

# 모바일 데이터베이스 환경의 신뢰성 보장 모바일 클라이언트-서버 프레임워크

주해종\*, 홍석주\*, 박영배\*

\*명지대학교 컴퓨터공학과 데이터베이스연구실  
e-mail:hjjoo@mju.ac.kr

## A Generic Framework for Reliable Mobile Client-Server System in Mobile Database Environments

Hae-Jong Joo\*, Suk-Joo Hong\*, Young-Bae Park\*

\*Dept of Computer Science, Myoung-Ji University

### 요 약

본 논문은 모바일 클라이언트-서버(Mobile Client-Server) 환경에서 모바일 데이터베이스 시스템 특성상 가질 수 있는 무선 네트워크의 약한 연결성 및 접속성 단절로 인한 데이터베이스 비축(Database Hoarding)과 관련된 문제, 공유 데이터(Shared Data)의 일관성(Consistency) 유지 문제, 그리고 로그(Log) 최적화 문제를 해결하기 위한 모바일 질의 처리 시스템(MQPS : Mobile Query Processing System)을 포함하는 새로운 모바일 클라이언트-서버 시스템을 제안하는데 목적이 있다.

### 1. 서론

무선 네트워크(Wireless Network)의 약한 연결성 및 접속단절, 모바일 클라이언트의 이동성, 모바일 클라이언트의 휴대성으로 인해 발생하는 모바일 데이터베이스 시스템(Mobile Database System) 관련 이슈들과 이 문제들을 해결하기 위한 연구들도 한창이다[1,2,3,7]. 이동 컴퓨팅은 언제 어디서나 원하는 모든 정보를 이용할 수 있는 사용자의 편의성이나 성능 면에서의 요구를 만족시키고 있지만, 데이터 관리 측면에서는 해결되어야만 하는 많은 문제점들을 안고 있다[1,3].

이러한 문제점을 해결하기 위해, 본 논문은 모바일 클라이언트-서버 환경에서 모바일 데이터베이스 시스템 특성상 가질 수 있는 무선 네트워크의 약한 연결성 및 접속성 단절로 인한 데이터베이스 비축(Database Hoarding)과 관련된 문제, 공유 데이터(Shared Data)의 일관성(Consistency) 유지 문제, 그리고 로그(Log) 최적화 문제를 해결하는 새로운 모바일 클라이언트-서버 시스템의 구성을 제안하는데 목적이 있다. 이 논문의 구성은 다음과 같다. 2장에서는 본 논문의 이론적 고찰을 위해 모바일 클라이언트-서버 구조와 모바일 데이터베이스 시스템

특성에 따른 문제 정의를 살펴본다. 3장에서는 모바일 데이터베이스 환경에 문제를 해결하기 위한 모바일 질의 처리 시스템(MQPS)을 포함하는 시스템의 요소 정의, 설계, 그리고 신뢰성 보장기법을 설명한다. 마지막으로 4장에서는 결론을 맺는다.

### 2. 관련 연구

#### 2.1 모바일 클라이언트-서버 모델 구조

(그림 1)의 모바일 환경에서의 일반적인 명령전달 구조는 응용 클라이언트(Application Client)가 무선망을 통해 유선망상의 서버로 요청(Request)하는 모델을 나타내며, Coda[1]에 의해 소개된 C-CA-S 모델은 소규모 UNIX 파일 시스템을 가진 서버와 대규모 클라이언트에 사용되며, 이 클라이언트 에이전트(CA)는 서버에 의해서 정상적으로 수행된 파일 시스템 연산들을 모바일 클라이언트를 대신에서 수행한다. 또한 C-SA-S 모델에서 서버 에이전트(SA)는 서버와의 인터페이스 작업을 수행하는 유선망에 존재하는 프록시(Proxy)와 같으며, 그러한 프록시를 서버 에이전트(SA)라고 부른다. 이와 같이 기존의 클라이언트-서버 모델의 기능을 향상시키기 위해서 클라이언트 또는 서버 측에 에이전트를 사용하였지만,

