

모바일 데이터베이스 관리 시스템의 성능향상을 위한 이동 에이전트 기반 그룹핑 기법

신 성 옥[†] · 정 동 원[†] · 이 정 옥^{**} · 백 두 권^{***}

요 약

무선 네트워크의 폭발적인 증가와 이동 통신 기기의 발전은 기존에 사용되어지던 유선 일변도의 사용방식을 무선 환경으로 급속히 확장, 이 전시키고 있다. 무선 환경의 발전에 따라 모바일 데이터베이스 사용의 필요성은 급격히 증가하고 있으며 이에 따른 많은 문제점도 발생하고 있다. 특히 현재의 모바일 데이터베이스는 동기화를 위하여 하나의 동기화 서버가 다수의 모바일 데이터베이스를 관리하는 중앙 집중형 방식으로 이는 전체 시스템의 성능을 저하하는 주요 요소이다. 또한 고장허용에 대한 해결책은 전혀 고려되어 있지 않다. 이러한 문제점을 해결하기 위하여 본 논문에서는 이동 에이전트를 이용한 모바일 데이터베이스 관리 시스템을 제안한다. 이 시스템은 모바일 데이터베이스의 그룹핑을 통하여 전체 시스템의 성능을 높이며 고장허용 관리 기능의 지원이 가능하다.

A Mobile Agent-based Grouping Method for enhancing Performance of Mobile Database Management Systems

SungOog Shin[†] · DongWon Jeong[†] · JeongOog Lee^{**} · DooKwon Baik^{***}

ABSTRACT

A rapid growth of mobile devices and a proliferation of wireless networks lead to changing the existing environment into a mobile environment. Expansion of mobile environment makes a pressing need of mobile databases, and there are several solutions to achieve it. But they still accompany with some problems. Especially, the current synchronization server that manages consistently many mobile database management systems is based on the centralized method to synchronize data between the mobile database systems. It is the major factor that causes the lower performance of the whole system. Furthermore, the fault tolerance has not been considered in the existing systems. Therefore, we suggest the mobile agent-based mobile database system in order to resolve the problems. The system, we propose in this paper, provides the high-performance of the whole system by mobile grouping, and also may support a fault-tolerance management method.

키워드 : 모바일 데이터베이스(Mobile Database), 에이전트(Agent), 일관성 유지(Consistency), 고장 허용(Fault-tolerance)

1. 서 론

최근 폭발적인 네트워크의 발전은 무선 네트워크 부분에서도 일어나고 있다. 무선 네트워크의 근간이 되는 무선 대역폭의 확장과 함께 무선 장비들의 발전으로 통신환경과 기기에 제한적이던 무선 환경이 대폭 개선되고 있다[1]. 이에 따라서 지금까지 한정적이던 무선기기의 데이터 처리량이 기하급수적으로 증가하고 있으며 단순한 단말기의 역할에서 하나의 독립적인 데이터 처리를 요구하는 디바이스로 발전하고 있다. 이러한 요구에 따라 무선기기에서 사용되는 모바일 데이터베이스의 요구는 폭발적으로 증가하고 있

며 현재 여러 제품이 상용화되었다.

기존의 모바일 데이터베이스 시스템은 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. 이러한 구조는 동기화 서버에 대한 많은 이동 기기의 잦은 접속으로 인한 과부하가 발생할 수 있으며 모바일 데이터베이스와의 동기화를 위해서 불필요한 자원이 소모된다. 또한 모바일 디바이스간의 동기화를 위해 별도의 동기화 서버가 필요하므로 자원의 낭비는 물론 융통성 있는 모바일 데이터베이스 시스템의 운영에 제약이 가지게 된다. 그리고 무선 환경의 특성중의 하나인 예측할 수 없는 사고상황에 대응할 수 있는 고장허용(fault-to-

[†] 준 회원 : 고려대학교 대학원 컴퓨터학과
^{**} 정 회원 : 건국대학교 컴퓨터·융용과학부 교수
^{***} 종신회원 : 고려대학교 컴퓨터학과 교수
 논문접수 : 2002년 8월 3일, 심사완료 : 2003년 1월 6일

lerance)과 비연결 동작(non-connection operation)에 대한 연구가 미흡한 실정이다.

이러한 문제점을 해결하기 위하여 이동 에이전트를 기반으로 하는 모바일 데이터베이스 관리 시스템을 제안한다. 이 시스템은 무선 환경에서의 일관성 유지와 비 연결 동작, 고장 허용을 제공 할 수 있는 기반 구조를 제공하며, 중앙 집중적인 동기화 서버의 문제점인 중앙 서버에 대한 부하 분산 정책 부재와 일관성 유지 서버의 추가적인 비용발생 및 중앙 집중적인 동기화를 해결하기 위해서 모바일 데이터베이스에 대한 그룹핑을 실시한다. 이 논문의 구성은 다음과 같다. 2장에서는 이 시스템을 구성하는데 사용된 요소 기술에 대해서 기술한다. 3장에서는 시스템에 대한 구조적인 설명과 함께 각 기능에 대하여 설명한다. 4장에서는 시뮬레이션 시스템을 구현하고 시뮬레이션 결과에 대한 분석 결과를 설명한다. 마지막으로 5장에서는 연구 성과와 함께 앞으로 추가되어야 할 연구 분야에 대하여 설명한다.

2. 관련 연구

이동 에이전트는 이질적인 기기 또는 동작 환경을 가지는 네트워크 사이를 이동하면서 사용자를 대신하여 자율적으로 특정한 작업을 수행하는 컴퓨터 프로그램이다[2]. 이러한 이동 에이전트의 개념은 기존에 연구되어왔던 프로세스 이동(process migration), 원격 평가(remote evaluation), 이동 오브젝트(mobile object) 기술에서부터 발생되었다[3-5]. 따라서 위에서 언급된 기술들이 지니는 장점과 함께 에이전트 본래의 지능적인 작업수행의 장점을 가진다. 이동 에이전트는 에이전트의 자율적인 경로선택, 데이터 처리 프로세스의 전송 및 실행, 독립적인 실행 환경, 이동 에이전트간 통신을 통한 작업 세분화 등의 기능으로 전통적인 클라이언트/서버 환경의 정적, 수동적 시스템으로 적용하기 어려운 이질적인 분산 데이터베이스 시스템과 무선 통신 환경에 적합한 환경을 제공한다[2, 6, 7].

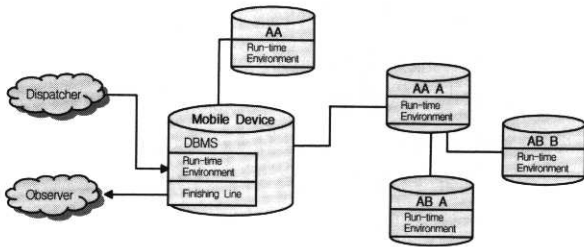
ad-hoc 네트워크는 불규칙한 이동 경로와 구성원을 가지는 이동 통신 환경에 대처할 수 있는 가장 효율적인 방법 중 하나이다. 각 모바일 데이터베이스는 하나의 ad-hoc 네트워크를 구성하여 동작한다. ad-hoc 네트워크는 중앙 집중화 된 관리나 표준화된 지원 서비스의 도움없이 임시 망을 구성하는 무선 이동 호스트들의 집합으로 백본 호스트나 다른 이동 호스트로의 연결을 제공하기 위한 고정된 제어 장치를 가지고 있지 않다. 분산운영과 동적인 네트워크 형태, 불규칙한 링크용량 그리고 소전력 기기에 사용된다는 특성을 가지고 있다[10]. ad-hoc network의 각각의 노드(이동 기기)들은 송신자/수신자를 역할을 함께 수행할 수 있으며 자신이 포함된 네트워크를 담당하는 홈 노드으로써 동작하는 것도 가능하다. ad-hoc network에 포함되는 구성원의

수가 많아질수록 송신자와 수신자와의 통신은 많은 시간적, 금전적 비용이 발생한다. 이러한 문제를 해결하기 위하여 물리적 이웃(physical neighbors)과 논리적 이웃(logical neighbors)을 설정한다. 물리적 이웃은 해당 노드가 전송할 수 있는 주변 노드들을 지정하고 논리적 이웃은 자신이 속한 네트워크의 홈 노드가 가지는 주변의 홈 노드를 지정한다. 물리적 노드의 전송 거리, 그리고 논리적 노드와 이웃한 홈 노드의 수를 조절하여 많은 구성원을 가지는 ad-hoc network를 효율적으로 관리하는 것이 가능하다[11]. 무선 환경에 ad-hoc 네트워크를 제공하면 배치와 재구성이 쉬우며 뛰어난 적응성을 가진 네트워크를 구성할 수 있다[9].

