀의 MSS, 그리고 현재 MSS와 이전 셀의 MSS 사이에서 주고 받는 메시지를 보여주고 있다. MSS는 주기적으로 Beacon 메시지를 셀에 방송하며, MH는 주기적으로 방송되는 메시지를 수신하게 된다.




[그림 3] 핸드오프시의 메시지 전송

본 논문에서는 주기적으로 방송되는 Beacon 메시지 내에 MH가 연결 상태를 판단하여 연결 상태 전이 예측 기법에 사용하고 캐쉬된 데이터의 일관성을 위해 다음 메시지 방송시간과 캐쉬 무효화 메시지를 포함한다. 따라서 MH는 이미 알려진 다음 방송시간 내에 메시지가 도착하는 경우, Fully Connected 상태로 판단하고, 방송예정 시간이후에는 타임아웃 방식으로 Weakly Connected 와 Disconnected 상태로 구분한다. [그림 4]는 [그림 3]의 주기적인 Beacon 메시지와 MH에 위치하는 캐쉬된 데이터의 자료구조 이다.




[그림 4] MSS 의 Beacon 메시지와 MH에 캐쉬된 데이터의 구조

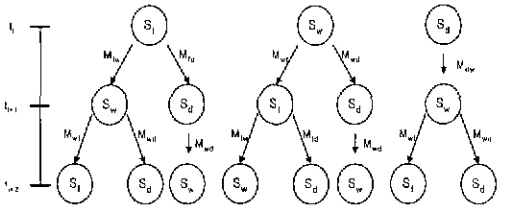
### 3.2 연결 상태 전이 예측 기법

본 절에서는 접속 상태가 전이되는 것을 예측하기 위한 방안으로 연결 상태에 대한 상태 전이 트리를 정의한다. MH는 현재 자신의 접속 상태를 판단하기 위해서 [그림 3]에 보여지는 MSS에서 MH로 방송되는 Beacon 메시지를 이용한다. 메시지에는 MSS에서 메시지를 방송한 시간과 다음번 방송시간을 포함하고 있기 때문에 다음 방송시간까지 현재의 상태를 유지하고 방송예정 시간에 메시지 수신이 이루어지게 되면 자신의 상태가 $S_f$ 상태에 속하는 것으로 판단한다. 만일 방송예정 시간이 초과하는 경우, $S_w$에 속한 상태라고 판단한다. 그리고 $S_w$ 상태에서도 타임아웃이 발생하면 $S_d$에 속한 상태라고 판단한다 [그림 5-(b)]는 윈도우 크기가 2인 상태 전이 트리이다. "Window Size = 2" 라는 것은 현재 상태에서 상태가 2회 변경된 후의 상태를 말한다. 윈도우 크기가 2인 이유는 윈도우 크기를 확장할 때 바탕이 될 수 있는 최소의 크기이기 때문이다. [그림 5-(b)]에 표기된 상태

와 메시지들은 다음과 같이 정의한다.

*f : Fully Connected , w : WeaklyConnected ,*
*d : Disconnected*
*State S = { $S_f$ , $S_w$ , $S_d$ }*
*Trans Msg M = {$M_{ij}$ | i∈source, j∈destination}*

**[그림 5 (a)] 상태 전이 트리 요소 정의**



**[그림 5-(b)] 상태 전이 트리(*Window size = 2*)**

[그림 5-(a)]에는 상태 전이 트리에서 사용되는 원소들을 정의하였다. S는 연결 상태 집합을 의미하고, M은 상태 전이를 의미한다. [그림 5-(b)]에는 집합 S의 원소인 연결 상태들에 따라 각각 상태 전이 트리를 정의하였다. 상태 전이 트리는 [그림 1]에서 구분한 상태 전이를 기준으로 하기 때문에 연결 상태가 변화하지 않는 경우를 포함하지 않는다. 따라서 상태가 전이되지 않는 경우와 $S_d$에서 직접 $S_f$로 전이하는 경우 , 그리고 MH가 속할 수 있는 상태 S에서 예측 윈도우 크기 2를 벗어나는 $S_d$는 재접속이 필요한 상태로 간주하고 상태 전이 트리에 포함하지 않는다. 따라서 $T_i$에서 $T_{i+2}$까지 연속으로 $S_d$가 나타나는 경로와 동일한 상태가 반복되는 경로는 포함하지 않는다. 위의 상태 전이 트리를 이용하면 마지막 단말 노드에 존재하게 되는 연결 상태를 알수 있다. 따라서 MH가 데이터 캐쉬을 위해 다음절에서 설명할 Prefetching 과 Hoarding 가운데 어떠한 기법을 위해 자원을 더 할당해야 하는지 판단을 위한 기준으로 사용한다. 더욱 세밀한 판단을 위해서 윈도우 크기를 확장 할 수 있는데, $T_i$에서 연결 상태 별로 정의된 윈도우 크기 2의 트리들을 $T_{i+1}$ 일 때의 상태에 맞는 윈도우 크기 2의 트리를 적용하여 윈도우의 크기를 확장한다. 그러나 윈도우 크기가 커지게 되면 $3^n$에 근접한 만큼 경우의 수가 증가할 수 있으므로, 적절한 크기의 윈도우를 결정하기 위해서는 크기별 실험이 이루어져야 한다.

