이동통신 HLR 시스템에서의 효과적인 색인 및 백업 기법

(Effective Index and Backup Techniques for HLR System in Mobile Networks)

김 장 환 † 이 충 세 ††
(Jang-Hwan Kim) (Chung-Sei Rhee)

요 약 HLR system은 이동전화 망에서 지속적으로 변하는 개별 가입자의 위치 정보를 관리한다. 이를 수행하기 위해, HLR database system은 table 관리 기능과 색인 관리 기능, 그리고 백업 관리 기능을 제공한다.

본 논문에서는, 이동 전화 번호(MDN:Mobile Directory Number)를 위한 적절한 색인 기법으로서 이단계 색인 기법의 사용과, 단말번호(ESN:Electronic Serial Number)를 위한 버켓 연결 해쉉 기법을 제안한다. 이동 전화 번호(MDN)와 단말번호(ESN)는 HLR database system에서 key로 사용된다. 또한 HLR database transaction의 특성을 고려한 효율적인 백업 방법을 제안한다. 이단계 색인 기법은 기존의 T 트리 색인 기법보다 검색 속도와 기억 공간 사용 효율 측면에서 우수하다. 버켓 연결 해쉉 기법은 기존의 변형된 선형 해쉉 기법보다 삽입과 삭제 시의 오버헤드가 적다. 제안한 백업 방법에서는, 빈번한 위치 등록 기능 수행으로 인해 야기되는 성능 저하 문제를 해결하기 위해 두가지 종류의 갱신 플래그를 사용하였다. 100만 가입자 수용시, 제안 기법을 사용하게 되면 기존 기법보다 메모리 사용량 절감(62% 이상), 디렉토리 증가 작업(25만 번 이상)제거, 백업 작업 감소(80% 이상)를 제공받게 된다.

키워드 : 이단계 색인 기법, 버켓 연결 해쉉 기법, 갱신 플래그

Abstract A Home Location Register(HLR) database system manages each subscriber's location information, which continuously changes in a cellular network. For this purpose, the HLR database system provides table management, index management, and backup management facilities.

In this thesis, we propose using a two-level index method for the mobile directory number(MDN) as a suitable method and a chained bucket hashing method for the electronic serial number(ESN). Both the MDN and the ESN are used as keys in the HLR database system. We also propose an efficient backup method that takes into account the characteristics of HLR database transactions.

The retrieval speed and the memory usage of the two-level index method are better than those of the T-tree index method. The insertion and deletion overhead of the chained bucket hashing method is less than that of the modified linear hashing method. In the proposed backup method, we use two kinds of dirty flags in order to solve the performance degradation problem caused by frequent registration-location operations.

For a million subscribers, proposed techniques support reduction of memory size(more than 62%), directory operations(250,000 times), and backup operations(more than 80%) compared with current techniques.

Key words : two-level index method, chained bucket hashing method, dirty flag

† 정 회 원 : 대덕대 IT계열 교수
jhkim@mail.ddc.ac.kr
†† 종신회원 : 충북대학교 컴퓨터과학과 교수
csrhee@cbucc.chungbuk.ac.kr
논문접수 : 2002년 5월 13일
심사완료 : 2002년 10월 23일

1. 서 론

이동 통신 망에서는 기존의 고정망에서와는 달리 가입자의 위치가 지속적으로 변하므로 이에 대한 위치 정보(Location Information)의 관리가 필요하다. HLR(Home

Location Register) 시스템은 이러한 가입자의 위치 정보 관리를 주요 기능으로 하여 현재 이동통신서비스 분야에서 호처리 및 부가서비스 지원 등의 중요한 역할을 하고 있다. 이러한 HLR 시스템은 향후 IMT2000(IMT: International Mobile Telecommunication) 및 위성이동통신 등의 분야로 그 기능이 확장되어 사용될 것으로 예상된다.

반도체 산업의 발달과 더불어 주기억장치의 용량이 커지고 가격이 많이 하락함에 따라서 데이타베이스 분야에서는 주기억장치를 최대로 활용하여 데이타베이스 시스템의 성능을 개선하는 주기억장치 데이타베이스 시스템(main memory database system)에 관한 연구가 활발히 진행되고 있다[1][2][3][4]. 주기억장치 데이타베이스 시스템은 디스크 기반 데이타베이스 시스템(disk resident database system)과는 달리 데이타베이스 전체가 주기억장치에 상주하고 있어서 데이타에 대한 신속한 액세스를 제공한다. 따라서 위치 정보의 변경 및 접근에 대한 실시간 처리와 대용량의 가입자 정보 관리를 목적으로 하는 HLR 시스템에서는 이러한 주기억장치 데이타베이스 시스템을 사용한다. 그러므로 향후 HLR 시스템에서 보다 많은 수의 가입자 수용 및 실시간 처리를 위해서 필수적으로 개선되어야 하는 부분은 HLR 시스템의 주기억장치 데이타베이스 시스템이다.

HLR 시스템에서 사용하는 주기억장치 데이타베이스 시스템은 이동 통신 망의 특성상 일반적인 주기억장치 데이타베이스 시스템과는 다른 다음과 같은 차이가 있다. 첫번째는 신속한 접근을 위하여 색인을 구성할 필요가 있는 키(key)가 가입자 정보 접근을 위한 이동 전화번호(MDN:Mobile Directory Number)와 단말번호(ESN:Electronic Serial Number)로 제한된다는 점이다. 두 번째는 시스템의 오류로 인하여 HLR 시스템에서 관리하는 가입자의 위치 정보가 손실되더라도 기존의 회복기법과 같은 데이타베이스 시스템의 도움없이 복구가 가능하다는 점이다. 이 외에도 부가서비스의 증가에 따른 동적인 스키마 변경의 필요성, 데이타베이스간의 가입자 이동(migration), 특수 부가서비스에 대한 지원 등이 있다. 따라서 이러한 특성이 HLR 데이타베이스 시스템에 반영될 필요가 있다.

본 논문에서는 위와 같은 HLR 시스템의 특성 중에서도 키의 사용 및 조건에 따른 백업의 필요성과 현재의 HLR 데이타베이스 시스템을 분석한 결과 발견한 문제점을 바탕으로 보다 나은 시스템 개발을 위해서 기존의 색인 방법 중에서 실용적으로 적용 가능한 빠른 색인 방법과 HLR 시스템의 성능을 고려한 효과적인 백업 방법을 제시한다.

사용을 제안하는 색인 방법은 기존의 HLR 데이타베이스 시스템에서 고려하지 않았던 제한된 키의 사용이라는 특징을 바탕으로 각 키의 특성을 고려하여 전화번호 및 단말번호를 위하여 각각 이 단계 색인을 응용한 색인 방법과 버켓 연결 해쉉(CBH:Chained Bucket Hashing)[5] 기법이다. 그리고 백업 방법으로는 HLR 데이타베이스에서 발생할 수 있는 트랜잭션의 종류를 구분하여 두 가지의 갱신 플래그(dirty flag)를 관리하여 백업을 수행하는 방법을 제시한다. 100만 가입자 수용의 경우,기존 기법에서의 전화번호 인덱스용 메모리양이 약 16M Byte였으나, 제안 기법은 약 5~6Mbyte로 처리할 수 있으며, 기존 기법에서의 전화번호 검색은 속도면에서 약 20회 이상의 비교 연산이 필요하나,제안 기법은 2회의 메모리 접근으로 처리할 수 있으며, 기존 기법에서의 단말 번호 초기 해쉬 구조 구성시 약 25만 번의 디렉토리 증가 작업이 과부하로 작용하였으나, 제안 기법은 디렉토리 증가 작업이 없으며, 기존 기법에서는 초당 약 800번 이상 발생하는 위치 등록으로 인한 실시간 백업이 과부하로 작용하였으나, 제안 기법은 위치 등록으로 인한 백업은 실시간으로 할 필요가 없게 되어 약 80%의 실시간 백업으로 인한 부하를 감소시킬 수 있는 효과가 있다.

본 논문의 구성은 다음과 같다. 제Ⅱ장에서는 HLR 시스템 및 HLR 데이타베이스 시스템에 대하여 소개한다. 제Ⅲ장에서는 기존의 색인 방법과 제안하는 색인 방법에 대하여 설명한다. 제Ⅳ장에서는 성능 향상을 고려한 효과적인 백업에 대하여 기술한다. 제Ⅴ장에서는 결론과 함께 앞으로의 IMT2000 service와 보안 기능 추가에 관련한 HLR 시스템의 개선 방향을 제시한다.

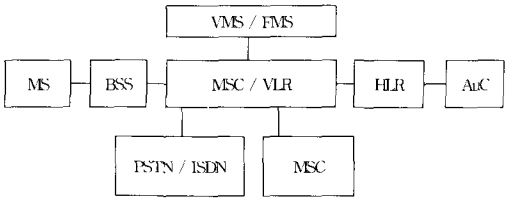
2. HLR 시스템 소개

2.1 HLR 시스템

HLR 시스템은 MSC/VLR(Mobile Switching Center, Visitor Location Register)과 연결되며, 그 주된 기능은 가입자에 대한 위치 정보의 관리를 통한 호처리 지원과 각 가입자별 부가서비스 정보의 관리 및 지원 등을 실시간으로 처리하는 것으로서 현재 AMPS(Advanced Mobile Phone System, 아날로그 방식) 및 CDMA(Code Division Multiple Access, 디지털 방식) 등 셀룰러 망(Cellular Network)에서 사용되고 있다.