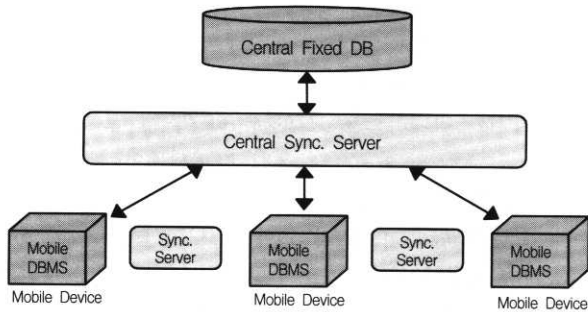
이동 환경과 이동 기기들의 저장 및 처리 용량의 부족, 데이터 전송을 위한 좁은 대역폭, 그리고 이동성의 증가에 따른 잦은 연결 에러는 이동 기기와 모바일 데이터베이스의 신뢰성과 효율성을 저하시키는 중요한 요소이다. 이를 해결하기 위한 방법으로 캐싱과 이동성-보조 스테이션(mobility-support station)이 있다. 캐싱은 이동 기기에서 비연결 상황이 발생하였을 때 자신의 캐쉬에 데이터의 변경과 처리를 임시적으로 저장하는 방법이다[12, 15, 16]. 비연결 상황에서 처리되는 데이터를 자신의 캐쉬안에 저장하면서 지역적 승인(locally commit)을 주어 처리한다. 이후 네트워크의 재 연결이 이루어지면 전체 네트워크에 대한 전역적 승인(global commit)을 획득하여 전체 네트워크의 동기화를 이룬다. 전역적 승인이 거부되었을 경우에는 지역적 승인으로 처리되었던 동작을 복구한다. 이동성-보조 스테이션은 네트워크 상에서 지역적 또는 논리적인 지역을 담당하는 이동성-보조 스테이션을 사용하여 지역안의 이동 기기를 관리하는 방법이다[13, 14]. 각 이동성-보조 스테이션은 자신이 담당하는 지역의 이동 기기에 대한 비연결 처리와 일관성 유지에 대한 책임을 진다. 하나의 그룹에 속한 이동기기의 예측된 부재에 대처하는 방법에는 basic sign-off, check-out, 그리고 relaxed check-out의 세 가지 방법이 있다[17]. basic sign-off는 부재중인 멤버를 ad-hoc 네트워크의 다른 멤버들은 손상이 아닌 일시적인 disconnect로 인식하며 부재중 멤버는 자신의 데이터베이스를 읽기전용 방식으로 사용한다. check-out과 relaxed check-out은 자신이 지정한 (check-out) 데이터베이스의 부분에 대해서는 자신만이 접근할 수 있는 권한을 주는 방식으로 relaxed check-out은 다른 멤버에게 지정된 데이터에 대한 읽기 권한만을 주는 방식이다.

데이터베이스 통합 에이전트 환경은 DBMS 내부에 에이전트가 동작할 수 있는 실행환경(agent run-time environment)을 작성하는 것이다[8]. 통합환경에서 에이전트는 DBMS가 가지는 기능들을 자신의 기능처럼 사용할 수 있다. 통합환경이 가져야 할 요소로서는 데이터를 처리하는 DBMS, 에이전트를 실제 구현, 실행 그리고 전송하기 위한 실

행환경, 에이전트를 생성할 수 있는 에이전트 운전자(dispatcher), 에이전트들의 결과를 추적하고 종료선에 대한 책임을 지는 관찰자(observer), 그리고 결과를 수집하는 종료선(finish-line)으로 구성된다. 통합 환경은 DBMS의 기본 기능에 에이전트가 공유함으로써 시스템 자원의 소모를 줄이고 직접적인 DBMS 기능의 사용으로 다른 외부 인터페이스를 사용하는 것보다 빠른 데이터 처리가 가능하다. 또한 외부 스키마에 쉽게 적용할 수 있는 기능을 제공한다. (그림 1)은 데이터베이스 통합 에이전트 환경에 대한 개요도이다.



(그림 1) 데이터베이스 통합 에이전트 환경



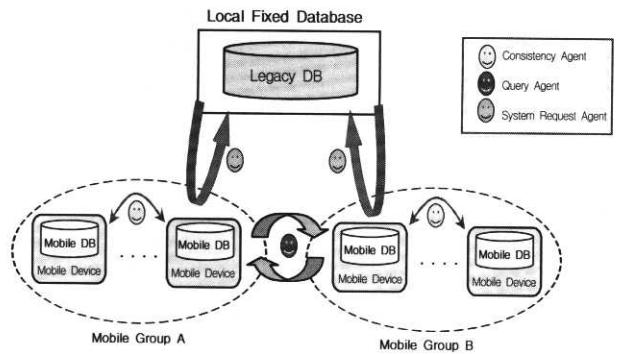
(그림 2) 기존의 모바일 데이터베이스 시스템의 동기화

(그림 2)는 현재 상용화된 모바일 데이터베이스들의 동기화에 대한 개요도이다. pointbase, Oracle, Aladin과 같은 많은 기업에서 모바일 데이터베이스를 사용할 수 있는 제품을 시판하고 있으나 현재까지는 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. Pointbase의 경우 PointBase Server, PointBase Embedded, PointBase Micro, PointBase UniSync의 4가지 제품군을 사용하여 모바일 데이터베이스 시스템을 운영하며 PointBase UniSync가 기존의 정적 데이터베이스와 모바일 데이터베이스, 그리고 모바일 데이터베이스 사이의 동기화를 책임진다. 대부분의 모바일 데이터베이스 업체들은 모바일 데이터베이스를 기존의 데이터베이스의 확장 개념으로 접근하고 있으며 중앙의 정적 데이터베이스와 일관성 유지를 위하여 하나의 중앙 싱크서버를 사용한다.

3. 이동 에이전트 기반의 모바일 데이터베이스 관리 시스템

3.1 시스템 개요

이번 절에서는 제안하는 시스템의 전체적인 구성과 각 부분의 역할에 대해서 기술하며 각 부분에 대한 상세한 설명은 다음절에서 설명한다. 이 시스템은 정적 데이터베이스의 데이터 일부를 사용하는 모바일 데이터베이스들을 대상으로 하며 정적 데이터베이스와 모바일 데이터베이스 사이의 데이터 동기화(Synchronization)는 주기적으로 발생한다고 가정한다.



(그림 3) 시스템 개요도

(그림 3)은 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 위해서는 3가지 단계를 거친다.

3.1.1 모바일 그룹내에서의 동기화

하나의 그룹에 속하는 각 모바일 데이터베이스들은 그룹 내의 다른 모바일 데이터베이스와 동기화를 유지한다. 모바일 데이터베이스의 데이터 변경은 그룹내의 다른 모바일 데이터베이스들에게 전달되고 중앙의 정적 데이터베이스와는 별개로 자신들의 데이터를 변경한다.

3.1.2 모바일 그룹간 동기화

각 모바일 그룹은 자신의 그룹에 속한 모바일 데이터베이스의 일관성을 유지하고 있으며 자신이 속한 그룹에서 관리되어지지 않은 데이터에 대한 처리는 해당 데이터를 포함하는 다른 모바일 그룹에 요청하여 처리한다. 중앙의 정적 데이터베이스는 각 모바일 그룹에 대한 정보를 관리하고 요청이 있을 때 요청한 모바일 그룹에 대한 정보를 제공한다.

3.1.3 중앙의 정적 데이터베이스 동기화

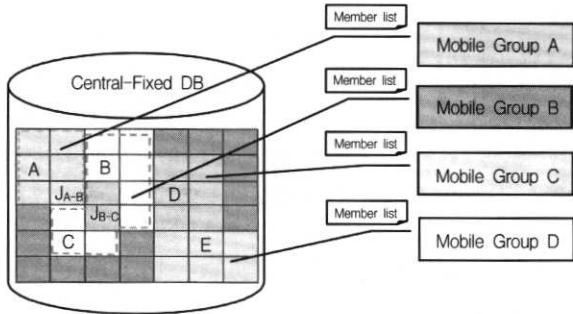
중앙의 정적 데이터베이스는 원칙적으로 지정된 시각에 각 모바일 그룹에서 데이터 백업을 받는 것으로 전체 시스템의 일관성을 유지한다. 하지만 필요시에는 해당 모바일 그룹을 호출하여 데이터 변경사항에 대한 정보를 획득함으로써 실시간적으로 일관성을 유지하는 것도 가능하다.

