

위치 기반 서비스에서 서술 기반의 시멘틱 프리페칭 기법을 이용한 동적 질의 처리

(Dynamic Query Processing Using Description-Based Semantic Prefetching Scheme in Location-Based Services)

강 상 원 [†] 송 의 성 ^{**}
(Sang-Won Kang) (Ui-Sung Song)

요 약 위치 기반 서비스는 질의하는 클라이언트의 위치에 따라 질의에 대한 결과를 제공한다. 위치 기반 서비스에서, 전통적인 캐싱과 프리페칭 같은 기법은 서버로부터의 데이터 전송과 질의 응답 시간을 감소시키기 위해 효과적인 접근 방법이다. 그러나 클라이언트의 다양한 이동성과 질의 패턴으로 이들 기법은 캐시의 비효율성과 네트워크 부하를 초래할 수 있다. 이러한 단점들을 해결하기 위해, 사용자가 요구한 질의와 질의 결과를 표현하는 세그먼트들의 집합으로 정의된, 서술을 이용하여 프리페칭 세그먼트 개념과 개선된 캐시 교체 전략을 함께 사용하는 시멘틱 프리페칭 기법을 제안한다. 이동 클라이언트는 시멘틱 프리페칭 영역이라고 불리는 서비스 영역에 진입할 때 서버로부터 미리 필요한 시멘틱 정보를 가져온다. 이동 컴퓨팅 환경에서 클라이언트는 위치 종속 데이터의 질의 처리를 위해 캐시에 미리 그 정보를 유지한다. 제안 기법의 성능은 사용자 이동성과 질의 패턴, 위치 종속 데이터의 분포, 적용 캐시 교체 전략과 같은 다양한 환경 변수와 관련 지어 조사된다. 시뮬레이션 결과는 제안 기법이 영역 질의와 최 근접 질의에 대해 기존 기법보다 효율적임을 보여준다. 추가로, 질의 처리에 두 질의를 동적으로 반영하여 적용하면 성능이 향상됨을 알 수 있다.

키워드 : 시멘틱 프리페칭, 영역 질의, 최 근접 질의, 캐시 교체, 지역성

Abstract Location-Based Services (LBSs) provide results to queries according to the location of the client issuing the query. In LBS, techniques such as caching and prefetching are effective approaches to reducing the data transmission from a server and query response time. However, they can lead to cache inefficiency and network overload due to the client's mobility and query pattern. To solve these drawbacks, we propose a semantic prefetching (SP) scheme using prefetching segment concept and improved cache replacement policies. When a mobile client enters a new service area, called semantic prefetching area, proposed scheme fetches the necessary semantic information from the server in advance. The mobile client maintains the information in its own cache for query processing of location-dependent data (LDD) in mobile computing environment. The performance of the proposed scheme is investigated in relation to various environmental variables, such as the mobility and query pattern of user, the distributions of LDDs and applied cache replacement strategies. Simulation results show that the proposed scheme is more efficient than the well-known existing scheme for range query and nearest neighbor query. In addition, applying the two queries dynamically to query processing improves the performance of the proposed scheme.

Key words : Semantic prefetching, Range query, Nearest neighbor query, Cache replacement, Locality

1. 서 론

무선 네트워크 환경에서 위치 기반 서비스(LBSs: Location based services)는 무선 단말의 위치 정보와 사용자의 위치에 따라 변하는 위치 종속 데이터(LDD: Location dependent data)를 이용하여 사용자에게 서비

[†] 학생회원 : 고려대학교 컴퓨터학과

swkang@disys.korea.ac.kr

^{**} 정 희 원 : 부산교육대학교 컴퓨터교육과 교수

ussong@bnue.ac.kr

논문접수 : 2007년 3월 20일

심사완료 : 2007년 6월 15일

스를 제공한다. 응급 서비스, 여행자 가이드, 길 찾기 정보, 지역 지도 정보, 자녀의 위치 추적 서비스 등이 LDD 기반 LBS의 대표적인 예이다[1,2]. 기본적으로 LBS는 사용자 프로파일 기반의 서비스를 제공한다. 사용자 프로파일에는 사용자의 식별자와 주소 같은 기본 정보뿐만 아니라, 사용자의 선호도, 이동 패턴, 질의 패턴과 같은 시멘틱 정보도 포함될 수 있다.

사용자 맞춤형 LBS를 위해 사용자 위치와 관련된 많은 연구가 이동 컴퓨팅 환경에서 수행되었다. 그러나, 기존 연구는 이동 클라이언트가 공간 질의(Spatial Query)를 하였을 때 서버가 이동 클라이언트의 위치에 따라 질의 결과를 제공해 주는 것에 초점을 두었기 때문에, 시멘틱 정보를 이용하여 사용자의 기호까지 반영한 특화된 맞춤형 서비스를 제공하기 위한 연구는 미미하였다. 기존 연구에서 사용자의 시멘틱 정보는 서버에서 유지되므로, 시멘틱 정보 처리는 서버 측에서만 가능하였다[3,4]. 이동 클라이언트에 의해 서버에 전송된 질의는 R-tree 색인, 검색, 모니터링 기법과 같은 서버 기반 질의 처리 방법을 이용하여 처리되었다[5-8]. 질의 처리 관점에서 이러한 방식은 서버로부터 이동 클라이언트가 질의한 결과를 얻기 위해 많은 네트워크 부하를 일으키게 된다.

본 논문은 이동 클라이언트가 고속 데이터 전송을 지원하는 시멘틱 프리페칭 영역(SPA: Semantic Prefetching Area)에 진입했을 때, 서버로부터 미리 LDD에 대한 시멘틱 정보를 얻도록 함으로써 이동 클라이언트의 서버에 대한 빈번한 질의 요구를 줄이고 통신 부하를 감소시키는 새로운 시멘틱 프리페칭(SP: Semantic Prefetching) 기법을 제안한다. 제안하는 기법에서 미리 전송하는 시멘틱 정보는 사용자가 이전에 요구한 질의와 질의 결과를 표현하는 세그먼트들의 집합으로 정의된, 서술을 통해 구성된다. 이동 클라이언트는 프리페칭 서술로 명명된 정보를 통해 사용자의 요구사항과 기호를 반영할 수 있다. 또한 서버와의 통신 부하를 효율적으로 줄이기 위해서 기존 캐시 교체 전략들을 개선한 캐시 교체 전략들도 제안한다. 제안한 SP 기법의 성능 측정을 위해 다양한 이동 클라이언트 유형과 여러 캐시 교체 전략들에 기반하여 시뮬레이션이 수행되었으며 캐시 적중률과 통신 비용 측면에서 기존의 기법보다 좋은 결과를 보여주었다.

본 논문은 다음과 같이 구성된다. 2장은 LBS를 위한 데이터 전송 기법, 캐시 관리 기법, 질의 처리와 관련된 최근의 관련 연구를 살펴본다. 3장은 본 논문에서 제안하는 SP기법을 설명한다. 4장은 SP 기법에서의 질의 처리 방법과 캐시 교체 전략들을 설명한다. 5장은 실험 모델과 다양한 이동 클라이언트 유형을 통한 시뮬레이

션 결과들을 분석한다. 마지막으로, 6장에서 결론을 맺는다.

2. 관련 연구

LDD 환경에서 이동 클라이언트와 서버간의 데이터 전송은 제한된 통신 대역폭 등으로 인해 과도한 네트워크 부하를 초래할 수 있다. 이동 클라이언트와 서버가 네트워크 부하를 줄이기 위해 부분적인 정보만을 교환한다면, 이동 클라이언트는 부정확한 정보를 유지하게 되고 결국 사용자 서비스 품질(QoS: Quality of Service)은 저하된다. 특히, 제한된 통신 대역폭은 서버가 이동 클라이언트에게 사용자가 관심 있어하는 LDD 정보를 원활히 전송시키지 못하게 할 수 있다. 이러한 문제의 해결을 위해, 프리페칭 기법[9,10], 방송 기법[11], 유연한 네트워크 구조[12] 등이 효율적인 데이터 전송을 위해 연구되어왔으며, 또한 사용자가 관심 있는 정보의 효율적인 관리를 위해 다양한 캐시 교체 전략들이 연구되었다[13,14].

사용자는 자신의 관심사항에 대해서 빈번히 질의하기 때문에 이동 클라이언트의 질의는 사용자의 기호를 반영한다. 그러므로, 제한적인 대역폭을 사용하는 이동 컴퓨팅 환경에서 사용자가 자주 질의할 것으로 예측되는 질의의 응답을 미리 전달받는 프리페칭 기법은 결과를 얻기까지의 평균 지연 시간과 통신 비용을 감소시킬 수 있다[9,10]. [11]에서는 방송 채널을 통해 객체에 대한 접근을 허용하는 키를 프리페칭하는 방법을 제안하였다. 이 방법은 사용자 접근율, 객체 갱신 빈도, 데이터 크기 등에 기반한 데이터 유포 기법을 사용하였으나, 특정 서비스에 대한 사용자의 기호나 관심사항이 반영된 질의는 처리할 수 없었다. 이동 컴퓨팅 환경에서는 질의를 수행한 사용자의 위치에 따라 서로 다른 결과를 얻을 수 있으므로, 시스템 관리자가 데이터의 공간적인 지역성을 반영하여 영역을 구분해 놓아야만 서버로부터 얻어진 위치 기반 질의 결과의 타당성을 보장받을 수 있다. 이것과 관련된 대표적인 연구로 구분된 영역에서 프리페칭 기법을 사용하여 위치 종속 질의를 처리하는 인포스테이션 시스템이 있다[12]. 인포스테이션 시스템은 이동 컴퓨팅 환경에서 사용자에게 대량의 데이터를 신속하게 전송할 수 있으나, 프리페칭을 위한 데이터 전송 시 사용자의 기호나 관심사항 등이 고려된 데이터 전송은 수행하지 못한다.

이동 컴퓨팅 환경에서 이동 클라이언트의 데이터 캐싱은 서버에 부하를 주지 않으면서 사용자의 데이터 접근을 용이하게 한다[15,16]. 페이지 캐싱 기법은 디스크 페이지를 단위로 RDBMS에서 널리 이용되었으나 질의 정보 자체를 유지하지는 못하며, 객체에 대한 키 값의

동일 선택 질의(equi-select query) 결과만을 보장한다 [17]. 그러나, 시멘틱 캐싱은 캐시에 데이터와 시멘틱 서술을 함께 유지한다[13,18]. [14,19,20]에서는 사용자의 현재 위치와 관심 속성값들을 수식으로 표현하여 시멘틱 영역으로 정형화시켰으며, 시멘틱 캐시를 사용하여 위치 기반 질의 처리에 대한 시멘틱 지역성도 반영하였다. 그러나, 사용자의 이동 패턴과 질의 패턴을 충분히 반영하지 못했고, 캐시에서 이동 클라이언트가 진입한 새로운 영역에 대한 정보를 얻을 수도 없었다.

사용자가 계속해서 움직이는 이동 컴퓨팅 환경에서 사용자의 위치에 따라 캐시 유효성을 보장하는 것은 중요한 문제이다. 비트 벡터 접근법이 캐시 일관성을 유지하고 유효 영역을 보장하기 위하여 제안되었다[21]. [14]에서는 이동 컴퓨팅 환경에서 LDD 질의를 위한 캐시 교체 전략을 제안하였다. FAR(Farthest away replacement) 캐시 교체 전략은 사용자의 현재 위치로부터 가장 멀리 떨어진 시멘틱 영역을 교체한다. 그러나, 다양한 접근 패턴과 높은 이동성의 사용자가 시멘틱 캐시를 이용하면 빈번한 위치 갱신으로 인해 캐시 내용은 유효하지 않게 된다. 그러므로, FAR 전략은 이동성과 접근 패턴을 결합한 경우 약점을 지니게 된다[13].

