

# 이동 데이터베이스 시스템에서 데이터의 위치와 영역 특성을 고려한 캐쉬 교체 기법

## (Cache Replacement Strategies considering Location and Region Properties of Data in Mobile Database Systems)

김 호 숙 <sup>†</sup>    용 환 승 <sup>\*\*</sup>  
(Ho-Sook Kim) (Hwan-Seung Yong)

**요 약** 최근 저가의 무선 통신 기술의 발달과 고성능 이동 컴퓨팅 장비의 보급에 따라 이동 컴퓨팅 시장은 점차로 확대되는 추세에 있다. 이동 컴퓨팅 환경에서의 제한된 대역폭, 잦은 단절과 배터리 제한 등의 제약성에 효율적으로 대처하기 위하여 여러 방법이 제안되었고, 특히 지구국에서 전송된 데이터 중 향후 사용 가능성이 높은 데이터를 캐쉬에 저장하고 이용하는 캐쉬 기법에 대한 많은 연구가 이루어지고 있다. 그러나 기존의 캐쉬 교체 기법들은 사용자의 이동성과 데이터의 공간 속성에 의한 특성을 고려하지 못하고 있기 때문에 그 효율성에서 한계를 가지고 있다.

본 논문은 이동 호스트의 위치 변화에 따라 이동 호스트의 캐쉬 내에 저장된 데이터의 가치와 의미가 변경되는 것을 보인다. 또한 지리(geographic) 데이터의 공간적 위치(location)와 지리 데이터가 영향을 미치는 공간적 범위 즉 영역(region)을 데이터의 공간 속성(spatial attributes)으로 정의하고, 시간에 따른 사용자의 이동성과 데이터의 공간 속성을 효과적으로 지원하는 새로운 캐쉬 교체 방법들을 제안하였다.

기존의 방법과 본 논문에서 제안한 캐쉬 교체 방법의 비교를 통하여 이동 호스트의 위치와 연관된 질의에 대한 성능 평가를 수행한 결과, 본 논문에서 제안한 캐쉬 교체 방법에 의한 캐쉬 적중률의 향상을 입증하였다. 또한 데이터 밀집도에 따라 캐쉬 교체 방법들의 성능이 변화함을 밝히고 이를 이용하여 이동 호스트가 지나가는 대상 지역의 데이터 밀집도에 따라 서로 다른 캐쉬 교체 방법의 선택이 필요함을 제시하였다.

**Abstract** The mobile computing service market is increasing rapidly due to the development of low-cost wireless network technology and the high-performance mobile computing devices. In recent years, several methods have been proposed to effectively deal with restrictions of the mobile computing environment such as limited bandwidth, frequent disconnection and short-lived batteries. Amongst those methods, much study is being done on the caching method - among the data transmitted from a mobile support station, it selects those that are likely to be accessed in the near future and stores them in the local cache of a mobile host. Existing cache replacement methods have some limitations in efficiency because they do not take into consideration the characteristics of user mobility and spatial attributes of geographical data.

In this paper, we show that the value and the semantic of the data, which are stored in the cache of a mobile host, changes according to the movement of the mobile host. We argue it is because data that are geographically near are better suited to provide an answer to a user's query in the mobile environment. Also, we define spatial location of geographical data and region, which is the spatial boundary that the geographic data has effect on, using the spatial attributes of data. Finally, we propose two new cache replacement methods that efficiently support user mobility and spatial attributes of data. One is based on the location of data and the other on the meaningful region of data.

From the comparative analysis of the performance of the previous methods and that of our cache replacement methods, we are able to show that our methods outperform others in that they improve the cache-hit ratio. Also, we show that performance varies according to data density-using this, we argue different cache replacement methods are required for regions with varying density of data.

<sup>†</sup> 학생회원 : 이화여자대학교 컴퓨터학과  
991COG04@cwaha.ac.kr

<sup>\*\*</sup> 종신회원 : 이화여자대학교 컴퓨터학과 교수  
hsyong@ewha.ac.kr

논문접수 : 1999년 8월 9일

심사완료 : 1999년 12월 15일

## 1. 서론

이동 컴퓨팅 환경(Mobile Computing Environment)은 사용자가 랩탑이나 노트북 등의 이동 가능한 장비를 휴대하고 무선 통신을 통해서 서버 컴퓨터의 자원과 함께 작업하는 환경을 의미한다[1,2]. 최근 이동 장비의 소형화, 고성능화와 함께 무선 통신의 기반 구축에 힘입어, 작업 위치의 제약 없이 항상 최신의 정보를 얻고, 이를 통해 좀 더 높은 차원의 정보 서비스를 창출하고자 하는 사용자들의 요구가 높아지면서 이동 컴퓨팅의 시장은 점차로 확대되고 있다[3].

이동 컴퓨팅 환경은 이동 지구국(Mobile Support Station)과 이동 호스트(Mobile Host 또는 Mobile Unit)로 구성된다. 이동 지구국은 고정 네트워크에 연결된 호스트들 중에서 이동 호스트에게 무선 통신 인터페이스를 지원하는 것으로서 데이터베이스 서버의 역할도 담당한다. 지구국이 담당하는 지리적 범위를 셀(cell)이라 하며 한 건물 내의 무선 LAN으로부터 휴대 전화망(cellular phone network)에 이르기까지 다양하다[4]. 이동 호스트는 음성과 다양한 문자 정보를 제공받을 수 있는 휴대용 전화기로부터 데이터베이스 시스템 기능을 포함하는 이동 가능한 단말 장치에 이르기까지 다양하다[5]. 점차적으로 작은 휴대 장비에도 데이터베이스 시스템의 기능을 추가하려는 경향이 있는데 이는 단절 상태에서 지속적으로 데이터 서비스를 지원할 수 있고, 기업 차원의 데이터베이스와의 동기화(synchronization)를 지원하며, 다양한 응용 프로그램을 쉽고 빠르게 개발할 수 있다는 장점에 바탕을 두고 있다 [6].

이동 컴퓨팅 환경에서는 무선 통신의 좁은 대역폭에 의해 전송 데이터 양의 제약을 받고, 전송 매체의 외부 간섭에 의한 잦은 단절이 발생할 수 있다. 이동 호스트는 시간에 따라 셀 간을 이동하면서 서로 다른 지구국과의 연결에 의해 트랜잭션을 지속하며, 전력 소모의 제한을 갖는다[2]. 이러한 특징을 고려하여 지구국에서 데이터를 전송할 때 자주 요구되는 데이터를 방송(broadcast)을 통해 전달함으로써 이동 호스트가 데이터를 요구하는데 필요한 에너지 소모를 줄이고 사용자 수의 증가에 효율적으로 대처하는 방법이 제안되었다[7]. 이와 함께 지구국에서 방송되는 데이터 중에서 향후 사용 가능성이 높은 데이터를 이동 호스트의 캐쉬에 저장하였다가 이후 사용자의 질의에 응답할 때 이용하는 캐쉬 기법이 도입되었다[8]. 캐쉬 데이터의 사용은 질의 응답 지연 시간을 줄이고 지구국과 이동 호스트와의 데이터 전송량을 줄이며, 정상 상태 뿐 아니라 통신이 단

절된 상태에서도 사용자의 질의에 대해 응답할 수 있는 장점을 갖는다[8].