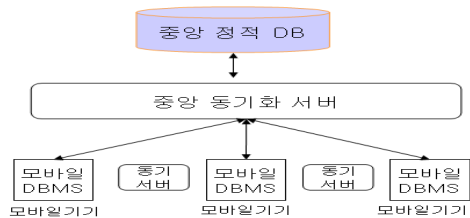
이동하는 동안 클라이언트 에이전트(CA)와 동작을 할 수 있도록 서버 또는 클라이언트 응용을 수정하기에 항상 수월한 것만 아니다. 이러한 난제를 해결하기 위해 클라이언트와 서버 양쪽에 에이전트를 두어, 쌍방의 에이전트간에 무선망과 유선망의 모든 통신을 담당하도록 한 것이 바로 C-I-S 모델이다.

일반적인 명령 전달 구조	
Client-Client Agent-Server 데이터 전달 구조 (C-CA-S)	
Client-Server Agent-Server 데이터 전달 구조 (C-SA-S)	
Client-intercept-Server 데이터 전달 구조 (C-I-S)	

(그림 1) 기존 모바일 클라이언트-서버 모델 구조

### 2.2 기존 모바일 데이터베이스 시스템 동기화

모바일 데이터베이스 업체들은 중앙의 데이터베이스와 각 모바일 데이터베이스 사이에 동기화 서버가 존재하여 전체의 동기화를 이루는 방식을 취하고 있다. 모바일 데이터베이스 사이의 동기화는 고려되어 있지 않거나 동기화를 위하여 모바일 데이터베이스 사이에 새로운 동기화 서버를 각각 위치시키는 방식을 취하고 있다[5]. 대부분의 모바일 데이터베이스 업체들은 모바일 데이터베이스를 기존의 데이터베이스의 확장 개념으로 접근하고 있으며 중앙의 정적 데이터베이스와 일관성 유지를 위하여 하나의 중앙 싱크서버를 사용한다[5].



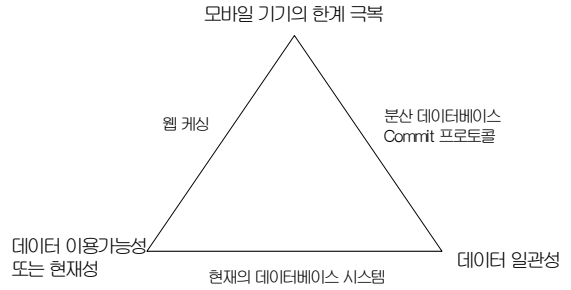
(그림 2) 기존 모바일 데이터베이스 시스템 동기화

### 2.3 모바일 데이터베이스 시스템 특성

(그림 3)과 같이 모바일 데이터베이스 환경에 적합한 데이터베이스 서비스를 설계할 때, 세가지 요소를 고려해야 한다[1,7].

즉 첫째, 본질적으로 모바일 기기가 지니고 있는 무선망에서의 한계를 극복해야 하며 둘째, 특정 데

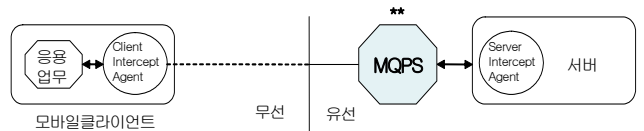
이터베이스 응용에서 해당 데이터의 이용가능성 또는 현재성이 보장되어야 하고 셋째, 모바일 기기에 표시되는 데이터는 데이터 서버 또는 데이터 웨어하우스로부터 사용되는 데이터 일관성을 유지해야 한다는 사실이다.



(그림 3) 모바일 데이터베이스 환경 세가지 고려 요소

### 3. MQPS 클라이언트-서버 시스템 구성

이 논문에서 제시하는 새로운 모바일 클라이언트-서버 시스템 구조는 클라이언트와 서버 양쪽에 에이전트를 가지는 시스템(C-I-S)이외에 모바일 클라이언트 지향적인 데이터 웨어하우스 기능을 가진 추가적인 개체로 모바일 질의처리 시스템(MQPS)으로 구성된다. 에이전트와 서버는 유선 네트워크를 통해 연결되어 있고, 에이전트는 모바일 클라이언트와 무선 네트워크를 통해 통신한다. 처리속도가 상대적으로 느리고 대용량 데이터 처리에 한계가 있는 모바일 클라이언트와 무선 네트워크를 통한 통신 부하를 줄이기 위한 구조이다.