[22]에서는 Voronoi 다이어그램을 기반으로 R-tree로 색인된 정적인 점들에 대해 이동 질의를 하여 최 근접 이웃을 찾아내는 방법을 제안하였다. [23]에서는 k개의 정적인 객체들과 이동 질의들을 처리하기 위한 k-최 근접 질의 처리 방법이 다루어졌다. [6]에서는 LBS에서 최 근접 질의와 원도우 질의를 처리하기 위한 방법이 제안되었다. 그러나, 위의 연구들은 사용자의 질의 패턴이나 특정 서비스에 대한 사용자의 기호나 관심사항이 반영된 질의 처리에는 한계를 지닌다. 따라서, 본 논문에서는 서술 기반의 시멘틱 프리페칭과 동적인 질의 처리를 통해 사용자의 기호나 관심사항을 고려하여 사용자의 질의를 효과적으로 처리할 뿐만 아니라, 효율적인 데이터 전송과 캐시 관리가 이루어지는 기법을 제안한다.

3. 시멘틱 프리페칭

3.1 시멘틱 프리페칭 기법의 동작과정

일반적으로, 이동 컴퓨팅 환경은 셀을 기반으로 하여 영역이 구분되고 기지국(Base station)을 통해 메시지 전송이 수행된다. 본 논문에서는 이러한 이동 컴퓨팅 환경을 기반으로 사용자가 특정 영역으로 이동할 때 필요한 정보를 미리 얻을 수 있도록 하는 논리적인 서비스 영역을 시멘틱 프리페칭 영역(SPA)으로 정의하였다. 이와 함께, 제안 기법을 수행할 때 전송되는 정보는 시멘틱 프리페칭 채널(SPC: Semantic Prefetching Channel)을 이용한다. 제안 기법에서 사용하는 채널은 기존에 사용

하던 기지국 채널(BSC: Base Station Channel)과 달리 많은 양의 데이터를 전송하는데 용이해야 한다. 본 논문에서 가정하는 환경은 최근 이슈가 되고 있는 HSDPA(High Speed Downlink Packet Access)나 Wibro와 같은 휴대 인터넷 기술이 적용된 환경에 적용될 수 있다. 제안하는 시멘틱 프리페칭 기법은 클라이언트-서버 모델을 기반으로 하며, SPA에서 이동 클라이언트와 서버는 다음과 같이 동작한다.

처음으로 동작을 시작한 이동 클라이언트가 특정 SPA로 최초 진입하면, 해당 SPA 서버는 이동 클라이언트의 정보를 가지고 있지 않으므로, 이동 클라이언트는 미리 전달 받을 정보가 없게 되어 캐시에는 저장할 데이터가 없게 된다. SPA에 상주하는 동안 이동 클라이언트는 서버에 필요할 때마다 LDD 질의를 수행하게 되고, 서버로부터 얻은 각 LDD 질의 정보와 질의 결과들은 시멘틱 세그먼트라고 불리는 형태로 시멘틱 캐시에 저장된다. 현재 상주하던 SPA에서 다른 SPA로 이동할 때, 이동 클라이언트는 상주하던 SPA에서 수행한 모든 LDD 질의 정보를 담고 있는 로그 파일을 서버에게 기지국 채널(BSC: Base Station Channel)을 이용하여 전송한다. 이후 이동 클라이언트는 새로운 SPA로 이동할 때마다 위의 과정을 반복한다. 이동 클라이언트가 예전에 방문했던 SPA를 재 방문하면, 서버는 이전에 전송 받았던 로그 파일내의 LDD 질의 정보를 분석하여, 각 질의 정보에 대해 프리페칭 세그먼트라고 불리는 서술 형태의 데이터를 생성한다. 각 프리페칭 세그먼트는 로그 파일내의 질의 정보 및 질의 결과와 관련된 내용을 담고 있다. 서버에 의해 생성된 프리페칭 세그먼트 집합은 고속의 다운링크 채널인 시멘틱 프리페칭 채널(SPC: Semantic Prefetching Channel)을 통해 이동 클라이언트에게 전송된다. 이동 클라이언트는 자신의 시멘틱 캐시에 저장되어 있는 시멘틱 세그먼트들과 서버로부터 얻은 프리페칭 세그먼트들을 서로 결합시켜 서술 테이블을 생성한 후 시멘틱 캐시에 저장한다. 현재 상황에 적합한 서술 테이블을 생성하여 저장하기 위해 필요에 따라 캐시 교체 전략이 사용된다. 이후 이동 클라이언트의 질의 결과는 먼저 시멘틱 캐시에 저장된 서술 테이블을 통해 얻게 되며 서술 테이블을 통해 얻지 못한 결과들은 서버에게 추가적으로 질의하여 얻게 된다. 서버를 통해 추가적으로 얻은 질의 결과들은 서술 테이블에 반영되어 서술 테이블에 추가되거나 기존 세그먼트 정보를 갱신하게 된다. 기존 시멘틱 캐싱(SC: Semantic Caching) 기법에서는 이동 클라이언트의 LDD 질의와 질의 결과들을 시멘틱 캐시에 저장하였다. 이 기법에서는 이동 클라이언트가 새 SPA 방문 바로 이전 영역의 SPA들에서 생성한 정보를 시멘틱 캐시에 유지

하기 때문에, 새로이 재 진입한 SPA에서 필요한 내용과는 맞지 않는 LDD 관련 정보를 가지고 있을 확률이 높다. 이 경우 빈번한 캐시 실패로 이동 클라이언트와 서버는 상호간에 빈번한 질의 요청과 질의 응답을 하게 되어 네트워크 부하를 가중시킨다. 또한, 질의 결과를 얻을 때까지 발생하는 지연 시간으로 즉각적인 서비스가 이루어지지 않아 QoS를 저하시키게 된다.

본 논문에서 제안한 시멘틱 프리페칭(SP: Semantic Prefetching) 기법은 이동 클라이언트가 SPA에 진입하고 나갈 때 기존 시멘틱 캐싱 방식에 비해 추가적인 네트워크 트래픽을 발생시킨다. 그러나, SPA로 진입할 때 서버로부터 미리 사용자의 기호나 관심 사항이 반영된 LDD 정보를 제공받기 때문에, 캐시 실패를 감소시킬 수 있어 네트워크 부하를 감소시키고 캐시 정보를 이용한 즉각적인 서비스도 제공해 주어 QoS를 향상시킬 수 있다.

3.2 시멘틱 프리페칭의 구성과 모델링

3.1절에서 나온 주요 용어들의 정의와 관련 개념들을 설명한 후 시멘틱 캐시에 서술 테이블이 어떻게 저장되고 구성되는지 설명한다.

LDD 질의 처리를 위해서는 이동 클라이언트와 LDD 데이터의 위치를 표현하는 방법이 필요하다. 본 논문에서는 정수 값을 가지는 2차원 공간 좌표를 이용하여 다음과 같이 위치를 표현한다. 공간 좌표를 구성하는 X와 Y에 의해 형성된 그리드 영역 $G = X \times Y$ 에서, LDD 데이터 집합 D_f 는 그리드 영역 G에 존재한다. D_f 의 각 구성요소와 사용자의 위치는 좌표 (L_x, L_y) 로 표현되며, $L_x \in X$ 이고 $L_y \in Y$ 이다.

이동 클라이언트는 SPA 영역을 벗어날 때, SPA에서 수행한 시멘틱 질의 정보를 담고 있는 로그 파일 L_f 를 서버에 전송한다. 이동 클라이언트가 SPA로 재 진입하면, 서버의 프리페칭 관리자는 L_f 를 분석하고 이동 클라이언트에게 전달할 내용들을 결정한다. 이때, 각 내용들은 프리페칭 세그먼트 $F_i = \langle F_r, F_a, F_p, F_l, F_c \rangle$ 의 형태로 표현되며 프리페칭 세그먼트의 집합인 프리페칭 서술 F_d 가 이동 클라이언트에게 전송된다. L_f 로부터 생성된 프리페칭 서술 F_d 는 이전 방문했을 때의 시멘틱 질의 정보와 질의 결과를 담고 있다. 프리페칭 세그먼트 F_i 의 F_r 과 F_a 는 내용의 관계와 속성, F_p 는 F_i 의 튜플들이 만족하는 SELECT 조건을 표현한다. F_i 은 이동 클라이언트의 위치를, F_c 는 F_i 의 결과를 저장하고 있는 메모리의 첫 페이지 번호를 표현하며, F_i 의 실제 내용은 페이지 단위로 저장된다.

$\langle F_r, F_a, F_p, F_l, F_c \rangle$ 로 표현되는 F_i 의 각 튜플은 수학적으로 다음과 같이 정의된다. $F_r \in L_f$, $F_a \subseteq A$ (A는 L_f 의 속성집합), $F_p = P_1 \vee P_2 \vee \dots \vee P_j \vee \dots \vee P_m$ 이며

각 P_j 는 $P_j = b_{j1} \wedge b_{j2} \wedge \dots \wedge b_{jk} \wedge \dots \wedge b_{jn}$ 로 비교 수식 b_{jk} 의 논리곱으로 표현된다. 비교 수식 b_{jk} 는 서버의 로그 파일 L_f 와 로그 파일의 속성 집합 A에서, $a \text{ op } c$ 의 형태로 표현되는 수식이다. 여기서, $a \in A$, $op \in \{\leq, <, \geq, >, =\}$, 그리고 c 는 지정 영역에 대한 상수 값이다. [24]에서 언급한 것처럼 비교 연산자 \neq 은 NP-hard 문제를 야기할 수 있기 때문에 본 논문에서는 비교 연산자 \neq 을 고려하지 않았다. F_i 은 (L_x, L_y) 로 이동 클라이언트의 위치를 표현하며, F_c 는 $F_c = \pi F_a \sigma F_p(F_r)$ 로 표현된다. 여기서 σ 는 SELECT 연산자이고, π 는 PROJECT 연산자를 의미한다. 프리페칭 세그먼트들은 비교 수식들을 포함하는 선택조건을 가진 SELECT-PROJECT 연산의 결과들을 알 수 있다. 서버에서 생성된 프리페칭 서술 F_d 는 내용과 색인으로 구성되며, 프리페칭 서술내의 모든 프리페칭 세그먼트 F_i 는 하나 또는 여러 개의 연결 페이지에 저장된다. 예제를 통해 프리페칭 서술 F_d 가 구성되는 방법을 설명한다.

예. restaurant (Rtype, Rname)과 wine bar(Wname, Wrank)이라는 두 가지 관계를 가지는 위치 종속 데이터베이스가 존재하고 사용자가 특정 SPA를 이전에 방문했을 때 다음과 같은 질의를 수행했다고 하자.

- 점심 시간에 위치 (32,40)에서 이동 클라이언트는 “3킬로미터 내에 있는 한정식 음식점의 이름을 모두 알려주세요.”라는 질의 Q_1 를 수행했다.
- 저녁 시간에 위치 (35,38)에서 이동 클라이언트는 “5킬로미터 내에서 가격이 50000원 이하인 와인을 판매하는 상점에 대한 정보와 상점 이름을 알려주세요.”라는 질의 Q_2 를 수행했다.

SPA를 재 방문하면, 이동 클라이언트는 서버로부터 표 1과 같은 프리페칭 서술 색인과 질의 Q_1 , Q_2 의 결과를 전송 받는다. 표 1에서 F_c 의 값 7, 11은 질의 Q_1 , Q_2 에 대하여, 만족하는 실제 내용을 저장하고 있는 페이지의 첫 번째 번호를 표현한다. 이 페이지 번호를 통해 Q_1 , Q_2 의 결과에 접근할 수 있다.