위의 3가지 일관성 유지 방법은 이후의 절들에서 세부적으로 설명되어진다.

3.2 모바일 데이터베이스 그룹핑

모바일 데이터베이스의 성능이 향상되고는 있지만 기존의 정적인 데이터베이스(static database)와 비교하여 작은 데이터 용량과 처리 능력을 가진 것은 확실하다[18]. 이러한 환경에서 모바일 데이터베이스는 정적 데이터베이스의 선택적인 일부만을 복사하여 사용하게 된다[17]. 이는 각 모바일 데이터베이스가 자신들이 주로 사용하는 데이터의 종류를 구별할 수 있다는 것이다. 이러한 사실에 근거하여 각 모바일 데이터베이스를 각각의 사용에 따라 분류하여 그룹핑 한다. 분류된 모바일 데이터베이스는 해당 모바일 그룹 안에 배치되며 그룹별로 데이터의 일관성 유지와 고장허용 정책이 적용된다.

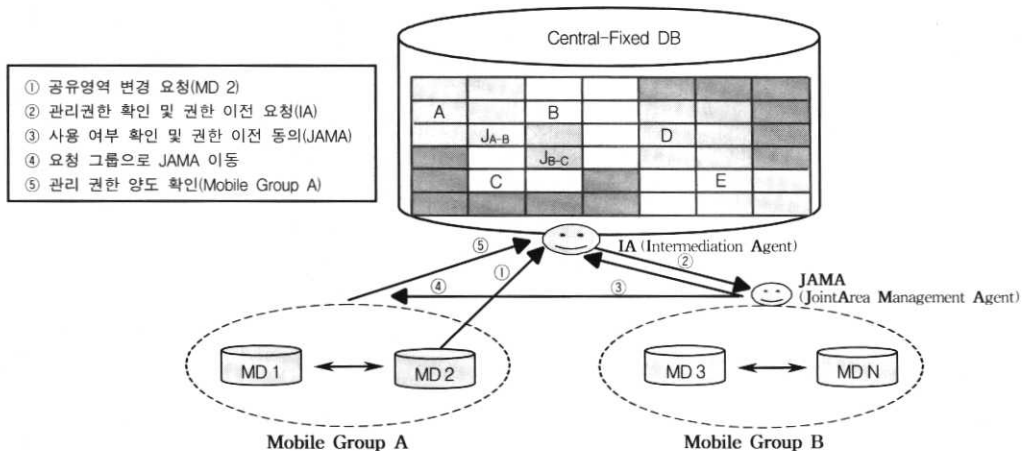
(그림 4)는 모바일 데이터베이스 그룹핑 개념을 도식적으로 보여준다. 각각의 모바일 데이터베이스는 정적인 중앙의 데이터베이스의 일부분을 사용한다. 이러한 모바일 데이터베이스의 사용 패턴을 분석하여 정적 데이터베이스를 일정한 그룹으로 분류한다. 초기 그룹핑 단계에서는 전체 모바일 데이터베이스의 수, 각 그룹에 소속되는 모바일 데이터베이스의 수, 그룹에 포함된 모바일 데이터베이스의 트랜잭션 발생빈도, 그룹이 포함되는 네트워크의 상태, 그룹이 포함하는 공유영역의 트랜잭션 발생빈도, 그리고 모바일 데이터베이스의 오류 발생빈도에 따라 각 정적 데이터베이스를 분류한다. 이를 기반으로 각 정적 데이터베이스 그룹에 속하는 모바일 데이터베이스를 하나의 모바일 그룹으로 배치한다. 이때 정적 데이터베이스는 각 모바일 데이터베이스에 그룹의 멤버에 대한 정보를 포함하는 member list를 통해서 그룹에 대한 정보를 제공한다. 모바일 그룹 안에서 각 모바일 데이터베이스는 ad-hoc 네트워크를 구성하여 개개의 데이터베이스 처리에 대한 데이터를 교환하며 각 모바일 그룹은 그룹 안에서 데이터의 일관성 유지와 고장허용 관리 기능을 제공한다.



(그림 4) 모바일 데이터베이스 그룹핑

3.3 그룹핑 기반 공유영역 관리 메커니즘

모바일 데이터베이스를 그룹핑함으로써 각 모바일 그룹은 자신들이 주로 사용하는 정적 데이터베이스의 한정된 일부분만을 관리할 수 있으며 일관성 유지에 소요되는 부하를 줄일 수가 있다. 또한 각 모바일 그룹 안에서는 그룹에 속한 각 모바일 데이터베이스의 데이터 일관성이 유지되기 때문에 현재 사용하는 모바일 데이터베이스와 같이 단순한 모니터 및 참조 기능이 아닌 하나의 독립된 데이터베이스로서의 기능 활용을 극대화 할 수 있다. 추가적으로, 중앙 집중적인 동기화 서버를 사용할 필요가 없이 각 모바일 데이터베이스는 동등한 위치를 가지는 ad-hoc 네트워크를 구성함으로써 각 이동 기기에 할당되는 부하의 균등 분할과



(그림 5) 공유영역 관리 메커니즘

함께 고장허용 관리 기능을 제공할 수 있는 기본 구조를 제공한다[9].

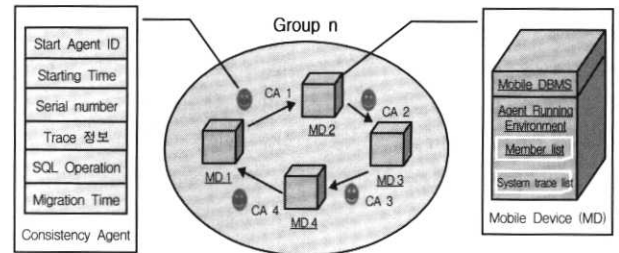
시스템 관리 측면에서 그룹핑에 있어서 가장 중요한 부분은 (그림 5)에서 JA-B로 표현되는 공유영역 (JointArea)의 관리이다. 공유영역은 각 모바일 그룹들 사이에 중복되어 요청되어진 데이터베이스 영역이다. 공유영역을 공유하는 모바일 그룹은 공유영역의 데이터 변경이 발생했을 경우 다른 그룹의 모바일 데이터베이스의 일관성을 함께 책임져야 한다. (그림 5)는 공유영역 변경 요청이 성공했을 때의 처리 단계를 보여준다. 공유영역 관리를 위해서 두 가지의 에이전트가 사용된다. 먼저 IA(Intermediation Agent)는 중앙 서버에 위치하여 공유영역 관리의 소유권이 어느 그룹에 속하는가를 추적하며 공유영역 관리 권한의 이동에 대한 책임을 진다. 공유영역 관리 권한을 가지는 그룹에 속한 모바일 데이터베이스만이 공유영역에 대한 변경을 실행할 수 있다. JAMA(JointArea Management Agent)는 공유영역 관리를 위임 받은 에이전트로 해당 그룹에 위치한다. IA는 공유영역에 대한 요청이 발생하면 JAMA의 위치를 파악하여 소유권 이동에 대한 요청을 실시한다. 공유영역을 공유하는 모바일 그룹의 수가 많아질수록 시스템 전체에 심각한 효율저하를 미칠 수 있다. 따라서 시스템 상에서 지정된 한계까지만 공유영역을 설정하도록 하며, 공유영역 특성상 제한된 그룹핑이 불가능할 경우 중앙 데이터베이스 상에서 직접 관리하도록 한다.

3.4 모바일 그룹내 일관성 유지

각 모바일 데이터베이스는 DBMS 기능과 함께 에이전트 실행 환경(run-time environment)을 제공한다[8]. 각각의 모바일 데이터베이스의 실행환경은 에이전트가 동작하고, 새로운 에이전트를 생성할 수 있는 환경을 제공한다. 모바일 그룹은 그룹 멤버에 대한 정보를 그룹핑 과정에서 획득하고 이를 일관성 유지의 기준 정보로 사용한다. 모바일 데이터베이스에서의 일관성 유지는 분산된 데이터베이스가 독립적인 데이터베이스로 작동하는데 가장 기본적이며 중요한 요소이다[1, 18].