현재까지 이동 컴퓨팅 환경의 여러 가지 제약점을 극복하기 위한 캐쉬 기법에 대한 연구는 주로 데이터의 접근 시간과 수정 빈도 등 데이터의 시간 속성을 중심으로 연구되었다[2,8]. 이러한 연구들은 이동 호스트가 시간에 따라 위치를 변화하는 특성과 이에 따라 캐쉬하고 있는 데이터들의 의미와 가치가 달라지는 것을 고려하지 못한 단점을 갖는다. 이동 호스트의 이동 방향과 범위를 예측하여 이를 서버에서 관리하는 방안이 Liu 등에 의해 연구되었다[9,10]. 그러나 이동 호스트의 숫자와 움직임에 대한 예측이 힘들고 다양하게 변하는 이동 컴퓨팅 환경에서는 사용자가 자신의 위치와 캐쉬 교체에 대한 책임을 지는 것이 바람직하며, 이동 호스트 스스로 자신의 위치 변화를 고려하여 캐쉬 교체를 수행하는 방법에 대한 연구가 필요하게 되었다. 일정한 지역 범위 내의 데이터들을 모아서 지리적으로 구분된 분산 데이터베이스를 구축하고 이를 질의에 이용함으로써 사용자의 위치와 데이터의 지역성(location dependent nature)을 고려하려는 연구가 수행되었다[11]. 그러나 이 연구는 같은 지역에 속한 데이터들도 각각의 데이터마다 영향을 미치는 고유한 범위를 갖는다는 점을 고려하지 못하였다. 그러므로 이동 호스트의 위치 이동과 데이터 자체의 공간적 속성을 고려하고, 이동하는 대상 지역의 지리 정보 변화에 효과적으로 대처하는 이동 호스트의 캐쉬 교체 기법이 필요하다.

본 논문은 기존의 캐쉬 교체 방법이 이동 호스트의 이동성과 데이터의 공간적 속성을 간과한 단점을 보완하기 위하여 첫째, 이동 호스트의 위치가 시간에 따라 변화하는 속성이 있고 이에 따라 이동 호스트 내에 캐쉬된 데이터에 대한 미래의 사용 가능성이 변경되는 점을 고려한 캐쉬 교체 기법을 제안한다. 둘째, 각각의 데이터의 공간상의 위치와 데이터가 영향을 미치는 공간적 범위 즉 데이터 영역을 데이터의 공간 속성으로 정의하고 이러한 속성을 고려한 캐쉬 교체 알고리즘을 제안한다. 이동 호스트의 위치 이동과 데이터의 공간 속성을 고려한 캐쉬 기법을 통해 캐쉬의 적중률 향상과 이동 호스트와 이동 지구국 사이의 데이터 요청 빈도의 감소, 질의에 대한 응답 시간의 단축 효과를 기대할 수 있다. 또한 본 논문에서 제안하는 캐쉬 교체 기법의 효율성을 검증하기 위하여 Informix Universal Server와 Informix Spatial Datablade Module을 사용하여 서대문구 지역의 데이터베이스를 구축하고 이동 호스트의 위치를 변경하면서 다양한 조건에서 본 논문에서 제안

한 캐쉬 교체 알고리즘의 성능을 평가하였다.

본 논문의 구성은 다음과 같다. 제2장은 이동 호스트의 위치 변화와 데이터의 공간 속성을 고려한 캐쉬 교체 기법을 제안하고, 제3장에서는 제안한 기법에 대한 성능 평가를 수행하며, 마지막으로 결론과 함께 앞으로의 연구 방향에 대해 기술한다.

## 2. 데이터의 공간 속성을 이용한 캐쉬 교체 기법

### 2.1 기본 가정과 용어 정의

본 절에서는 논문에서 사용한 기본 가정과 알고리즘 기술에 필요한 용어를 먼저 정의한다. 논문은 기본적으로 Babara가 제안한 이동 컴퓨팅 환경의 모델을 이용하며[2], 다음의 몇 가지를 가정한다. 첫째, 한 지구국은 자신이 담당하는 셀 영역내의 이동 호스트들에게 자신의 데이터베이스 내에 있는 데이터 정보를 방송한다. 둘째, 이동 호스트는 지구국의 셀 내에서 항상 자신의 위치를 알 수 있다. 셋째, 이동 호스트의 캐쉬 용량은 서버의 데이터베이스 용량보다 적기 때문에 각각의 이동 호스트는 서버에서 방송하는 데이터의 정보 중 적합한 데이터를 선택하여 자신의 캐쉬에 저장한다.

지구국에서 관리하는 셀 내에 위치하는 이동 호스트와 지리 데이터를 그림 1과 같은 표현을 사용하여 설명하도록 한다.

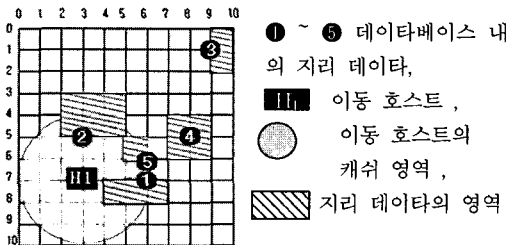


그림 1 셀 내에 위치하는 이동 호스트와 지리 데이터의 예

[정의 1] 데이터 위치  $DL_i$  (Data Location)는 데이터  $D_i$ 의 공간상의 위치로 한 점의 좌표로 표시한다

$$DL_i = ( X_i, Y_i )$$

각각의 데이터는 공간상의 위치와 더불어 해당 데이터가 의미를 갖는 논리적 범위를 갖는데 이러한 공간적 범위를 데이터 영역이라고 정의한다. 예를 들어 주식 값

의 범위는 한 국가 단위이며, 날씨 정보는 한 지역에 국한되고, 교통 상황 정보는 좀 더 좁은 범위에서 의미를 갖는다[3,7]. 동일하게 셀 내의 다양한 데이터도 각각 영향을 미치는 공간적 범위를 갖는다. 예를 들면 사방 1Km 마다 설치되어 있는 공중 전화 박스가 의미를 갖는 범위는  $1Km^2$  이지만 사방 5Km 마다 위치하고 있는 주유소가 의미를 갖는 범위는  $25Km^2$ 으로 공중 전화의 데이터 영역과 비교하여 25배의 크기를 갖는다고 볼 수 있다. 현실 세계에서 공간상의 데이터 영역은 다양한 다각형의 형태를 갖지만 본 논문에서는 해당 다각형을 둘러싸고 있는 최소 경계 사각형(MBR: Minimum Boundary Rectangle)으로 표현하고, 좌측 하단과 우측 상단 두 점의 좌표로 표현한다. 데이터 영역을 원이 아닌 MBR로 정의한 것은 데이터가 영향을 미치는 범위를 결정하는 데에는 거리 뿐 아니라 지형적, 행정적인 다른 요인이 함께 작용하기 때문이다. 즉 두개의 데이터가 이동 호스트를 중심으로 같은 거리에 위치하고 있더라도 강이나 산으로 막혀 있거나, 일방 통행 등 교통상의 특징에 따라서 실제로 데이터가 영향을 미치는 범위가 달라질 수 있고, 동이나 구 등의 행정 구역의 분할에 따라 담당 범위가 결정될 수도 있기 때문이다.

[정의 2] 데이터 영역  $DR_i$  (Data Region)은 데이터  $D_i$ 가 영향을 미치는 공간적 범위로 좌측 하단과 우측 상단 두 점의 좌표로 표시한다.

$$DR_i = ( X_1, Y_1, X_2, Y_2 )$$

[정의 3] 데이터의 속성은 ID, Location, Region, Class로 구성된다. 이때 데이터 ID는 하나의 데이터베이스 내에서 유일한 정수 값으로 데이터를 구분하는 키 값이며, 데이터의 Class 속성은 주차장, 호텔, 병원 등 데이터의 용도에 따른 의미적 분류이다.