프리페칭 세그먼트 F_i 가 표현된 형식과 유사하게 질의 Q와 시멘틱 세그먼트 S_i 도 표현될 수 있다. 질의 Q는 $\langle Q_r, Q_a, Q_p, Q_l, Q_c \rangle$ 와 같이 표현하며 질의는 결과를 얻기 전이므로, $Q_c = \emptyset$ 이다. 시멘틱 세그먼트 S_i 는 $S_i = \langle S_r, S_a, S_p, S_l, S_c \rangle$ 와 같이 표현한다. 시멘틱 세그먼트의 집합인 시멘틱 서술 S_d 에는 이동 클라이언트가 거쳐간 SPA 영역에서 질의했던 질의 관련 시멘틱 정보와 질의 결과가 저장되며, 시멘틱 서술 S_d 도 프리페칭 서술과 유사하게 내용과 색인으로 구성된다. 프리페칭 세그먼트와 시멘틱 세그먼트는 동일한 유형을 가지는 형태이지만, 서버와 이동 클라이언트의 시멘틱 정보에 대한 구별을 위해, 본 논문에서는 서버 측 시멘틱 정보는 프리페

표 1 프리패칭 서술 색인의 구성 예

F	F_r	F_a	F_p	F_l	F_c
F_1	Restaurant	{Rname, Rtype}	$(L_x-3 \leq Rxposition \leq L_x+3)$ $\wedge (L_y-3 \leq Ryposition \leq L_y+3)$ $\wedge Type*=2$	32,40	7
F_2	Wine bar	{Wname, Wrank}	$(L_x-5 \leq Wxposition \leq L_x+5)$ $\wedge (L_y-5 \leq Wyposition \leq L_y+5)$ $\wedge (Price < 50000)$	35,38	11

*Type: 1-유럽식, 2-한정식, 3-일식, 4-중식, 5-멕시코식

칭 세그먼트라고 부르고, 이동 클라이언트 측의 시멘틱 정보는 시멘틱 세그먼트라고 부른다.

서버로부터 수신한 F_d 와 S_d 의 효율적인 관리를 위해 모든 프리패칭 세그먼트와 시멘틱 세그먼트들의 합으로 구성되는 서술 테이블 $\Gamma(\Gamma = A \cup B, A = \forall F_i, B = \forall S_i)$ 을 시멘틱 캐시에 저장한다. 서술 테이블 Γ 가 생성되기 이전의 S_d 내의 S_i 와 F_i 가 서로 동일한 형태를 가지면서 Γ 에 포함되기 때문에, 본 논문에서는 이후 용어상의 혼동이 없는 한 Γ 의 원소를 시멘틱 세그먼트라고 부른다. 일반적으로, 이동 클라이언트의 이동성이나 질의 유형과 같은 특성에 따라, 특정 LDD 질의의 결과 데이터들은 다른 LDD 질의의 결과들보다 빈번하게 사용될 수 있다. 이런 결과들을 일정 기간 동안 이동 클라이언트의 시멘틱 캐시에 유지하는 것은 캐시의 효율성을 높이는 데 도움이 된다. 또한, 질의 Q 에 대해 이동 클라이언트가 Γ 로부터 그 결과를 얻을 수 있다면, 서버와의 통신 비용 없이 빠른 응답을 사용자에게 제공할 수 있으므로, 효율적인 Γ 의 구성은 매우 중요하다. 그러므로, 본 논문에서는 Γ 의 각 시멘틱 세그먼트에 서술 매개변수의 집합, $\kappa = \{t, f, l\}$ 을 추가한다. 여기서 t, f, l 은 접근 시간, 빈도, 지역성을 각각 나타낸다.

Γ 내의 시멘틱 세그먼트 $S'_i = \langle S'_r, S'_a, S'_p, S'_l, S'_c \rangle$ 와 질의 Q 는 서로 밀접한 관련이 있다고 하자. 즉, 시멘틱 세그먼트 $S'_i = \langle S'_r, S'_a, S'_p, S'_l, S'_c \rangle$ 과 질의 $Q = \langle Q_r, Q_a, Q_p, Q_l, Q_c \rangle$ 가 $S'_r = Q_r, S'_a = Q_a$, 그리고 $S'_p = Q_p$ 의 관계라면, S'_i 과 Q 는 관계와 속성이 일치하고, Q 의 질의 영역과 S'_i 의 시멘틱 영역도 완전히 일치하여 전체적으로 겹침을 의미한다. 우리는 이러한 관계를 Equivalent한 관계라고 부른다. Q 의 질의 결과는 질의 정돈(query trimming) 처리에 의해 일치 질의(probe query)와 나머지 질의(remainder query)라고 불리는 두 가지 종류로 나뉘어진다. 일치 질의는 S'_i 에 의해서 만족되는 Q 의 질의 결과 부분을 나타내며, 나머지 질의는 S'_i 에서 찾을 수 없는 질의 결과 부분을 나타낸다. Q 와 S'_i 가 Equivalent한 관계일 때에는 Q 는 S'_i 으로부터 질의 결과를 모두 얻을 수 있기 때문에 질의 결과는 일치 질의이다. Q 가 S'_i 으로부터 일부의 질의 결과를 얻

는 경우에는 Related한 관계가 있다고 하며, Q 의 질의 결과는 일치 질의와 나머지 질의로 나누어진다. Q 와 S'_i 가 Related 관계일 경우에는 나머지 질의 처리를 위해 추가적인 통신 비용이 발생한다.

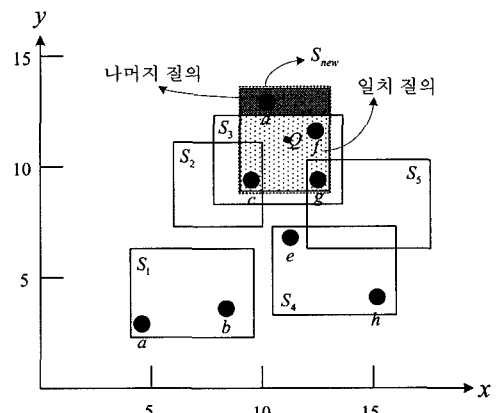
4. 질의 처리와 캐시 교체 전략

이 장에서는 시멘틱 프리패칭 기법이 적용된 환경에서, 이동 클라이언트의 영역 질의(range query)와 최근접 질의(nearest neighbor query)가 처리되는 세부적인 질의 처리 과정과 캐시 교체 전략을 설명한다.

4.1 시멘틱 영역 질의 처리

영역 질의(range query)에 대한 세부적인 질의 처리 과정을 살펴보자. 그림 1은 영역 질의의 예를 나타낸다. 이 그림에서 각 사각형 영역은 서술 테이블 Γ 의 시멘틱 세그먼트 영역을 표현하며, 각 시멘틱 세그먼트내의 a부터 h까지의 점들은 LDD 객체를 표현한다. 질의 Q 의 질의 영역은 어두운 사각형 영역이며, 점으로 구성된 사각형은 일치 질의, 빛금으로 구성된 사각형은 나머지 질의를 각각 의미한다.

질의 처리 결과는 서술 테이블 Γ 의 시멘틱 세그먼트 영역들과 질의 영역의 겹침 정도에 의존적이다. 겹침 정



서술 테이블 $\Gamma = \{S_1, S_2, S_3, S_4, S_5, S_{new}\}$

그림 1 영역 질의 처리의 예

도에 따라, 질의 Q 의 처리는 세가지 경우로 나누어질 수 있다. 첫 번째는 질의 결과 모두를 이동 클라이언트의 Γ 로부터 얻을 수 있는 경우이다. 즉, 질의 영역과 시멘틱 세그먼트 영역이 완전히 겹치고 질의 결과가 일치 질의인 경우이다. 두 번째는 질의 결과의 일부만을 Γ 로부터 얻을 수 있는 경우이다. 즉, 그림 1과 같이 질의 Q 의 질의 영역이 시멘틱 세그먼트 S_2, S_3 , 그리고 S_5 의 영역들과 부분적으로 겹치는 경우이다. 설명의 편의를 위해서, S_3 와 겹치는 질의 영역에 대해서만 생각한다. 이 경우에 질의 Q 의 질의 영역은 Γ 로부터 질의 결과를 얻을 수 있는 일치 질의 영역과 서버에게 나머지 질의를 하여 추가적인 결과를 얻을 수 있는 나머지 질의 영역으로 나누어진다. 완전한 질의 결과는 이 두 영역으로부터 얻을 수 있다. 세 번째는 질의 결과를 Γ 로부터 전혀 얻을 수 없는 경우이다. 즉, 질의 영역과 시멘틱 세그먼트 영역이 전혀 겹치지 않는 경우이다. 이 경우에는 질의 Q 가 서버에 전달되어야만 질의 결과들을 얻을 수 있다. 그림 1에서, 이동 클라이언트는 객체 c, f, g 의 정보를 시멘틱 캐시로부터 얻을 수 있지만 객체 d 의 정보는 나머지 질의를 수행하여야만 얻을 수 있다. 질의 Q 에 대한 영역 질의 결과는 객체 c, d, f, g 의 시멘틱 정보가 되며, 질의 결과는 새로운 시멘틱 세그먼트 S_{new} 로 서술 테이블 Γ 에 추가된다. 이러한 방법을 사용하여 유사 질의 발생시 이동 클라이언트의 시멘틱 캐시 유용성을 높일 수 있다.

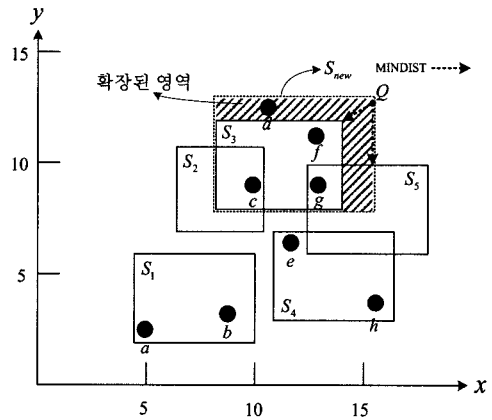
앞서 언급한 것처럼 Γ 내의 각 시멘틱 세그먼트들은 서술 매개변수 집합 k 를 포함한다. 시멘틱 세그먼트 영역과 질의 영역의 겹침 정도에 따라, k 의 각 매개변수는 다음과 같이 갱신된다. 만일 질의 Q 가 Γ 내의 시멘틱 세그먼트와 겹칠 때, 겹친 시멘틱 세그먼트에 대한 접근 시간은 질의한 시간으로 교체된다. 그리고, 겹친 시멘틱 세그먼트의 빈도 값은 1씩 증가한다. 마지막으로, 나머지 질의 영역 크기에 대한 일치 질의 영역 크기의 비율을 나타내는 지역성은 식 (1)과 같이 갱신된다.

$$l = l' + \frac{M}{R}, \quad (1)$$

여기서 l' 은 지금까지 유지된 지역성의 값, M 은 질의 영역과 시멘틱 세그먼트 영역간에 겹친 일치 질의 영역의 크기, R 은 질의 영역과 시멘틱 세그먼트 영역간에 겹치지 않은 나머지 질의 영역의 크기를 의미한다.

4.2 시멘틱 최 근접 질의 처리

이 절에서는 최 근접 질의(NNQ)에 대한 세부적인 질의 처리 절차를 설명한다. 최 근접 질의는 영역 질의와 함께 공간 데이터 처리를 위해 주로 사용되어 왔다 [6,25]. 그림 2는 최 근접 질의 처리의 예를 보여준다. 이 그림에서, 사각형 영역은 Γ 의 시멘틱 세그먼트들의



서술 테이블 $\Gamma = \{S_1, S_2, S_3, S_4, S_5, S_{new}\}$

그림 2 최 근접 질의 처리의 예

영역을 나타내며, a부터 h까지 점 정보로 표현된 LDD 객체들을 포함한다. 영역 질의와 달리, 최 근접 질의 Q 의 영역은 작은 점으로 표현되며 질의를 수행한 위치를 의미한다. Q 에서 시작된 두 점선은 질의와 시멘틱 세그먼트 영역간의 최소 거리를 나타낸다.

최 근접 질의 결과는 이동 클라이언트의 질의 지점이 시멘틱 세그먼트들 내에 포함되는지의 여부에 따라 달라진다. 만일 질의 지점이 시멘틱 세그먼트 내에 존재하면, 질의 결과는 질의 지점에서 가장 가까운 질의 조건을 만족하는 객체로 결정된다. 그림 2와 같이 질의 지점이 시멘틱 세그먼트 외부에 존재하면, MINDIST 연산 [26]을 사용하여 먼저 질의 지점에서 가장 가까운 시멘틱 세그먼트인 S_3 가 선택된다. 가장 가까운 시멘틱 세그먼트가 선택된 후에, 이동 클라이언트는 질의 지점과 선택된 시멘틱 세그먼트의 영역을 포함하는 최소 사각형 영역을 새로운 시멘틱 세그먼트 S_{new} 로 생성한다. 이 때, S_{new} 의 확장 부분(그림 2의 빗금 친 부분)은 영역 질의 처리의 나머지 질의처럼 서버에 요청하여 얻어진다. 이동 클라이언트는 S_{new} 에서 네 개의 객체 c, d, f, g 를 얻게 되며 최종적으로 S_{new} 로부터 질의 지점에서 가장 가까운 객체 f 를 얻게 된다. 이 S_{new} 는 영역 질의에서처럼 서술 테이블 Γ 에 추가된다.