그림 1은 HLR 시스템과 관련된 이동 통신 망의 구성요소를 나타낸다[6]. 그림에서 HLR 시스템은 MSC/VLR과 공통선 신호방식(CCS7:Common Channel Signalling

No.7)에 의하여 연결되는 MSC/VLR로부터 가입자에 대한 위치 정보의 갱신 및 조회 요구 등에 대한 처리 기능을 제공하며, 가입자 정보 변경의 경우에는



MS	Mobile Station	FMS	Fax Mail System
BSS	Base Station System	HLR	Home Location Register
MSC	Mobile Switching Center	AuC	Authentication Center
VLR	Visitor Location Register	PSTN	Public Switched Telephony Network
VMS	Voice Mail System	ISDN	Integrated Service Digital Network

그림 1 이동 통신 망에서의 HLR 시스템

이를 MSC/VLR로 통보하는 기능도 수행한다. 또한 MSC/VLR과 인증센터(AuC:Authentication Center)간의 메시지를 전달하는 기능도 포함하고, 경우에 따라서 MSC/VLR을 통하여 VMS/FMS와 같은 부가서비스 시스템과 연동하기도 한다.

일반적으로 HLR 시스템을 구성하는 하드웨어 요소는 그림 2와 같이 크게 세 가지 부분으로 나눌 수 있다. 먼저 주제어 컴퓨터(MC:Main Computer)는 가입자 정보관리, 공통선 신호방식 및 이동 통신 응용부(MAP:Mobile Application Part)에 대한 처리 등 HLR 시스템의 주요 기능을 수행한다. 주제어 컴퓨터는 그 중요성을 고려하여 CPU 보드, 메모리 보드 등 모든 하드웨어 유니트가 이중화 혹은 다중화된 장애 허용 컴퓨터(Fault-Tolerant Computer)를 사용한다. 다음으로 MUX는 MSC/VLR과의 E1/T1 접속을 지원하는 기능을 수행하며, OMS (Operation and Maintenance Subsystem)는 주제어 컴퓨터 및 MUX와 연결되어 운용자가 HLR 시스템을 보다 편리하게 관리하고 사용할 수 있는 기능을 제공한다.

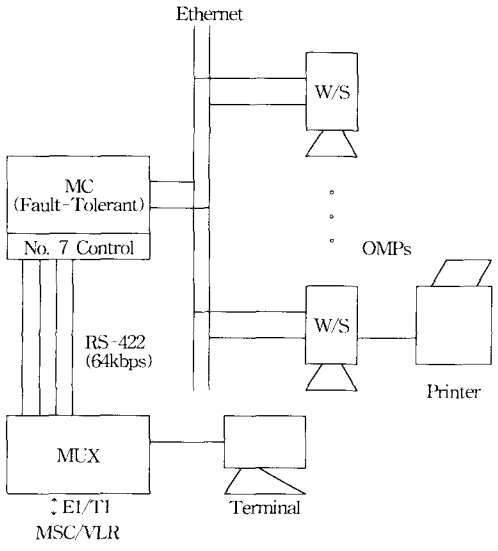


그림 2 HLR 시스템 하드웨어 구조

HLR 시스템의 소프트웨어는 MSC/VLR과의 연동을 위한 부분으로 프로토콜의 하위 계층인 메시지 전달부(MTP:Message Transfer Part)와 신호연결 제어부(SCCP:Signalling Connection Control Part), 상위계층인 문답처리 응용부(TCAP:Transaction Capabilities Application Part)와 이동 통신 응용부를 처리하는 응용서비스요소 처리부(ASE:Application Service Entities)가 있다. 그리고 가입자 정보에 대한 추가, 삭제, 검색, 갱신 등의 기능을 제공하는 데이타베이스 시스템(Data base System), 장애 및 상태 그리고 통계 등을 처리하는 운용유지 보수부(Operation and Maintenance Part), 그리고 HLR 시스템의 편리한 사용을 위한 운용자 관리부(User Interface Part)등으로 구성되어 있다. 다음의 그림 3은 HLR 시스템의 소프트웨어 구성 요소를 나타낸다.

그림 3에서 데이타베이스 시스템은 HLR 시스템의 주된 기능인 위치등록, 호처리 및 부가서비스 정보의 관리와 직접적으로 관련이 되는 중요한 부분이다. 이러한 HLR 데이타베이스 시스템의 특성은 실시간성(real-time)을 요구하므로 이를 위하여 주기억장치 데이타베이스 시스템을 사용한다. 현재 이러한 데이타베이스 시스템과 관련된 실시간 데이타베이스 시스템[7]에 대한

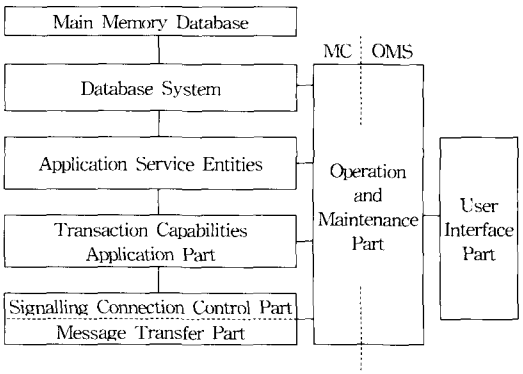


그림 3 HLR 시스템 소프트웨어 구조

연구가 활발히 진행되고 있다.

HLR 데이타베이스 시스템에서 제공하는 기능에는 삽입, 삭제, 갱신 등의 가입자 정보의 관리를 위한 테이블 관리 기능, 가입자 정보에 빠르게 접근하기 위한 인덱스 관리 기능, 주기억장치의 휘발성(volatile)을 고려한 디스크로의 주기적인 백업 기능 등이 있다. 다음 절에서 이러한 HLR 데이타베이스 시스템의 기능과 특징에 대하여 자세히 설명한다.

2.2 HLR 데이타베이스 시스템

HLR 데이타베이스 시스템에서 관리하는 정보는 가입자 정보, 국번 정보, 도난 가입자(stolen subscriber) 정보 등으로 그 종류가 매우 제한되어 있다. 이중 가입자 정보는 우선 순위, 과금 종류 등 기본 정보와 가입자가 위치한 장소를 나타내는 위치 정보, 각종 부가서비스 정보 등을 포함한다. 가입자 정보를 제외한 나머지 정보는 자주 쓰이지 않으며, 빈번하게 발생하는 작업(operation)은 가입자 정보 중에서도 위치 정보에 대한 변경 및 접근이다. 스키마 측면에서는 가입자 정보 중에서 부가서비스 정보를 제외한 대부분의 정보가 초기에 설계된 스키마에서 변화되지 않는다.

현재 HLR 데이타베이스 시스템에서는 위에서 언급한 정보를 바탕으로 가입자 테이블, 국번 테이블, 도난 가입자 테이블이 있으며 이들 테이블은 일차원 배열(array) 형태로 UNIX에서 지원하는 공유 메모리(shared memory)에서 관리된다. 따라서 각 튜플은 고정길이(fixed length)를 갖는 속성(attribute)으로 구성되는 자료 구조(data structure)로 표현되고, 튜플 식별자(tuple identifier)는 배열의 색인 번호(index number)로 나타나므로 응용 프로그램(application program)에서는 정해진 자료 구조와 테이블 번호 및 튜플 식별자를 사용하여 각 튜플에 접근한다. 이러한 테이블들이 관리되는 공유 메모리를 메모리 SLD(System Load DATA)라 하고, 메모리 SLD는 백업 방법에 의하여 디스크 SLD에 저장된다.

사용하는 테이블의 종류가 제한되어 있고 또한 이들 테이블에 대한 작업도 제한되어 있으므로 각 테이블의 튜플에 빠르게 접근하기 위한 색인 방법도 제한된 키에 대해서만 지원된다. 국번 테이블의 경우는 튜플의 수가 고정되어 있으므로 키 값이 테이블의 색인 번호로 사용되고, 도난 가입자 테이블의 경우는 실제 상황에서 튜플 수가 매우 적으므로 순차적인 색인 방법을 사용한다. 따라서 별도의 색인 방법의 사용은 튜플의 수가 수십만에서 백만개 이상을 수용해야 하는 가입자 테이블의 전화번호와 단말 번호에 국한되며, 현재 HLR 데이타베이스 시스템에서 사용하는 색인 방법은 엔진 프로세스내의 자료 구조로서 구현되어 있다.

HLR 데이타베이스를 사용하는 부분은 크게 두 가지 그룹으로 나눌 수 있다. 첫 번째는 응용서비스요소 부분의 프로세스들로서 주로 가입자의 위치 정보에 대한 갱신 및 조회 등 매우 빠른 응답을 요구하는 그룹이고, 두 번째 그룹은 가입자의 추가, 삭제 등과 관련된 응용프로세서로 응답 시간보다는 정보가 손실되지 않아야 하는 특성을 갖는 그룹이다. 주기억장치가 휘발성의 특성을 가지므로 두번째 그룹의 요구 사항을 만족하기 위해서는 각 작업에 대하여 디스크로의 백업과 오류 발생시의 회복이 필요하다. 현재 HLR 데이타베이스 시스템에서는 엔진 프로세스가 변경이 일어난 페이지에 대하여 갱신 플래그(dirty flag)를 1로 하고, 백업 프로세스가 지속적으로 갱신 플래그를 검사하여 그 값이 1인 페이지에 대하여 디스크로 백업을 수행하는 방법을 사용한다.