현재의 모바일 데이터베이스는 정적 데이터베이스와 모바일 데이터베이스 사이에 하나의 동기화 서버를 두어 전체 시스템의 동기화를 유지한다. 이러한 구조는 모바일 데이터베이스의 수가 많아질 경우 동기화 서버에 집중된 접속으로 과부하가 발생한다. 또한 시스템 전체의 효율이 떨어질 수 있고 중앙의 동기화 서버의 이상은 전체 시스템의 이상을 야기하는 신뢰적이지 못한 구조이다. 모바일 데이터베이스의 비연결 처리를 위해서는 자신이 가지고 있는 모든 데이터에 대해서 정적 데이터베이스와 동기화 작업을 요구함으로써 시스템의 효율을 극도로 떨어뜨린다. 모바일 데이터베이스간 동기화는 중앙의 동기화 서버를 통해서 이

루어진다. 일부 모바일 데이터베이스 업체에서는 모바일 데이터베이스간의 동기화를 위하여 모바일 데이터베이스간 동기화 서버를 제공한다. 이러한 경우 모바일 데이터베이스의 증가에 따라 기하급수적으로 많은 동기화 서버가 필요하게되므로 자원의 낭비와 함께 유동적인 모바일 데이터베이스 운영에 장애가 된다.



(그림 6) 모바일 그룹 안에서의 일관성 유지

(그림 6)은 모바일 그룹 안에서 일관성을 유지하는 가장 간단한 예로서 단일 트랜잭션에 대하여 어떠한 손상도 발생하지 않는다고 가정한다. 각 모바일 데이터베이스는 DBMS와 에이전트의 실행환경을 가지며 두 개의 list를 유지한다.

- member list : 그룹에 속한 멤버들에 대한 정보를 가지며 전송시킬 모바일 데이터베이스를 선택하는 데이터 중에 하나로 사용한다. member list는 중앙의 고정된 데이터베이스에서 각 모바일 그룹이 분리될 때 각각의 모바일 데이터베이스에 통보되며 모바일 그룹 멤버의 변화가 있을 경우에는 변경 결과를 중앙의 데이터베이스에 통보함으로써 일관된 정보를 유지한다.
- System trace list : 정적 데이터베이스에서 분리된 이후부터 모바일 데이터베이스의 트랜잭션 정보를 저장하여 고장허용 관리와 정적 데이터베이스로의 백업에 사용한다.

그룹 안의 모바일 데이터베이스에서 SQL 질의(update, delete, insert)에 의한 데이터의 변화가 발생하면 해당 질의와 발생 시간, 자신의 정보를 포함한 이동 에이전트를 전송한다. 이동 에이전트를 전송 받은 모바일 데이터베이스는 자신의 데이터를 갱신하고 member list를 기반으로 다른 모바일 데이터로 전송한다. 이러한 단계를 반복하여 그룹의 멤버를 모두 거치면서 일관성을 유지한다. 이 논문에서는 그룹 안에 일관성 유지를 위한 정책과 구현 방법에 대한 세부적인 설명은 하지 않는다.

모바일 그룹을 사용한 일관성의 유지는 전체 모바일 데이터베이스를 분할하여 관리함으로써 그룹단위의 처리가 가능하다. 그룹단위의 처리로 하나의 모바일 데이터베이스 또는 하나의 모바일 그룹의 오류가 다른 부분의 처리에는 영향을 미치지 않는 신뢰성 있는 시스템의 구현이 가능하다. 또한 그룹간에 질의 에이전트를 사용하여 자신이 관리하지 않는 데이터에 대한 일관성 있는 처리를 허용한다. 모바일

그룹을 사용한 일관성 유지는 좀더 신뢰성 있고 확장 가능한 융통성 있는 구조를 제공한다.

3.5 고장 허용(fault-tolerance)

무선 통신 환경의 특성상 물리적 또는 시스템적 요소로 인한 잦은 고장이 발생할 수 있다[19]. 따라서 이러한 상황에 대처할 수 있는 고장허용의 관리 능력은 신뢰성 있는 시스템을 구축하기 위한 가장 중요한 요소 중의 하나이다. 현재의 모바일 데이터 베이스 시스템은 고장허용에 대하여 고려를 거의 하고 있지 않다. 모바일 데이터베이스의 고장허용에서 고려되어 할 점은 무선 통신의 특성인 고비용의 데이터 전송, 일관성 유지를 위한 빠른 처리 등이 있다. 이 시스템에서 고장허용을 관리하기 위해서는 다음 두 가지 종류의 에이전트를 사용한다.

- ① 모니터 에이전트 : 모바일 데이터베이스나 이동 기기의 상황을 주기적으로 점검하여 정상 상태로 복귀되었을 때 손상시에 중지되었던 작업을 수행하고 손상 관리 에이전트를 호출한다.
- ② 손상 관리 에이전트 : trace list를 정상상태의 다른 모바일 데이터베이스와 비교하여 손상상태에서 누락된 정보를 반환함으로써 일관성을 유지한다.

손상의 종류에는 크게 연결 손상(connection fail)과 기기 손상(device fail)로 구분할 수 있다[20]. 이를 보다 상세히 분류하면, 연결 손상은 수신 손상, 송신 손상으로 구분되며 기기 손상은 에이전트, 에이전트 실행환경 및 이동 기기의 손상으로 구분된다.

모바일 데이터베이스에서의 손상은 세 가지 경우로 분류된다.

3.5.1 송신 손상 또는 에이전트 실행환경, 에이전트의 손상
 에이전트 기반 모바일 데이터베이스 시스템에서는 락커 패턴을 적용한 체크포인트(checkpoint) 기법[21]을 사용하여 이동 에이전트의 이동 전에 자신의 데이터를 저장한다. 네트워크 상황이나 기타 이유로 인하여 송신이 손상 됐을 경우 네트워크 상황을 파악하는 모니터 에이전트를 생성한다. 모니터 에이전트는 전송하려고 하는 모바일 데이터베이스와 이동 에이전트에 대한 정보를 가지고 주기적으로 전송 상태를 확인한다. 모니터 에이전트의 생성과 함께 시스템은 다른 모바일 데이터베이스를 선택하여 전송을 시도한다. 일정 시간 이상의 송신 손상이 발생하거나 이동 기기, 에이전트 실행환경의 이상이 발생하였을 경우 시스템은 더 이상 일관성을 유지하기 힘들다고 판단하고 손상 관리 에이전트를 생성한다. 이 에이전트는 해당 모바일 데이터베이스의 변경을 금지시키고 비연결 처리만을 허용한다. 또한 주기적으로 상태를 확인하고 이상 상태가 회복되었을 경우 system

trace list를 다른 이동 에이전트로 전송하여 이상 기간동안 진행된 트랜잭션에 대한 동기화를 이룬다. 동기화가 이루어진 후에 데이터베이스에 대한 제한을 풀고 정상적인 동작을 실행하게 된다.

3.5.2 수신측의 이동 기기에 손상

수신측의 손상으로 전송이 불가능 할 경우 time-out을 적용하여 일정 시간을 시도한 뒤 응답이 없을 경우 송신측 모바일 데이터베이스는 모니터 에이전트를 생성하여 상대방의 이상 상태를 모니터링하면서 다른 모바일 데이터베이스를 선택하여 전송한다.

3.5.3 송신측의 에이전트 실행환경이나 혹은 실행환경의 손상

제안한 시스템에서 모바일 데이터베이스는 점검 단위의 전송이 무사히 이루어졌을 경우 점검 노드에 해당하는 송신 모바일 데이터베이스에 전송 완료 메시지를 송신한다. 이 경우에는 빠른 일관성의 유지를 위하여 exactly-once property [22]를 고려하지 않는다. 일정 시간동안 전송 완료 메시지가 도착하지 않으면 점검 노드는 점검 단위를 줄여서 전송을 실시한다. 점검단위가 1이 되었을 경우 수신 모바일 데이터베이스는 내부 손상이 있다고 인식하고 다른 모바일 데이터베이스로 전송한다. 고장허용 관리 기능은 일관성 유지의 중요한 요소로서 시스템의 신뢰성을 높이며 비연결 처리를 가능하도록 하여 준다.

4. 시뮬레이션 시스템 구현

이 연구에서는 모바일 데이터베이스 관리 시스템 중에서 모바일 데이터베이스의 그룹화가 이루어하는 그룹핑에 대해서 중점적으로 논의한다. 또한 그룹핑 시스템 부분만을 실제로 구축하고 일관성 유지 및 고장 허용 부분은 그룹핑 시스템 내부에 시뮬레이션 부분으로 포함되도록 구현한다.

4.1 시뮬레이션 시스템 개요

이 논문에서 제안하는 모바일 데이터베이스 관리 시스템의 여러 기능 중 그룹핑 기능은 가장 중요한 부분 중 하나이며 전체 시스템에서 가장 먼저 수행되어야 하는 기능이다. 그룹핑은 관리되어지는 각각의 모바일 데이터베이스들을 어떠한 방법으로 효율적으로 그룹화하여 관리할 것인지를 결정하는 단계이다. 그룹핑은 실제적으로 모바일 데이터베이스 관리 시스템의 효율성을 결정하는 척도가 될 뿐 아니라 일관성 유지 및 다른 기능에도 많은 영향을 준다. 그룹핑 시스템에서는 모바일 시스템의 특성상 고가의 장비와 시스템의 구축이 필수적이고 시스템의 구축을 위하여 소요되는 부가되는 자원의 소요가 예상됨으로 모바일 환경에 대한 부분은 시뮬레이션으로 대체한다.