최 근접 질의도 영역 질의처럼 각 시멘틱 세그먼트들은 서술 매개변수 집합 k 를 가진다. 접근 시간과 빈도에 대한 갱신 방법은 영역 질의에서와 동일하다. 그러나, 최 근접 질의에서 지역성에 대한 갱신 방법은 영역 질의와는 다르다. 최 근접 질의에서 M 은 질의 지점에서 MINDIST 연산에 의해 선택된 가장 가까운 시멘틱 세그먼트 영역, R 은 새로운 시멘틱 세그먼트 S_{new} 에서 M 을 제외한 영역을 각각 표현한다.

4.3 캐시 교체 전략

이전에 방문했던 SPA로 진입할 때마다 이동 클라이언트는 서버로부터 전송 받은 프리페칭 서술 F_d 를 이용하여 서술 테이블 Γ 를 갱신하기 때문에 이동 클라이언트의 현재 상황에 적합한 서술 테이블의 구성을 위해서는 효율적인 캐시 교체 전략이 필요하다. 본 논문에서는 세 가지의 캐시 교체 전략을 제안한다.

- 1) Semantic least recently used(S-LRU): 기존 LRU 전략의 변형으로 서술 테이블 Γ 에는 가장 최근에 참조된 시멘틱 세그먼트들만 유지한다. Γ 내에서 가장 오래된 접근 시간을 가지는 Γ 의 시멘틱 세그먼트들이 F_d 로 교체된다.
- 2) Semantic least frequently used(S-LFU): 기존 LFU 전략의 변형으로 Γ 에는 빈번히 참조된 시멘틱 세그먼트들만 유지한다. Γ 에서 가장 참조되지 않은 시멘틱 세그먼트들이 F_d 로 교체된다. 그러나, 시멘틱 세그먼트들이 F_d 보다 높은 빈도를 가지면 교체하지 않는다.
- 3) Preference priority replacement(PPR): SPA에서 공간 지역성과 빈도수를 고려한 캐시 교체 전략으로 Γ 의 모든 시멘틱 세그먼트들은 식 (2)와 같은 선호도를 유지한다.

$$Pref(S_i) = \frac{\omega \times l + (1 - \omega) \times f}{Q_{num}}, \quad (2)$$

f 와 l 은 빈도와 지역성에 대한 값을 각각 나타낸다. 시멘틱 세그먼트 S_i 에 대하여, Q_{num} 은 질의 수를, $\omega(0 \leq \omega \leq 1)$ 은 지역성에 대한 가중치를 표현한다. 빈도 f 와 지역성 l 만으로도 클라이언트의 시간적 공간적 특성을 각각 고려할 수 있기 때문에, 접근 시간 t 는 PPR에서 사용되지 않는다.

4.4 알고리즘 요약

시멘틱 프리페칭 기법의 알고리즘은 크게 서버와 이동 클라이언트에서 동작하는 알고리즘으로 나눌 수 있다. 이동 클라이언트의 알고리즘은 질의 종류에 따라 영역 질의 처리와 최 근접 질의 처리 알고리즘으로 나뉘어진다.

그림 3은 이동 클라이언트에게 보낼 프리페칭 서술 F_d 를 결정할 수 있는 서버 알고리즘을 보여준다. 이동 클라이언트가 SPA로 진입할 때마다 서버는 먼저 이동 클라이언트의 식별자를 검사한 후 로그 파일 L_f 를 이용하여 프리페칭 세그먼트들의 합 $\sum_{i \in I} F_i$ 를 F_d 로 결정한

```

프리페칭 서술 결정 알고리즘

사용 기호
Fi : 프리페칭 세그먼트 Fd : 프리페칭 서술
Lf : 로그 파일 logm : 로그 메시지

입력: 이동 클라이언트의 위치와 식별자(identifier) 출력: Fd

Procedure:
if a mobile client is in a semantic prefetching area then
  Check user's identifier;
  if Lf ← NULL then
    Fi ← NULL;
  else
    for each prefetching segment Fi in log file Lf do
      Fd ← ∪i ∈ I Fi, where Fi ∈ Lf;
    end for
  else
    if the server receives additional log message logm then
      Update Lf;
    end if
  end if
end if
    
```

그림 3 서버 알고리즘

다. 이동 클라이언트가 SPA를 벗어날 때 서버로 보낸 로그 메시지 log_m 을 이용하여 서버는 로그 파일 L_f 를 갱신한다.

그림 4는 이동 클라이언트의 시멘틱 영역 질의 처리 알고리즘을 나타낸다. 영역 질의 Q_i 는 서술 테이블의 시멘틱 세그먼트들과 비교되어 세가지 경우에 따라 서로 다른 처리를 진행한다. 질의 영역과 시멘틱 세그먼트 영역의 겹침 정도에 따라 영역 질의 Q_i 의 결과는 일치 질의 Q_{probe} 와 나머지 질의 $Q_{remainder}$ 로 나누어진다. Q_{probe} 영역은 R_{buffer} 에 저장되며, 서버로 요청된 $Q_{remainder}$ 의 결과도 계속해서 R_{buffer} 에 저장된다. 이것은 Q_i 의 결과로 서술 테이블에 S_{new} 로 저장되며 S_{new} 내에서 질의 결과 R_i 를 얻을 수 있다.

그림 5는 시멘틱 최 근접 질의를 처리하기 위한 이동 클라이언트에서 사용되는 알고리즘을 보여준다. 최 근접 질의 Q_i 가 요청되면 이동 클라이언트는 먼저 Γ 의 시멘틱 세그먼트들 영역에 질의 지점이 포함되는지를 확인한다. 특정 시멘틱 세그먼트에 포함된다면 시멘틱 세그먼트 영역내의 객체 중 질의 조건을 만족하는 최단 거리의 최 근접 객체를 찾는다. 그렇지 않다면, 질의 지점과 Γ 의 시멘틱 세그먼트들 중 가장 가까운 시멘틱 세그먼트들을 찾아 우선순위 큐에 저장한다. 우선순위 큐의 원소를 하나씩 꺼내어 질의 지점을 포함하는 확장된 영역의 시멘틱 세그먼트를 생성한 후 생성된 시멘틱 세그먼트 영역내의 객체 중 질의 조건을 만족하는 최단 거리의 최 근접 객체를 찾는다.

세만틱 영역 질의 처리 알고리즘

사용 기호
 Q_i : 영역 질의 R_i : Q_i 의 질의 결과
 S_i : 세만틱 세그먼트
 Q_{probe} : Γ 의 세만틱 정보와 일치하는 질의 결과 영역에 대한 세만틱 정보
 $Q_{remainder}$: Γ 에서 얻을 수 없는 질의 결과 영역에 대한 세만틱 정보
 R_{buffer} : 일시적인 결과 저장용 버퍼
 S_{new} : 새롭게 확장되어 생성된 세만틱 세그먼트

입력: Q_i 출력: R_i

Procedure:
 $Q_i \leftarrow EMPTY$;
 if a mobile client issues Q_i then
 for each S_i overlapped with Q_i do
 if Q_i totally retrieves the result in S_i then
 $Q_{probe} \leftarrow Q_i, R_i \leftarrow Q_{probe}$;
 else if Q_i partially retrieves the result in S_i then
 Q_i is trimmed as Q_{probe} and $Q_{remainder}$;
 while $Q_{remainder} \neq NULL$ do
 $R_{buffer} \leftarrow Q_{probe}$;
 Send $Q_{remainder}$ to the server ;
 Save the result of $Q_{remainder}$ to R_{buffer} ;
 end while
 $S_{new} \leftarrow R_{buffer}, R_i \leftarrow S_{new}$;
 add S_{new} to Γ ;
 end for
 if Q_i does not retrieves the result in any S_i then
 Send Q_i to the server ;
 Save the result of Q_i to S_{new} ;
 $R_i \leftarrow S_{new}$;
 add S_{new} to Γ ;
 end if

그림 4 시멘틱 영역 질의 처리

5. 성능 평가

본 장에서는 이동 컴퓨팅 환경하에서 다양한 실험을 통해 시멘틱 프리페칭 기법에 대한 성능을 기존의 시멘틱 캐싱 기법의 성능과 비교한다[14,19].

5.1 실험 모델

본 논문에서 제안하는 시스템은 그림 6과 같이 질의 처리, 캐시 관리를 담당하는 이동 클라이언트 부분과 시멘틱 프리페칭 관리, 프리페칭 서술의 전송, 기타 나머지 작업을 담당하는 서버 부분으로 나뉜다. 이동 클라이언트는 영역 질의(RQ: Range Query)와 최 근접 질의(NNQ: Nearest Neighbor Query)를 생성하는 질의 생성기, 질의들을 분석하는 질의 분석기, 질의가 캐시에서 일치하는지를 확인하여 일치 질의를 결정하고, 서버에 나머지 질의를 재전송하는 캐시 관리자, 캐시 교체 전략에 따라 서술 테이블을 갱신하는 캐시 저장소의 네 가지 모듈로 구성된다. 서버 부분 중 프리페칭 서버 부분은 이동 클라이언트로부터 받은 로그 파일을 저장하는 로그

세만틱 최 근접 질의 처리 알고리즘

사용 기호
 Q_i : 최 근접 질의 R_i : Q_i 의 질의 결과
 $Count(Y)$: Y의 원소 개수 반환 함수
 $Loc(Q_i)$: 질의한 이동 클라이언트의 좌표를 반환하는 함수
 $GetArea(S_i)$: 세만틱 세그먼트 S_i 의 영역을 반환하는 함수
 $GetShortestElement(X,Y)$: X와 가장 가까운 Y의 원소이면서 Q_i 의 질의 조건을 만족하는 원소를 반환하는 함수
 $priority_queue[]$: Γ 의 세만틱 세그먼트 원소 중 $Loc(Q_i)$ 와 가장 가까운 세만틱 세그먼트들을 거리 순서대로 저장하는 자료구조
 $CreateSemanticSeg(X,Y)$: X와 Y의 영역을 포함하는 새로운 세만틱 세그먼트를 생성한 후 생성된 세만틱 세그먼트를 반환 함수
 입력: Q_i 출력: R_i

Procedure:
 $Criterion \leftarrow true$;
 if a mobile client issues Q_i then
 if $\exists S_i \in \Gamma$ such that $(Loc(Q_i) \in GetArea(S_i))$ then
 $R_i \leftarrow GetShortestElement(Loc(Q_i), S_i)$;
 end if
 if $(\forall S_i \in \Gamma \text{ such that } (Loc(Q_i) \in GetArea(S_i)) \vee (R_i = NULL))$
 then
 for j=1 to $Count(\Gamma)$ do
 $priority_queue[j] \leftarrow GetShortestElement(Loc(Q_i), \Gamma)$;
 end for
 while $Criterion \wedge (1 \leq j \leq Count(priority_queue[]))$ do
 $S_k \leftarrow CreateSemanticSeg(Loc(Q_i), priority_queue[j])$;
 $R_i \leftarrow GetShortestElement(Loc(Q_i), S_k)$;
 if $R_i \neq NULL$ then
 $Criterion \leftarrow false$;
 add S_k to Γ ;
 end if
 end while
 end if
 end if

그림 5 시멘틱 최 근접 질의 처리

저장소, 로그 파일을 분석하는 로그 분석기, 이동 클라이언트에게 전달될 프리페칭 서술을 결정하는 프리페칭 관리자의 세가지 모듈로 구성되며 로컬 서버는 프리페칭 서버가 수행하지 않는 나머지 작업들을 처리한다.

