

이동 컴퓨팅 환경에서 이동 호스트의 자치성 증대를 위한 선택적 캐쉬 일관성 유지 기법

김 희 숙[†] · 황 병 연^{††}

요 약

현재 이동 컴퓨팅 환경에서 이동 호스트가 사용하는 캐쉬는 전력의 한계와 제한된 대역폭이라는 단점을 보완할 수 있는 중요한 장치로서 사용된다. 하지만 서버 데이터와의 일관성을 유지해야 하는 부담을 가지고 있다. 본 논문에서는 선택적 캐쉬 일관성 유지 기법을 제안한다. 서버는 캐쉬 상태 테이블, 데이터 접근 테이블을 유지하여 이동 호스트에서 미래 사용성이 높은 데이터에 대한 선별을 통해 효율적인 방송을 가능하게 한다. 또한 오랜 접속 단절 이후에 연결된 이동 호스트의 요청에 따라 서버가 보관하고 있던 지난 방송 정보를 전달하고 비동기적인 방송을 수행함으로써 다수의 캐쉬 데이터가 버려지는 것을 방지하고, 최신의 데이터를 보유할 수 있도록 한다. 따라서 이동 호스트는 서버에 대한 의존도를 줄일 수 있게 되어 자치성 증대라는 효과를 얻을 수 있음이 실험을 통해서 보여진다.

Selective Cache Consistency Scheme to Enlarge Autonomy of Mobile Host in Mobile Computing Environments

Hee-Sook Kim[†] · Byung-Yeon Hwang^{††}

ABSTRACT

The cache used by mobile host is an important device that recovers the weak points of limited power and bandwidth, in mobile computing environments. However, it has to stand and maintain the consistency with the server data. In this paper, we propose a 'Selective Cache Consistency Scheme.' The server allows an effective broadcasting by selecting data of high usability using 'Cache State Table' and 'Data Access Table.' Moreover, this scheme prevents the loss of data that may occur by a long period of disconnection, by asynchronous broadcasting and transmitting those broadcast data preserved in the server. This also allows user to possess the latest data. Through experiments, we have found that the enlargement of autonomy is possible by reducing the dependence of server.

키워드 : 이동 컴퓨팅(Mobile Computing), 자치성(Autonomy), 선택적 캐쉬 일관성(Selective Cache Consistency)

1. 서 론

무선 네트워크 환경의 기술적인 발전과 함께 휴대용 컴퓨터와 무선 단말기의 급속한 보급이 이루어지면서 이동 컴퓨팅 환경에 대한 연구가 보다 활발하게 진행되고 있다. 이에 따라 사용자들은 시간과 장소에 구애 받지않고 컴퓨팅 작업을 수행하며 데이터베이스에 접근하는 것이 가능하게 되었다. 하지만 이동 컴퓨팅은 통신 대역폭(bandwidth)이 제한되어 있고, 외부간섭이나 이동 호스트의 이동으로 인한 접속 단절로 데이터 손실이 발생할 수 있다. 또한 배터리를 사용하기 때문에 전력 제한으로 인해 통신이 자유롭지 못한 제약이 있다[1, 2]. 이러한 단점을 극복하기 위해 여러 기법에 대한 연구가 꾸준히 수행되고 있다.

서버는 방송(broadcast) 기법[3]의 사용으로 제한된 대역폭

을 효율적으로 이용하게 하고, 이동 호스트에서는 캐쉬(cache)를 사용하여 접속 단절 상태에서도 지속적으로 작업을 수행하며 서버로의 데이터 요청을 줄일 수 있기 때문에 서버에 대한 의존도를 감소시켜 자치성을 확보할 수 있다. 따라서 효율적인 클라이언트 자원 활용이 가능해지지만 서버 데이터와의 일관성 유지를 해야 하는 부담을 가지게 된다[4].

효율적인 일관성 유지를 위해 본 논문에서는 선택적 캐쉬 일관성 유지 기법을 제안한다. 제안된 기법은 서버에 두 개의 캐쉬 관련 테이블을 유지하여 이동 호스트가 가지고 있지 않은 데이터에 대한 불필요한 전달을 방지하고 이동 호스트에서 사용될 확률이 높은 데이터에 대한 정확한 선별을 통한 방송 메시지 전송을 통해 캐쉬 적중률을 향상시킨다. 특히, 오랜 접속 단절 후에 서버와의 통신이 재개된 이동 호스트는 서버 요청을 통해 단절 기간동안 수신하지 못했던 정보를 사용하고 가장 최근의 갱신 내용을 비동기적으로 방송함으로써 모든 이동 호스트가 최신의 정보를 얻을 수 있게 한다.