[정의 4] 이동 호스트( $H_i$ )의 속성은 이동 호스트의 중심 좌표( $X_{h_i}, Y_{h_i}$ )와 이동 호스트의 캐쉬가 담당하는 지역의 범위를 결정하는 반지름( $R_i$ )으로 구성된다.

$$H_i = ( X_{h_i}, Y_{h_i}, R_i )$$

[정의 5] 이동 호스트  $H_i$ 의 캐쉬 크기( $C_{max_i}$ )는 캐쉬 내에 저장 가능한 데이터의 개수이다.

[정의 6] 데이터 밀집도는 이동 호스트의 캐쉬 영역( $CR_i$ )에 포함되는 대상 지역의 데이터의 개수이다.

위의 정의에 따라 그림 1의 지리 데이터에 대한 속성을 나타낸 것은 표 1과 같다. 그림 1에서 이동 호스트

$H_1$ 은 (3, 7, 3)으로 표기되며 이때 데이터의 위치를 중심으로 하는 해당 지역의 데이터 밀집도는 2이고, 데이터의 영역을 중심으로 하는 해당 지역의 데이터 밀집도는 3이다.

표 1 그림 1의 지리 데이터에 대한 데이터베이스 테이블

ID	Location	Region	Class
1	(6,7)	(4,8,7,7)	주차장
2	(3,5)	(2,5,5,3)	주차장
3	(9,1)	(9,2,10,0)	호텔
4	(8,5)	(7,6,9,4)	주차장
5	(6,6)	(5,6,6,5)	병원

**2.2 이동 호스트의 위치 변화와 데이터의 위치를 고려한 캐쉬 교체 기법**

이동 호스트는 시간에 따라 위치를 변화하는 속성을 갖는데, 질의가 요청된 지리적 장소(이동 호스트의 위치)에 따라 질의의 결과 값이 달라지며, 이에 따라 셀 내에 존재하는 지리 데이터가 갖는 의미와 중요도가 달라진다. 그림 2는 이러한 상황을 설명하는 예이다. (3, 7)의 위치에 있는 이동 호스트  $H_1$ 의 경우 "가장 가까운 곳에 위치한 주차장의 ID와 Location을 조회하라."는 질의 Q1의 결과는 (2, (3, 5))이지만, (7, 3)의 위치에 있는 이동 호스트  $H_2$ 의 경우 같은 질의에 대한 결과는 (4, (8, 5)) 이다. 이와 같은 결과는 같은 데이터를 이용한 동일한 질의에 대해서도 이동 호스트의 현재 위치에 따라 다른 결과 값을 나타내는 것을 보여주며, 이에 따라 이동 호스트 내에 캐쉬 되는 데이터에 대한 의미와 가치를 다르게 평가해야 하는 필요성을 보여준다. 이동 호스트가 서버에서 방송하는 데이터 중 일부를 자신의 캐쉬에 저장할 때, 이동 호스트의 현재 위치에 따라서 이동 호스트의 캐쉬 영역 안에 데이터의 위치가 포함되는 데이터를 선택한다면,  $H_1$ 의 캐쉬 내에는 1, 2의 데이터가 캐쉬 될 것이며,  $H_2$ 의 캐쉬 내에는 3, 4의 데이터가 저장되어 있게 된다.

위에서 살펴본 이동 호스트의 위치에 따른 캐쉬 내용 변화의 필요성을 이동 호스트의 캐쉬 교체 알고리즘에 적용시켜 보자. 이동 호스트가 시간에 따라 위치를 이동하고 이에 따라 새로운 데이터가 자신의 캐쉬 영역 내로 들어오게 되면 좀 더 사용 가능성이 높은 데이터를 자신의 캐쉬에 저장하기 위해 prefetch를 수행할 수 있다. 또한 사용자의 질의에 대한 데이터가 이동 호스트의

캐쉬에 없는 경우 서버로부터 해당 데이터를 받아서 자신의 캐쉬 내용을 수정한다. 이때 이동 호스트의 캐쉬 크기가 한정되어 있으므로 적절한 캐쉬 교체 알고리즘을 수행한다. 캐쉬 교체 알고리즘은 캐쉬 내의 데이터 각각에 대한 캐쉬 교체 점수(CRS: Cache Replacement Score)를 계산하여 이용하는데 데이터의 위치를 기준으로 캐쉬 교체를 수행할 때 캐쉬 교체 점수는 이동 호스트  $H_i$ 의 현재 위치의 중심 좌표( $X_{hi}, Y_{hi}$ )와 데이터  $D_j$ 의 위치 ( $X_j, Y_j$ )사이의 거리로 정의한다. 즉 이동 호스트는 자신의 캐쉬 내에 있는 각각의 데이터에 대해서 교체 점수를 계산하여 새로 캐쉬 내로 삽입할 데이터보다 교체 점수가 높은 (거리가 더 멀리 있는) 기존의 데이터를 캐쉬에서 삭제하고 캐쉬 교체 점수가 낮은 새 데이터를 삽입한다.

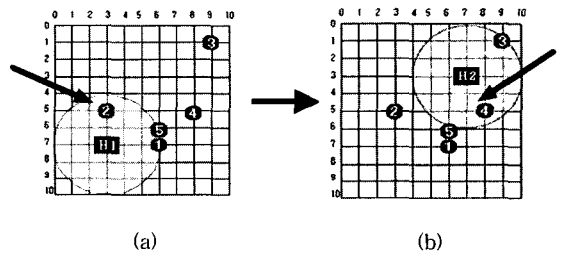


그림 2 이동 호스트 위치에 따른 질의 Q1의 결과 변화와 데이터 위치 기준의 캐쉬 변화 :  
 (a)  $H_1$ 의 위치가 (3, 7)일 때 질의 Q1의 결과는 (2, (3, 5))이며  $H_1$ 의 캐쉬에는 캐쉬 영역과 겹쳐지는 데이터 1, 2가 저장되어 있다.  
 (b)  $H_2$ 의 위치가 (7, 3)일 때 질의 Q1의 결과는 (4, (8, 5))이며  $H_2$ 의 캐쉬에는 데이터 3, 4가 저장되어 있다.

[정의 7] 이동 호스트  $H_i$ 의 캐쉬에서 데이터의 위치를 기준으로 한 데이터  $D_j$ 의 데이터 캐쉬 교체 점수(CRS\_L: Cache Replacement Score based on Location of Data)는 이동 호스트  $H_i$ 의 중심 좌표 ( $X_{hi}, Y_{hi}$ )와 데이터  $D_j$ 의 위치 ( $X_j, Y_j$ ) 사이의 거리이다.

$$CRS\_L(D_j) = \sqrt{(X_{hi} - X_j)^2 + (Y_{hi} - Y_j)^2}$$

그림 3은 CRS\_L을 이용한 캐쉬 교체 알고리즘이다. 이때 사용되는 변수의 의미는 다음과 같다.

