

분산 모바일 임베디드 시스템 기반의 새로운 위치정보 관리 시스템

김 장 환*

Performance Enhancement Architecture for HLR System Based on Distributed Mobile Embedded System

Jang-Hwan Kim*

요 약

모바일 망에서는 지속적으로 변하는 개별 가입자의 위치 정보가 관리되어야 한다. 이와 같은 분산 환경 속에서 위치 정보 관리의 핵심 역할을 수행하는 내장형 시스템인 HLR system은 table 관리 기능과 색인 관리 기능, 그리고 백업 관리 기능을 제공한다.

본 논문에서는, 현재 사용되고 있는 HLR system의 문제점을 철저한 분석을 통해 파악하여 그 문제의 해결 방안을 제시하고, HLR system을 위한 새로운 구조를 제시하였다.

HLR system에서는 가입자 정보의 실시간 접근과 갱신을 제공하기 위해 주기억장치 데이터베이스 시스템이 사용되고 있다. 따라서, 더 나은 실시간 기능을 제공하기 위해서, 가입자의 정보를 보다 신뢰성 있게 관리하기 위해서, 보다 많은 가입자에게 편의를 제공하기 위해서는 이의 성능 개선이 요구된다.

본 논문에서는, 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. 버킷 연결 해싱 기법은 기존의 변형된 선형 해싱 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.

가입자 수용시, 제안 기법을 사용하게 되면 기존 기법보다 메모리 사용량 절감(62% 이상), 디렉토리 증가 작업 제거, 백업 작업 감소(80% 이상)를 제공받게 된다.

Key Words : 임베디드 시스템, 스레드, 회복, 로깅, 트랜잭션

ABSTRACT

In mobile cellular network, the ever-changing location of a mobile host necessitates the continuous tracking of its current position and efficient management of location information. A database called Home Location Register(HLR) plays a major role in location management in this distributed environment, providing table management, index management, and backup management facilities.

The objectives of this paper are to identify the problems of the current HLR system through rigorous analysis, to suggest solutions to them, and to propose a new architecture for the HLR system. In the HLR system, a main memory database system is used to provide real-time accesses and updates of subscriber's information. Thus it is suggested that the improvement be made to support better real-time facilities, to manage subscriber's information more reliably, and to accommodate more subscribers.

In this paper, I 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. Insertion and deletion overhead of the chained bucket hashing method is less than that of modified linear hashing method. In the proposed backup method, I use two kinds of dirty flags in order to solve the performance degradation problem caused by frequent registration-location operations.

Performance analysis has been performed to evaluate the proposed techniques based on a system with subscribers. The results show that, in comparison with the current techniques, the memory requirement is reduced by more than 62%, directory operations, and backup operation by more than 80%.

Keywords : embedded system, thread, direcovery, logging, transaction

* 대덕대 IT계열 교수

논문번호 : KICS2004-10-212, 접수일자 : 2004년 10월 2일

I. 서론

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. 또한 위치 기반 서비스의 확대와 함께 다양한 형태의 위치 서버 기기의 수요가 확산될 것으로 예상된다. 특히 국가가 신성장 동력 산업으로 추진하고 있는 이동통신 임베디드 시스템, 홈 네트워크, Telematics 분야에서의 대부분의 응용은 위치 정보 관리의 기능을 필수로 요구하는 것들이다. 이러한 HLR 시스템은 향후 Ubiquitous Network 및 위성이동통신 등의 분야로 그 기능이 확장되어 사용될 것으로 예상된다.

반도체 산업의 지속적 발달로 저전력, 초소형의 기억 장치를 탑재한 각종 위치 기반 서비스 기기 및 응용이 계속 출시되고 있다. 반도체 산업의 발달과 더불어 주기억장치의 용량이 커지고 가격이 많이 하락함에 따라서 데이터베이스 분야에서도 주기억장치를 최대로 활용하여 데이터베이스 시스템의 성능을 개선하는 주기억장치 데이터베이스 시스템(main memory database system)에 관한 연구가 활발히 진행되어 오고 있다[1][2][3][4]. 주기억장치 데이터베이스 시스템은 디스크 기반 데이터베이스 시스템(disk resident database system)과는 달리 데이터베이스 전체가 주기억장치에 상주하고 있어서 데이터에 대한 신속한 액세스를 제공한다. 따라서 위치 정보의 변경 및 접근에 대한 실시간 처리와 대용량의 가입자 정보 관리를 목적으로 하는 HLR 시스템에서는 이러한 주기억장치 데이터베이스 시스템을 사용한다[5][6][10].