본 논문의 구성은 다음과 같다. 2장에서 기존 연구들과

* 본 연구는 2003년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

† 준 회원 : 가톨릭대학교 대학원 컴퓨터공학과

†† 종신회원 : 가톨릭대학교 컴퓨터정보공학부 교수

논문접수 : 2003년 7월 1일, 심사완료 : 2003년 7월 15일

그 문제점을 살펴보고, 3장에서는 본 논문에서 제안하는 선택적 캐쉬 일관성 유지 기법을 기술한다. 4장에서는 기존의 캐쉬 일관성 유지 기법과 제안한 기법의 성능 평가를 수행하고, 5장에서 결론을 맺는다.

2. 관련 연구

방송 타임 스탬프(Broadcasting Timestamps) 기법[5]은 일정기간 동안에 갱신된 데이터의 식별자와 해당 데이터가 마지막 갱신된 시점에 대한 타임 스탬프의 쌍으로 이루어진 무효화 보고를 방송한다. 그러나 접속 단절 등의 이유로 자신의 현재 캐쉬 내용이 유효한 것인지를 확인할 수 없을 경우 이동 호스트는 자신의 캐쉬 내용을 모두 무효화하게 되며, 빈번하게 사용되는 데이터에 대한 무효화 메시지가 방송되면 이동 호스트들로부터 집중적인 캐쉬 요청이 발생할 수 있다. 또 이동 호스트는 자신의 캐쉬 데이터를 사용하기 전에 데이터의 유효성 여부를 확인하기 위해 다음 무효화 방송이 올 때까지 기다려야 하므로 질의 처리가 늦어지는 단점이 있다.

SCC(Simple Checking Caching) 기법[6]은 무효화 보고만으로 캐쉬의 유효성 확인이 불가능할 경우 이동 호스트가 서버로 유효성 확인 요청을 하는 기법이다. 이동 호스트는 유효성 확인을 원하는 데이터의 식별자와 무효화 보고를 받은 마지막 시점의 타임 스탬프 값을 서버로 전송하게 되며, 서버에서는 이에 해당하는 정보를 요청한 이동 호스트에게 전송한다. 하지만 이 기법은 캐쉬 내에 데이터의 수가 많을 경우 심한 대역폭 소모를 가져오게 된다.

SGC(Simple Grouping Caching) 기법은 SCC 기법의 단점을 보완하기 위해 데이터들을 그룹화하고, 그룹의 식별자를 전송하여 유효성 확인 요청을 한다. 하지만 이동 호스트 내의 그룹 데이터가 전부 갱신되지 않았을 경우에도 서버에서 그룹이 무효화되면, 이동 호스트내의 갱신 데이터가 속해 있는 그룹 내의 모든 데이터가 무효화되는 잘못된 무효화(false invalidation)가 발생한다.

UIR(Updated Invalidation Report) 기법[7]은 주기적인 무효화 방송 사이에 추가적인 갱신 정보를 보내줌으로써 질의 처리가 지연되는 문제를 해결할 수 있다. 하지만 이동 호스트가 받게 되는 정보의 양이 불필요하게 많아지거나 접속단절이 오랫동안 지속되는 경우에는 기존의 기법과 마찬가지로 효율적인 대처 방안이 없는 것이 단점이다.

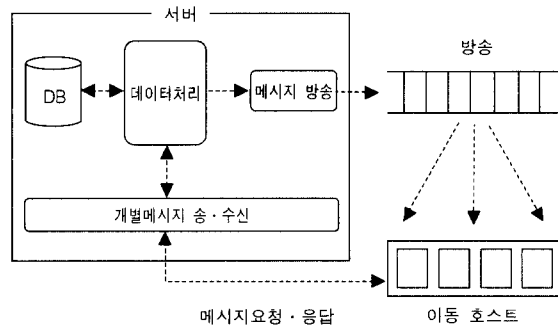
적응적 캐쉬 일관성 유지 기법[8]은 무선 네트워크의 연결 상태에 따른 가용 대역폭의 변화에 따라 무효화와 전파 메시지의 방송 비율을 동적으로 조절한다. 네트워크가 안정되어 가용 대역폭이 충분히 확보되면 경계값을 재설정하여 전파 메시지의 방송에 많은 양을 할애하여 캐쉬 요청을 감소시키고 서버의 응답 시간을 단축시키도록 한다. 잦은 부분 단절 및 접속 단절이 발생하여 가용 통신 대역폭이 감소하면, 무효화 메시지의 방송 비율을 늘려 메시지 파손 및 손실 확률을 최소화한다. 하지만 이동 호스트의 갱신과 서

버의 갱신을 구별하지 않기 때문에 이동 호스트에 존재하지 않는 데이터에 대한 서버 측의 빈번한 갱신으로 인해 전파가 수행될 경우 불필요한 데이터의 방송으로 인한 대역폭의 낭비가 발생하며, 오랜 접속 단절 이후에는 기존의 기법들과 동일한 문제점을 가지게 된다.