그룹핑 시스템은 실제로 적용가능한 시스템이다. 모바일 환경에 관한 부분은 시뮬레이션으로 대체하고 시뮬레이션 결과를 그룹핑 시스템의 입력으로 사용하여 에이전트를 이용한 그룹핑의 자동화를 구현한다.

그룹핑 단계에서 고려되어야 할 요소는 다음과 같다.

- ① 전체 모바일 데이터베이스의 수
- ② 각 그룹당 모바일 데이터베이스의 수
- ③ 그룹에 포함된 모바일 데이터베이스의 트랜잭션 발생 빈도
- ④ 그룹이 할당받은 지역의 네트워크 상태
- ⑤ 그룹이 가지는 공유영역(JA)의 트랜잭션 발생 빈도
- ⑥ 모바일 데이터베이스의 오류 발생 빈도

(그림 7)은 그룹핑 시스템의 시뮬레이션을 위한 프로세스 흐름도를 표현한다.

초기 입력에서 각 모바일 데이터베이스는 모바일 데이터베이스 수와 공유영역의 한도를 적용 받아 해당 그룹으로 그룹핑된다. 시스템 상에서 배치된 모바일 데이터베이스는 정해진 입력 빈도에 따라 트랜잭션이 발생한다. 동일한 네트워크 및 환경을 유지하기 위해서 초기 그룹핑시에 각 그룹 및 모바일데이터 베이스에 적용된 환경을 함께 입력받는다. 이러한 환경 요소를 유지함으로써 그룹핑후 같은 환경에서의 시뮬레이션으로 시스템의 효율성 증가를 측정한다. 모바일 데이터베이스의 트랜잭션이 발생하면 (그림 7)의 단계를 거쳐 하나의 트랜잭션을 완료하며 각 트랜잭션의 처리 결과 및 과정은 시스템의 로그 파일에 저장된다. 트랜잭션은 입력시에 주어진 빈도 정보를 이용하여 각 그룹별, 모바일 데이터베이스 별로 발생하며 발생된 트랜잭션은 시스템에서 독립적으로 적용된다.

4.2 시뮬레이션

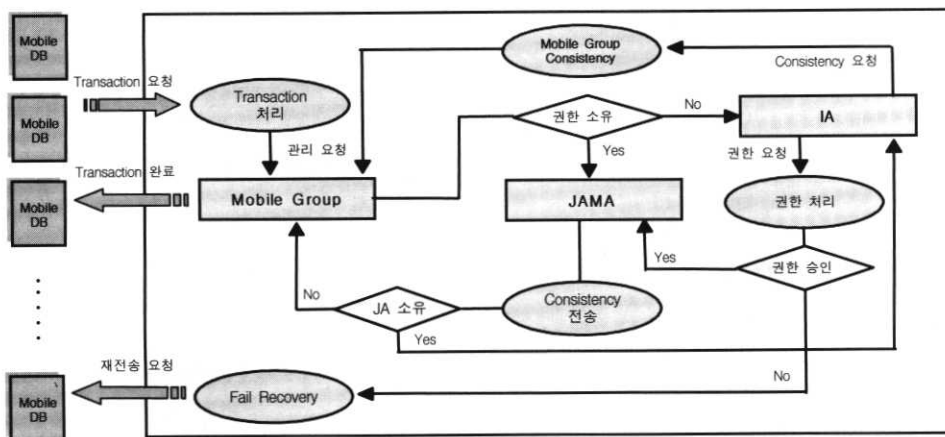
이 논문에서 제안하는 모바일 데이터베이스 관리 시스템

중 모바일 환경과 관련된 부분은 시뮬레이션 구현한다. 모바일 환경에 포함되는 부분은 모바일 데이터베이스간의 통신 기능과 같은 네트워크상에서 발생하는 트랜잭션과 일관성 유지, 고장 허용의 기능이다. 이러한 기능은 비용적, 시간적 측면에서 실제적으로 구현되기 어려운 부분이며 이상의 부분에 대한 것은 향후 연구에서 추가하도록 한다. 시뮬레이션은 하나의 트랜잭션에 대한 모바일 환경에서의 동작을 기준으로 한 것으로 각 모바일 데이터베이스에 대한 트랜잭션 요청과 트랜잭션 처리 결과는 그룹핑 시스템에서 담당한다. (그림 8)은 시뮬레이션 부분의 동작을 나타내는 hierarchical extended Petri Net이다[23]. 이 Petri Net은 time Petri Net의 개념을 이용하여 트랜잭션 시간을 측정한다. 처음 request가 발생 했을 때, 즉 모바일 데이터베이스가 트랜잭션을 요청하면 시스템은 @의 time stamp를 가진다. 이후 에이전트의 요청 처리 시간과 네트워크 전송이 이루어질 경우, 그리고 시스템에서 계산이 이루어질 경우 각 place와 transaction마다 정해진, 또는 시스템에서 선택되어진 time slot값이 추가된다. (그림 8)은 전체 시스템에 대한 Petri Net이다. 각 place는 시스템의 상태와 단위 기능을 나타내며 transaction은 place에서 점화 될 수 있는 반응 조건을 표현한다. 또한 이 Petri Net은 계층적으로 이루어져 있다. Consistency in Groups, Fail Recovery, Gain Authority, Consistency through groups의 4개의 place는 내부에 하부 Petri Net을 소유한 sub-place이다. 다음은 (그림 8)의 Petri Net을 pseudo code로 나타낸 것이다. 그룹핑 시스템의 아래의 pseudo code를 기반으로 구현된다.

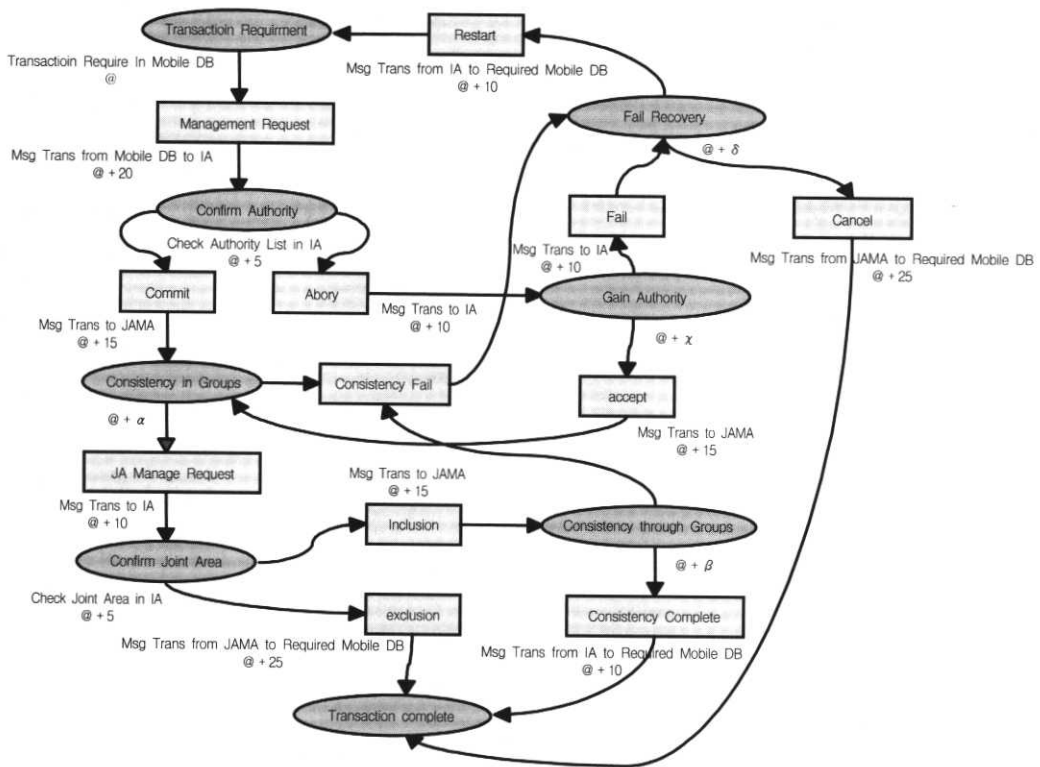
```

BEGIN
GET Request_Message /* Information of Requesting Mobile
Database */

FIND Authority List in IA
SET Authority.JAMA = Authority_Owner_Group
IF (Authority.JAMA == Request_Message.Group)
{
    
```