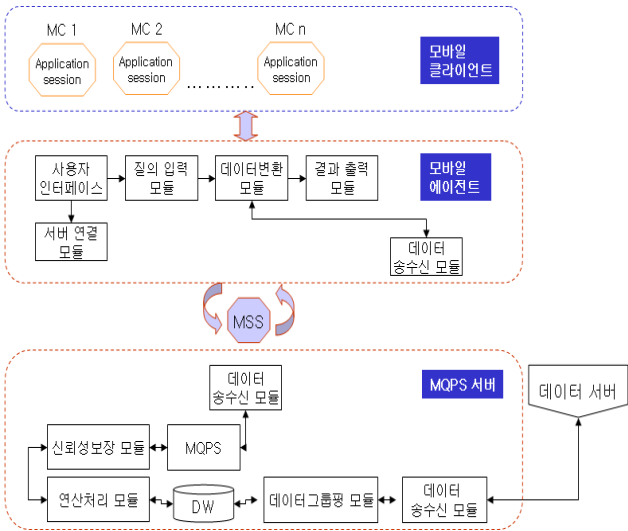


(그림 4) MQPS를 포함하는 모바일 클라이언트-서버 구조

#### 3.1 시스템 구성요소

제안 시스템은 모바일 클라이언트, 모바일 에이전트와 MQPS 서버로 구성되며, (그림 5)와 같은 구조를 갖는다. 모바일 컴퓨팅 환경은 모바일 클라이언트(MC)라고 부르는 모바일 컴퓨터들과 컴퓨터들의 유선 네트워크들로 구성된다. 모바일 클라이언트는 모바일 지원 스테이션(MSS)이라고 하는 컴퓨터를 통해 컴퓨터의 유선 네트워크와 통신을 수행한다.

각 MSS는 자신이 지원 가능한 지리적 영역, 즉 셀 내부의 모바일 클라이언트들을 관리한다. 모바일 에이전트는 실시간으로 모바일 클라이언트의 요청에 관한 질의를 요청하고 그 결과를 확인 할 수 있다.

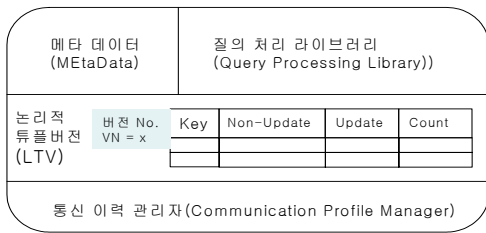


(그림 5) MQPS 클라이언트-서버 시스템 구조

모바일 클라이언트의 구성은 사용자 인터페이스, 서버 연결 모듈, 질의 입력 모듈, 데이터 변환 모듈, 결과 출력 모듈, 데이터 송수신 모듈로 되어 있다. MQPS 서버는 데이터 송수신 모듈, 데이터 그룹핑 모듈, 데이터웨어하우스(DW), 연산처리 모듈, 데이터의 동기화와 일관성을 유지하는 신뢰성 보장 모듈, 그리고 MQPS로 구성되어 있다.

### 3.2 MQPS 설계

MQPS는 접속단절이 발생하면 사용하였던 뷰에서 발생한 데이터 변화를 보관하고, 뷰의 복잡한 버전을 유지 관리하도록 다음과 같은 구성요소를 가지고 있어야 한다. 이러한 요소들을 포함하는 MQPS의 내부구조는 (그림 6)과 같다.