3. 선택적 캐쉬 일관성 유지 기법

3.1 개요

기존의 캐쉬 일관성 유지 기법들을 사용할 경우 사용 빈도가 높은 데이터에 대한 무효화의 발생이나 오랜 접속 단절 등으로 인한 문제점이 발생할 수 있다. 이러한 문제를 극복하고 이동 호스트의 자치성을 증대시키기 위한 방법으로 본 장에서는 선택적 캐쉬 일관성 유지 기법을 제안한다.



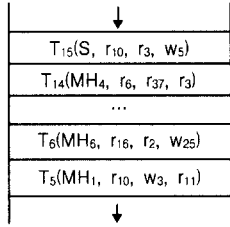
(그림 1) 캐쉬 일관성 유지기법의 정보 전달 구조

(그림 1)은 본 논문에서 제안하는 기법의 정보 전달 구조이다. 그림과 같이 방송을 위한 채널과 메시지 요청 등을 처리하기 위한 개별적인 통신 채널은 분리되어 있다.

이동 호스트는 캐쉬 데이터를 사용하여 트랜잭션을 수행하기 때문에 서버와는 달리 데이터의 유효성을 검사하는 과정이 필요하다. 사용하려는 데이터가 캐쉬에 존재하지 않거나 무효화된 경우, 서버에 데이터 요청을 한다. 요청된 데이터는 서버의 데이터 처리 모듈을 통해 해당 이동 호스트로 전달된다.

데이터를 수집한 이동 호스트는 트랜잭션을 수행하고, 서버에 트랜잭션을 전송하여 완료(commit)요청을 한다. 서버는 트랜잭션에 사용된 데이터들이 최신 버전인지를 검사하여 최신 버전일 경우 완료 트랜잭션 큐(Committed Transaction Queue : 이하 CTQ)에 트랜잭션 정보를 넣고 이동 호스트에 완료 메시지를 전송한다. 최신 버전이 아닐 경우는 최신 버전의 데이터와 철회(abort)메시지를 해당 이동 호스트에 전송하여 최신 데이터로 트랜잭션을 다시 수행할 수 있도록 한다.

(그림 2)는 완료된 트랜잭션들을 담는 CTQ이다. 트랜잭션을 수행한 이동 호스트의 정보와 사용된 데이터에 대한 정보를 모두 가지고 있다. 서버는 CTQ 내의 정보를 바탕으로 방송 메시지를 구성한다. 이동 호스트는 이 방송 정보를 수신하여 캐쉬 데이터의 유효성 여부를 판단할 수 있게 된다.



(그림 2) 완료 트랜잭션 큐(CTQ)

서버에 이동 호스트들의 캐쉬에 대한 상태 정보와 데이터 접근 빈도에 대한 정보를 담고있는 테이블을 유지함으로써 이동 호스트에 존재하지 않는 데이터에 대한 방송을 하지 않아 불필요한 대역폭 낭비를 방지하고 사용 가능성이 높은 데이터에 대한 변별력을 높여 효율적인 방송을 수행할 수 있도록 한다. 또한, 오랜 접속 단절 후에 서버와 연결된 이동 호스트는 서버 요청 방법을 통해 지난 방송 정보들을 활용하여 캐쉬 데이터가 버려지는 것을 막을 수 있다. 본 논문에서 제안하는 기법을 사용하게 되면, 이와 같은 효율적인 캐쉬의 사용을 통해 이동 호스트의 자치성을 증대시킬 수 있게 된다.

3.2 테이블을 사용한 데이터 선별 기법

데이터에 대한 이동 호스트의 사용 가능성을 예측하여 방송될 데이터를 효율적으로 선별하기 위해서 서버는 캐쉬 상태 테이블과 데이터 접근 테이블을 유지한다. 갱신 트랜잭션은 서버와 이동 호스트 모두에서 발생하고 보통 서버에서는 이동 호스트보다 많은 갱신이 발생한다. 때때로 어떤 이동 호스트도 가지고 있지 않은 데이터에 대한 서버 갱신이 발생할 수 있다. 이 경우, 모든 갱신에 대해 방송을 수행하게 되면 이동 호스트에 존재하지 않는 데이터에 대한 불필요한 방송을 하게 되므로 대역폭 낭비를 초래한다. 이동 호스트에 보유된 데이터의 갱신 내용에 대해서만 방송을 수행하게 되면 불필요한 정보의 방송으로 인한 대역폭 낭비를 방지하는 효과를 얻을 수 있다. <표 1>의 캐쉬 상태 테이블(Cache State Table : 이하 CST)은 각 이동 호스트 내에 존재하는 캐쉬 데이터에 대한 목록을 유지하고 있는 테이블이다. 이 테이블을 통해 서버는 이동 호스트에 존재하는 데이터만을 선별해 방송할 수 있다.