- $D_i$  : 캐쉬 내에 새로 삽입할 데이터
- $C_{cur}$  : 현재 캐쉬 내에 저장된 데이터의 총 개수

$D_{high}$  : 현재 캐쉬 내에 있는 데이터 중에서 가장 큰 캐쉬 교체 점수를 갖는 데이터

```

Replace_Cache_Loc (Di)
{
  if (Ccur < Cmax) then {
    Di를 캐쉬에 삽입한다. ;
    Ccur = Ccur + 1; }
  else { /* 캐쉬의 교체가 필요한 경우 */
    캐쉬내에 존재하는 모든 데이터에 대하여 CRS_L을
    재 계산한다. ; .....㉔
    Dhigh을 찾는다. ;
    Dhigh를 캐쉬에서 제거한다. ;
    Di를 캐쉬에 삽입한다. ;
    CRS_L(Di)를 계산한다. ; ..... ㉕
  } /*else*/
} /*end of Replace_Cache_Loc */
    
```

그림 3 데이터 위치를 기준으로 한 캐쉬 교체 알고리즘

이동 호스트의 캐쉬 크기가 4일 때 이동 호스트가 위치를 변화함에 따라서 Replace\_Cache\_Loc 알고리즘을 적용한 예가 그림 4에 나타나 있다.

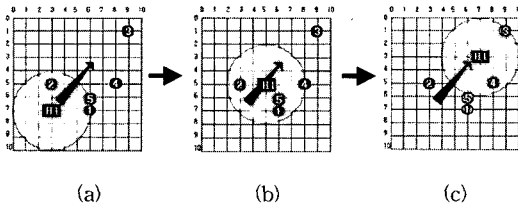


그림 4 CRS\_L을 이용한 캐쉬 교체 알고리즘의 적용 예 :

- (a)  $H_1$ 의 위치가 (3, 7)일 때 캐쉬에는 데이터 1, 2가 저장된다.
- (b)  $H_1$ 의 위치가 (5, 5)일 때 캐쉬에는 데이터 1, 2, 4, 5가 저장된다.
- (c)  $H_1$ 의 위치가 (7, 3)일 때 캐쉬 크기가 4이므로 2번 데이터가 삭제되고 3번 데이터가 삽입되어 캐쉬에는 데이터 1, 3, 4, 5가 저장된다.

### 2.3 이동 호스트의 위치 변화와 데이터의 영역을 고려한 캐쉬 교체 기법

이동 데이터베이스 환경에서 데이터 영역을 고려할

때 이동 호스트의 위치에 따른 질의 결과의 변화를 살펴보자. 그림 5는 지리 데이터에 대해서 데이터 영역을 고려하는 경우 데이터 위치만을 고려하는 것과 다른 결과를 보여주는 예이다. 즉 각 주차장의 규모나 편리성, 이동의 편리성 등을 고려한 데이터 영역을 빗금 친 사각형으로 표현 할 때 이동 호스트의 위치가 (3, 7)일 때 “가장 가까운 곳에 위치한 주차장의 ID와 Location을 조회하라.”는 질의 Q1에 대해서 데이터의 위치만을 고려한 (a)경우는 (2, (3, 5))가 리턴 되지만 데이터의 영역을 고려한 (b)의 경우는 (1, (6, 7))이 리턴 된다.

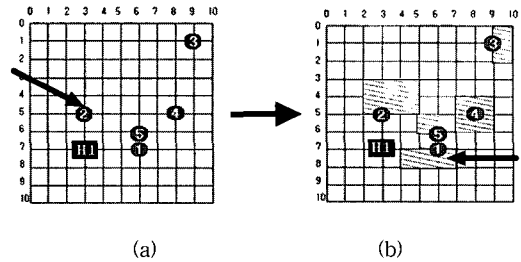


그림 5 데이터 영역을 고려할 때 질의 Q1의 결과 변화 :

- (a) 데이터의 위치만을 고려한 경우 질의 Q1의 결과는 (2, (3, 5)) 이다.
- (b) 데이터의 영역을 고려한 경우 질의 Q1의 결과는 (1, (6, 7)) 이다.

이동 호스트가 셀 사이를 이동할 때는 각 지구국 사이의 핸드 오프 작업이 이루어지고 이동 호스트 내의 캐쉬 된 데이터 항목에 대해서 무효화시키는 등의 변경이 필요하다. 그러나 셀 사이를 이동할 때마다 캐쉬를 무효화하는 것은 아직 유효한 데이터 정보를 함께 수정하게 되므로 결국 필요 없는 네트워크 전송량의 증가와 캐쉬의 수정을 발생시키게 된다. 그러므로 각 데이터마다 데이터가 유효한 공간적인 범위를 갖고 이 범위를 벗어날 때만 캐쉬에서 제거하면 캐쉬의 효율을 높일 수 있다[7]. 이동 호스트의 캐쉬 크기가 한정되어 있으므로 셀 간을 이동할 때 뿐 아니라 하나의 셀 내에서도 이동 호스트의 위치가 변화할 때 캐쉬 내용의 변경이 필요하다. 이때 캐쉬의 효율성을 높이기 위해서 데이터 영역이 캐쉬가 담당하는 공간 범위를 벗어 날 때 캐쉬에서 제거하는 방법을 적용할 수 있다.

그림 6은 데이터의 영역을 고려할 때 이동 호스트의

위치에 따른 이동 호스트의 캐쉬 내용의 변화를 나타낸다. 이동 호스트 H<sub>1</sub>과 H<sub>2</sub>의 캐쉬 반지름은 3이고 캐쉬 크기는 4이다. 이동 호스트는 서버가 방송하는 데이터의 정보를 듣다가 자신의 캐쉬 영역과 데이터의 영역이 겹쳐지는 데이터를 캐쉬 한다고 하면 이동 호스트의 위치가 (3, 7)인 H<sub>1</sub>의 경우 캐쉬 내에는 1, 2, 5의 데이터가 캐쉬 되어 있고, 위치가 (7, 3)인 H<sub>2</sub>의 경우 2, 3, 4, 5의 데이터를 캐쉬하게 된다.

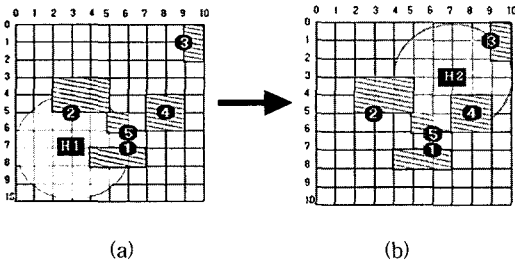


그림 6 이동 호스트 위치에 따른 데이터 영역 기준의 캐쉬 변화 :

- (a) H<sub>1</sub>의 위치가 (3, 7)일 때 캐쉬 내에는 1, 2, 5의 데이터가 저장된다.
- (b) H<sub>2</sub>의 위치가 (7, 3)일 때 캐쉬 내에는 2, 3, 4, 5의 데이터가 저장되게 된다.