### 3.3 예측 결과를 이용해 환경에 적응하는 캐쉬 기법

Prefetching은 앞으로 사용될 데이터들에 대한 정보를 수집하기 위해 제기된 연산들에 대한 분석을 해야하고, Hoarding은 사용자 패턴을 저장하는 HDB(Hoarding Database)를 분석하여 데이터를 캐쉬에 저장하기 때문에 캐쉬에 대한 참조율이 매우 낮을 수 있다. 따라서, Prefetching에 의해 저장되는 데이터는 Hoarding에 의해 저장되는 데이터보다 참조될 확률이 높은 데이터들이다. 본 모델에서는 Prefetching에 의해 캐쉬된 데이터들은 반드시 일관성을 유지해야 하는 데이터로 정의하고 Hoarding에 의해 캐쉬된 데이터는 핸드오프시 MH에서 우선적으로 무효화하는 데이터로 간주하는 정책을 사용한다. 앞 절에서 설명한 상태 전이 트리를 사용하여 루트 노드의 현재 연결 상태와 단말 노드에 있는 상태들의 빈도를 구한 후, 현재 상태가 $S_w$이거나 전체 상태들의 빈도 가운데 $S_w$가 제일 높은 경우와 $S_d$의 빈도가 높게 나타나는 경우에는 캐쉬 참조율이 높은 Prefetching에 자원을 더 할당하고, $S_f$인 경우에는 Hoarding에 자원을 더 할당한다. 제안한 캐쉬 기법에서는 연결 상태 전이 예측에 따라 두 가지 캐쉬 기법이 모두 사용되기 때문에 저장된 데이터의 무효화 방법 또한 구별하여 사용한다 따라서 [그림 4]의 캐쉬된 데이터의 자료구조에 이러한 기법을 통해 데이터가 캐쉬 되었는지 구분하기 위한 필드를 추가하였다. 그리고 캐쉬 된 데이터의 유효성 검사의 효율성을 높이기 위해 Prefetching 기법으로 캐쉬 된 데이터 가운데 마지막 유효성 메시지를 수신한 데이터의 시간을 저장하는 필드를 추가하여 핸드오버시 이전 MSS에서 수행하는 유효성 검사의 시점을 줄여주도록 하였다. 이 필드들은 핸드오프시 새로운 셀의 MSS에게 전달될 내용을 감소시킴으로써 신속한 핸드오버가 수행되어 일시적인 접속 단절이 일어나는 경우에 접속 재개를 위한 정보를 갱신하는데 효과적으로 사용될 수 있다.

## 4. 결론 및 향후 연구과제

이동 컴퓨팅 환경에서는 연결 상태가 불안하기 때문에 일정한 전송 대역폭을 갖는 클라이언트/서버 환경에서의 캐쉬 기법이 직접 사용될 수 없다

본 논문에서는 연결 상태가 변화하는 이동 컴퓨팅 환경에서의 연결 상태 전이 트리를 구성하고, 구성된 트리를 이용하여 동적으로 캐쉬 기법을 전환하는 연결 상태 전이에 적응하기 위한 데이터 캐쉬 기법을 제안하였다 이 기법은 이동 컴퓨팅 연결 상태 전이 환경에서 사용자 질의에 대한 응답시간을 향상시키고, 연결 상태 전이로 인해 무효화되는 데이터들의 대상을 감소시킬 수 있다.

향후에는 본 논문에서 제안된 상태 전이 트리를 이용하여, 상태 전이에 적응하면서도 시스템의 부하를 최소화 할 수 있는 윈도우 크기를 실험을 통해 결정해야 할 것이다.

## 참고문헌


[1] Daniel Barbara, "Mobile Computing and Databases-A Survey", *IEEE Transaction on Knowledge and Data Engineering*, Vol 11, No. 1, 1999

[2] Hong Va Leong and Antonio S,. "Database Caching Over the Air-Storage", *The Computer Journal*, Vol. 40, No. 7, 1997

[3] A. Prasad Sistla, Ouri Wolfson and Yixiu Huan, "Minimization of Communication Cost Through Caching in Mobile Environments", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 4, 1998

[4] Brian D. Noble, et.al., "Agile Application-aware adaptation for Mobility", The *Proceeding of 16th ACM Symposium on Operating System Principles*, October 1994.

[5] Micheal J, Franklin, Micheal J. Carey and Miron Livny, "Transactional Client-Server Cache Consistency : Alternetives and Perfomances ", *ACM Transaction on Database Systems*, Vol. 22, No. 3, 1997