<표 1> 캐쉬 상태 테이블(CST)

데이터 식별자	MH ₁	MH ₂	MH ₃	MH ₄	MH ₅	MH ₆	...
D1	1	1	0	0	1	0	
D2	1	1	1	1	0	1	
D3	0	0	0	0	0	0	
D4	1	0	1	1	0	1	
...							

이동 호스트가 새로운 데이터를 요청하게 되면 서버에서는 데이터를 전송해주면서 CST의 해당 값을 1로 세팅한다.

이동 호스트는 데이터의 유효성 확인 불가 상태나, 새로운 데이터의 입력으로 인해 캐쉬에서 기존의 데이터가 버려지게 되면 서버로 데이터 discard 메시지를 전송하고 서버는 CST의 해당 값을 0으로 세팅한다. 모든 이동 호스트가 0값을 가지게 되는 데이터는 서버에서 갱신이 발생하더라도 방송되지 않는다.

<표 1>의 경우 D3은 어떤 이동 호스트도 이 데이터를 캐쉬에 가지고있지 않기 때문에 서버측 트랜잭션에 의해 갱신이 발생하더라도 이를 이동 호스트에 방송할 필요가 없다. (그림 3)은 이동 호스트의 캐쉬 관리 알고리즘이다.

```

CACHE ( Order, Data )
{
  if ( Order is INSERT )
  {
    if ( cache is full )
    {
      send message to Server < Di is discard >
      // Di는 무효화된 데이터 중 수신 시점이 가장 오래된 데이터
      delete Di
      insert Data
    }
    else
      insert Data
  }
  else if ( Order is UPDATE )
  {
    delete Di
    insert Data
  }
  else if ( Order is DISCARD )
  {
    send message to Server < Di is discard >
    delete Di
  }
}
    
```

(그림 3) 이동 호스트의 캐쉬 관리 알고리즘

<표 2>의 데이터 접근 테이블(Data Access Table : 이하 DAT)은 CTQ에 존재하는 트랜잭션들에 사용된 데이터에 대한 서버와 이동 호스트의 접근 횟수를 기록한 테이블이다. CTQ에 존재하는 데이터에 대한 정보만을 사용하기 때문에 최근에 사용된 데이터들을 가지고 접근 횟수를 기록할 수 있게 된다. 따라서 서버는 DAT를 유지함으로써 방송될 데이터의 전파와 무효화 여부를 결정할 수 있게 된다.

<표 2> 데이터 접근 테이블 (DAT)

데이터 식별자	이동 호스트 읽기	이동 호스트 갱신	서버 갱신
D1	0	0	15
D2	8	8	10
D3	12	4	1
D4	8	8	2
...			

```

CTQ receive Ti // CTQ는 트랜잭션 큐
If (CTQ is Full)
{
  ( D's Acc_Num of Ti-n ) - 1 // n은 큐의 크기
  delete Ti-n // Acc_Num은 접근 횟수
  ( D's Acc_Num of Ti ) + 1 // D는 트랜잭션에 사용된 데이터
}
    
```

```

insert Ti
}
else
{ (D's Acc_Num of Ti) + 1
insert Ti
}
    
```

(그림 4) DAT 구성 알고리즘

(그림 4)는 DAT 구성 알고리즘이다. DAT는 데이터에 대한 이동 호스트의 읽기, 갱신, 그리고 서버의 갱신 횟수를 기록하게 된다. 서버에서의 읽기는 이동 호스트로 방송될 데이터를 선별하는데 영향을 미치지 않기 때문에 기록하지 않는다.

전파될 데이터와 무효화할 데이터를 구분하기 위해 전체 갱신 데이터를 전파 그룹과 무효화 그룹으로 나눈다. 그리고 갱신된 데이터들에 대한 접근 횟수를 가지고 정렬을 수행한다. 정렬 결과 접근 횟수가 많은 데이터부터 차례로 전파 메시지의 비율만큼 전파 그룹으로 옮겨지게 된다. 갱신된 데이터들은 각 단계를 거치며 전파와 무효화 두 그룹으로 나누어지게 된다. 위에서 살펴본 두 테이블을 사용함으로써 이동 호스트에서 사용될 확률이 높은 데이터에 대한 보다 정확한 선별이 가능해지고, 이를 통해 캐쉬 적중률 향상과 데이터 요청 감소의 이점을 얻을 수 있다.