본 논문은 그림 6과 같이 프리페칭 세그먼트들(프리페칭 서술)의 효율적인 전송을 위해 시멘틱 프리페칭 채널(SPC: Semantic Prefetching Channel)이라는 고속의 전용채널이 사용된다고 가정한다. 기지국 채널(BSC: Base Station Channel)은 로그 정보 전송과 나머지 질의에 대한 요구와 응답을 위해 사용된다. 또한, 3.1절의 동작 과정에서 언급한대로 이동 클라이언트는 어떤 SPA라도 방문할 수 있으며, 처음 방문하는 SPA에서도 알고리즘은 수행된다. 한 번 이상 방문한 SPA의 경우, 로그 파일의 유무에 따른 초기 데이터 생성과정을 통해 제안 기법의 적용이 가능하다. 프리페칭 서버에는 이동 클라이언트의 로그 파일이 존재한다고 가정한다. 본 논문에서는 무선 통신 상에서 발생할 수 있는 메시지 손실은 고려하지 않으며, 서버는 이동 클라이언트의 정보를 유지할 수 있는 충분한 용량과 처리 능력을 보유하

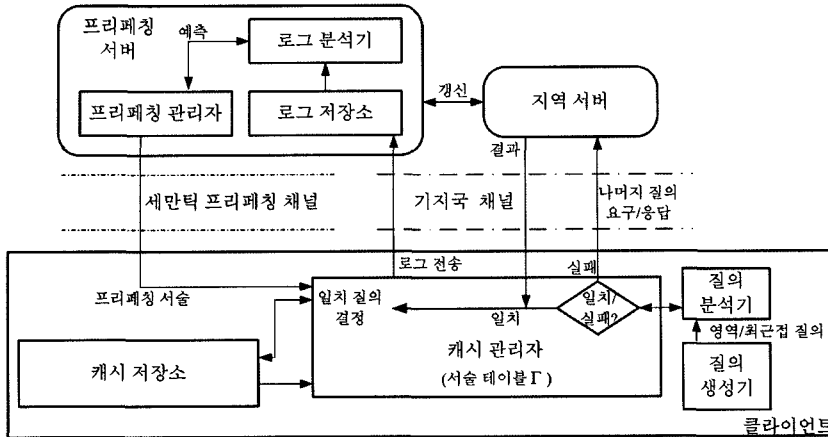


그림 6 실험 모델

고 있다고 가정한다.

5.2 시뮬레이션 환경

시뮬레이션의 수행을 위해 표 2의 매개변수들을 사용하였으며, 10000개의 물리적인 셀 기반 네트워크 영역과 논리적인 25개의 SPA 네트워크 영역을 적용하였다. 논리적인 각 SPA는 물리적인 20×20의 400개의 셀로 구성되며, 이 물리적인 셀 영역과 논리적인 SPA 영역을 함께 사용하여 중복 네트워크(Overlay network)로 설정하였다. 이동 클라이언트는 이동한 영역에 해당하는 셀 기반 네트워크와 SPA 네트워크를 동시에 사용하여 제안 기법을 수행한다.

쏠림 계수 θ 를 가지는 지프 분포[27]를 이용하여 LDD를 흩어진 경우(scattered)와 쏠린 경우(skewed)로 배치한다.

흩어진 경우에는, $\theta = 0$ 으로 400개의 LDD를 SPA에서 무작위로 생성한다. 쏠린 경우에는, $\theta = 0.8$ 로 400개의 LDD를 생성시킨다. 이동 클라이언트는 SPA에서 주유소, 음식점, 호텔, 도서관, 주점과 같은 다섯 가지 종

류의 질의 중 하나를 TOTAL_QUERY_NUM 개 만큼 발생시킨다. 질의는 유사한(similar) 질의와 다양한(various) 질의의 두 가지 유형으로 나뉜다. 유사한 질의들은 0.8의 쏠림 계수를 사용하는 지프 분포를 이용하여 생성하고, 다양한 질의들은 0의 쏠림 계수 값을 적용하여 생성한다.

이동 클라이언트들의 이동성은 주어진 시간 동안에 방문한 SPA의 수에 따라 낮은(low) 이동성을 가지는 경우와 높은(high) 이동성을 가지는 경우로 분류하였다. 표 3은 이동성과 질의 유형의 조합으로 구성된 네 종류의 이동 클라이언트 유형들을 보여준다. 낮은 이동성으로 인해 적은 수의 SPA만 이동하고 유사한 종류의 질의를 하는 유형 A의 이동 클라이언트는 서버로부터 받은 프리페칭 서술을 이용하여 캐시에서 좋은 결과를 얻을 수 있다. 낮은 이동성과 다양한 종류의 질의를 하는 유형 B의 이동 클라이언트는 제한된 캐시 용량으로 인해 다양한 패턴의 질의에 충분히 응답할 수 있는 결과를 가지지 못하므로 캐시 실패가 자주 일어날 수 있다.

표 2 매개변수 설정

매개변수	설명	설정값
NUM_CELL	사각 셀의 수	100 × 100
NUM_SPA	SPA의 수	25
NUM_LDD	LDD의 수	400
TYPE_LDD	LDD의 유형 수	5
TOTAL_QUERY_NUM	질의 수	100~2000
QUERY_NUM_PER_SPA	SPA당 질의 수	10
CACHE_SIZE	이동 클라이언트의 캐시 크기	512kbytes
DATA_A_DOWN_SPC	다운링크 SPC 채널을 통해 전송되는 시멘틱 프리페칭 데이터의 크기: $D(a)_{down}$ [SPC]	100kbytes
DATA_B_UP_BSC	업링크 BSC 채널을 통해 전송되는 질의 데이터의 크기: $D(b)_{up}$ [BSC]	3.75kbytes
DATA_B_DOWN_BSC	다운링크 BSC 채널을 통해 전송되는 질의 결과 데이터의 크기: $D(b)_{down}$ [BSC]	3.75kbytes
DATA_C_UP_BSC	업링크 BSC 채널을 통해 전송되는 로그 파일의 크기: $D(c)_{up}$ [BSC]	10kbytes
LOCALITY_WEIGHT	지역성 가중치	0.7

표 3 이동 클라이언트 유형 분류

항목 \ 유형	A	B	C	D
이동성	낮음	낮음	높음	높음
질의 패턴	유사	다양	유사	다양

높은 이동성으로 주어진 시간 동안 많은 여러 지역을 방문할 수 있고 유사한 질의 패턴을 가지는 유형 C의 이동 클라이언트는 유사한 종류의 질의를 하는 특징을 가지지만, 높은 이동성 때문에 빈번한 캐시 교체가 이루어져 캐시 실패가 발생할 수 있다. 마지막으로, 많은 SPA를 이동하면서 다양한 종류의 질의를 수행하는 유형 D의 이동 클라이언트는 다른 경우에 비해 좋은 성능을 가지지 못한다.

5.3 성능 척도

제안 기법(SP)과 기존 기법(SC)의 성능 분석을 위한 기준으로 캐시 적중률의 변형인 질의 절감율(QSR: Query saving ratio)과 네트워크 부하를 사용한다.

5.3.1 질의 절감율

일반적으로, 캐시 적중률(CHR: Cache hit ratio)은 다음과 같이 계산된다.

$$CHR = \sum_i h(q_i) \times \frac{1}{Q_{num}}, \quad h(q_i) = \begin{cases} 1 & \text{if cache hit} \\ 0 & \text{if cache miss} \end{cases} \quad (3)$$

q_i 는 질의를, $h(q_i)$ 는 질의에 대하여 캐시에서 적중인지 실패인지를 나타내고, Q_{num} 은 전체 질의 수를 표현한다. 그러나, 식 (3)은 제안 기법에서 정확한 측정요소로 사용될 수 없다. 왜냐하면, 부분적인 캐시 적중이 일어난 경우에는 나머지 질의가 발생하여 추가적인 비용이 발생함에도 불구하고 캐시가 적중한 것으로 판단하기 때문이다. 따라서, 제안 기법의 정확한 성능 측정을 위해 캐시 적중률뿐만 아니라 나머지 질의에 대한 양도 고려한 질의 절감율(QSR: Query Saving Ratio)을 도입한다. QSR은 다음의 식 (4)와 같이 계산된다.

$$QSR = \left(\sum_i h(q_i) \cdot \frac{M(q_i)}{(M(q_i) + R(q_i))} \right) \times \frac{1}{Q_{num}},$$

$$h(q_i) = \begin{cases} 1 & \text{if cache hit} \\ 0 & \text{if cache miss} \end{cases} \quad (4)$$

$M(q_i)$ 와 $R(q_i)$ 는 새로운 질의 q_i 의 사각형 영역이 로컬 캐시의 세그먼트가 저장하고 있는 사각형 영역과 겹칠 때, 캐시에서 일치되는 부분과 일치되지 않는 부분의 공간 영역의 범위를 각각 양적으로 표현한 것이다. 즉, 새로운 질의 q_i 에 대하여 로컬 캐시에서 결정되는 일치 질의와 추가적인 네트워크 사용으로 해결하는 나머지 질의의 상대적인 비율을 질의의 양으로 표현하였다.

5.3.2 네트워크 트래픽

본 절에서는 이동 클라이언트가 어떤 특정 SPA로 들

어가고 나가는 동안 일어나는 SP기법과 SC기법에서의 네트워크 부하를 수식을 통해 분석한다. 이동 클라이언트가 SPA로 들어가면 SPC를 사용하여 프리페칭 서술 데이터를 받고 SPA를 떠나면 BSC를 사용하여 로그 파일을 전송한다. SPA 내에서 발생한 질의의 결과 전체를 캐시에서 찾을 수 있다면, 통신 비용이 발생하지 않는다. 그러나, 질의 결과의 일부분만 캐시에 있는 Γ 에서 찾게 되면, 부분적인 실패에 대한 질의 결과를 얻기 위해 네트워크 부하가 일어난다. Γ 에서 전부 실패가 발생하면, BSC를 통해 서버에 질의를 한다. SP 기법에서 평균 네트워크 트래픽 양은 다음과 같이 계산된다.

$$SP_{traffic} = D(a)_{down}[SPC] + \hat{Q}_{num} \cdot (1 - QSR) \cdot (D(b)_{up}[BSC] + D(b)_{down}[BSC]) + D(c)_{up}[BSC] \quad (5)$$

$D(a)_{down}[SPC]$ 는 다운링크 SPC를 경유하여 전송되는 시멘틱 프리페칭 데이터의 양이고, $D(b)_{up}[BSC]$ 와 $D(b)_{down}[BSC]$ 는 업링크 BSC를 통해 서버에 전송되는 질의 데이터의 양과 다운링크 BSC를 통해 전송되는 질의 결과 데이터의 양을 각각 나타낸다. $D(c)_{up}[BSC]$ 는 업링크 BSC를 통해 서버에 전송되는 로그 파일의 양을 표현한다. \hat{Q}_{num} 은 하나의 SPA내에서 발생하는 평균 질의 수이다.

SC 기법은 이동 클라이언트가 SPA를 들어가고 나가는 동안에 어떤 네트워크 트래픽도 발생하지 않는다. 네트워크 트래픽은 이동 클라이언트의 캐시에 질의 결과가 없을 때 발생한다. SC 기법에서의 평균 네트워크 트래픽 양은 다음과 같이 계산된다.

$$SC_{traffic} = \hat{Q}_{num} \cdot (1 - QSR) \cdot (D(b)_{up}[BSC] + D(b)_{down}[BSC]) \quad (6)$$

$SP_{traffic}$ 과 $SC_{traffic}$ 의 수식을 비교해 보면, SP 기법은 SC 기법보다 더 많은 네트워크 트래픽을 발생시킬 수 있음을 알 수 있다. 그러나, SP 기법은 SC 기법보다 더 높은 QSR 값을 가질 수 있다. 이 경우에는 SC 기법보다 더 적은 네트워크 트래픽을 초래할 수 있다. 다시 말해, SP 기법은 SPA의 출입 시 추가적인 네트워크 트래픽을 발생시키지만, SPA의 진입 시 최신의 캐시를 구성할 수 있기 때문에 캐시 실패에 의해 발생하는 추가적인 네트워크 트래픽을 줄일 수 있다. 반면에, SC 기법은 이전 SPA의 내용을 캐시에 계속 유지하므로, 새로운 SPA에서는 캐시 내용이 적합하지 않아 빈번한 캐시 실패를 초래할 수 있다.