그림 4는 위에서 설명한 HLR 데이타베이스 시스템의 일반적인 구조를 보여준다. 그림에서 엔진 프로세스는 메모리 SLD의 관리 및 색인 구조의 관리를 통하여 각 응용 프로세스의 요구를 수행하는 기능을 하며, 백업 프로세스는 엔진 프로세스에 의하여 갱신된 메모리 SLD의 페이지를 지속적으로 디스크 SLD로 백업하는 기능을 수행한다. 그리고 응용 프로세스는 라이브러리 형태로 주어진 함수를 호출하여 엔진 프로세스에 데이타베이스 관련 요구를 전달하고 그 결과를 받으며, 데이타 생성기(data generator)는 주어진 형태의 스키마 정의를 사용하여 초기 디스크 SLD를 생성하는 기능을 수행한다.

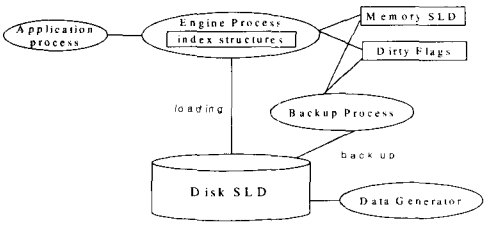


그림 4 HLR 데이타베이스 시스템 구조

3. 색인 방법에 대한 연구

3.1 데이타베이스 시스템에서의 색인 방법

색인 구조로 고려되는 많은 방법은 크게 두 가지 종류로 구분된다. 첫 번째는 키의 순서가 유지되는 AVL 트리, B-트리, T 트리 등을 들 수 있으며, 두 번째는 선형 해싱(linear hashing), 확장 해싱(extendable hashing)

등 해쉬 색인으로 랜덤 접근의 특성을 갖는다[8]. 본 절에서는 이러한 두 가지 종류의 색인 방법에 대하여 살펴본다.

색인 구조로 널리 사용되고 있는 방법은 트리 구조이다. 트리를 사용한 색인 방법의 종류에는 이진 트리(binary tree)[5]에서 트리의 깊이(depth)를 균형적으로 유지하므로 O(logN)의 트리 순회(tree traversal)로 검색이 가능한 AVL트리, 디스크 기반인 경우를 고려하여 디스크 입출력을 최소로 하며 트리의 장점을 적용한 B-트리와 B^+-트리[9][10][11]가 있으며, 주기억장치 데이타베이스 시스템을 위해서는 이진 탐색으로 AVL 트리의 빠른 검색의 특성과 갱신 및 저장 효율이 좋은 B-트리로부터 만들어진 T 트리[8]가 있다.

위에서 설명한 트리 구조의 특징은 키의 종류에 크게 관계없이 비교적 일정한 트리 순회 및 노드내의 이진 탐색으로 검색이 가능하고, 키가 순서대로 저장됨으로써 키의 순서에 따른 순차적 접근이 가능하다는 점이다. 반면에 키의 삽입이나 삭제가 일어난 경우에 트리 및 트리내의 노드 상태에 따라서 일어나는 작업의 양이 달라지므로 응답 시간이 일정하지 않은 단점이 있다.

트리와 함께 널리 사용되고 있는 색인 방법은 해쉬 기법이다. 해쉬 기법에는 정적인 구조를 지니며 충돌(collision)이 발생했을 경우에 버켓을 체인으로 연결함으로써 해결하는 버켓 연결 해쉬[5] 방법이 있고, 키의 개수가 가변적임을 고려하여 디렉토리의 변화를 가능하게 하여 동적인 구조를 사용하는 확장 해쉬 방법과 선형 해쉬[12] 방법이 있다.

해쉬 방법의 경우는 트리와는 달리 순차적인 검색이 가능하지 않으나 이론적으로 한번의 접근에 의하여 검색이 가능하고 종류에 따라서는 삽입 및 삭제 작업에 대하여 비교적 일정한 시간을 보장받을 수 있다. 그러나 키의 특성과 예상 갯수를 정확히 고려하여 해쉬 함수(hash function)를 선택하지 않을 경우, 잦은 충돌로 인하여 저장 효율이 떨어질 뿐 아니라 검색을 위하여 접근 횟수가 많아지고 삽입 및 삭제 작업시 많은 시간이 걸릴 수 있다.

3.2 기존 HLR 데이타베이스 시스템에서의 색인 방법

현재 운용(operation)되고 있는 HLR 시스템의 데이타베이스 시스템에서 전화 번호와 단말 번호를 위하여 제공되는 색인 방법은 T 트리와 변형된 선형 해쉬(modified linear hashing)방법이다. 본 절에서는 이 두 색인의 특징과 구현된 방법 그리고 문제점에 대하여 살펴보고 HLR 시스템 관점에서 두 방법을 살펴본다.

현재 구현된 T 트리와 변형된 선형 해쉬 기법에서 사용하는 메모리 영역은 데이타베이스가 아닌 엔진 프로세스 내의 자료 구조로 되어 있다. 따라서 엔진 프로세스 구동 시에 매번 새롭게 초기 색인 구조를 구성해야 한다. 이러한 구조를 사용하면 색인에 대해 별도의 백업 기능이 필요하지 않으며, 복잡한 동시성 제어 기능 등이 필요하지 않은 장점이 있다. 그러나 색인 구조의 구성을 위하여 엔진 프로세스의 초기화 과정이 매우 길어지는 문제점이 있다.

색인 방법을 평가하는데 사용하는 중요한 요소는 검색 속도, 삽입 및 삭제 시의 속도, 그리고 사용하는 공간의 양, 즉 주기억장치 데이타베이스 시스템에서의 메모리 양이다. HLR 데이타베이스 시스템에서는 제한된 주기억장치 용량을 고려해야 하므로 메모리 양을 최소로 하는 것이 중요한 요소가 되며, HLR 시스템의 특성상 전화 번호를 키로 사용한 삽입 및 삭제의 빈도수는 이를 사용한 검색 빈도수에 비하여 무시할 정도이므로 검색 속도가 중요한 평가 요소이다. 반면에 단말 번호를 키로 사용한 경우는 검색과 동일한 빈도수로 삽입 및 삭제가 일어나므로 삽입 및 삭제의 속도가 중요한 평가 요소가 된다. 따라서 본 절에서는 전화 번호를 위한 T 트리에서는 필요한 메모리 양과 검색 속도에 대하여 분석하고 단말 번호를 위한 해쉬 기법에 대해서는 메모리 양과 삽입 및 삭제 속도에 대하여 살펴본다.

3.2.1 T 트리 색인 방법

T 트리는 AVL트리와 B-트리의 장점을 결합하여 만든 트리이다. AVL 트리는 이진 탐색 트리로서 검색이 빠르며, 삽입과 삭제의 결과로 불균형(unbalance)이 발생하면 회전 알고리즘(rotation algorithm)에 의해 재균형(rebalance)을 이루도록 한다. 그러나 AVL 트리에서는 각 노드에 많은 양의 제어 정보(control data) 및 자식 및 부모 노드에 대한 포인터가 필요하므로 저장 효율이 떨어지는 단점이 있다. T 트리는 AVL 트리의 장점을 유지하면서 노드 하나에 여러 개의 키를 저장하여 저장 효율을 높이고, 삽입 및 삭제가 하나의 노드 안에서 발생하는 경우가 많으므로 AVL 트리에 비하여 재균형이 덜 발생하도록 한 것이다.

그림 5는 T 트리의 전체적인 구조와 T 노드의 구조를 나타낸다. 그림에서와 같이 T 트리는 균형된 구조를 가지며, 각 T 노드는 <키, 튜플식별자>로 구성되는 아이템들 이외에 제어 정보와 각 서브 트리의 루트 노드인 자식 노드 및 부모 노드에 대한 포인터로 구성되어 있다. 이러한 구조에서 AVL 트리의 알고리즘을 기반으로 검색, 삽입, 삭제에 대한 알고리즘이 개발되어 있다[8].

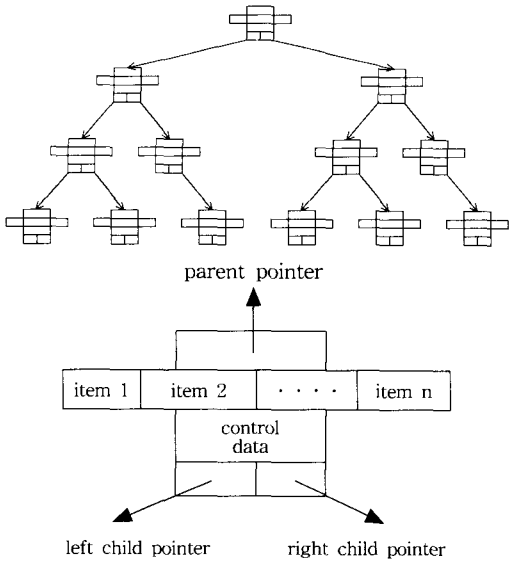


그림 5 T 트리 구조와 T 노드 구조.

현재 기존의 T 트리는 위에서 설명한 방법을 그대로 적용하였으며, 모든 T 노드가 엔진 프로세스내의 변수로서 구성되어 있다. 따라서 엔진 프로세스가 처음에 구동될 때는 주기억장치의 메모리 SLD 내용을 기반으로 반복적인 삽입 작업을 수행하여 초기 트리를 구성한다. T 노드의 할당(allocation)은 필요할 때마다 정해진 수만큼씩 이루어지며, 각 노드는 16개 혹은 32개의 아이템을 최대로 가진다.