데이터 위치(DL<sub>i</sub>) 대신 데이터 영역(DR<sub>i</sub>)을 기준으로 그림 3의 캐쉬 교체 알고리즘을 수정하여 보자. 그림 3의 ㉔, ㉕에서 CRS<sub>L</sub> 대신 데이터의 영역을 기준으로 한 캐쉬 교체 점수(CRS<sub>R</sub>)를 이용하여 캐쉬 교체를 수행하도록 한다. CRS<sub>R</sub>은 이동 호스트의 중심 좌표와 데이터 영역간의 거리를 기준으로 캐쉬 교체 점수를 계산하는 방식으로써 공간상의 한 점(X<sub>h</sub>, Y<sub>h</sub>)과 사각형(X<sub>1</sub>, Y<sub>1</sub>, X<sub>2</sub>, Y<sub>2</sub>) 사이의 거리를 계산하는 함수 DIS(X<sub>h</sub>, Y<sub>h</sub>, X<sub>1</sub>, Y<sub>1</sub>, X<sub>2</sub>, Y<sub>2</sub>)를 이용한다.

```
DIS (Xh, Yh, X1, Y1, X2, Y2)
/* 공간상의 점(Xh, Yh)과 사각형(X1, Y1, X2, Y2) 사이의 거리 */
{
  case1 : 점이 사각형 내부에 위치하는 경우
    DIS = 0
  case2 : 점이 사각형의 각 변을 연장한 십자부분 내부에 있는 경우
    DIS = 점으로부터 가장 가까운 사각형의 변까지의 거리
}
```

```
case3 : 점이 사각형의 각 변을 연장한 십자부분 외부에 있는 경우
DIS = 점으로부터 가장 가까운 사각형의 꼭지점까지의 거리
}
```

그림 7 공간상의 한 점과 사각형 사이의 거리를 계산하는 함수

DIS 함수를 이용하여 데이터 영역을 고려한 캐쉬 교체 점수를 다음과 같이 정의한다.

[정의 8] 데이터 영역을 고려한 캐쉬 교체 점수(CRS<sub>R</sub>: Cache Replacement Score based on Region of Data)는 이동 호스트의 중심 좌표(X<sub>h</sub>, Y<sub>h</sub>)와 각 데이터 영역(X<sub>1</sub>, Y<sub>1</sub>, X<sub>2</sub>, Y<sub>2</sub>) 사이의 거리이다.

$$CRS\_R(D_i) = DIS (X_h, Y_h, X_1, Y_1, X_2, Y_2) \quad \blacksquare$$

그림 8은 이동 호스트의 캐쉬 크기가 4일 때 CRS<sub>R</sub>을 이용하여 캐쉬 교체 알고리즘을 적용할 때의 단계별 캐쉬 내용 변화를 나타낸다

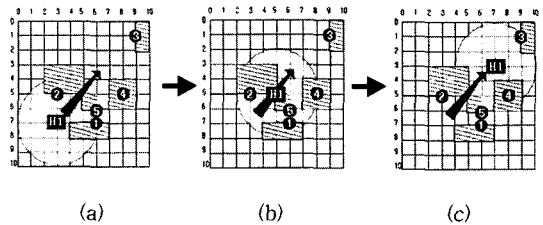


그림 8 CRS<sub>R</sub>을 이용한 캐쉬 교체 알고리즘의 적용 예 :

- (a) H<sub>1</sub>의 위치가 (3, 7)일 때 캐쉬에 저장된 데이터는 1, 2, 5
- (b) H<sub>1</sub>의 위치가 (5, 5)일 때 캐쉬에 저장된 데이터는 1, 2, 4, 5
- (c) H<sub>1</sub>의 위치가 (7, 3)일 때 캐쉬 크기가 4이므로 1번 데이터가 삭제되고 3번 데이터가 삽입되어 캐쉬에 저장된 데이터는 2, 3, 4, 5 이다.

그림 8의 (C)에서 데이터의 위치로만 비교하면 1번 데이터가 2번 데이터보다 이동 호스트의 위치 (7, 3)에 더 가깝지만 데이터의 영역이 큰 2번 데이터가 이동 호

스트의 캐쉬에 오랫동안 남아있게 된다. 또한 5번 데이터의 영역은 (a)부터 (c)까지 지속적으로 캐쉬 영역 내에 남아 있는데 이는 데이터 영역의 크기가 작더라도 이동 호스트의 이동 경로 상에 존재하는 경우 계속 의미 있는 데이터로 남겨지는 것을 보여주는 예이다 .

### 3. 성능평가

본 장에서는 논문에서 제안한 방법들의 성능 평가를 위하여 기존의 캐쉬 교체 방법과 제안한 두 가지 캐쉬 교체 방법을 비교하여 다음과 같은 성능 평가를 수행한다. 첫째, 이동 호스트 캐쉬 크기의 변화에 따른 캐쉬의 적중률 변화를 살펴보고 이를 통하여 제안한 캐쉬 교체 방법의 성능 향상을 입증한다. 둘째, 데이터 분포가 서로 다른 지역을 이동 호스트가 통과하는 경우, 제안한 캐쉬 교체 방법들의 적중률을 비교함으로써 지역의 데이터 밀집도에 따른 캐쉬 교체 모델들의 성능 변화를 살펴본다.

#### 3.1 제안된 모델

본 논문에서 제안한 방법들의 성능을 비교 평가하기 위하여 표 2와 같이 세 가지 모델을 설정한다.

표 2 성능 평가에 사용할 캐쉬 교체 모델

제안모델	캐쉬 교체의 기준	적용CRS
CR_LRU	데이터의 삽입/수정 시간	LRU
CR_LOC	데이터의 위치와 이동 호스트와의 거리	CRS_L
CR_REG	데이터의 영역과 이동 호스트와의 거리	CRS_R

CR\_LRU (Cache Replacement by LRU) 모델은 기존의 캐쉬 교체 알고리즘 가운데 LRU 방법을 이용하여 이동 호스트의 캐쉬 데이터를 교체하는 방법이다. 즉 캐쉬 내에 저장되어 있는 모든 데이터에 대하여 데이터 접근(access) 시간에 대한 정보를 관리하면서, 캐쉬 교체가 필요한 경우 가장 오랫동안 접근되지 않은 데이터를 캐쉬에서 삭제하는 방법으로 캐쉬 교체 모델의 대표적인 방법중 하나이다.

CR\_LOC (Cache Replacement by LOCation of data) 모델은 2장에서 제안한 CRS\_L을 기준으로 캐쉬를 교체하는 모델로서 캐쉬 내의 각각의 데이터는 이동 호스트의 현재의 위치와 데이터 위치와의 거리에 대한 정보를 유지한다. 이동 호스트의 캐쉬 교체가 필요한 경우에 현재의 이동 호스트의 위치와 가장 거리가 먼 데이터가 캐쉬에서 삭제된다.

CR\_REG (Cache Replacement by REGion of

data) 모델은 2장에서 제안한 CRS\_R을 기준으로 캐쉬를 교체하는 모델로서 캐쉬 내의 각각의 데이터는 이동 호스트의 현재의 위치와 데이터 영역간의 거리에 대한 정보를 유지한다. 이동 호스트의 캐쉬 교체가 필요한 경우에 현재의 이동 호스트의 위치로부터 데이터의 영역이 가장 멀리 위치한 데이터가 캐쉬에서 삭제된다.

CR\_LOC와 CR\_REG의 경우 이동 호스트의 위치가 변경됨에 따라 기존의 캐쉬 내에 들어있는 데이터 각각에 대한 캐쉬 교체 점수가 변하게 되는데, 이러한 변경 정보를 이동 호스트의 움직임에 따라 실시간으로 반영하는 것은 수많은 계산의 반복을 발생시킨다. 특히 움직임이 많고 질의가 적은 사용자의 경우에는 많은 자원과 에너지의 낭비를 초래하게 된다. 본 논문에서는 이동 호스트의 위치 변경에 따른 데이터의 공간 속성을 반영하면서 동시에 캐쉬 교체 점수 계산의 반복에 의한 자원의 낭비를 막기 위해서, 이동 후 호스트의 위치로부터 이전의 캐쉬 교체 점수를 계산했던 이동 호스트의 위치까지의 거리가 캐쉬 반지름 보다 큰 경우에 한해서 이동 호스트 내의 데이터에 대한 캐쉬 교체 점수를 재 계산한다.