5.4 시뮬레이션

먼저, 이동 클라이언트의 유형과 질의 수를 달리하면서 영역 질의와 최 근접 질의에 대한 SP와 SC 기법의 QSR값을 구한다. 그 다음으로 LRU, FAR[23], S-LRU, S-LFU, PPR과 같은 다양한 캐시 교체 전략들에 대한 QSR값을 구한다. 마지막으로, 이동 클라이언트의 유형

에 따라 SP와 SC 기법에서의 네트워크 트래픽 양을 측정하고 성능을 분석한다.

5.4.1 이동 클라이언트 유형에 대한 QSR 분석

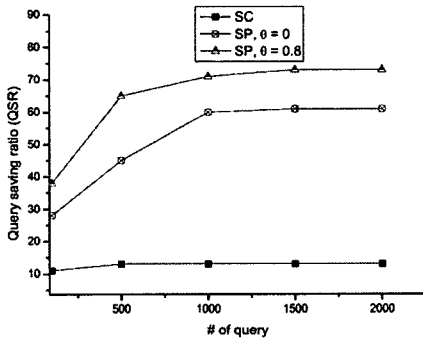
이 절에서는 100부터 2000개까지의 영역 질의와 최근접 질의를 사용하여 네 가지 이동 클라이언트 유형에 대한 SP와 SC 기법들의 QSR 을 분석한다. 또한, LDD 분포에 대한 QSR의 변화를 알아보기 위해 흠어진 LDD($\theta = 0$)와 쏠린 LDD($\theta = 0.8$)의 두 LDD 집합을 사용하며, 캐시 교체 전략으로는 PPR을 사용한다.

영역 질의에 대한 QSR의 성능을 보여주는 그림 7로부터 모든 이동 클라이언트 유형에 대하여 SP 기법이 SC 기법보다 30-60%정도 더 높은 QSR 을 가짐을 알 수 있다. 이것은 이동 클라이언트 캐시를 최신으로 유지시키는 시멘틱 프리페칭 서술 때문이다. SP 기법에서 흠어진 LDD와 쏠린 LDD를 비교할 때, 쏠린 LDD가 흠어진 LDD보다 약 8-15%정도 높은 QSR을 가짐을 알 수 있다. 이러한 결과는 복잡한 공간 특성을 가지는 흠어진 LDD보다 효율적으로 캐시를 만들 수 있는 쏠린 LDD에 대하여 SP 기법이 더 효율적임을 보여준다. 그림 7에서, 낮은 이동성과 유사한 질의 패턴을 가지는 유형 A인 경우가 다른 유형의 경우보다 더 높은 QSR을 가짐을 알 수 있다. 낮은 이동성의 이동 클라이언트들은

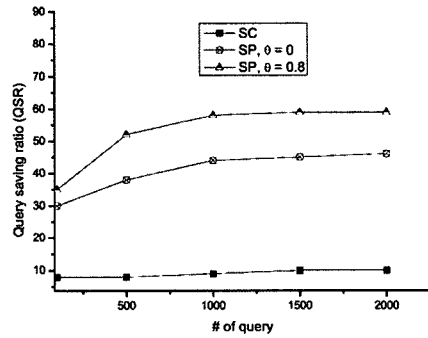
이동 량이 많지 않기 때문에 서술 테이블에 저장된 시멘틱 세그먼트의 지역성 정확도가 향상되어 캐시의 서술 테이블로부터 질의 결과를 얻을 수 있다. 또한, 유사한 질의 패턴은 서술 테이블에 저장된 예전 질의 결과로부터 현재 질의의 결과를 얻을 확률을 높여 높은 QSR을 가지게 한다.

그림 8은 최 근접 질의에 대한 QSR의 성능을 보여준다. 결과들은 그림 7과 유사하지만, 영역 질의와 최 근접 질의의 처리 방법 차이로 인해 QSR 값에서 약간의 차이가 있다.

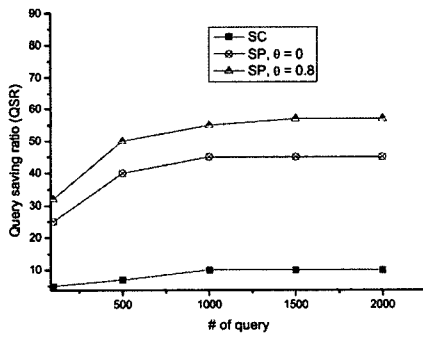
그림 7과 8에서 유형 B와 C의 경우에는 유사한 QSR 을 보인다 하지만, 그림 7과 8에서 유형 A와 D의 경우는 특별한 의미를 부여할 수 있는 분석이 가능하다. 그림 7(a)에서, SP 기법에 대한 유형 A의 QSR은 1000개의 질의 후에 60-70%이지만, 그림 8(a)에서는 65-77%의 QSR을 보인다. 즉, 유형 A에 대해 최 근접 질의가 영역 질의보다 더 높은 QSR을 가진다. 한편, 그림 7(d)는 SP 기법에 대한 유형 D의 QSR이 1000개의 질의 후에 38-45%이고 그림 8(d)는 약 25-35%의 QSR을 보여준다. 유형 A와는 대조적으로, 유형 D에 대해 영역 질의가 최 근접 질의보다 더 높은 QSR을 가진다. 이러한 결과는 이동 클라이언트 유형과 적용되는 질의 방법



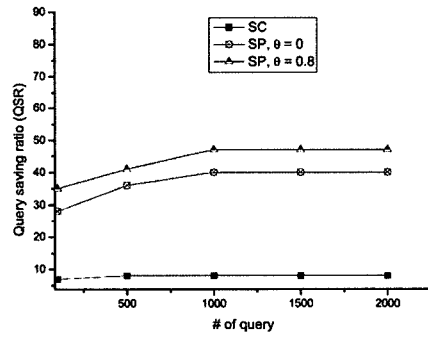
(a) 유형 A



(b) 유형 B



(c) 유형 C



(d) 유형 D

그림 7 영역 질의인 경우의 질의 절감율

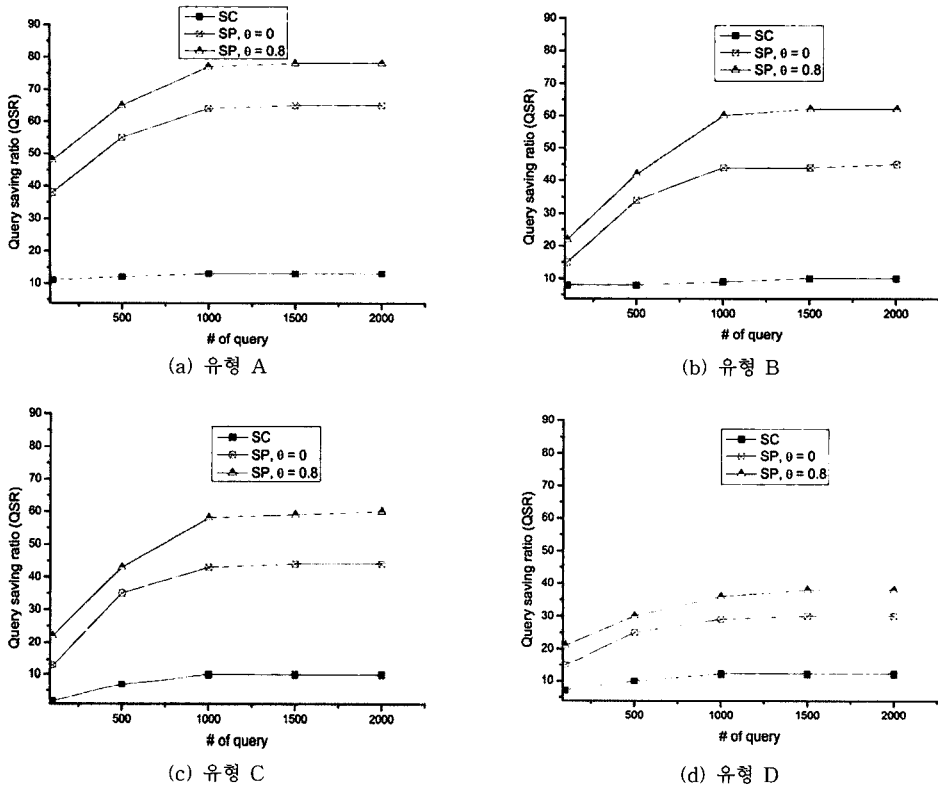


그림 8 최 근접 질의인 경우의 질의 절감율

은 서로 상당히 밀접한 관련이 있음을 암시한다.

유형 A의 이동 클라이언트는 낮은 이동성과 유사한 질의 특성으로 질의 지점과 시멘틱 세그먼트의 겹침이 자주 발생할 수 있다. 그림 9는 유형 A에서 영역 질의 또는 최 근접 질의가 발생하는 경우의 예를 보여준다. 이 그림에서 화살표는 네 개의 질의 (Q_1, Q_2, Q_3, Q_4)가 발생한 경로를 나타낸다. 그림 9에서 유형 A의 이동 클라이언트는 낮은 이동성과 질의 유사성으로 영역 질의와 최 근접 질의에서 질의 영역 또는 질의 지점이 시멘틱 세그먼트 영역들과 겹치고 있다. 최 근접 질의의 경우는, 질의 지점이 시멘틱 세그먼트 내에 있으면 가장 가까운 객체를 찾고 질의 처리는 캐시에서 끝나버린다. 하지만, 영역 질의에 대해서는 질의 지점이 시멘틱 세그먼트 내에 있더라도, 나머지 질의가 발생하면 질의 처리는 바로 끝나지 않고, 나머지 질의를 다시 서버에 전달해야 한다. 따라서, 유형 A의 경우에는 나머지 질의에 대한 추가적인 요구 때문에 그림 7(a)와 8(a)에서처럼 영역 질의의 QSR이 최 근접 질의보다 5-7%가량 낮게 나온다. 반면에, 높은 이동성과 다양한 질의를 가지는 유형 D의 경우에는, 유형 A의 경우보다 질의 영역이나 지점이 시멘틱 세그먼트 영역과 덜 겹치게 된

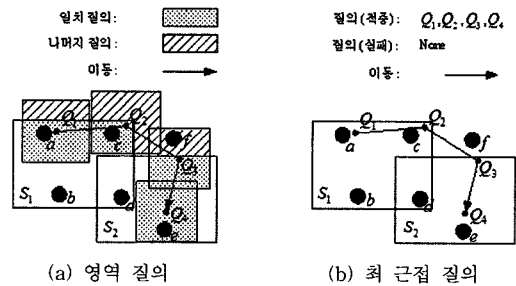
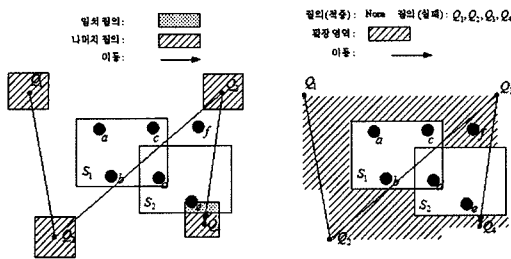


그림 9 낮은 이동성과 유사한 질의의 예

다. 그림 10은 네 개의 질의가 시멘틱 세그먼트들과 거의 겹치지 않는 경우의 예를 보여준다. 영역 질의의 경우에 발생하는 나머지 질의와 최 근접 질의의 경우에 발생하는 영역 확장 부분에 대한 시멘틱 정보는 서버에 요구되어야 한다. 그림 10(a)에서 보여지는 것처럼 영역 질의의 경우에 이동 클라이언트는 완전한 질의 결과를 얻기 위해서, 빗금친 사각형 부분에 대한 정보를 서버에 요구한다. 그러나, 최 근접 질의의 경우에는 그림 10(b)에서와 같이 질의 지점과 시멘틱 세그먼트를 포함하는 확장 부분의 시멘틱 정보가 서버에 요구된다. 이 영역의 차이에서 알 수 있듯이, 유형 D에 대해 최 근접



(a) 영역 질의 (b) 최 근접 질의

그림 10 높은 이동성과 다양한 질의의 예

질의로부터 발생하는 네트워크 트래픽은 영역 질의에서 발생하는 트래픽보다 훨씬 크다. 그러므로, 그림 7(d)와 8(d)의 SP 기법에서처럼 최 근접 질의는 확장 부분에 대한 네트워크 트래픽 때문에 영역 질의보다 10% 정도 낮은 QSR을 가진다. 이러한 결과로부터 낮은 이동성과 유사한 질의 특성을 가지는 유형 A의 이동 클라이언트들에서는 최 근접 질의의 경우에, 높은 이동성과 다양한 질의를 가지는 유형 D의 이동 클라이언트들에서는 영역 질의의 경우에 보다 높은 QSR을 얻을 수 있음을 알 수 있다.