B-트리의 저장 효율과 동일하게 생각했을 때 T 트리에서 각 T 노드의 저장 효율은 50%에서 100%가 된다. 따라서 랜덤한 분포에서 69%를 각 T 노드의 저장 효율이라 하고 전체키의 수를 N, 노드내 아이템의 수를 M, 각 노드의 크기를 S라 했을 때, 사용되는 메모리의 양은 $N \times S/(M \times 0.69)$와 같이 나타난다. 이 수식과 현재 데이타베이스 시스템에서 사용되는 노드의 크기인 176바이트 및 노드 당 16개의 아이템 수를 사용하여 *HLR* 시스템의 요구 사항인 100만 가입자 수용을 위해 필요한 T 트리의 메모리 양을 계산하면 약 16Mbyte가 나온다. 이 메모리 양은 다음 절에서 소개하는 전화번호를 위한 색인 방법에서의 메모리 양과 비교했을 때 상당히 많은 양이다.

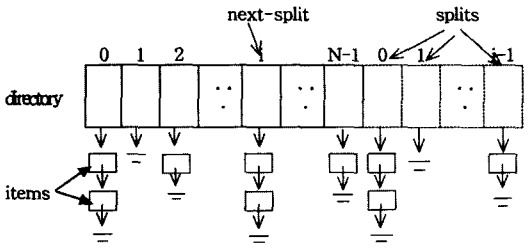
T 트리에서의 검색 속도는 트리 순회를 하는데 필요한 비교 연산과 노드내에서의 이진 탐색에 필요한 비교 연산으로 나타낼 수 있는데 이는 이진 탐색 방법과 같이 평균적으로 $\log_2 N$ 이상의 연산이 필요하다. 따라서 위와 같이 100만 가입자 수용을 예상하면 평균적으로 $\log_2 10^6$, 즉 약 20회 이상의 비교 연산이 필요하다. 이러한 비교연산 횟수는 다음절에서 소개하는 전화 번호를 위한 색인 방법에서의 비교 연산 횟수에 비하여 많음을 알 수 있다.

3.2.2 변형된 선형 해쉬 방법

변형된 선형 해쉬 방법은 디렉토리의 선형적인 증가를 특성으로 하는 해쉬 방법을 주기억장치의 특성에 맞도록 개선한 것으로 T 트리와 함께 기존 HLR 데이타베이스 시스템의 색인 방법으로 제공된다. 이미 언급한 바와 같이 해쉬에서 필요한 자료들도 엔진 프로세스내의 변수로서 구현되어 있으며, T 트리와 마차가지로 프로세스가 구동될 때 반복적인 삽입에 의하여 초기 해쉬 구조가 만들어진다.

변형된 선형 해쉬 기법은 그림 6과 같이 해쉬 함수가 디렉토리가 증가 혹은 감소함에 다라 계속 변하게 된다. 디렉토리는 총 아이템 수와 디렉토리 크기의 비가 일정 비율 이상이 되면 증가하고 관계된 아이템들이 분배되며, 일정 비율 이하가 되면 감소하고 관련된 아이템들이 통합된다. 디렉토리의 크기가 변하면 그림에서와 같이 적용되는 해쉬 값도 동적으로 변하게 된다.

기존의 *HLR* 데이타베이스 시스템에서의 해쉬 방법은 위와 같은 일반적인 방법을 그대로 적용하였으나 디렉토리 및 아이템과 관련된 메모리의 할당 및 반환을 용이하게 하기 위하여 그림 7과 같이 두 단계의 디렉토리 계층을 두었으며, 아이템을 일정한 개수씩 할당되어 풀(pool)에서 관리된다.



hash-value = h(x) mod N

if hash-value < next-split then hash-value = h(x) mod(2*N)

그림 6 변형된 선형 해쉬 기법

변형된 선형 해쉬 기법에서는 해쉬 기법의 특성에 따라 해쉬 함수가 잘 설정된 경우에 두세 번의 메모리 접

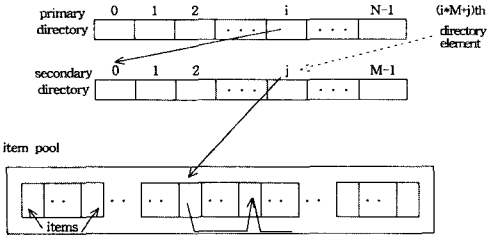


그림 7 구현된 해싱 기법

근으로 검색이 가능한 특성을 갖는다. 또한 일정한 분포로 분배된다는 가정 하에서 필요한 메모리의 양도 <키, 튜플식별자>로 구성되는 아이템의 수에 선형적으로 비례하므로 다른 방법에 비하여 그 양이 적다고 볼 수 있다. 이 해싱 기법이 적용되는 단말 번호는 가입자가 추진될 때 번호의 중복을 조사하기 위해서 사용되고 이에 따라 가입자의 추가 및 삭제시 키에 대한 삽입 및 삭제가 발생하게 된다. 따라서 주로 고려해야 하는 사항은 검색과 동일한 비율로 발생하는 삽입 및 삭제에서의 오버헤드를 들 수 있다.

변형된 선형 해싱 기법에서는 디렉토리가 선형적으로 증가하기 때문에 확장 해싱 기법에 비하여 키의 수에 따른 적당한 메모리 사용을 유지할 수는 있으나 디렉토리의 증가 및 감소에 관련된 작업이 더 많이 일어난다. 특히 엔진 프로세스가 새롭게 구동될 때마다 매번 초기 해쉬 구조를 구성해야 하므로 삽입에 따른 디렉토리의 증가가 큰 오버헤드로 작용한다.

해쉬 함수가 일정한 분포로 해쉬 값을 생성한다고 가정하고 디렉토리가 증가하는 시점에서 하나의 디렉토리 요소의 속하는 아이템의 수를 N이라 하고 삽입되는 키의 개수를 M이라 하면, 개략적으로 M/N번의 디렉토리 증가가 일어난다. 여기에 구현해서 사용한 N=4와 최종 단계의 예상 수용 가입자 수인 100만의 경우를 계산하면 약 25만 번 이상의 증가가 일어나고, 이는 엔진 프로세스가 초기 구조를 구성할 때 심각한 수준의 오버헤드로 작용한다.

3.3 HLR 데이타베이스 시스템을 위해 제안하는 색인 방법

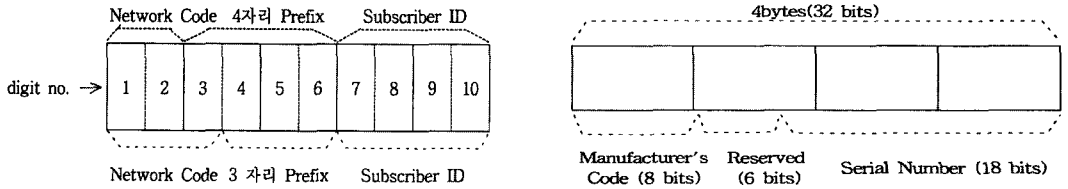
HLR 데이타베이스 시스템은 매우 특수한 목적을 갖는 시스템으로서 이미 설명한 바와 같이 색인을 지원할 필요가 있는 키가 전화 번호와 단말 번호로 제한되어 있다. 본 절에서는 우선 이들 두 속성의 쓰임과 구성 그리고 특성에 대하여 살펴보고, 이를 바탕으로 전화 번호를 위해서는 메모리 사용량과 검색 속도를 고려하여, 단말 번호를 위해서는 삽입 및 삭제시의 오버헤드를 고려하여 사용을 제한하는 색인 방법에 대하여 설명한다.

3.3.1 HLR 데이타베이스 시스템에서 키의 종류 및 특성

두 개의 속성이 키로 사용되는 목적은 다음과 같다. 먼저 전화 번호는 제 2장에서 설명한 응용서비스 요소 처리부의 프로세스들이 MSC/VLR로부터 전달된 전화 번호로 각 가입자 튜플에 대한 위치 정보의 변경 및 접견을 수행하기 위하여 사용된다. 도한 운용자에 의한 정해진 명령어 혹은 고객관리센터로부터 전달된 명령어에 의해 가입자를 추가 및 삭제할 때 사용되기도 한다. 그러나 이러한 가입자의 추가 및 삭제는 위치정보의 변경 및 접근에 따른 검색에 대하여 매우 작은 비율이라고 볼 수 있다.

단말 번호의 경우는 가입자의 신규 가입 시에 해당 가입자의 단말 번호가 중복되지 않도록 하기 위하여 사용되므로 전화 번호에 비하여 사용 빈도수가 비교적 적음을 알 수 있다. 그리고 가입자의 가입 및 해지 시에 키에 해당하는 단말 번호의 추가 및 삭제가 일어나므로 검색에 비하여 삽입 및 삭제에 따른 오버헤드가 크게 작용하는 특성을 갖는다.

전화 번호는 그림 8의 (a)와 같이 총 5Byte, 즉 열 디지트의 BCD(Binary Coded Decimal) 형태로 구성된다[6][13]. 맨 앞은 망 식별 번호로 HLR 시스템내의 모든 가입자가 동일한 값을 가진다. 다음은 국 번호(prefix number), 나머지 네 디지트는 가입자 번호(subscriber identifier)로서 해당 망내에서의 가입자를 구분하는데 사용된다. 따라서 HLR 데이타베이스 시스



(a) 전화번호의 구조 (b) 단말번호의 구조

그림 8 전화번호 및 단말번호의 구조

템에서 고려할 필요가 있는 의미 있는 부분은 망 식별 번호를 제외한 나머지이고, 가입자의 증가에 따라 국번이 네 디지트인 경우도 여덟 디지트를 넘지는 못한다.