- 1 단계 : 이동 호스트의 읽기 횟수와 갱신 횟수를 더한 값으로 내림차순 정렬하여 비교
- 2 단계 : 이동 호스트의 읽기 횟수로 내림차순 정렬하여 비교
- 3 단계 : 서버의 갱신 횟수로 오름차순 정렬하여 비교
- 4 단계 : 식별자의 번호 비교

3.3 서버 요청을 통한 캐쉬 일관성 유지 기법

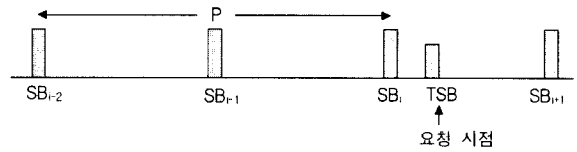
서버는 테이블을 사용한 방송 데이터 선별 기법을 통해 구성된 갱신 정보를 주기적으로 방송함으로써 서버와 이동 호스트의 캐쉬 데이터를 일관성 있게 유지한다. 하지만 오랜 접속 단절 후에 서버와의 통신이 재개된 이동 호스트의 경우 단절 기간 동안 방송을 수신할 수 없으므로 현재 캐쉬 데이터가 유효한지 확인할 수가 없다. 따라서 본 논문에서 서버와 접속이 재개된 이동 호스트는 서버로 캐쉬 데이터의 유효성 확인을 요청하고, 서버는 이동 호스트의 요청에 대비하여 지난 방송 정보를 유지하는 기법을 사용한다.

<표 3> 서버가 보관하고 있는 방송 정보

방송 시점	타임스탬프	방송 데이터 식별자							
SB _{i-2}	9:00	1	10	15	22				
SB _{i-1}	9:20	30							
SB _i	9:40	3	7	10	15	17	32	35	39
TSB	9:45	1	7	19					

서버가 지난 방송 시점별로 정보를 보관하고 있는 모습

을 <표 3>에서 보여주고 있다. (그림 5)는 주기적인 방송 시점(SB)과 이동 호스트의 서버 요청으로 인한 비동기적 방송시점(TSB)을 보여주고 있다. 서버가 보관하게 되는 방송 주기(P)는 알고리즘의 변수이다.



(그림 5) 갱신 데이터 정보들의 방송 시점

이동 호스트가 오랜 접속 단절 후에 서버와 통신을 재개하게 되면 서버에 유효성 확인을 요청한다. 요청을 받은 서버는 보관하고 있던 일정 기간(P) 동안의 지난 방송 정보를 구성하여 요청한 이동 호스트로 전송해 데이터의 갱신 여부를 알려주게 된다. 이동 호스트는 방송 시점의 타임스탬프와 갱신이 발생한 데이터의 식별자로 구성된 정보를 받아 해당 데이터를 무효화하게 된다.

이동 호스트가 접속 단절되기 전, 마지막 방송을 받은 시점부터 유효성 확인 요청으로도 데이터 갱신 여부를 알 수 없는 시점(SB_{i-2}) 이전까지의 데이터는 모두 캐쉬에서 버려진다. 즉, 방송을 수신하지 못한 구간의 데이터는 그 갱신 여부에 상관없이 버려지게 되는 것이다.

서버는 마지막 방송 시점(SB_i)부터 요청이 들어온 시점까지의 갱신 정보를 모든 이동 호스트에 비동기적으로 방송(TSB)해 주어 이동 호스트들이 최신의 데이터를 유지할 수 있도록 해준다. 다음 주기의 방송(SB_{i+1})에서는 비동기 방송 이후의 갱신 정보를 방송하게 된다.

이 방법을 사용함으로써 이동 호스트는 오랜 접속 단절로 방송을 수신하지 못한 경우에도 접속 후에 갱신 정보를 얻을 수 있다. 따라서 기존의 일관성 유지 기법들에 비해 캐쉬에서 버려지는 데이터가 적고 최신의 갱신 정보를 수신할 수 있으므로 캐쉬 손실을 줄일 수 있다.