(그림 7) 시뮬레이션 프로세스



(그림 8) Hierarchical Extended Petri Net

```

PROCESSING Consistency in Request Group
SET Consistency_Result = Return Processing Result
IF (Consistency_Result == Fail)
    PROCESSING Fail Recovery
ELSE IF (Consistency_Result == Complete)
{
    SET JA_Inclusion_List = JA of Request Mobile Database
    WHILE (Number of JA_Inclusion_List)
    {
        READ one of JA_inclusion_List
        PROCESSING Consistency between Groups
        SET Consistency_Result = Return Processing Result
        IF (Consistency_Result == Fail)
        {
            PROCESSING Fail Recovery
        }
    }
    END WHILE
    PROCESSING Transition Complete
}
END IF
    
```

<표 1>은 시뮬레이션에 사용되는 시뮬레이션 파라미터이다. 모바일 네트워크 환경에서 발생하는 요소로는 Mobile Database, JAMA, IA 사이에서 메시지 전달에 필요한 시간이 있다. 각 파라미터는 메시지 전달에 필요한 전송 횟수와 전송에 필요한 처리량을 기준으로 time slot를 할당 받는다. 다른 하나는 시스템 상에서의 처리를 위해서 소요되는 요소이다. 처리를 위한 파라미터는 시스템 처리 속도와 중복

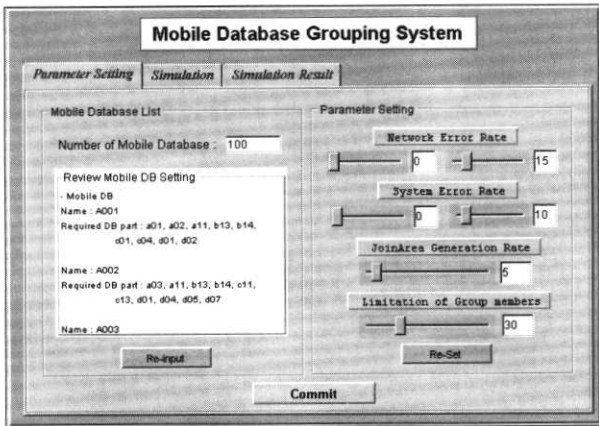
처리의 가능성을 기준으로 time slot를 할당 받는다. 시뮬레이션에서 가변적인 요소들로는 네트워크 상황 또는 시스템에 대한 환경적인 요소이다. 이와 관련된 파라미터들은 실제적으로 사용되고 있는 상용, 범용 시스템의 일반적인 수치를 기준으로 지정된다. 이러한 가변적인 파라미터들은 일정한 범위내로 제한되지만 그룹핑 시스템 상에서 사용자가 임의로 입력 가능하다. 입력된 데이터들을 기반으로 시뮬레이션이 실행되며 파라미터의 값을 변화시키는 것으로 특정 모바일 환경을 반영하는 것이 가능하다.

<표 1> 시뮬레이션 파라미터

파라미터	시뮬레이션 값
모바일 데이터베이스에서 IA로 메시지 전송 시간	20 time slot
IA에서 권한 확인 시간	5 time slot
IA에서 JAMA로 메시지 전송 시간	15 time slot
JAMA에서 IA로 메시지 전송 시간	10 time slot
IA에서 JA 확인 시간	5 time slot
JAMA에서 요청한 모바일 데이터베이스로 메시지 전송 시간	10 time slot
IA에서 요청한 모바일 데이터베이스로 메시지 전송 시간	25 time slot
네트워크 에러율	10~30%
권한 보유율	(1/Group)×100
시스템 에러율	0~15%
공유영역 생성 비율	그룹핑 시스템 내에서 계산
그룹 멤버의 수	그룹핑 시스템 내에서 계산

4.3 시뮬레이션 시스템 구현

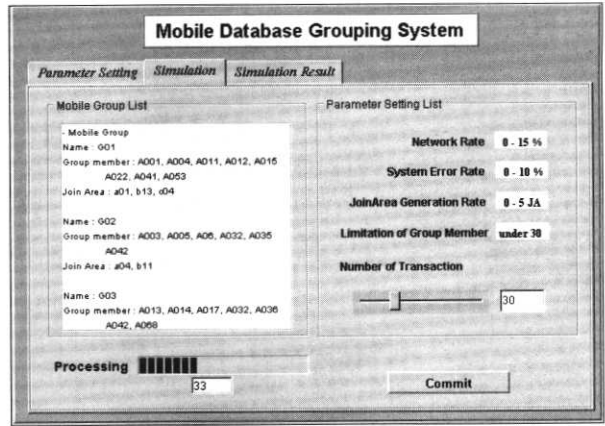
이 논문에서 실제적인 구현은 그룹핑 시스템이다. 모바일 환경에 관련된 부분은 그룹핑 시스템에서 그룹핑에 사용할 입력 데이터로써 시뮬레이션을 통해서 이루어진다. 시뮬레이션 시스템은 window2000 환경에서 JBuilder 3.0를 사용하여 구현되었다. 초기 입력 파일에서 모바일 데이터베이스와 모바일 환경에 대한 초기값을 입력받는다. 입력파일에는 각 모바일 데이터베이스가 필요로 하는 중앙 데이터베이스의 번호가 저장되고 이를 기준으로 시스템은 그룹핑을 실시하여 초기 그룹을 설정한다. 설정된 그룹과 공유영역을 바탕으로 시스템 내부에서 시뮬레이션을 실시하고 시뮬레이션 결과를 이용하여 시스템에서는 좀 더 효율적인 모바일 데이터베이스 관리를 실시 할 수 있는 그룹을 재 설정한다. 시스템의 효율성 측정을 위해서 전번 시뮬레이션에서 사용한 시뮬레이션 파라미터 유지하거나 다른 환경에 적용시키기 위해서 새로운 입력 파일을 입력받거나 시뮬레이션 파라미터를 재 설정하는 것도 가능하다.



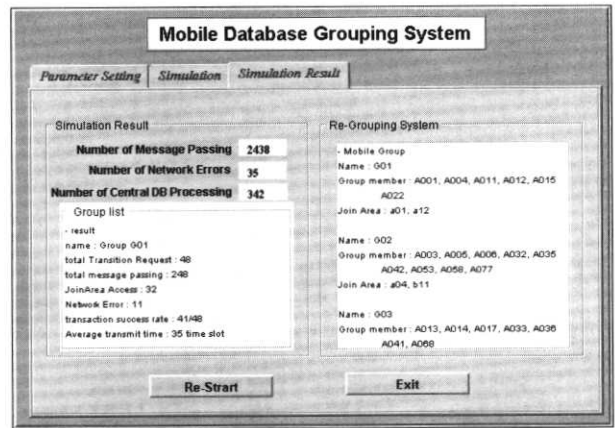
(그림 9) 파라미터 설정 인터페이스

(그림 9)는 시스템의 파라미터 입력 부분이다. 사용자가 임의로 제작한 입력 파일에는 모바일 데이터베이스와 각각의 모바일 데이터베이스가 필요로 하는 중앙 데이터베이스의 부분, 그리고 시뮬레이션에 필요한 파라미터가 지정되어 있다. 입력 파일과는 별도로 네트워크나 시스템, 그리고 공유영역에 대한 시뮬레이션 파라미터를 지정할 수 있다. 파라미터에 대한 설정이 끝나면 시스템은 입력된 모바일 데이터베이스에 대한 그룹을 설정한 뒤 시뮬레이션 부분으로 이동한다.

(그림 10)의 시뮬레이션 부분에서는 주어진 환경에서 모바일 환경에 대한 시뮬레이션을 실시한다. 화면 좌단에는 시스템에서 처리된 그룹에 대한 정보가 출력된다. 출력된 정보는 파일로 저장되어 시뮬레이션 후 효율성 측정에 사용 가능하다. 우측에는 시뮬레이션 파라미터와 시뮬레이션에서 실행될 총 트랜잭션의 수를 지정할 수 있다. 설정이 완료되면 시스템은 설정된 파라미터에 따라 시뮬레이션을 수행한다.



(그림 10) 시뮬레이션 인터페이스



(그림 11) 시뮬레이션 결과 인터페이스

(그림 11)은 시뮬레이션 결과에 대한 부분이다. 시뮬레이션이 완료되면 시스템 효율성 측정에 사용할 수 있는 전체 메시지 전송량과 네트워크 에러율, 중앙 데이터베이스의 집중도와 각 그룹에 대한 세부 정보를 출력한다. 화면 우측은 시뮬레이션 결과를 바탕으로 새로운 그룹을 설정한 결과이다. 그룹 재 설정의 영향을 미치는 요소는 다음과 같다.