HLR 시스템에서 저장하는 가입자의 전화 번호는 일정한 규칙에 의하여 관리되는데 이는 HLR 시스템이 특정 지역을 중심으로 가입자를 수용하므로서 나타나는 규칙이다. 망을 관리하는 서비스 사업자는 특정한 범위의 국번을 각 지역별로 할당하게 되고, 이에 따라 해당 지역의 HLR 시스템은 특정한 범위의 국번만을 수용하게 된다. 그리고 치국 계획에 따라 특정 국번의 가입자 번호가 어느 정도 사용되기 이전까지는 다른 국번을 사용하지 않는다. 따라서 특정 HLR 시스템에서 관리하는 국번의 수는 해당 시스템의 가입자 수용 능력에 따라 일정한 비율로 정해진다.

이러한 치국 계획의 특성에 의하여 100만 가입자를 수용하는 HLR 시스템의 경우 국번 사용율을 1.0으로 했을 경우 100개의 국번이 필요하며, 0.75로 했을 경우 134개, 0.5로 했을 경우는 200개의 국번 만을 수용하는 구조이면 되고, 일반적으로는 0.7 혹은 0.8 이상의 사용율이 되었을 때 다른 국번을 사용하기 시작한다. 그러므로 이러한 전화 번호의 특성을 고려하여 적합한 색인을 선택할 필요가 있다.

단말 번호의 경우는 그림 8의 (b)와 같이 여덟 비트의 생산자코드(manufacturer's code)와 열 여덟 비트의 일련 번호, 그리고 사용이 보류(reserved)된 여섯 비트로 구성되어 있다. 가입자의 증가에 따라 사용이 보류된 여섯 비트는 규격 확장에 따라 사용이 보류된 여섯 비트는 규격 확장에 따라 사용이 예상되고 있으며, 이를 고려하면 여덟 비트 이상의 생산자 코드와 열 여덟 비트 이상의 일렴 번호로 구성될 것이고 그 합은 결국 네 바이트 양의 정수(unsigned integer)안에서 표현된다.

단말 번호가 결국 지정된 범위의 양의 정수로 표현됨으로서 이를 UNIX 시스템에서 제공되는 랜덤 함수(random function)의 시드(seed)로 사용하므로서 비교적 일정한 분포의 해쉬 값을 얻을 수 있다. 따라서 이러한 특징을 바탕으로 삭제 및 삽입시의 오버헤드가 작은 해쉬 기법을 선택할 필요가 있다.

3.3.2 전화 번호를 위한 색인 방법

전화 번호는 국번과 가입자 번호로 구성되고 HLR 시스템의 특성에 따라 수용하는 가입자 수가 결정되므로 해당 HLR 시스템이 관리해야하는 국번호의 수가 제한된다. 그리고 전화 번호를 사용한 검색이 실시간에 이루어져야 할 뿐 아니라 그 값에 관계없이 매우 안전적인 검색 속도를 유지해야 한다. 이러한 특징을 고려하였을 경우 이 단계 색인 방법을 응용하여 적용할 수 있는 방법을 생각할 수 있다. 이 방법에서는 해쉬 기법에서와 같은 충돌이 없고 트리와 같이 많은 비교 연산이 필요하지 않기 때문에 매우 일정한 검색 속도를 유지할 수 있을 뿐 아니라 사용되는 메모리 양도 실용적으로 수용이 가능하다.

이 색인 방법에서 첫 번째 색인은 HLR 시스템에서 수용할 수 있는 모든 국번을 갖고, 두 번째는 각 국번별로 수용하게 되는 가입자 번호를 색인으로 갖는다. 그림 9는 이러한 방법을 보여준다. 그림에서 국번 엔트리와 가입자 번호 엔트리는 모두 고정된 길이의 속성을 가지므로 제한된 연산에 의하여 <키,튜플식별자>를 알아낼 수 있다. 그리고 가입자 번호 엔트리의 수는 수용 가능한 총 가입자 및 치국에 사용하는 국번 수용율을 사용하여 초기에 결정할 수 있다.

이 색인 방법을 사용할 경우 검색은 다음의 절차에 의하여 이루어진다. 먼저 검색을 요구한 전화 번호는 의미 있는 국번호 및 가입자 번호로 구분이 된다. 분리된 국번호는 카탈로그 정보에 의하여 지정된 위치의 주기억장치에 배열 형태로 존재하는 국번 엔트리의 색인 번호로 사용되어 해당 국번이 지정하는 가입자 엔트리의 위치를 알 수 있다. 다음에는 분리한 가입자 번호를 해당 가입자 엔트리의 색인 번호로 사용함으로서 검색이 종료된다.

이러한 매우 간단하며 일정한 연산 법칙 및 두 번의 메모리 접근으로 검색이 이루어지기 때문에 그 속도가 매우 빠를 뿐 아니라 주어진 전화 번호의 값이나 현재 수용 가입자 수에 관계없이 일정한 속도의 검색이 가능하다. 이러한 검색 속도는 제 3.2절에서 소개한 트리 방법에 의한 20회 이상의 연산 및 이에 따른 메모리 접근에 비하여 월등히 빠른 것이다. 또한 충돌에 의해 키마다 검색 속도가 달라질 수 있는 해쉬 기법보다도 효과적이라 할 수 있다.

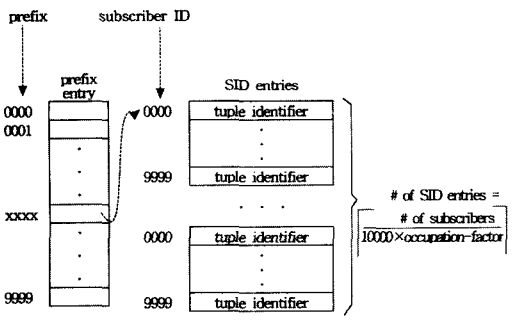


그림 9 전화 번호를 위한 색인 방법

위와 같은 색인 방법에 의하여 필요한 메모리의 양은 다음과 같이 계산될 수 있다. 먼저 국번호 엔트리를 위하여 필요한 양은 수십 Kbyte 이내로 가입자 번호 엔트리에 비하여 그 양이 매우 작으므로 무시할 수 있다. 그리고 가입자 번호 엔트리의 양은 각 가입자 번호 요소 크기와 국번당 가입자 수와 국번의 예상 수용율의 곱으로 나눈 것에 해당한다. 여기에 현재 용량인 100만 가입자, 0.7~0.8의 예상 수용율을 적용하면 125~143개의 엔트리의 수가 계산된다. 그리고 색인 번호가 곧 전화 번호인 키로 사용되므로 가입자 번호 요소의 크기는 튜플식별자인 네 바이트로 나타나며 국번당 만개의 가입자 번호 수를 적용하면 약 5~6Mbyte의 메모리 양이 필요하다. 따라서 이 단계 색인을 위하여 필요한 메모리의 양은 앞 절에서 살펴본 T 트리에 비하여 적은 양으로 구현이 가능함을 알 수 있다.

트리에 비하여 이 색인 방법이 지원할 수 없는 기능은 키의 순서에 따른 순차적인 접근(sequential access)을 생각할 수 있다. 그러나 HLR 시스템에서는 매우 제한적인 명령어만 제공하므로 이러한 일반적인 순차적인 검색 기능을 요구하지 않는다. 다만 이와 비슷한 명령어로는 국번별로 가입자 통계를 요구하는 경우가 있는데, 이는 위에서 설명한 색인 방법으로도 충분히 지원이 가능한 수준이다.

지금까지 살펴본 바와 같이 전화 번호를 사용한 색인 방법은 키의 특성을 고려했을 경우 이 단계 색인을 이용한 방법이 검색 속도면이나 메모리 사용량에서 우수함을 알 수 있다. 또한 그 구조가 매우 간단하여 구현이 간단하고 추후 관리가 용이하다. 반면에 이 방법에서는 서비스 사업자의 치국 계획이 위에서 설명한 일반적인 방법을 따르지 않을 경우 많은 국번을 수용할 수가 없다는 단점이 있다.

3.3.3 단말 번호를 위한 색인 방법

현재 단말 번호를 위하여 사용되고 있는 변형된 선형 해싱 기법의 경우는 가입자의 수가 늘어남에 따라 잦은 디렉토리의 증가로 인한 오버헤드가 큰 문제이다. 또한 엔진 프로세스가 재시동될 때마다 새로 구성되어야 하는 구조로 인하여 이 오버헤드는 더욱 큰 문제점으로 인식되고 있다. 따라서 키와 HLR 시스템의 특성을 고려하여 디렉토리의 증가와 같은 오버헤드가 적은 해싱 기법을 사용해야 한다.

이미 살펴본 바와 같이 색인에서 키로 사용되는 단말 번호는 지정된 범위의 양의 정수로 나타나고, 이를 UNIX 시스템의 랜덤 함수의 시드로 사용함으로서 비교적 일정한 분포의 해쉬 값을 얻을 수 있다. 그리고 색인에서 수용해야 하는 단말 번호의 수는 궁극적으로 HLR 시스템에서 수용해야 하는 가입자 수와 동일한 수로 고정되어 있으므로 초기에 색인에 필요한 자료 구조를 할당할 수 있다. 이러한 조건에서 삽입 및 삭제에 따른 오버헤드를 최소로 하는 색인 방법은 단순한 구조의 버켓 연결 해싱 기법이다.