#### 3.2 시스템 구현 환경 및 데이터베이스 구축

데이터의 공간 속성을 고려한 캐쉬 교체 알고리즘의 성능 평가를 위한 시스템은 Sun 사의 Solaris 2.6 기반의 객체-관계 DBMS인 Informix Universal Server를 사용하였다[12]. Database로부터 공간 데이터를 효율적으로 접근하게 하는 공간 데이터 베이스 엔진으로 사용한 Informix Spatial Datablade Module은 Informix Universal Server Database의 기존의 질의와 함께 2차원 연산을 지원하기 위해 9개의 data type들과 45개의 SQL function들을 제공하며 공간 색인을 위해 R-Tree를 지원한다[13]. 그밖에 프로그램에 필요한 다양한 함수 구현을 위해 Informix Datablade API와 ESQL/C, SPL function을 사용하였다.

지구국의 데이터베이스에서 관리하는 대상 지역은 서대문구 대현동을 중심으로 한 가로 5.5km 세로 3.3km의 직사각형 영역이다. 100m를 한 단위로 하여 대상 지역의 데이터에 대한 x, y 위치 좌표를 설정하고 이를 기반으로 서버의 지리 정보 데이터베이스를 구축하였다. 데이터의 위치와 영역을 포함하는 서버의 지리 정보 테이블(MAP\_SERVER TABLE)은 다음과 같다.

```
Create table map_server
(
```

```

id          smallint not null , /* 데이터를 구분하는
           primary key */
location    sp2Pnt , /* 위치 좌표 (X, Y) */
region      sp2Box , /* 데이터의 영역 (X1 ,Y1 ,X2 ,Y2 ) */
class       char(3) , /* 용도를 나타내는 분류 */
name        varchar(20), /* 데이터의 이름 */
attributes  set (varchar(20) not null) /* 데이터의 특징을 저장
                                           장 */
);
    
```

그림 9 MAP\_SERVER TABLE 정의

데이터베이스에서 데이터의 위치는 x, y 좌표 값으로 결정하고, 각 데이터의 영역은 전체 대상 지역의 범위 안에 같은 용도를 갖는 데이터의 개수에 반비례하도록 Class별로 크기를 결정하였다. 즉 대상 지역의 범위 내에 같은 용도를 갖는 데이터가 많은 경우는 데이터 수가 작은 경우 보다 상대적으로 작은 크기의 데이터 영역을 갖게 된다.

$$\begin{aligned}
 \text{전체 대상 지역의 크기} &= a \\
 \text{같은 Class를 구성하는 데이터의 개수} &= b \\
 \text{Class별 데이터의 영역 크기} &= \frac{a}{b} \Rightarrow c \\
 \text{Class별 데이터의 가로, 세로의 길이} &= \sqrt{c} \Rightarrow d \\
 \text{해당 데이터의 위치} &= (X_i, Y_i) \text{ 일 때} \\
 \text{데이터 영역은} &= (X_i - d, Y_i - d, X_i + d, Y_i + d) \text{의 사각형} \\
 &\text{으로 결정된다.}
 \end{aligned}$$

그림 10 데이터의 영역을 결정하는 방법

성능 평가에서 사용할 질의 테이블은 전체 200개의 질의로 구성되어 있고 이 가운데 질의를 수행한 결과 오로지 하나의 데이터만이 선택되는 exact match 형태의 질의가 170개 존재하며, 결과 값이 리스트로 나타나는 형태의 질의가 30개 있다. 질의 테이블(QUERY TABLE)은 다음과 같다.

```

Create table query
(
  qid          smallint not null , /* 질의를 구분하는 key */
  query_state  char(400) ,
  answer_id    set (smallint not null) ,
  select_cnt   smallint ,
  /* 질의문 */
  /* 질의 결과 데이터 집합 */
  /* 질의 선택 횟수 */
);
    
```

```

rel_location  sp2Box
              /* 질의 결과의 데이터 위치와 연관된 지역 */
rel_region    sp2Box
              /* 질의 결과의 데이터 영역과 연관된 지역 */
);
    
```

그림 11 QUERY TABLE 정의

### 3.3 성능 평가 프로그램

성능 평가를 위한 프로그램은 이동 호스트의 캐쉬 크기와 초기 좌표 값 등의 파라미터를 입력받는 초기화 수행으로 시작하며, 이동 호스트의 위치가 지구국에서 서비스하는 셀 영역을 벗어나는 때까지 수행을 반복한다. 먼저 이동 호스트가 위치하는 지역과 연관된 질의들을 선택한다. 선택된 질의 리스트를 수행하는 과정에서 이동 호스트의 캐쉬 내에 저장된 데이터의 적중률을 평가한다. 한 위치에서 선택된 질의를 모두 수행한 후 위치를 이동하는데 이때 변경 후의 위치가 이전의 캐쉬 교체 점수를 계산한 위치에서의 캐쉬 영역 반지름을 벗어난 경우, 이동 후 위치에서의 캐쉬 영역에 대한 prefetch를 수행한다. 이동 호스트의 위치가 셀 영역을 벗어나면 각 위치별 적중률의 평균을 계산하고 그 결과를 기록한다.

성능 평가의 기준인 적중률(hit ratio)은 발생한 전체 질의 수에 대하여 이동 호스트의 캐쉬에 질의에 해당하는 데이터가 존재하는 비율로 측정한다.

[정의 9] 캐쉬 적중률 =

$$\frac{\text{캐쉬내에 해당 데이터가 존재하는 경우의 질의 수}}{\text{전체 질의 수}}$$

캐쉬의 적중률을 계산하는 질의 수행 과정을 자세히 살펴보면 그림 12와 같다. 먼저 이동 호스트는 발생한 질의를 자신의 캐쉬 데이터를 이용하여 수행한다. 자신의 캐쉬 내에 해당 데이터가 있는 경우에는 hit수를 증가시키고 다음 질의를 수행한다. 만약 자신의 캐쉬에 질의에 대한 데이터가 존재하지 않으면 지구국의 데이터베이스 서버로 질의를 전달하고, 서버에서 질의 수행 후 결과를 이동 호스트로 전달한다. 이때 이동 호스트의 캐쉬에 비어 있는 저장 공간이 있으면 캐쉬 교체 없이 데이터의 삽입이 수행되나 캐쉬의 저장 공간이 모두 사용되고 있는 경우에는 캐쉬 교체가 발생한다. 이때 사용하는 캐쉬 교체 방법에 따라 CR\_LRU, CR\_LOC, CR\_REG 방법을 구분하게 된다.

성능 평가의 환경은 서로 다른 크기의 캐쉬를 갖는



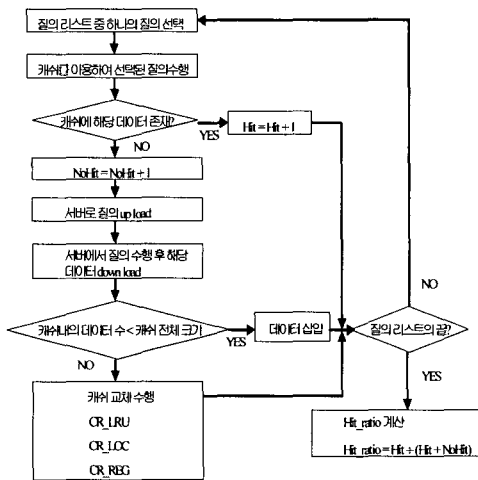


그림 12 질의 수행 순서도

다양한 종류의 이동 호스트들이 데이터의 밀집도가 다른 지역을 이동하면서 각 지역별로 발생하는 질의를 수행하는 것으로 캐쉬 교체 모델별로 이동 호스트의 캐쉬 적중률을 측정한다. 이동 호스트의 캐쉬 조건과 시간에 따른 위치 이동을 조절하기 위하여 성능 평가에 사용한 파라미터들은 표 3과 같다.