4. 성능 평가

4.1 실험 모델

본 논문에서는 실험 언어로 JAVA J2SDK 1.4.1을 사용하였고, Windows 2000 Server 환경하의 서버와 Windows XP를 사용하는 클라이언트 두 대를 사용하여 실험을 수행하였다. 서버는 유선 네트워크에 연결되어있고, 클라이언트는 서버와 무선으로 연결된다. 두 대의 클라이언트에 실험에 사용되는 이동 호스트의 수를 동수로 나누어 실험하였다. 접속 단절의 원인에 구별 없이 동일한 단절 상태로 취급하기 때문에 셀 내로 새로이 진입하거나 셀 밖으로 나가는 이동 호스트는 없다고 가정하였으며, 모든 이동 호스트가 동일한 캐쉬 내용을 가지고 실험을 시작한다. 사용되는 주요 파라미터의 초기 설정값은 <표 4>와 같다.

<표 4> 실험 파라미터

파라미터	설정치(단위)
서버 내 데이터베이스 크기	1000(개)
이동 호스트의 개수	10(개)
방송 주기(L)	20(초)
캐쉬 크기	150(개)
방송 윈도우 크기(w)	1
서버가 보관하고있는 지난 방송 주기(P)	3
각 이동 호스트가 수행하는 트랜잭션 수	50(개)
방송 메시지 중 전파 메시지의 비율	0.2
접속 단절 시간(DT)	60(초)
트랜잭션 큐의 크기	30
식별자의 크기	64(bits)
데이터의 크기	1024(bytes)

파라미터의 값은 측정하고자 하는 성능 평가 요소에 따라 다르게 설정될 수 있으며, 파라미터 값의 변화 내용과 실험에 부가적으로 사용되는 파라미터들은 성능 평가 수행 시에 추가적으로 설명하도록 한다. 본 실험에 사용되는 서버와 이동 호스트간의 메시지의 종류는 총 8가지로 분류되고 <표 5>와 같이 구성된다.

<표 5> 실험에 사용되는 메시지 구성

이동 호스트 생성 메시지	
종류	형태
데이터 요청	$\langle \text{flag, MH}_{id}, (D_{id}, D_{id}, \dots D_{id}) \rangle$
트랜잭션 완료 요청	$\langle \text{flag, MH}_{id}, W(D_{id}, \text{Data}, \text{TS}) R(D_{id}, \text{Data}, \text{TS}) \dots R(D_{id}, \text{Data}, \text{TS}) \rangle$
유효성 확인 요청	$\langle \text{flag, MH}_{id}, L_BTS \rangle$
데이터 discard 알림	$\langle \text{flag, MH}_{id}, (D_{id}, D_{id}, \dots D_{id}) \rangle$
서버 생성 메시지	
종류	형태
방송	$\langle \text{flag, BTS, P}((D_{id}, \text{Data})(D_{id}, \text{Data}) \dots), I(D_{id}, D_{id} \dots) \rangle$
트랜잭션 완료 응답	$\langle \text{flag, MH}_{id}, C \rangle$ or $\langle \text{flag, MH}_{id}, \text{TS}, A(D_{id}, \text{Data}) \dots (D_{id}, \text{Data}) \rangle$
유효성 확인 응답	$\langle \text{flag, MH}_{id}, (\text{BTS}, D_{id}, D_{id}, D_{id}) \dots (\text{BTS}, D_{id}, D_{id}) \rangle$
데이터 요청 응답	$\langle \text{flag, MH}_{id}, \text{TS}, (D_{id}, \text{Data}) \dots (D_{id}, \text{Data}) \rangle$

표에서 flag는 메시지 종류를 구분해주는 값이고, MH_{id}는 이동 호스트의 식별자이다. D_{id}는 데이터의 식별자, Data는 실제 데이터를 의미한다. 트랜잭션 완료 요청 메시지에는 데이터의 식별자 앞에 갱신을 나타내는 'W'나 읽기를 나타내는 'R'을 붙여 구분을 가능하게 하고, 타임 스탬프를 보냄으로써 데이터의 유효성 여부를 확인할 수 있도록 했다. L_BTS는 이동 호스트가 수신한 마지막 방송 시점의 타임 스탬프를 알려준다. 이 값을 사용하여 유효성 확인 요청 응답을 보낼 메시지들의 구간을 설정할 수 있다. 방송메시지에서 'P'는 전파될 데이터를, 'I'는 무효화되는 데이터임을 나타낸다. 트랜잭션 완료 응답은 완료된 경우 완료임을 알려주는 'C'를 보내주고, 철회된 경우 철회를 알리는 'A'와

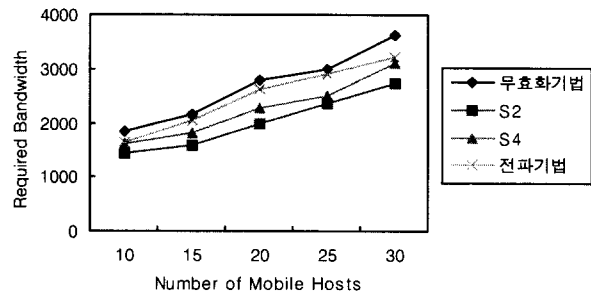
함께 데이터 식별자와 데이터를 보내 트랜잭션을 다시 수행할 수 있도록 한다.