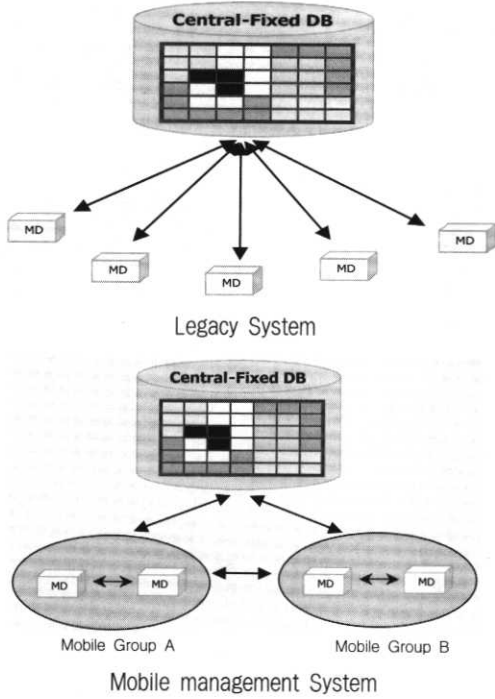
- 트랜잭션 요청 횟수/메시지 전송 횟수
- 네트워크 상태
- 트랜잭션에 소요된 평균 시간
- 공유영역 접근 횟수
- 트랜잭션 성공률

각 요소에 따라 그룹의 모바일 데이터베이스를 조절하거나 그룹을 분리한다. 공유영역의 경우는 접속 횟수를 최소화 할 수 있도록 그룹의 멤버를 조절한다.

4.4 시뮬레이션 결과 분석

중앙 데이터베이스의 부하를 줄이는 것이 모바일 데이터베이스 관리 시스템의 가장 중요한 기능중 하나이다. 현재 제품화되어 있는 모바일 데이터베이스들은 모바일 데이터

베이스와 중앙 데이터베이스 사이에 하나의 일관성 유지 서버를 위치시킨다. 또한 각 모바일 데이터베이스들 사이의 동기화를 위해서는 모바일 데이터베이스 양단간 각각 일관성 유지 서버를 두는 방법과 중앙의 일관성 유지 서버가 전체를 관리하는 방식을 취한다. 이러한 방법은 중앙서버에 대한 부하 분산 정책 부재와 일관성 유지 서버의 추가적인 비용 발생 문제를 초래했다. 이를 해결하기 위해서 모바일 데이터베이스 관리 시스템은 그룹핑 정책을 수행한다.

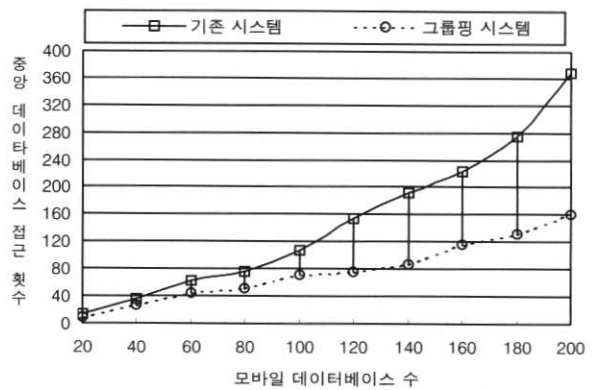


(그림 12) 기존의 시스템과 모바일 데이터베이스 관리 시스템 비교

(그림 12)는 모바일 데이터베이스 관리 및 상호 데이터 일관성 유지를 위한 기존의 중앙 집중형 관리 시스템과 모바일 데이터베이스 관리 시스템의 시스템 개요도이다. 각각의 모바일 데이터베이스가 중앙의 서버에 접근하는 기존의 방식과 달리 모바일 데이터베이스 관리 시스템은 하나의 그룹에서 그룹을 대표하는 하나의 모바일 데이터베이스만이 중앙 서버에 접근한다.

(그림 13)은 기존 시스템과 모바일 데이터베이스 관리 시스템의 중앙 데이터베이스 접근 횟수를 시뮬레이션한 결과이다. 기존의 중앙 집중형 시스템과는 다르게 그룹핑 시스템은 중앙 데이터베이스에서 그룹핑을 관리해야 할 필요가 있다. 이에 따라 그룹에 포함되는 모바일 데이터베이스의 수가 많아질수록 그룹핑 관리에 필요로 하는 중앙 데이터베이스에 대한 접근이 추가적으로 발생한다. 또한 그룹핑 시스템에서 그룹에 속하는 모바일 데이터베이스의 수가 증가할수록 그룹안에서 수용해야하는 공유영역은 증가한다. 공유영역과 그룹에 포함된 모바일 데이터베이스의 수는 중

양 데이터베이스 접근에 있어서 모순되는 관계(trade-off)를 가진다. 즉 그룹 안에 포함된 모바일 데이터베이스의 수가 커질수록 중앙 데이터베이스에 대한 접근은 줄어들지만 포함되는 공유영역은 커진다. 공유영역이 커질수록 공유영역 관리를 위한 중앙 데이터베이스 접근은 많아진다. 따라서 (그림 13)의 시뮬레이션은 이러한 상황을 고려하여 무선 네트워크 환경에서 일반적으로 적용되는 네트워크 에러율 10~30%, 시스템 에러율 0~10%, 그리고 한 그룹에 속한 최대 모바일 데이터베이스의 수는 15개 이하로 제한한 상황에서 실시하였다. 각 시뮬레이션 요소들은 두 시스템에 동일하게 적용되었고 3번 시뮬레이션 한 결과의 평균값이다. 기존의 시스템은 모바일 데이터베이스의 수가 증가할수록 접근 횟수가 급격히 증가함을 볼 수 있다. 모바일 데이터베이스 관리 시스템은 그룹의 수에 변화가 없을 때 약간의 증가 또는 현상을 유지한다. 기존의 시스템에서 중앙 데이터베이스의 접속 횟수는 모바일 데이터베이스의 수보다는 그룹의 수에 직접적으로 영향을 받는다. 그러나 접속횟수의 증가는 기존의 시스템에 비하여 극히 작은 수치를 나타낸다. 모바일 데이터베이스의 수가 증가할수록 두 시스템간의 격차가 커지는 것으로 그룹핑이 중앙 데이터베이스 집중 현상을 줄일 수 있다는 것을 알 수 있다. 또한 시스템 에러율이 클수록 고장 허용을 위한 추가적인 동작의 발생이 발생한다. 기존 시스템의 경우 고장 허용을 위한 재전송은 자신이 관리하는 모든 모바일 데이터베이스에 대하여 이루어 지지만 모바일 데이터베이스 관리 시스템은 에러가 발생한 그룹에 대해서만 재전송이 이루어지므로 시스템이나 네트워크 에러율이 클수록 기존 시스템과 모바일 데이터베이스 관리 시스템의 격차가 더 커지는 것을 알 수 있다.



(그림 13) 기존 시스템과 그룹핑 시스템 시뮬레이션 결과 분석

<표 2>는 초기 입력 파일을 이용한 그룹핑 환경에서의 시뮬레이션 결과와 초기 환경의 결과를 적용하여 시스템에서 다시 그룹핑한 시스템 조정 환경에서의 시뮬레이션 결과를 비교한 것이다. 시뮬레이션은 총 100개의 모바일 데이터베이스, 200번의 트랜잭션을 발생시켰고 네트워크 에러율

은 15%, 시스템 에러율은 5%, 마지막으로 그룹당 모바일 데이터베이스의 제한은 20개로 설정하였다. 표에서 보여지듯이 시스템을 조정한 후에 총 전송 횟수와 트랜잭션 처리 시간이 짧아 졌음을 알 수 있다. 트랜잭션당 전송횟수의 감소로 동일한 조건에서 시스템 전체의 일관성 유지를 위한 전송이 감소를 알 수 있다. 초기 설정된 공유 영역에서의 과잉 접근 때문에 시스템 효율이 떨어지는 것을 방지하기 위하여 공유 영역 감소를 목적으로 그룹이 분할되었다.

〈표 2〉 그룹핑 시스템 적용 결과

구 분	초 기 환 경	시스템 조정 환경
그룹수	13	15
평균공유영역수	3.3	2.5
총 전송 횟수	2218	1840
평균 트랜잭션 처리시간	87 time slot	61 time slot
중앙 데이터베이스 접근 횟수	211	213
트랜잭션당 전송횟수	11.09	9.2

위의 분석 결과로 그룹핑 시스템에서 네트워크 상황에 맞도록 그룹을 재 설정하여 전체 시스템의 효율을 높인다는 것을 알 수 있었다. 또한 네트워크 환경을 변화하고 동일한 시뮬레이션을 반복하면 시뮬레이션 환경에 따라 다른 방식으로 그룹핑 함으로서 시스템의 효율을 높인다.