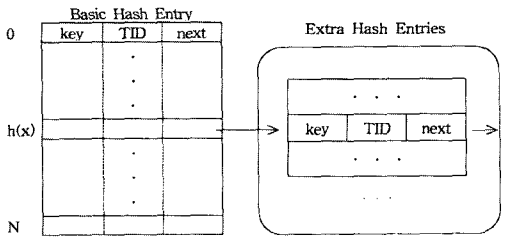
버켓에 수용되는 단말 번호의 수를 하나라 하고, 단말 번호를 x, 수용해야하는 가입자 수를 N이라 했을 경우 버켓에 대한 색인 번호는

$$bucket_no = random(srandom(x)) \bmod N$$

과 같이 나타난다. 그리고 랜덤 함수의 특성에 의하여 이 번호는 비교적 일정한 분포로 나타날 것이므로 충돌이 일어날 확률도 매우 적을 것이다. 그러나 해싱에서는 반드시 충돌을 고려해야 하므로 이를 해결하기 위하여 동일한 버켓 번호를 가지는 단말 번호를 체인으로 연결하는 방법을 생각할 수 있다. 이 방법은 최종적인 키의 수가 정해져 있고 비교적 일정한 분포의 해쉬 값을 얻을 수 있으므로 적용이 가능한 방법이다.

그림 10은 이러한 해싱 방법을 보여준다. 그림에서 기본 해쉬 엔트리(basic hash entry)는 엔진 프로세스가 구동되는 초기에 고정되어 할당되는 영역이며, 부가 해쉬 엔트리(extra hash entry)는 충돌에 의해 연결되는 버켓이 존재하는 곳으로서 일정 개수씩 동적으로 할당되는 영역이다. 충돌이 일어난 경우는 기본 해쉬 엔트리에서 시작하여 순차적으로 부가 해쉬 엔트리를 검색함으로서 해당하는 튜플식별자를 알아낼 수 있다.

기존의 HLR 데이타베이스 시스템에서 사용하고 있는 변형된 선형 해싱 기법이 잦은 디렉토리의 증가로 인하여 엔진 프로세스가 구동될 때마다 오버헤드를 갖는 반면에 최종 가입자 수가 고정된 점을 고려한 버켓 연결 해싱 기법은 디렉토리의 증가와 같은 동적인 변화가 적으므로 초기 오버헤드가 적다. 그리고 비록 충돌 및 부가 해쉬 엔트리 할당에 의하여 삽입, 검색 및 삭제시에



h(x) = random(srandom(x)) mod N, N = # of subscribers

그림 10 단말 번호를 위한 색인 방법

일정한 속도를 제공하지는 못하지만 단말 번호를 사용하는 경우가 번호의 중복 감지, 가입자의 신규 가입 및 기존 가입자의 해지에 국한됨을 고려할 때 이 점은 크게 문제가 되지 않는다.

지금까지 살펴본 바와 같이 단말 번호를 사용한 해쉉 방법의 겨우도 위의 전화번호의 경우와 마찬가지로 키의 특성과 이에 필요한 기능을 고려했을 때 버켓 연결 해쉉 기법이 가장 적합한 해쉉 기법으로 생각되고 또한 이 방법은 그 구조 및 구현이 간단하며 추후 관리도 용이한 장점을 갖는다.

4. 백업 방법에 대한 연구

본 장에서는 현재 HLR 데이타베이스 시스템에서 사용하고 있는 백업 방법의 문제점을 바탕으로 새로운 효과적인 백업 방법을 제안한다. 이를 위하여 제 4.1절과 제 4.2절에서는 각각 일반적인 주기억장치 데이타베이스에서의 백업 및 회복 방법과 기존 HLR 시스템에서의 백업 방법에 대하여 살펴보고, 제 4.3절에서 설명하는 HLR 데이타베이스 시스템에서의 트랜잭션의 종류와 특성을 바탕으로 제 4.4절에서 제안하는 효과적인 백업 방법을 설명한다.

4.1 주기억장치 데이타베이스 시스템에서의 백업

HLR 데이타베이스 시스템에서의 백업의 필요성과 백업이 필요한 작업에는 기존의 주기억장치 데이타베이스 시스템과는 다른 특징이 있다. 그러나 HLR 데이타베이스 시스템은 결국 주기억장치 데이타베이스 시스템으로 볼 수 있으므로, 본 절에서는 주기억장치 데이타베이스 시스템에서의 백업 및 회복의 필요성과 기존의 기법인 MM-DBMS와 Dali 시스템에서의 회복 기법에 대하여 간략히 소개한다.

주기억장치 데이타베이스 시스템에서 전원 차단이나 운영체제(operating system) 등의 오류가 발생하면 주기억장치의 휘발성 특성에 의하여 데이타베이스의 내용이 손실된다. 백업은 이러한 오류가 발생했을 경우에 회복(recovery)을 위하여 실시한다. 기존의 데이타베이스 시스템에서는 백업 기법으로서 주로 로깅(logging)과 체크포인팅(check-pointing)을 이용한 회복 기법을 주로 사용한다.

로깅이란 시스템이 정상적으로 동작하는 상황에서 트랜잭션의 상태 변화 및 데이타 변동 과정을 비휘발성 장치에 기록하는 기능을 말한다. 체크포인팅이란 오류로 인하여 시스템이 재시동하는 경우 빠른 시간내에 일관성(consistency)을 유지하는 데이타베이스를 재구성하기 위하여 주기적으로 최근의 데이타베이스의 상태를 재시동되는 경우 트랜잭션 일괄성을 유지하는 가장 최근의 데이타베이스 상태를 구성해주는 기능이 필요하다[14].

주기억 시스템에서의 회복 기법의 예로는 MM-DBMS와 Dali 시스템에서의 회복 기법을 들 수 있다. MM-DBMS는 Wissconsin 대학에서 설계된 주기억장치 데이타베이스 시스템으로 관계형 데이타 모델을 지원하며, 색인 구조로 T 트리와 변형된 선형 해쉉 기법을 제공한다. MM-DBMS에서는 회복에 관련된 작업을 수행하는 별도의 프로세서(processor)를 사용하며, 로그 버퍼 영역으로서 비휘발성 메모리를 사용하고 있다[15].

Dali 저장시스템은 AT&T Bell Leb.에서 개발한 주기억장치 저장시스템이다. Dali 저장시스템은 운영체제의 도움으로 디스크 영역을 주기억장치 영역과 연결하는 기억장치-배치(memory-mapped)된 구조를 사용한다[3]. 이 방법에서는 디스크에서 관리되는 데이타베이스의 구조와 주기억장치에 상주하는 데이타베이스의 구조를 동일하게 관리할 수 있다.

4.2 HLR 데이타베이스 시스템에서의 백업

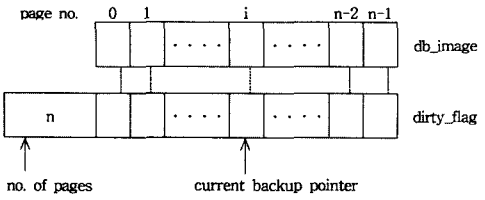
앞 절에서 살펴보았듯이 주기억장치 데이타베이스 시스템에서 반드시 고려해야하는 문제점의 하나는 파손시의 회복이다. HLR 시스템에서도 가입자의 추가 및 삭제, 부가서비스 등록 및 해지 등의 기능은 시스템의 오류로 인하여 손실되어서는 안되는 정보이므로 백업 기능이 필요하다.

현재 HLR 데이타베이스 시스템에서는 이러한 백업을 수행하기 위하여 엔진 프로세스와는 별도로 백업 프로세스(backup process)를 두고 있다. 이 백업 프로세스는 엔진 프로세스에 의하여 갱신된 메모리 SLD의 내용을 페이지 단위로 지속적으로 디스크 SLD로 백업하는 방법을 사용한다. 백업과 재시동시의 회복을 위하여 로깅이나 체크포인팅과 같은 기법을

사용하지 않으며 특별히 트랜잭션의 개념도 지원하지 않는다.

다음의 그림 11은 백업 프로세스가 백업하는데 메모리 SLD와 이를 위한 갱신 플래그, 그리고 다음에 백업을 수행할 포인터의 구조와 간략한 알고리즘을 나타낸다. 그림에서 db-image는 가입자 테이블을 비롯한 테이블이 저장되어 있는 메모리 SLD를 나타내는데, 지정된 크기의 페이지 단위로 입출력이 일어나며 따라서 각각의 페이지가 dirty-flag의 하나의 요소에 해당한다.

엔진 프로세스는 응용 프로세스의 요구를 받아 튜플의 추가, 삭제, 갱신, 검색 등의 필요한 데이타베이스 작업을 수행한 후 메모리 SLD에서 갱신된 페이지에 해당하는 갱신 플래그를 1로 변경한다. 별도의 프로세스가 동



do forever

find the nearest dirty page from the current pointer;

if there is no dirty page **then** sleep one second;

else write the page into the disk and move the pointer;

od;

그림 11 기존의 백업 방법

작하는 백업 프로세스는 지속적으로 갱신 플래그를 참조하며 엔진 프로세스에 의하여 1로 변경된 페이지를 디스크 SLD로 백업하고 플래그를 다시 0으로 변경하는 과정을 반복적으로 수행한다.

이러한 백업 방법을 사용하므로써 가입자의 추가 및 삭제, 부가서비스의 등록 및 해지 등의 작업이 수행된 후, 변경된 메모리 페이지의 내용이 디스크에 반영될 수 있다. 그러나 실제 상황에서 페이지가 갱신되는 주된 원인은 위치등록(registration notification)에 의한 변경이고, 다음 절에서 설명하듯이 HLR 시스템의 특성상 이러한 변경은 매번 신속하게 백업할 필요가 없는 정보이다.