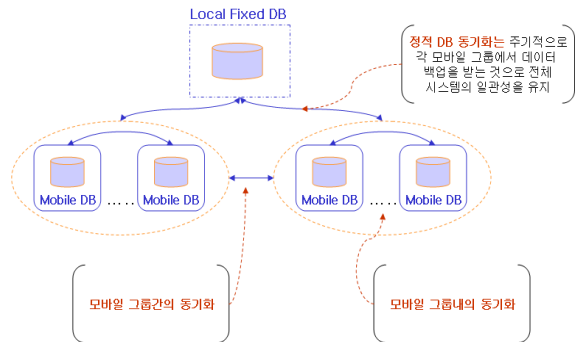
(그림 6) 설계된 MQPS의 내부 구조

### 3.3 신뢰성 보장 기법

#### 3.3.1 모바일 데이터베이스 시스템 3단계 동기화

기존 모바일 시스템의 동기화는 정적 데이터베이스의 데이터 일부를 사용하는 모바일 데이터베이스들을 대상으로 하며, 종적 데이터베이스와 모바일 데이터베이스 사이의 데이터 동기화는 주기적으로 발생한다고 가정한다. (그림 7)은 기존 모바일 데이터베이스 동기화 시스템 전체에 대한 개요도이다.

정적 데이터베이스는 단일 혹은 분산 데이터베이스로서 하나의 통합된 정보를 모바일 데이터베이스에게 제공한다. 또한 주기적인 동기화뿐만 아니라 필요에 따라 시스템 요청 에이전트를 이용하여 모바일 그룹간의 동기화를 이룬다. 각 모바일 그룹은 그룹 안에서 동기화를 유지하면서 동기화가 이루어진 다른 그룹과 질의 에이전트를 통하여 필요한 데이터를 처리하고 전체 시스템의 동기화를 처리한다. 모바일 데이터베이스 시스템 전체의 동기화를 위해 모바일 그룹내에서의 동기화, 모바일 그룹간의 동기화, 그리고 중앙의 정적 데이터베이스 동기화 3가지 단계를 거친다.



(그림 7) 모바일 데이터베이스 시스템의 3단계 동기화

#### 3.3.2 모바일 데이터베이스 그룹핑

모바일 데이터베이스 환경의 성능이 향상되고는 있지만 기존의 정적인 데이터베이스와 비교하여 작은 데이터 용량과 처리 능력을 가진 것은 확실하다. 이러한 환경에서 모바일 데이터베이스는 정적 데이터베이스의 선택적인 일부만을 복사하여 사용하게 된다[5]. 이는 각 모바일 데이터베이스가 자신들이 주로 사용하는 데이터의 종류를 구별할 수 있다는 것이다. 이러한 사실에 근거하여 각 모바일 데이터베이스를 각각의 사용에 따라 분류하여 그룹핑한다. 분류된 모바일 데이터베이스는 해당 모바일 그룹안에 배치되며 그룹별로 데이터의 일관성 유지와 현실성 유지 정책이 적용된다.

#### 3.3.3 모바일 데이터베이스 일관성 유지

MQPS의 또 다른 기능 설계 중에 하나는 MC의 사용 데이터의 일관성을 보장하기 위하여 트랜잭션이 발생한 후에 뷰와 데이터 소스를 사상(Mapping)하는 작업이 필요하다. 이를 위해서 본 논문에서는 다음과 같은 (정의 1)하에 MC의 사용 데이터 일관성 보장을 유지하는 방법으로 사용한다.

**(정의 1) 데이터 일관성 보장 정의**

데이터 소스 상태 벡터  $ssv_k = [ds1_k, ds2_k, \dots, ds_{m_k}]$ 는 주어진 트랜잭션  $T_k$ 가 실행된 후에 모든 데이터 소스의 상태를 포함한다고 하자. 그러면  $T_k$  트랜잭션을 수행한 후에 뷰에서 데이터 소스 상태  $m$ 까지의 사상(Mapping)이 다음과 같이 존재한다 :

1. 각 뷰의 상태  $vs_j$ 는 임의의  $k$ 에 대하여 타당한 상태  $m(vs_j) = ssv_k$ 를 반영한다.
2. 만약  $vs_j = vs_i$  이면,  $m(vs_j) < m(vs_i)$ 이다.