5.4.2 캐시 교체 전략 분석

이 절에서는 본 논문에서 제안한 PPR, S-LRU, S-LFU 캐시 교체 전략과 기존의 LRU, FAR 캐시 교체 전략의 QSR을 비교한다. 5.3.1절에서와 같이, 영역 질의와 최 근접 질의에 대하여 두 가지 종류의 LDD집합이 성능 평가에 사용된다. 이동 클라이언트의 유형으로는 낮은 이동성과 유사한 질의 패턴을 가지는 이동 클라이언트 유형 A를 적용한다. PPR 전략에 대한 지역성 가중치는 0.7을 적용한다.

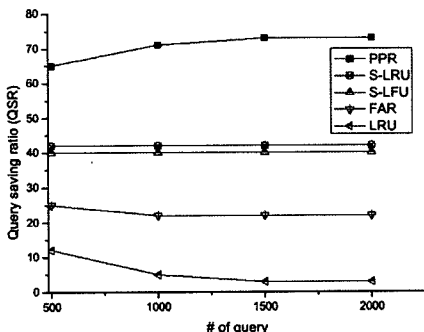
그림 11은 영역 질의에 대한 캐시 교체 전략의 QSR을 보여준다. 1000개의 질의 이후에, 다섯 가지 전략에 대한 QSR 값 모두는 큰 변화 없이 일정한 값을 유지한

다. 그 이유는 일정 개수 이상의 질의 후에 유형 A의 시멘틱 캐시는 안정화되기 때문이다. 이 그림에서 제안한 세가지 전략 (S-LRU, S-LFU, PPR)이 기존의 다른 두 전략 (LRU와 FAR)보다 높은 QSR을 보임을 알 수 있다. 캐시를 관리할 때 시멘틱 세그먼트에 대한 지역성 반영으로 PPR 전략은 S-LRU와 S-LFU 전략보다 더 높은 QSR값을 가진다. 그림 12의 최 근접 질의의 경우도 영역 질의의 경우와 유사한 경향을 보여준다. 더욱이, 그림 11과 12에서 풀린 LDD($\theta = 0.8$)가 풀어진 LDD($\theta = 0$)보다 더 높은 QSR을 가짐을 알 수 있다. 따라서, LDD가 지역적으로 집중 배치될수록, QSR은 점점 높아짐을 알 수 있다. 또한, 두 그림으로부터 영역 질의와 최 근접 질의에 대해 LDD의 복잡성과 상관없이 다른 전략들보다 PPR전략이 더 높은 QSR을 가짐을 알 수 있다.

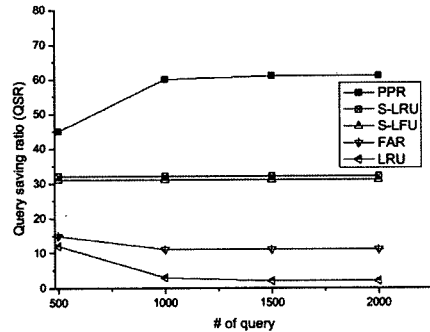
5.4.3 네트워크 트래픽 분석

그림 13은 PPR이 적용된 영역 질의의 경우에 SP와 SC 기법에 의해 생성되는 네트워크 트래픽을 보여준다. 이 그림에서, 큰 막대 사각형은 전체 네트워크 트래픽의 양, 작은 막대 사각형은 업로드 트래픽의 양을 보여준다. 그림 13으로부터, 모든 유형에 대해 SP 기법이 SC 기법보다 적은 네트워크 트래픽을 발생시킴을 알 수 있다. 흥미로운 사실은, 식 (5)와 (6)에서 SP 기법의 네트워크 트래픽 양은 추가적인 네트워크 트래픽 ($D(a)_{down}$ [SPC]과 $D(c)_{up}$ [BSC]) 때문에 SC 기법의 네트워크 트래픽 양보다 더 많아 보였다. 그러나, 그림 13은 SP 기법의 전체 네트워크 트래픽과 업로드 트래픽이 SC 기법보다 더 낮음을 보여준다. 그 이유는 SP 기법이 $D(a)_{down}$ [SPC]와 $D(c)_{up}$ [BSC]와 같은 추가적인 네트워크 트래픽을 회생하는 대신 높은 QSR 값을 얻어 서버와의 통신 횟수를 감소시켰기 때문이다.

그림 14는 PPR이 적용된 최 근접 질의의 경우에 SP와 SC 기법에 의하여 생성되는 네트워크 트래픽을 보여준



(a) 풀린 LDD



(b) 풀어진 LDD

그림 11 영역 질의에 대한 캐시 교체 전략의 비교

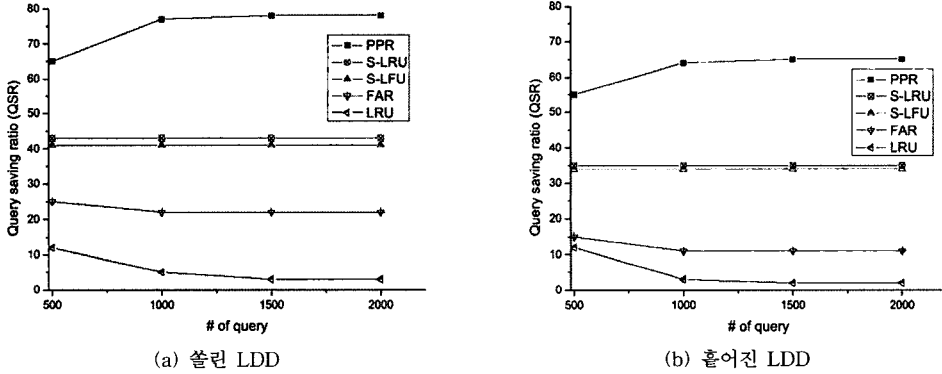


그림 12 최 근접 질의에 대한 캐시 교체 전략의 비교

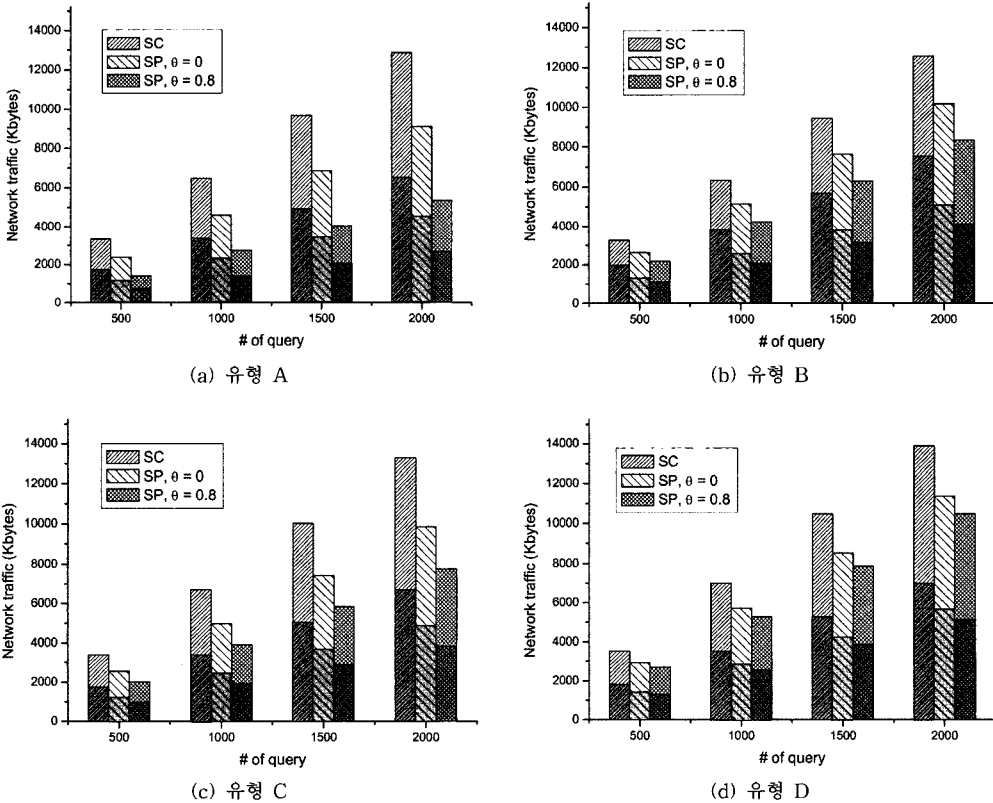


그림 13 네트워크 트래픽의 비교(영역 질의)

다. 그림 14의 결과들은 그림 13과 유사하다.

5.4.4 질의 처리 분석

질의를 하는 사용자는 질의의 결과에 관심이 있을 뿐 질의 처리 방법에는 관심이 없기 때문에, 사용자의 질의를 두 가지 질의 처리 방법 중 어느 방법으로 처리해도 상관없는 경우가 종종 있다. 예를 들어, “근처 주유소의 위치를 알려주세요” 라는 사용자 질의는 영역 질의 처리 방법과 최 근접 질의 처리 방법 중 어느 방법을 사

용하여 결과를 얻든 사용자가 원하는 결과를 제공할 수 있다. 이런 경우에 이동 클라이언트는 사용자의 이동 패턴이나 질의 특성을 고려하여 두 방법 중 어느 방법을 사용하는 것이 더 효율적인지를 선택하여 처리를 진행해야 한다.

그림 15는 PPR이 적용된 환경에서 영역 질의 처리 방법에 대한 다른 질의 처리 방법의 상대적인 네트워크 트래픽 비율을 보여준다. 즉, 모든 질의를 영역 질의

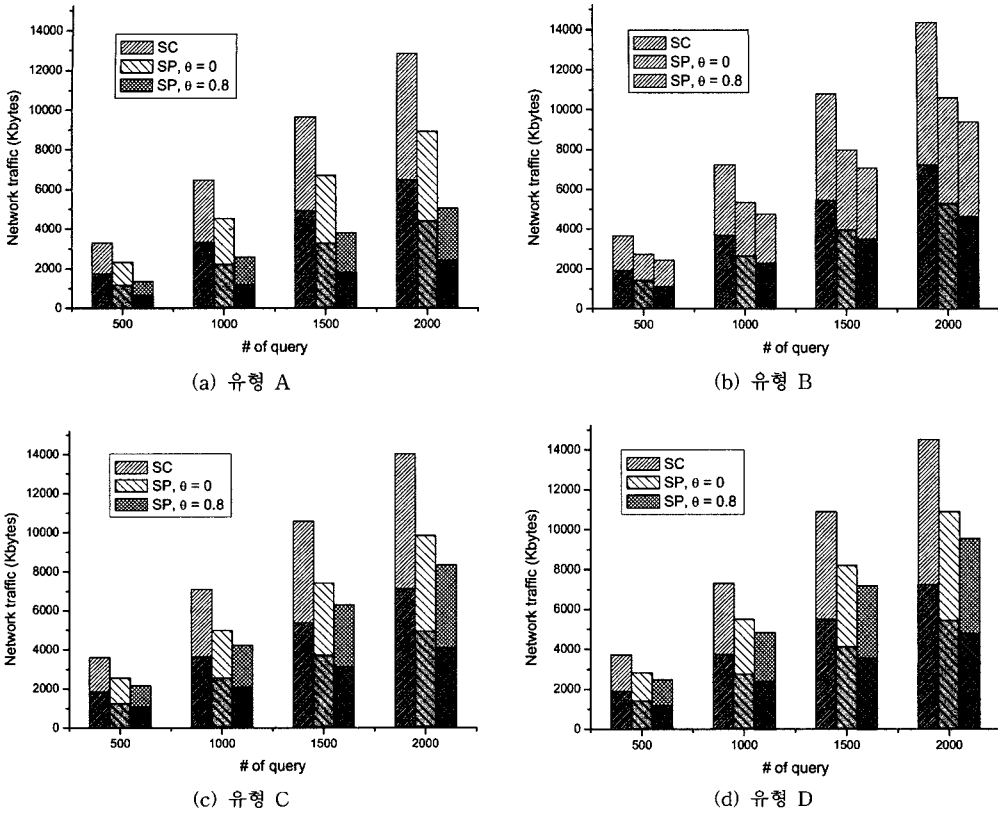


그림 14 네트워크 트래픽의 비교(최 근접 질의)