이러한 백업 방법은 적은 수의 가입자를 수용하고 있는 경우에도 별다른 문제를 발생시키지 않으나, 가입자 수가 증가하고 이에 따라 많은 위치등록이 발생할 경우 HLR 시스템의 성능에 매우 심각한 영향을 미칠 것으로 예상된다. 따라서 가입자의 추가 및 삭제와 같은 기능과 위치등록에 의한 위치정보의 변경과 같은 기능은 백업 방법을 다르게 할 필요가 있다.

본 절에서는 기존의 HLR 데이타베이스 시스템에서의 백업 방법을 살펴보고 이에 따른 문제점을 지적하였다. 다음 절에서는 발견된 문제점을 바탕으로 보다 효과적인 백업 방법을 설계하기 위하여 HLR 시스템에서 필요한 트랜잭션의 종류와 특성에 대하여 설명한다.

4.3 HLR 데이타베이스 시스템에서의 트랜잭션의 종류 및 특성

HLR 시스템의 주된 기능을 수행하는 응용서비스 요소 부분에서는 위치정보의 갱신을 위한 위치등록, 호처리를 위한 루팅정보 요구(routing information inquiry) 등의 기능을 대표적으로 고려할 수 있다. 이들 두 기능은 전체 응용서비스 요소 기능의 대부분(99% 이상)을 차지한다. 이중에서 루팅정보 요구의 경우는 전화 번호를 사용하여 가입자 테이블에서 해당 가입자 튜플을 검색한 후 가입자의 위치정보를 MSC/VLR에 전달하는 기능을 하므로 별도의 데이타베이스 변경 기능이 필요하지 않다. 그러나 위치 등록의 경우는 가입자의 전화 번호로 해당 가입자 튜플을 검색한 후, 전화 번호와 함께 MSC/VLR로부터 전달된 위치정보를 해당 가입자 튜플에 반영시키는 기능을 하므로 데이타베이스의 변경 기능이 수행된다.

위치등록에 의한 가입자 튜플의 위치정보의 변경은 데이타베이스 사용의 대부분을 차지하는 응용서비스 요소 기능 중에서도 80% 이상을 차지할 정도로 빈번하게 발생하는 중요한 기능이다. 100만 가입자 수용의 경우 초당 약 800번 이상의 위치 등록이 발생하나, 이러한 위치정보의 변경은 HLR 시스템의 특성상 신속한 백업이 필요하지 않은 정보이다. 그 이유는 시스템의 오류로 인하여 위치정보가 손실되더라도 다음 번의 위치등록에 의하여 비교적 빠른 시간 안에 정보의 복구가 자동적으로 이루어질 수 있기 때문이다. 또한 이러한 위치정보의 손실을 대비하여 MSC/VLR-HLR간 신호방식 기준에 "불확실한 위치자료 통보(unreliable roamer data directive)"라는 응용서비스 요소를 특별히 포함하고 있다.

오류로 인하여 HLR 시스템의 재시동이 발생한 경우에는 HLR 시스템으로부터 관련된 모든 MSC/VLR로 불확실한 위치자료 통보가 전달되고, 각 MSC/VLR에서는 해당 HLR 시스템으로 발생하는 위치등록 기능을 여과 없이 전달하므로서 신속한 복구가 이루어질 수 있도록 한다. 따라서 모든 가입자에 대하여 시간당 한번 이상 발생하는 위치등록의 특성에 의하여 재시동이 일어난 후 한시간 이내에 각 가입자의 위치정보가 바르게 복구되므로 별도로 신속한 백업이 필요로 하지 않는다.

위치등록에 의해 변경된 정보가 반드시 백업되지 않아도 되는 반면에 운용자 혹은 고객 관리 센터로부터의 요구에 의하여 이루어지는 신규 가입자의 등록 및 기존 가입자 해지, 부가서비스의 등록 및 삭제 등의 가입자 관련 운용 기능은 관련된 정보가 손실되어서는 안되는 기능이다.

신규 사업자의 등록과 같은 가입자 관련 운용 기능은 HLR 시스템의 운용자에 의하여 혹은 HLR 시스템과 연결된 고객 관리 센터의 요구에 의하여 그 기능이 수행되고, 오류로 인하여 HLR 시스템이 재시동되었을 경우에도 정보가 유지되어야 관리하고 있는 가입자들에

대한 지속적인 서비스가 가능하다.

지금까지 살펴보았듯이 정보에 대한 백업 여부를 기준으로 판단하였을 때, HLR 데이타베이스 시스템에서는 크게 두 가지 종류의 트랜잭션이 존재한다. 주기억장치 데이타베이스 시스템에서 성능에 직접 관련된 부분은 주기억장치에 비하여 상대적으로 접근 속도가 매우 늦은 디스크 작업이고 백업은 디스크 작업에 기반을 두고 있다. 따라서 디스크 작업에 따른 오버헤드를 줄이기 위해서는 이들 두 가지 종류의 기능에 대하여 백업과 관련 작업을 구분하여 수행할 필요가 있다.

본 절에서는 HLR 데이타베이스 시스템에서 발생하는 트랜잭션을 백업 관점에서 두 가지로 구분하였다. 다음 절에서는 이러한 특성을 고려하여 두 가지의 갱신 플래그를 사용함으로서 보다 효율적으로 백업할 수 있는 방법을 제안한다.

4.4 HLR 데이타베이스 시스템을 위한 효과적인 백업 방법

HLR 시스템에서는 하드웨어를 신뢰성 있는 장애허용 컴퓨터를 사용할 뿐 아니라 데이타베이스 작업의 대부분을 차지하는 위치정보는 손실되더라도 빠른 시간 안에 복구가 가능한 특성이 있다. 따라서 HLR 시스템의 성능을 극대화시키기 위하여 필수적인 기능을 수행하며 매우 제한적인 백업 방법이 필요하다.

본 절에서는 지금까지 살펴본 두 가지 종류의 트랜잭션에 대하여 방법을 달리하는 백업 방법을 제안한다. 신속한 백업을 필요로 하지 않으며 빠른 응답 시간을 요구하는 위치등록과 같은 기능에 대한 주기적인 백업 방법과 신규 가입자 등록의 경우와 같이 정보가 손실되지 않도록 신속한 백업이 이루어져야 하는 방법이 그것이다.

빈번하게 발생하는 위치등록에 의하여 변경되는 가입자의 위치정보는 손실에 대하여 회복이 자동적으로 매우 빨리 이루어지는 특성을 가진다. 그러나 각 가입자에 대하여 위치등록이 자주 일어나는 반면에 HLR 시스템이 재시동된 경우 각 가입자에 대한 위치정보가 다음번의 위치등록에 의하여 복구되기 이전이라도 해당 가입자의 오류전 위치정보를 사용하면 호를 성공시킬 수 있는 가능성이 매우 높다.

HLR 시스템이 재시동되기 이전의 위치정보를 사용하기 위해서는 각 가입자에 대하여 가장 최근의 위치정보를 백업하여 사용하는 것이 바람직하다. 그러나 지금까지 살펴보았듯이 위치정보에 대해 매번 백업을 수행할 경우에 HLR 시스템의 전체 성능에 매우 심각한 영향을 미치는 점을 지적하였다. 그리고 HLR 시스템이 특정 지역을 기반으로 하기 때문에 잦은 위치등록 횟수에 비하여 대다수 가입자의 위치정보가 쉽게 변하지 않는 특성을 가지고 있다. 따라서 위치정보의 변경과 같은 데이타베이스 갱신 정보는 하루 혹은 이틀 단위의 주기적인 백업에 의하여 그 정보를 유지할 필요가 있다.

주기적인 백업은 다음과 같은 순서에 의하여 이루어질 수 있다. 먼저 응용 프로세스의 위치정보 변경 요구를 받아 엔진 프로세스는 데이타베이스 갱신 기능을 수행한 후 갱신된 페이지에 해당하는 갱신 플래그를 1로 변경한다. 이렇게 갱신 플래그가 1로 변경된 페이지에 대해서는 기존의 백업 방법과는 달리 운용자의 요구나 시스템의 사용이 가장 적은 최한시에 백업 프로세스에 의하여 디스크에 반영되는 방법을 취한다.

위치정보가 주기적인 백업이 가능한 반면에 신규 가입과 같은 관련 운용 기능은 작업 내용이 즉시 백업되어야 한다. 이를 위해서는 크게 두 가지 방법을 생각할 수 있는데 첫 번째는 기존의 백업 프로세스의 방법을 그대로 사용하는 경우이고, 두 번째는 트랜잭션 개념을 도입하여 가입자 관련 운용 기능에 대하여 트랜잭션 별로 로깅을 수행하고 오류시 로그와 주기적인 백업에 의하여 저장된 디스크 SLD를 사용하여 회복을 수행하는 방법이다.

기존의 백업 프로세스의 방법을 사용하기 위해서는 엔진 프로세스가 가입자 관련 운용 기능에 의하여 변경된 페이지에 대해서는 위치정보의 변경과는 달리 갱신 플래그를 2로 변경과, 백업 프로세스는 지속적으로 갱신 플래그를 검사하여 2로 변경된 페이지를 디스크 SLD로 백업하는 기능을 수행한다. 이 방법은 기존의 구조를 크게 변경하지 않고 구현이 가능한 장점을 가지고 있다. 그러나 정보의 변경과 백업이 별개의 프로세스에 의하여 별도로 이루어지기 때문에 오류에 의한 재시동이 일어났을 경우 정보의 불일치(inconsistency)가 일어날 위험성을 가지고 있다.