**3.3.4 모바일 데이터베이스 현실성 유지**

MQPS의 다른 기능 설계 중에 하나는 MC의 현실 데이터의 이용가능성을 보장하기 위하여 서버측의 DBMS 에이전트와 연동되어 모니터링 뷰를 생성하여 현실성을 재계산하는 방법이다. 이를 위해 데이터 서버 DS1, DS2, DS3 각각의  $r1(a,b)$ ,  $r2(b,c)$ ,  $r3(c,d)$  테이블이 있고, 이들 테이블을 조인하여 데이터의 현실성을 측정하기 위한 모니터링 뷰가 생성되어 있다고 가정하자. (그림 8)은 DBMS 에이전트와 모니터링 뷰가 연동되어 데이터의 현실성 평가를 하는 알고리즘이다.

```

public class Coordinating DBMS-agent() {
// Query DS2.r2
SELECT DS2.r2.b, DS2.r2.c ;
FROM DS2.r2 ;

B = b and C = c ;
local variable version number v = 0 ;

Start view recomputation: {
    Supply query trip plan (query tree)
    with Query DS2.r2 results.
    Launch view evaluation DBMS-agent
    with query trip plan and version number v = 0
} //end recomputation

Begin Monitoring: {
Every (Monitor_time) {
    SELECT DS2.r2.b, DS2.r2.c
    FROM DS2.r2

    if (B ≠ b or C ≠ c) {
        version number v = v + 1 ;
        Start view recomputation: {
            Supply query trip plan (query tree)
            with Query DS2.r2 results ;
            Launch view evaluation DBMS-agent
            with query trip plan
            and current version number v ;
        } //end recomputation
    } //end if
} //end Every
} //end Monitoring
}

```

(그림 8) 데이터 현실성 평가 알고리즘

**3.3.5 로그 최적화**

현재의 모바일 환경에서는 MC에서 발생한 변경 내용을 MC에 로그로 기록하였다가 MC의 단절 후 재통합 시 사용하기 위해 로그를 사용하고 있다. 이는 또한 무선 통신 환경의 특성상 물리적 또는 시스템적 요소로 인한 잦은 고장이 발생할 때에도 활용가

능하다[5]. 이러한 상황에 대처할 수 있는 고장허용의 관리 능력은 신뢰성 있는 시스템을 구축하기 위한 가장 중요한 요소 중의 하나이다. 아울러 로그의 크기를 최소화하는 것은 MC의 로컬 메모리 절약, 재통합 동안 무선 대역폭 전송시간 단축에도 유리한 조건이 된다.

**4. 결론**

본 논문에서는 모바일 데이터베이스 환경에서 무선망의 약한 연결성과 접속단절 시에 모바일 컴퓨팅 과정에서 발생할 수 있는 데이터 처리의 안정성과 사용 데이터의 일관성 유지, 그리고 대용량 데이터 사용시의 효율적인 모바일 기기의 한계성 극복을 위해 MQPS를 포함하는 새로운 모바일 클라이언트-서버 시스템을 제안하였다.

향후의 과제로는 제안시스템 설계 및 구현을 위한 방안을 제시하고, 시뮬레이션을 통해 기존의 방법보다 우수함에 대한 검증이 필요하다.

**참고문헌**

- [1] 주해중, 박영배, “모바일 데이터베이스 환경의 신뢰성 보장 질의처리 시스템 설계”, 정보처리학회 논문지D 12권 4호, pp.521-530, 2005.8
- [2] 이재우, “모바일 데이터베이스 응용 사례”, 데이터베이스연구회, 17권 3호, pp.115-118, 2001.9
- [3] 최미선, 김영국, “이동(Mobile) 데이터베이스 개요 및 연구 현황”, 데이터베이스연구회, 17권 3호, pp.3-16, 2001.9
- [4] Margaret H. Dunham and Vijay Kummer, "Impact of Mobility on Transaction Management", Proceeding of the International Workshop on Data Engineering for Wireless and Mobile Access, pp.14-21, August 1999
- [5] H. Joanne and A. Divyakant, "Planned Disconnections for Mobile Database", Proceedings of IEEE 11th international workshop, 2000
- [6] Sanjay Kummer Madria, Bharat K. Bhargava, "A Transaction Model to Improve Data Availability in Mobile Computing", Distributed & Parallel Databases 10(2):127-160, 2001
- [7] G. Walborn and P. K. Chrysanthis, "Proceeding in Mobile Database Applications", In Proceeding of the 14th Symposium on the Reliable Distributed Systems, September 1995