표 3 입력 파라미터

파라미터	값 (하한/상한)	설명
Xhcur	0 / 55	이동 호스트의 X 좌표 값
Yhcur	0 / 33	이동 호스트의 Y 좌표 값
NextX	-10 / 10	이동 호스트의 X 방향 이동 크기
NextY	-10 / 10	이동 호스트의 Y 방향 이동 크기
Cmax	10 / 100	이동 호스트 캐쉬 크기 (단위: 데이터 개수)
CRradius	1 / 10	이동 호스트 캐쉬 반지름
Query_cnt	1 / 70	한 위치에서 발생하는 질의 수
ALPA	0 / 5	캐쉬 영역에 대한 질의 분산도 (해당 지역에서 발생하는 질의가 이동 호스트의 캐쉬영역을 벗어나는 정도)
CR_type	0, 1, 2	캐쉬 교체 방법 0: CR_LRU 1: CR_LOC 2: CR_REG

### 3.4 성능 평가 결과

캐쉬 크기 변화에 따른 캐쉬 교체 방법간의 성능을

평가하기 위하여 CR\_LRU, CR\_LOC, CR\_REG 모델을 함께 비교하였다. 그림 13은 캐쉬 크기를 30부터 90까지 변경하면서 캐쉬 교체 방법별 적중률을 구한 결과이다. 이때 사용한 캐쉬 반지름의 크기는 6이고 한 지역에서 발생한 질의의 수는 40이다.

그림 13은 이동 호스트의 위치와 관련된 질의 발생 빈도가 높은 이동 컴퓨팅 환경에서, 데이터의 공간 속성을 이용한 CR\_LOC 방법과 CR\_REG 방법이 기존의 CR\_LRU 방법에 비해 전반적으로 높은 적중률을 나타내는 결과를 보여준다.

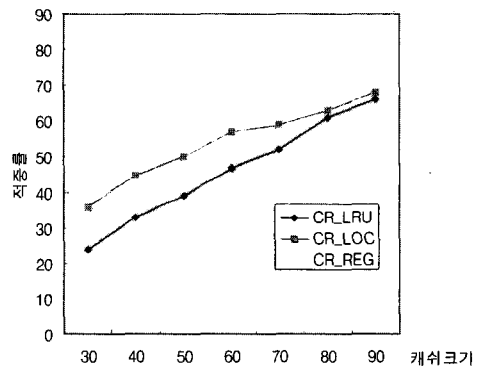


그림 13 캐쉬 크기 변화에 따른 캐쉬 교체 방법별 성능 평가: 데이터의 공간 속성을 기준으로 캐쉬 교체를 수행한 CR\_REC 방식과 CR\_LOC 방식이 기존의 CR\_LRU 방식에 비해 높은 적중률을 나타낸다.

또한 CR\_LRU 방법의 적중률이 캐쉬 크기의 증가와 비례하여 커지는데 비해서 CR\_LOC와 CR\_REG 방법은 캐쉬 크기가 커지면서 적중률의 증가 비율이 점차 줄어드는 결과를 보인다. 이는 이동 호스트의 캐쉬 크기가 이동 호스트가 담당하고 있는 영역의 데이터 수보다 커지는 경우에는 데이터의 공간 속성에 의해 선택된 데이터 외의 데이터까지 포함하게 됨으로써 캐쉬 적중률에 미치는 영향이 줄어들기 때문이다.

지리 정보 데이터베이스의 데이터 분포는 지역에 따라 서로 다른 데이터 밀집도를 갖는다. 캐쉬 크기가 고정되어 있는 이동 호스트가 서로 다른 데이터의 밀집도를 갖는 지역을 통과할 때 캐쉬 교체 방법에 따른 성능을 평가하기 위해서 데이터의 밀집도가 다른 6가지 이동 경로를 선택하여 각각의 경로에 대해 CR\_LOC와 CR\_REG 방법을 이용한 성능 평가를 수행하였다. 즉

데이터의 밀집도가 높은 지역을 통과하는 3가지 경로를 밀집 경로 1, 2, 3으로 하고 밀집도가 낮은 지역을 통과하는 3가지 경로를 희소 경로 1, 2, 3으로 설정하여 실험하였다. 대상 지역의 데이터 분포와 실험에서 설정한 밀집 경로와 희소 경로는 그림 14와 같다.

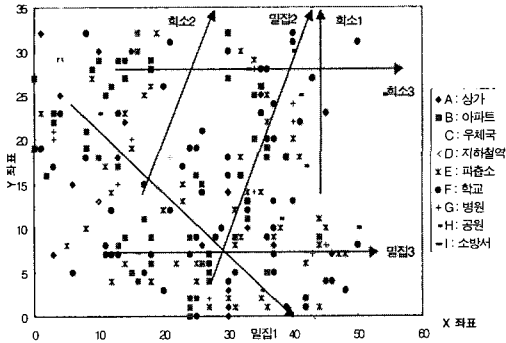


그림 14 데이터 분포와 이동 경로

데이터의 밀집도에 따라서 캐쉬 교체 방법들의 적중률을 비교한 결과는 그림 15와 같다. 이동 호스트의 캐쉬 크기에 비해 통과하는 지역의 데이터 수가 많은 데이터 밀집 지역에서는 CR\_LOC 방법이 CR\_REG 방법에 비해 높은 적중률을 나타낸 반면, 데이터의 수가 희박한 데이터 희소 지역에서는 CR\_REG 방법이 CR\_LOC 방법에 비해 높은 성능을 나타내었다. 이러한 결과는 일정한 범위의 영역을 처리할 때 캐쉬에서 담당하는 데이터의 양이 CR\_LOC 방법에 비해 CR\_REG 방법이 더 많기 때문이다. 그러므로 이동 호스트가 자신의 캐쉬 크기보다 더 많은 데이터가 존재하는 지역을 통과하는 경우에는 이동 호스트의 영역과 겹쳐지는 데이터를 캐쉬 내에 모두 저장하지 못하는 상황이 발생하고, 이에 따른 지속적인 캐쉬 교체 발생에 의해 CR\_REG 방법의 성능이 떨어지는 결과를 나타낸다.

4. 결론 및 향후 연구 과제

최근 차세대 고속 무선 통신망의 기반 구축과 이동 컴퓨팅 단말기의 보급 확대를 기반으로 위치의 제약에서 벗어나 좀 더 다양한 정보 서비스를 실시간으로 제공받으자 하는 사용자들의 요구가 늘어나고 있고 이를 지원하기 위한 많은 연구가 수행되고 있다. 그러나 이동 컴퓨팅 환경은 좁은 대역폭과 잦은 단절, 전력 소모의 제한과 시간에 따른 이동성이라는 특징으로 인하여 기존의 데이터베이스를 기반으로 하는 응용 시스템과는

다른 많은 문제점을 갖고 있다. 본 논문에서는 이러한 문제들을 해결하기 위하여 기존에 제안된 캐쉬 정책에 대하여 살펴보고, 기존의 방법들이 사용자의 이동성과 데이터의 공간 속성의 특징을 지원하기에는 많은 단점을 갖고 있음을 지적하였다.