5. 결론 및 향후 연구

본 연구의 목적은 기존의 중앙 집중형 모바일 데이터베이스 시스템의 중앙 집중화 동기화와 일관성 유지, 고장허용 능력 부재 등 기존 모바일 데이터베이스 시스템의 문제점을 해결하고 모바일 데이터베이스를 신뢰성 있고 효율적으로 관리할 수 있는 시스템을 제안하는 것이다. 이 관리 시스템은 각 모바일 데이터베이스가 요청하는 정적 데이터베이스의 부분 공유 데이터를 바탕으로한 논리적인 그룹핑을 통하여 중앙 데이터베이스 시스템의 부하를 분산하고 중앙 집중적 동기화 서버의 미사용과 함께 기존의 모바일 데이터베이스 시스템에서 제공하지 않는 고장허용 관리 기능을 제공함으로써 좀 더 신뢰성 있는 서비스를 제공한다. 또한 각 모바일 그룹별로 분리된 관리를 실시함으로써 정적 데이터베이스에 대한 데이터 백업을 쉽게 제공하도록 하였다. 그리고 제안된 시스템의 그룹핑 시스템과 모바일 환경 시뮬레이션 툴을 개발하여 제안된 시스템을 실제의 상황에 적용하였다. 시뮬레이션의 결과로 대규모 모바일 데이터베이스 환경에서 제안된 시스템의 효율성과 그룹핑 시스템을 사용한 재 그룹핑을 통해 모바일 데이터베이스 시스템의 성능 향상을 확인하였다.

향후 연구과제로는 모바일 그룹 안에서 이루어지는 일관성 유지에 대한 보다 세밀한 정책수립 및 전체 시스템의

구현 및 모바일 그룹간, 정적 데이터베이스와 모바일 그룹간의 데이터 교환에 대한 연구가 필요하다.

참 고 문 헌

- [1] E. Pitoura and B. Bhargava, "Maintaining Consistency of Data in Mobile Distributed Environments," the 15th International conference on Distributed Computing System, 1995.
- [2] Hartmut Vogler, Alejandro Buchmann, "Using Multiple Mobile Agents for Distributed Transactions," Proceedings of the Third International Conference of Cooperative Information Systems, Copyright(c), 1998.
- [3] E. Jul, H. Levy, N. Hutchinson, "Fine-grained mobility in the Emerald system," ACM Trans. Comput. Sys., 1998.
- [4] M. Powell, B. Miller, "Process Migration in DEMOS/MO," Proceedings of the Ninth ACM Symposium on Operating Systems Principles, 1983.
- [5] J. Stamos, D. Gifford, "Remote evaluation," ACM Trans. Comput. sys., 1990.
- [6] B. Goldschmidt, Z. Laszlo, "Mobile Agents in a Distributed Heterogeneous Database System," Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, 2002.
- [7] C. Chen, J. Huo, Z. Mai, "JMSAS : A Mobile Software Agent System," Proceedings of the 31st International Conference on Technology of Object-Oriented Language and Systems, 1998.
- [8] E. Weippl and J. Altmann, "Mobile Database Agents for Building Data Warehouses," IEEE 11th International Workshop, 2000.
- [9] M. Conti, S. Giordano, "Mobile Computing and Wireless Networks Mobile Ad-hoc Networking," Proceedings of the 34th Annual Hawaii International Conference on System Sciences, 2001.
- [10] L. Xiang-Yang, W. Peng-Jun, "Constructing minimum energy mobile wireless networks," ACM SIGMOBILE Mobile Computing and Communications Review, 2001.
- [11] S. Bandyopadhyay, K. Paul, "Evaluating the performance of mobile agent-based message communication among mobile hosts in large ad hoc wireless network," Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems, 1999.
- [12] L. SangKeun, H. Chong-Sun, Y. HeongChang, "Supporting transactional cache consistency in mobile database systems," Proceedings of the ACM international workshop on Data engineering for wireless and mobile access, 1999.
- [13] Y. Wai Gen, J. Donahoo, E. Omiecinski, "Scaling replica maintenance in intermittently synchronized mobile databases," ACM Press, 2001.

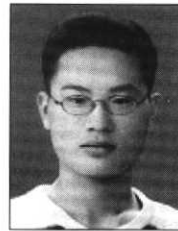
- [14] Walborn, G. D, Chrysanthis, P. K, "Supporting semantics-based transaction processing in mobile database applications," Proceedings of the 14TH Symposium on Reliable Distributed Systems, 1995.
- [15] Shirish H. Phatak, B. R. Badrinath, "Multiversion Reconciliation for Mobile Databases," 15th International Conference on Data Engineering, 1999.
- [16] H. Wen-Chi, S. Meng, Z. Hongyan, W. Hong, "An optimal construction of invalidation reports for mobile databases," Proceedings of the tenth international conference on Information and knowledge management, 2001.
- [17] H. JoAnne and A. Divyakant, "Planned Disconnections for Mobile Database," Proceedings of IEEE 11th international workshop, 2000.
- [18] V. Leong, S. Antonio, "On adaptive caching in mobile databases," Proceedings of the ACM symposium, 1997.
- [19] N. Badache, M. Hurfin, R. Macedo, "Solving the Consensus Problem in a Mobile Environment," IRISA, November, 1997.
- [20] P. Stefan and S. Andre, "Modeling Fault-Tolerant Mobile Agent Execution as a Sequence of Agreement Problem," IEEE 19th Symposium, 2000.
- [21] J. Kumar and M. Satyanarayanan, "Disconnected Operation in the Coda file system," ACM transactions on Computer system, 10, 1992.
- [22] J. Gray and A. reuter, "Transaction Processing : Concept and Techinques," Morgan Kaufman publisher, 1993.
- [23] A. Zaslavsky, L. Yeo, S. Lai, B. Mitelman, "Petri Nets Analysis of Transaction and Submitter Management Protocols in Mobile Distributed Computing Environment," Proceedings of the 4th International Conference on Computer Communications and Networks (ICCCN '95).



신 성 욱

e-mail : stnoble@software.korea.ac.kr
 2001년 고려대학교 컴퓨터학과(학사)
 2001년 고려대학교 대학원 컴퓨터학과
 (석사)
 2003년~현재 고려대학교 대학원 컴퓨터
 학과(박사)

관심분야 : 데이터베이스, 에이전트, 정보통합 등



정 동 원

e-mail : withimp@software.korea.ac.kr
 1997년 군산대학교 전산과(학사)
 1998년 한국전자통신연구원(위촉 연구원)
 1999년 충북대학교 전산과(석사)
 1999년 ICU 부설 한국정보통신교육원
 전임강사

2000년 (주)지구넷 선임연구원
 2000년~현재 고려대학교 대학원 컴퓨터학과 박사과정
 2001년~현재 (주)라임미디어 테크놀로지 연구원
 2002년~현재 TTA TC 08 SG 02 특별위원
 관심분야 : 분산 컴퓨팅, 이동 에이전트 시스템 성능 분석, 에이
 전트 모델링, 메타데이터 레지스트리 기반의 정보
 통합 방법론, 정보 표준화 등



이 정 욱

e-mail : ljo@kku.ac.kr
 1992년 고려대학교 컴퓨터학과(학사)
 1994년 고려대학교 대학원 컴퓨터학과
 (석사)
 2001년 고려대학교 대학원 컴퓨터학과
 (박사)

2002년~현재 건국대학교 자연과학대학 컴퓨터·응용과학부 조
 교수
 관심분야 : 정보통합, 인공지능, 데이터베이스, 정보검색 등



백 두 권

e-mail : baik@software.korea.ac.kr
 1974년 고려대학교 수학과(학사)
 1977년 고려대학교 대학원 산업공학과
 (석사)
 1983년 Wayne State Univ. 전산학 석사
 1985년 Wayne State Univ. 전산학 박사

1986년~현재 고려대학교 컴퓨터학과 교수
 1989년~현재 한국정보과학회 이사·평의원
 1991년~현재 한국시물레이션학회 이사·부회장·감사
 1991년~현재 ISO/IEC JTC1/SC32 국내위원회 위원장
 2002년~현재 고려대학교 정보통신대학 학장
 관심분야 : 데이터베이스, 소프트웨어공학, 시물레이션, 메타
 데이터, 정보통합, 정보보호 등