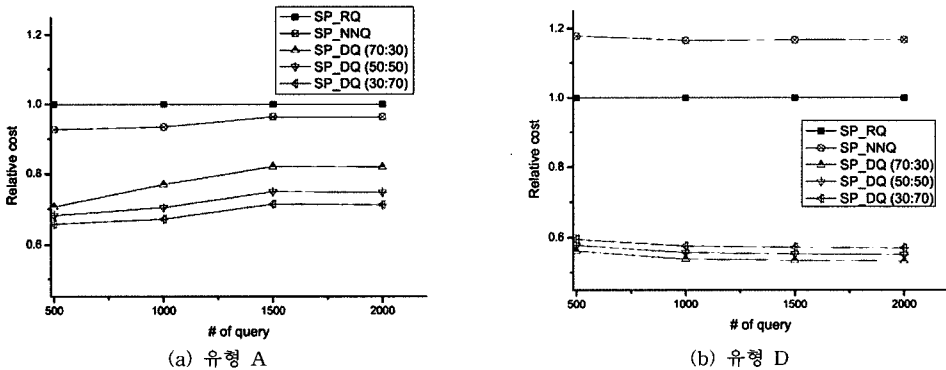


그림 15 다양한 질의 처리 성능 비교

(SP_RQ) 방법으로 처리할 때 발생하는 네트워크 트래픽 비용을 1로 놓았을 때 다른 질의 처리 방법의 상대적인 네트워크 트래픽 비용을 보여준다. 1보다 작은 값을 가지는 질의 처리 방법은 영역 질의 처리 방법보다 적은 네트워크 트래픽이 발생함을 의미한다. 그림 15의 SP_DQ는 영역 질의와 최 근접 질의 처리 방법을 혼용해서 사용하는 것을 의미하며, 괄호 안의 두 값은 영역 질의와 최 근접 질의 처리의 비율을 나타낸다. 그림

15(a)에서 낮은 이동성과 유사한 질의를 가지는 이동 클라이언트의 경우에는 최 근접 질의(SP_NNQ)로 처리하는 것이 영역 질의(SP_RQ)로 처리하는 것보다 좋은 성능을 가짐을 알 수 있다. 그러나, 모든 질의를 최 근접 질의(SP_NNQ)로 처리하는 것보다는 최 근접 질의 방법으로 처리하는 비율이 영역 질의 방법으로 처리하는 비율보다 상대적으로 높은 SP_DQ(30:70)를 사용하는 것이 가장 우수한 성능을 가짐을 알 수 있다. 그림

15(b)에서는 높은 이동성과 다양한 질의 특성을 가지는 이동 클라이언트의 경우에는 영역 질의(SP_RQ)가 최 근접 질의(SP_NNQ) 보다 더 좋은 성능을 가지며, SP_DQ (70:30)가 가장 좋은 성능을 보임을 알 수 있다. 그림 15의 결과로부터, 두 가지 질의 처리 방법을 모두 사용할 수 있는 동적인 질의 상황에서는 사용자의 이동성과 질의 유형을 고려하여 영역 질의와 최 근접 질의 처리 방법을 혼용하여 사용하는 것이 특정 질의 처리 방법 하나만 사용하는 경우보다 더 효율적임을 알 수 있다.

6. 결론

제한적인 대역폭을 사용하는 이동 컴퓨팅 환경에서 시멘틱 데이터를 처리하는 것과 관련된 기술들은 빠른 질의 응답 처리와 서버와의 통신에 기인한 네트워크 트래픽 부하 감소를 위해 이동 클라이언트 캐시를 효과적으로 활용하는 방향으로 연구가 진행되어왔다.

본 논문에서 제안한 시멘틱 프리페칭 기법은 서버로부터 미리 전달 받아 캐시에 저장해 놓은 시멘틱 정보를 이용하여 질의에 대한 처리를 수행한다. 이 기법은 이동 클라이언트의 캐시로부터 질의 결과를 얻을 수 있기 때문에 결과를 얻는데 걸리는 지연 시간과 서버와의 통신 비용을 감소시킨다. 또한, 시멘틱 질의에 대한 캐시의 효율성을 측정하는 방법으로 기존의 캐시 적중률 대신 QSR라고 불리는 새로운 측정 기준을 도입하였고 기존의 캐시 교체 전략을 개선한 세가지 캐시 교체 전략 S-LRU, S-LFU, PPR을 제안하였다. 시멘틱 프리페칭 기법을 위해 제안된 알고리즘들은 캐시에 저장된 시멘틱 서술들을 활용하여 영역 질의와 최 근접 질의를 효율적으로 처리할 수 있었다. 시뮬레이션에서는 이동 클라이언트의 이동성과 질의 유형, 위치 종속 데이터의 분포 형태를 고려하여 제안 기법과 기존 기법의 성능 비교를 수행하였으며, 시뮬레이션 결과는 제안 기법인 시멘틱 프리페칭 기법이 기존 기법보다 효율적임을 보여주었다. 캐시 교체 전략들의 성능 비교에서는 PPR 캐시 교체 전략이 영역 질의와 최 근접 질의 모두에 대해 최상의 성능을 가짐을 알 수 있었다. 또한, 동적인 질의 처리 상황에서는 이동 클라이언트의 이동성과 질의 유형을 고려하여 영역 질의와 최 근접 질의 처리 방법을 혼용하여 사용하는 것이 보다 네트워크 트래픽 감소에 효과적임을 알 수 있었다.

결론적으로, 우리가 제안한 시멘틱 프리페칭 기법과 캐시 교체 기법은 LDD 환경에서 영역 질의와 최 근접 질의에 대해 높은 QSR, 낮은 네트워크 트래픽 부하, 향상된 캐시 성능과 효율적인 질의 처리를 제공하였다. 본 논문은 클라이언트의 효율적인 질의 처리에 중점을 두었지만, 향후 연구과제로 사용자의 수가 증가할 경우 발

생할 수 있는 서버의 부하를 개선시키는 추가 연구 수행이 필요하다.

참고 문헌

- [1] A. Seydim, M. Dunham, V. Kumar, Location dependent query processing, in: Proc. Second ACM Int. Workshop on Data Engineering for Wireless and Mobile Access, 2001, pp. 47-53.
- [2] A. Seydim, M. Dunham, A location dependent benchmark with mobility behavior, in: Proc. Int. Database Engineering Applications Symposium, 2002, pp. 74-85.
- [3] M. Frankin, Challenges in ubiquitous data management, in: Proc. of Informatics'01, 2001, pp. 24-33.
- [4] E. Kaasinen, User needs for location-aware mobile services, Personal and Ubiquitous Computing 7 (1) (2003) 70-79.
- [5] D. Lin, C. Jensen, B. Ooi, S. Saltenis, Efficient indexing of the historical, present, and future positions of moving objects, in: Proc. Mobile Data Management, 2005, pp. 59-66.
- [6] J. Zhang, M. Zhu, D. Papadias, Y. Tao, D. Lee, Location-based spatial queries, in: Proc. 22th ACM SIGMOD International Conference on Management of Data, 2003, pp. 443-454.
- [7] X. Yu, K. Pu, N. Koudas, Monitoring k-nearest neighbor queries over moving objects, in: Proc. of the 21st Int. Conf. on Data Engineering, 2005, pp. 631-642.
- [8] V. Almeida, R. Guting, Supporting uncertainty in moving objects in network databases, in: Proc. the 13th ACM Int. Symposium on Advances in Geographic Information Systems, 2005, pp. 31-40.
- [9] G. Liu, Exploitation of location-dependent caching and prefetching techniques for supporting mobile computing and communications, in: Proc. the Sixth Int. Conf. on Wireless Communications, 1994, pp. 11-13.
- [10] N. Oren, A survey of prefetching techniques, Tech. Rep. TR-Wits-CS-2000-10, Southern Methodist University (2000).
- [11] G. Cao, Proactive power-aware cache management for mobile computing systems, IEEE Transactions on Computer 51 (6) (2002) 608-621.
- [12] T. Ye, H.-A. Jacobsen, R. Katz, Mobile awareness in a wide area wireless network of infostations, in: Proc. The Fourth Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking, 1998, pp. 109-120.
- [13] K. Lee, H. Leong, A. Si, Semantic query caching in a mobile environment, Mobile Computing and Communications Review 3 (2) (1999) 28-36.
- [14] Q. Ren, M. Dunham, Using semantic caching to manage location dependent data in mobile computing, in: Proc. the Sixth Annual Int. Conf. on

- Mobile Computing and Networking (MOBICOM'00), 2000, pp. 210-221.
- [15] B. Chan, A. Si, H. Leong, A framework for cache management for mobile databases: Design and evaluation, Distributed and Parallel Databases 10 (1) (2001) 23-57.
- [16] D. Barbara, Sleepers and workaholics: Caching strategies in mobile environments, in: Proc. ACM SIGMOD Int. Conf. on Management of Data, 1994, pp. 1-12.
- [17] N. Roussopoulos, An incremental access method for viewcache: Concept, algorithms, and cost analysis, ACM Transactions on Database Systems 16 (3) (1991) 535-563.
- [18] L. Li, K. Birgitta, N. Pissinou, K. Makki, Strategies for semantic caching, in: Proc. 12th Int. Conf. on Database and Expert Systems Applications, 2001, pp. 284-298.
- [19] Q. Ren, M. Dunham, V. Kumar, Semantic caching and query processing, IEEE Transactions on Knowledge and Data Engineering 15 (1) (2003) 192-210.
- [20] S. Dar, M. Franklin, B. Jonsson, D. Srivatava, M. Tan, Semantic data caching and replacement, in: Proc. 26th Int. Conf. on Very Large Data Bases, 1996, pp. 330-341.
- [21] J. Xu, X. Tang, D. Lee, Performance analysis of location-dependent cache invalidation schemes for mobile environments, IEEE Transactions on Knowledge and Data Engineering 15 (2) (2003) 474-488.
- [22] B. Zheng, D. Lee, Semantic caching in location-dependent query processing, in: Proc. of the 7th Symposium on Spatio-temporal Databases, 2001, pp. 97-116.
- [23] Z. Song, N. Roussopoulos, K-nn search for moving query point, in: Proc. of the 7th Symposium on Spatio-temporal Databases, 2001, pp. 79-96.
- [24] X. Sun, N. Kamel, L. Ni, Processing implication on queries, IEEE Transactions On software Engineering 15 (10) (1989) 1168-1175.
- [25] N. Roussopoulos, S. Kelley, F. Vincent, Nearest neighbor queries, in: Proc. 14th of ACM SIGMOD International Conference on Management of Data, 1995, pp. 71-79.
- [26] A. Papadopoulos, Y. Manolopoulos, Nearest Neighbor Search: A Database Perspective, Springer-Verlag, New York, 2005.
- [27] G. Zipf, Human Behaviour and the Principle of Least Effort, Addison-Wesley, 1949.



강 상 원

1998년 2월 고려대학교 전산학과 졸업 (이학사). 2003년 2월 고려대학교 대학원 컴퓨터학과(이학석사). 2003년 3월~현재 고려대학교 대학원 컴퓨터학과 박사과정 관심분야는 Mobile Data Management, Semantic-based Data Approach, Location-Based Services, Query Processing, RFID 등



송 의 성

1997년 2월 고려대학교 컴퓨터학과 졸업 (이학사). 1999년 2월 고려대학교 대학원 컴퓨터학과 졸업(이학석사). 2005년 2월 고려대학교 대학원 컴퓨터학과 졸업(이학박사). 2006년 3월~현재 부산교육대학교 컴퓨터교육과 전임강사. 관심분야는 이동 컴퓨팅, 네트워크 이동성, 위치 기반 서비스, 이동 에이전트, 분산시스템 등