4.2 실험 결과

본 절에서는 제안하는 기법(S2, S4)과 타임스탬프를 사용하는 무효화 기법, 메시지 전파 기법에 대한 비교를 통한 성능 평가를 수행한다. S2는 전파 비율을 0.2로 정한 기법이며, S4는 전파 비율이 0.4이다.

4.2.1 대역폭 소모량(Required Bandwidth)

대역폭 소모량은 제한된 대역폭을 사용해야 하는 이동 컴퓨팅 환경에 있어서 성능을 평가하는 가장 중요한 지표 중의 하나이다. 본 논문에서 평가하게 되는 대역폭 소모량은 방송 메시지로 소모되는 대역폭과 서버와 이동 호스트 사이의 개별적인 메시지 처리에 사용되는 대역폭을 더한 값을 사용하였다.



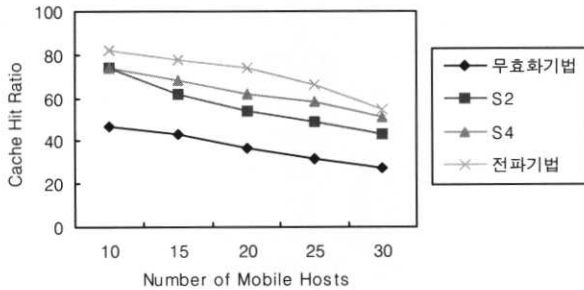
(그림 6) 이동 호스트 수에 따른 대역폭 소모량 (HotData = 20%, HotData 갱신률 = 80%)

(그림 6)은 이동 호스트수의 변화에 따른 대역폭 소모량을 살펴본 것이다. 무효화 기법은 방송 메시지로 인한 대역폭 소모는 가장 적지만, 개별적인 데이터 요청이 증가하면서 대역폭 소모가 심해진다. 특히 사용 빈도가 높은 데이터가 갱신된 경우 무효화로 인해 다수의 이동 호스트에서 캐쉬 요청이 발생하므로 이동 호스트의 수가 많을수록 급격한 소모량 증가를 보이게 된다. 전파기법의 경우 무효화 기법과는 반대로 방송 대역폭의 소모가 큰 반면 개별적인 데이터 요청으로 인한 대역폭 소모는 가장 적다. S2 기법과 S4 기법은 무효화 기법이나 전파 기법보다 평균적으로 적은 대역폭을 소모했다. 이는 갱신 내용의 방송에 있어 사용 빈도가 높은 데이터에 대한 전파가 부분적으로 일어나기 때문에 방송 메시지를 수신한 이동 호스트는 서버로의 요청 없이도 새로운 데이터를 가질 수 있게 된다. 따라서 이동 호스트의 수가 증가되어도 무효화 기법처럼 대역폭 소모량이 급격하게 증가하지 않음을 확인할 수 있다.

4.2.2 캐쉬 적중률(Cache Hit Ratio)

캐쉬 적중률은 이동 호스트의 자치성을 판단하는 중요한 척도가 될 수 있다. 캐쉬 적중률이 높을수록 서버에 대한 데이터 요청이 감소하고, 캐쉬 손실도 줄어들게 되어 서버에 대

한 의존도가 감소하게 된다. 따라서 본 논문에서는 이동 호스트의 자치성을 나타내는 지표로 캐쉬 적중률을 사용하였다.



(그림 7) 이동 호스트 수의 변화에 따른 캐쉬 적중률 (P = 3, DT = 60)

(그림 7)은 이동 호스트 수에 따른 캐쉬 적중률을 보여준다. 무효화 기법의 경우 갱신된 데이터에 대한 무효화만을 수행하므로 캐쉬가 유효한 데이터를 가지고 있지 않을 확률이 높아 캐쉬 적중률이 좋지 않다. 또한 캐쉬 데이터의 유효성 여부를 판단할 수 없는 경우 캐쉬 내의 모든 데이터를 버리게 되므로 캐쉬 적중률이 크게 감소하게 된다. 전파 기법은 갱신 내용을 직접 방송을 통해 모두 전송해줌으로써 가장 높은 캐쉬 적중률을 나타낸다. 반면 S2 기법과 S4 기법은 접속 단절 이후에 재접속이 된 경우에도 접속 단절된 시간과 서버가 보관하고 있는 지난 방송 정보 주기가 동일하기 때문에 캐쉬 손실이 없고, 유효성 확인 요청 후에 최근의 갱신 내용을 얻을 수 있기 때문에 무효화 기법보다 우수한 캐쉬 적중률을 보인다.