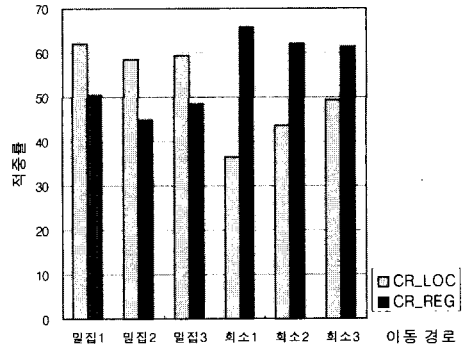


그림 15 데이터 밀집도에 따른 캐쉬 교체 방법별 성능 평가 : 데이터 밀집 지역을 이동하는 경우 CR\_LOC 방법이 우수하며, 데이터 희소 지역을 이동하는 경우에는 CR\_REG 방법이 우수한 결과를 나타낸다.

본 논문에서는 시간에 따른 사용자의 이동성과 데이터의 공간 속성을 지원하면서 다양한 밀집도를 갖는 지역을 이동하는 이동 호스트에 대하여 효율적으로 데이터베이스 서비스를 지원하기 위한 캐쉬 교체 방법으로 CR\_LOC와 CR\_REG의 방법을 제안하였다. CR\_LOC 방법은 이동 호스트의 캐쉬 데이터를 교체할 때 이동 호스트의 현재 위치와 데이터의 위치와의 거리를 계산하여 이동 호스트와 가장 먼 거리에 위치하는 데이터를 캐쉬에서 삭제하는 방법이다. CR\_REG 방법은 각각의 데이터의 영역을 고려한 것으로 이동 호스트의 캐쉬 데이터를 교체할 때 이동 호스트의 현재 위치와 데이터의 영역 사이의 거리를 계산하여 이동 호스트와 데이터의 영역이 가장 먼 데이터를 캐쉬에서 삭제하는 방법으로써 영역이 큰 데이터는 영역이 작은 데이터에 비해서 오랫동안 캐쉬에 남아 있게 된다.

제안된 방법들의 성능을 비교, 평가하기 위하여 데이터의 공간 속성을 포함한 데이터베이스를 구성하고, 이동 호스트가 대상 지역을 이동하면서 이동 호스트의 위치와 연관된 질의에 대한 처리를 수행하는 환경 하에서

기존의 캐쉬 교체 방법 중 대표적인 LRU 방법과 함께 세 가지 성능 평가를 수행하였다. 성능 평가 결과 첫째, 이동 호스트의 위치와 연관된 질의에 대하여 논문에서 제안한 CR\_LOC, CR\_REG 방법 모두 CR\_LRU 방법에 비해 높은 적중률을 보였다. 둘째, 대상 지역의 데이터 밀집도에 따른 캐쉬 교체 방법들의 적중률을 비교한 결과 이동 호스트의 캐쉬 크기에 비해 과다하게 데이터 수가 많은 데이터 밀집 지역을 통과 할 때는 CR\_LOC 방법이 CR\_REG 방법에 비해 높은 적중률을 나타낸 반면 데이터의 수가 희박한 데이터 희소 지역을 통과하는 경우에는 CR\_REG 방법이 CR\_LOC 방법에 비해 높은 성능을 나타내었다. 이러한 결과는 대상 지역의 데이터 밀집도에 따라서 적절한 캐쉬 교체 알고리즘을 선택함으로써 이동 호스트의 캐쉬에 대한 적중률을 높일 수 있는 방법을 제시하고 있다. 이를 통하여 이동 호스트와 지구국 사이의 데이터 요청 빈도를 감소시키며, 사용자의 질의에 대한 응답 시간을 단축하여 전체 시스템의 성능을 향상시킬 수 있다.

향후 서로 다른 캐쉬 크기를 갖는 이동 호스트들이 이동함에 따라서 지속적으로 대상 지역의 밀집도가 변화하는 이동 데이터베이스 환경에서 이동 호스트의 캐쉬 크기와 지역의 대역폭 그리고 해당 지역의 데이터 밀집도에 따라 prefetch를 최소화하면서 최상의 적중률을 달성할 수 있는 캐쉬 교체 방법에 대한 연구가 필요하다.

### 참 고 문 헌

- [1] Rafael Alonso, Henry F. Korth, "Database System Issues in Nomadic Computing," *Proceedings of the ACM SIGMOD*, pages 388-392, 1993.
- [2] Daniel Babara, Tomasz Imielinski, "Sleepers and Workaholics: Caching Strategies in Mobile Environments," *Proceedings of the ACM SIGMOD*, pages 1-12, 1994.
- [3] Tomasz Imielinski, Henry F. Korth, "Introduction to Mobile Computing," *Mobile Computing*, Kluwer Academic Publishers, pages 1-43, 1996.
- [4] Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, McGraw-Hill, pages 722-726, 1997.
- [5] Sybase, *Embedded Database Technology: UltraLite*, Sybase Press, 1998.
- [6] Frank Stajano, Alan Jones, "The Thinnest of Clients: Controlling It All via Cellphone," *A Publication of the ACM SIGMOBILE*, pages 46-53, 1998.
- [7] Tomasz Imielinski, S. Viswanthan, "Wireless Publishing: Issues and Solutions," *Mobile Computing*, Kluwer Academic Publishers, pages 299-329, 1996.
- [8] Boris Y. Chan, Antonio Si, Hong V. Leong, "Cache Management for Mobile Databases: Design and Evaluation," *IEEE CS Data Engineering*, pages 54-63, 1998.
- [9] George Y. Liu, General Q. Maguire, JR. , "A Mobile-Aware Dynamic Database Caching Scheme for Wireless Mobile Computing and Communications," *Distributed and Parallel Database 4*, pages 271-288, 1996.
- [10] George Liu, "Exploitation of Location-dependent Caching and Prefetching Techniques for Supporting Mobile Computing and Communications", The 6th International Conference on Wireless Communications, 1994.
- [11] Margaret H. Dunham, Vijay Kumar, "Location Dependent Data and its Management in Mobile Databases," *Mobility in Database and Distributed System*, 1998.
- [12] Informix, *Informix Universal Server Guide to SQL: Tutorial Version 9.1*, Informix Press, 1997.
- [13] Informix, *Informix Spatial Datablade Module: User's Guide*, Informix Press, 1997.



김 호 숙

1993년 이화여자대학교 전자계산학과 학사. 1993 ~ 1997년 삼성SDS 전임연구원. 1999년 이화여자대학교 컴퓨터학과 공학석사. 1999년 ~ 현재 이화여자대학교 컴퓨터학과 박사과정. 관심분야는 공간 데이터베이스 시스템, 데이터 마이닝



용 환 승

1983년 서울대학교 컴퓨터공학과 학사. 1985년 서울대 대학원 컴퓨터공학과 공학석사. 1985년 ~ 1989년 한국전자통신연구소 연구원. 1994년 서울대 대학원 컴퓨터공학과 공학박사. 1994년 서울대 컴퓨터신기술공동연구소 특별연구원. 1995년 ~ 현재 이화여자대학교 컴퓨터학과 부교수. 관심분야는 객체-관계 데이터베이스 시스템, 멀티미디어 데이터베이스, 데이터 웨어하우징 및 OLAP.