다음으로 현재 별도로 제공하고 있지 않는 트랜잭션 개념을 도입하여 로깅 및 이를 이용한 회복을 수행하는 방법은 시스템이 재시동되었을 경우에 정보의 불일치가 나타나지 않고 데이타베이스가 일관성 있는 상태로 회복될 수 있는 장점이 있는 반면에 기존의 엔진 프로세스 구조를 많이 수정 및 보완해야 하는 구현상의 어려움이 있다.

정보의 불일치는 심각한 문제를 일으킬 소지를 가지고 있으나 가입자 관련 운용 작업 내역은 고객 관리 센터와 운용자 작업 기록에 의하여 자동적으로 로깅이 되고 이 정보는 특정 기간동안 관리가 된다. 따라서 기존의 백업 프로세스를 사용한 회복에 의하여 특정 가입자

에 대하여 정보 불일치가 발생했을 경우에 이 작업 내역을 사용하여 데이타베이스를 일관성 있는 상태로의 회복이 가능하다.

결국 기존 HLR 시스템의 소프트웨어 구조를 크게 변경하지 않으며 실제 운용중인 시스템 전체의 성능을 향상시킬 수 있는 방법은 위치정보의 변경과 가입자 운용 기능에 따른 정보의 변경에 대하여 갱신 플래그의 값을 달리함으로써 백업 프로세스가 해당 갱신 페이지를 디스크에 저장하는 시점을 조절하는 방법을 사용할 수 있다.

본 절에서는 기존 시스템의 백업 방법의 문제점에 대하여 살펴보고, HLR 데이타베이스 시스템에서 발생하는 두 가지 종류의 트랜잭션 특성을 고려하여 기존의 소프트웨어 구조를 크게 바꾸지 않으며 성능을 향상시킬수 있는 백업 방법을 제안하였다. 이 방법에 의하여 가입자 수가 많아짐으로서 발생하는 실제 운용 상황에서의 시스템 성능 저하 문제점을 해결할 수 있다.

5. 결 론

본 논문에서는 현재 이동통신 망에서 운용중인 HLR 시스템의 주기억장치 데이타베이스 시스템에서 발견된 문제점을 바탕으로 성능을 고려한 효과적인 색인 방법과 백업 방법을 제안하였다. 이러한 개선점은 향후 IMT2000 service 및 위성통신에서의 HLR 시스템 개발에도 도움이 될 것으로 생각한다.

사용을 제안하는 색인 방법은 HLR 데이타베이스 시스템에서 키로 사용되는 전화 번호와 단말 번호의 특성을 고려하여 각각 이 단계 색인을 응용한 방법과 버켓 연결 해쉉 기법이다. 전화 번호를 위한 이 단계 색인 방법은 검색 속도 및 저장 효율 측면에서 기존의 T 트리에 비하여 우수함을 살펴보았다. 다음으로 단말 번호의 경우 잦은 디렉토리의 증가와 같은 오버헤드를 갖는 기존의 변형된 선형 해쉉 방법에 비하여 버켓 연결 해쉉 기법이 보다 효과적임을 살펴보았다. 또한 이들 두 가지 방법은 그 구조가 간단하여 구현 및 운용중 관리가 용이한 장점을 가지고 있다.

본 논문에서는 기존의 백업 방법이 위치등록에 의한 위치정보의 변경과 가입자 관련 운용 기능에 의한 정보 변경을 구분하지 않고 백업함으로서 발생하는 시스템 성능 저하의 문제를 해결하기 위하여 이들 두 가지 종류의 변경 기능을 구분하는 보다 효과적인 백업 방법을 제안하였다. 위치정보와 같이 자동적으로 복구가 가능한 정보의 변경은 주기적인 백업만을 수행하고, 가입자 관련 운용 기능에 의한 정보의 변경은 기존과 같이 신속하게 백업함으로써 정보의 손실을 막는 방법을 사용한다.

100만 가입자 수용의 경우,기존 기법에서의 전화번호 인덱스용 메모리양이 약 16M Byte였으나, 제안 기법은 약 5~6Mbyte로 처리할 수 있으며, 기존 기법에서의 전화번호 검색은 속도면에서 약 20회 이상의 비교 연산이 필요하나,제안 기법은 2회의 메모리 접근으로 처리할 수 있으며, 기존 기법에서의 단말 번호 초기 해쉬 구조 구성시 약 25만 번의 디렉토리 증가 작업이 과부하로 작용하였으나, 제안 기법은 디렉토리 증가 작업이 없으며, 기존 기법에서는 초당 약 800번 이상 발생하는 위치 등록으로 인한 실시간 백업이 과부하로 작용하였으나, 제안 기법은 위치 등록으로 인한 백업은 실시간으로 할 필요가 없게 되어 약 80%의 실시간 백업으로 인한 부하를 감소시킬 수 있는 효과가 있다.

향후 IMT2000 service로의 진화에 기반한 mobile commerce에 있어서 가장 중요한 요구 사항은 보안 기능일 것이다. 이러한 기능 수행을 위해 현재의 HLR DB Scheme과 관련하여 변화되어야 할 최소한의 attribute는 다음과 같다. session key 설정 및 상호 인증 protocol 수행을 위해서, 상호 인증 algorithms 수행의 결과로 산출된 결과값, 산출된 session key, 사용된 random number 각각을 저장할 수 있는 attribute들이 추가되어져야 한다. 향후의 IMT2000 서비스로의 진화와 함께 이질적 network들과의 roaming이 본격화되기 전에 HLR과 VLR간에 교환되는 session key 및 인증 관련 변수들에 대한 효과적인 보안 기능 개발이 절실하다.

personal mobility 지원에 따른 signalling traffic의 증가를 최소화시키기 위해 단말의 상태 정보를 구분 관리하기 위한 attribute 추가도 필요하다. 또한 새로운 부가서비스 지원을 위하여 동적인 스키마 변경을 지원하기 위한 연구가 필요하고, 가입자 수가 증가함에 따라 발생하는 HLR 시스템간의 가입자 이동에 대한 고려가 필요하며, 가입자 테이블과는 별도로 관리가 필요한 특수 부가서비스에 대한 연구가 있어야 한다.

참 고 문 헌

[1] DeWit, D.J. et al, "Implementation Techniques for Main Memory Database Systems," Proc. Intl. Conf. on Management of Data, ACM SIGMOD, Boston, USA, pp. 1-8, June, 1984.

[2] Garcia-Molina, H., "Main Memory Database Systems: An Overview," IEEE Trans. on Knowledge and Data Engineering, Vol. 4, No. 6, pp. 509-516, 1992.

[3] Jagadish, H. V. et al, "Dali: A High Performance

Main Memory Storare Manager," Proc. 20th Intl. Conf. on Very Large Data Bases, Santiago, Chille, pp. 48-59, Sept., 1994.
[4] Peter, M. G. et al, "PRISMA/DB: A Parallel, Main Memory Relational DBMS," IEEE Trans. on Knowledge and Data Engincering, Vol. 4, No 6, Dec. 1992.
[5] Knuth, D., The Art of Programming, Vol. 3, Sorting and Searching Addison-Wesley, 1973.
[6] 한국통신기술협회, MSC/VLR-HLR/AC간 MAP 신호 방식 표준, Jul. 1997.
[7] Son, S. H., Advances In Real-Time Systems, Prentice Hall, 1995.
[8] Lehman, T.J. and Carey, M.J., "A Study of Index Structures for Main Memory Database Mana gement Systems," Proc. 12th Intl. Conf. on Very Large Data, Kyoto, Japan, pp. 294-303, Aug. 1986.
[9] Bayer, R. and McCreight, C., "Organization and Maintenance of Large Ordered Indexes," Acta Infomatica, Vol. 1, No. 3, pp. 290-306, 1972.
[10] Bayer, R., "Symmetric Binary B-Trees: Data Structure and Maintenance Algorithms," Acta Infomatica, Vol. 1, No. 3, pp. 290-306, 1972.
[11] Comer, D., "The Ubiquitous B-Tree," ACMI Computing Surveys, Vol. 1, No. 2, pp. 121-137, 1979.
[12] Litwin, W., "Linear Hashing: A New Tool For File and Table Addressing," Proc. 6th Intl. Conf. on Very Large Data, Montreal, Canada, pp. 212-223, Oct. 1980.
[13] TIA/EIA, IS-41(A), Initial Version. Jan. 1991.
[14] Jagadish, H. V., Silberschatz, A. and Sudarshan, S., "Recovering from Main-Memory Lapses," Proc. 19th Intl. Conf. on Very Large Data Bases, Dublin, Ireland, pp. 391-404, Aug.. 1993.
[15] Lehman, T.J. and Carey, M.J., "A High Performance Memory-Resident Database System," Proc. 12th Intl. Conf. on Management of Data, San Francisco, USA, pp. 104-117, 1987.

김 장 환
1980년 서울대학교 경제학 학사. 1997년 한국과학기술원 전산학 석사. 2003년 충북대학교 전산학 박사. 1984년~1988년 쌍용정보통신 연구원. 1988년~1993년 Qnix Data System 연구원. 1993년~1998년 SK Telecom 연구원. 1998년~현재 대덕대 IT계열 교수. ITU-R, WIPO member, 전자상거래 관리사. 관심분야는 Mobile & Wireless Communication, Performance Analysis of Networks, Database System, Information Security, Mobile Multimedia, Mobility Managements, 알고리즘 및 계산이론 정보통신 경제 예측

이 충 세
1984년 University of South Carolins, 전산학 박사. University of North Dakota, 전산학과 조교수. 동아대학교 경영정보학과 교수. 관심분야는 결함허용, 알고리즘 및 전문가 시스템