5. 결 론

본 논문에서는 이동 컴퓨팅 환경에서 이동 호스트의 캐쉬 일관성 유지 기법들에 대해 살펴보고 문제점을 파악하여, 이를 보완하기 위한 선택적 캐쉬 일관성 유지 기법을 제안하였다. 제안한 기법에서는 방송 메시지를 전파 메시지와 무효화 메시지를 혼용하여 사용하기 때문에 전파 기법을 사용했을 때와 무효화 기법을 사용했을 때의 장점들을 겸용시킬 수 있었으며, 추가로 데이터에 대한 두 개의 테이블을 유지하여 이동 호스트에서 사용될 확률이 높은 데이터의 선별을 통한 효율적인 방송을 수행하게 된다. 또한 오랜 접속 단절 후에도 이동 호스트가 지난 방송 정보를 청취할 수 있는 기회를 제공하여 캐쉬 손실을 줄일 수 있다. 이러한 부가적인 정보들은 서버에서 유지되므로 서버가 대부분의 부하를 가지게 되어 이동 호스트의 부하를 최소화하였다.

실험을 통하여 제안한 기법이 무효화 기법을 사용했을 때와 전파 기법을 사용했을 때보다 대역폭 소모가 적어 대역폭 낭비나 지나친 점유를 막을 수 있었고, 평균적으로 높은 캐쉬 적중률을 보여줌으로써 이동 호스트의 자치성을 증대할 수 있게 되었음을 보였다.

향후 방송 채널의 대역폭의 변화에 유연하게 대처하여 전

파와 무효화의 비율을 동적으로 설정할 수 있도록 알고리즘을 개선하고, 이동 호스트의 서버 요청으로 인한 업링크 메시지를 줄일 수 있는 기법에 대한 연구가 지속되어야 할 것이다. 또한, 현재 사용하고 있는 일괄처리 트랜잭션보다 효율적인 트랜잭션 모델의 적용에 대한 연구가 필요하다.

참 고 문 헌

- [1] R. Alonso and H. F. Korth, "Database Systems Issues in Nomadic Computing," Proc. of the ACM SIGMOD Conf. on Management of Data, pp.388-392, 1993.
- [2] G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," IEEE Computer, Vol.27, No.6, pp.38-47, 1994.
- [3] S. Acharya and M. Franklin, "Balancing Push and Pull for Data Broadcast," Proc. of the ACM SIGMOD Conf. on Management of Data, pp.183-194, 1997.
- [4] J. Cai, K. Tan and B. Ooi, "On Incremental Cache Coherency Schemes in Mobile Computing Environments," Proc. of the IEEE Int. Conf. on Data Engineering, pp.114-123, 1997.
- [5] D. Barbara and T. Imielinski, "Sleepers and Workaholics : Caching Strategies in Mobile Environment," Proc. of the ACM SIGMOD Conf. on Management of Data, pp.1-24, 1994.
- [6] K. Wu, P. Yu and M. Chen, "Energy-Efficient Caching for Wireless Mobile Computing," Proc. of the IEEE Int. Conf. on Data Engineering, pp.114-123, 1997.
- [7] G. Cao, "A Scalable Low-Latency Cache Invalidation Strategy for Mobile Environments," Int. Conf. on Mobile Computing and Networking, pp.200-209, 2000.
- [8] 김희숙, 황병연, "이동 컴퓨팅 환경에서 접속 상태를 고려한 선택적 캐쉬 일관성 유지 기법," 제19회 정보처리학회 춘계학술대회논문집, 2003.



김 희 숙

e-mail : hs_kim@catholic.ac.kr

2001년 가톨릭대학교 컴퓨터공학과(공학사)
2003년 가톨릭대학교 컴퓨터공학과(공학석사)

관심분야 : 모바일 데이터베이스, 실시간 데이터베이스, XML, 전자상거래



황 병 연

e-mail : byhwang@catholic.ac.kr

1986년 서울대학교 컴퓨터공학과(공학사)
1989년 한국과학기술원 전산학과(공학석사)
1994년 한국과학기술원 전산학과(공학박사)
1999년~2000년 Univ. of Minnesota

Visiting Scholar

1994년~현재 가톨릭대학교 컴퓨터정보공학부 교수

2002년~현재 가톨릭대학교 정보통신원장

관심분야 : 공간 데이터베이스(GIS), XML, 모바일 데이터베이스, 전자상거래