HLR 시스템에서 사용하는 주기억장치 데이터베이스 시스템은 이동 통신 망의 특성상 일반적인 주기억장치 데이터베이스 시스템과는 다른 다음과 같은 차이가 있다. 시스템의 오류로 인하여 HLR 시스템에서 관리하는 가입자의 위치 정보가 손실되더라도 기존의 회복기법과 같은 데이터베이스 시스템의 도움없이 복구 가능하다는 점이다. 이 외에도 부가서비스의 증가에 따른 동적인 스키마 변경의 필요성, 데이터베이스간의 가입자 이동(migration), 특수 부가서비스에 대한 지원 등이 있다. 따라서 이러한 특성이 HLR 데이터베이스 시스템에 반영될 필요가 있다.

이동통신 시스템은 현재의 단일 이동성 서비스 방식으로부터 개인 이동성 서비스로 진화해 갈 것으로 예측된다. 또한, 국가간 로밍 서비스와 정보보호 기능 수행까지 담당해야 하는 방향으로 진화해 가고 있다 [8]. 따라서 이러한 서비스를 지원해야 하는 신호망의 중추적 역할을 수행해야 하는 위치등록기가 처리해야 하는 트래픽 메시지(traffic message)는 계속적으로 늘어날 수 밖에 없다[9]. 동일한 장비 비용으로 상대적으로 보다 많은 가입자를 유연하게 수용할 수 있는 위치등록기가 망 구성상 경쟁 우위를 차지하게 될 것이다. 따라서 이동통신 망 임베디드 시스템으로서의 위치등록기의 특성을 분석하여 동일한 장비 비용 대비 성능을 향상시킬 수 있는 구조를 연구하게 되었다. 본 논문에서는 위와 같은 HLR 시스템의 특성 중에서도 조건에 따른 백업의 필요성과 현재의 HLR 시스템을 분석한 결과 발견한 문제점을 바탕으로 보다 나은 시스템 개발을 위해서 실용적으로 적용 가능한 S/W 구조와 HLR 시스템의 성능을 고려한 효과적인 백업 방법을 제시한다.

사용을 제안하는 백업 방법은 HLR 데이터베이스에서 발생할 수 있는 트랜잭션의 종류를 구분하여 두 가지의 갱신 플래그(dirty flag)를 관리하여 백업을 수행하는 방법을 제시한다. 위치등록에 의한 위치 정보 변경은 신속한 백업을 수행하지 않고, 반면에 가입자 관련 운용 기능은 그 기능별, 즉 가입자 추가, 가입자 삭제, 부가서비스 등록, 부가서비스 삭제 등으로 그 기능을 구분하여 트랜잭션으로 만들었다. 이들 트랜잭션은 각각 별도의 메시지로 구분되어 운용유지보수 시스템이나 고객 관리 센터의 응용 프로세스에서 엔진 프로세스로 전달되고, 엔진 프로세스에 의하여 그 기능이 수행되며, 수행 결과만을 다시 응용 프로세스에 전달하도록 하였다. 기존 기법에서는 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. 현재의 백업 방법은 단순히 변경된 메모리 SLD 블록을 디스크에 기록하는 과정으로 데이터베이스 일관성 유지에 문제가 있음을 지적하였다. 이 문제를 해결하기 위해 가입자 관련 운용 기능에 대해서 트랜잭션 개념의 도입과 로깅과 체크포인팅을 통한 회복 기법을 제시하였다. 그리고 기존의 많은 Process로 인한 잦은 메시지 통신과 문맥 전환 기능 수행으로 인한 성능 저하를 개선하기 위해, 통합 가능한 Process들을 쓰레드 기법을 사용하여 통합하는 구조를 제안한다. 새로운 엔진 프

로세스에는 기존의 초기화 처리부, 질의 처리부, 테이블 관리부, 색인 관리부가 그 기능이 개선되어 포함되어 있으며, 회복 관리부와 로깅 관리부가 새롭게 추가되어 있다. 이 구조는 기존 엔진 프로세스의 골격을 유지하면서 성능과 안정성을 향상시키는 방향으로 설계하였다.

본 논문의 구성은 다음과 같다. 제Ⅱ장에서는 HLR 시스템과 HLR 시스템 S/W 구조에 대해 분석한 내용을 기술한다. 제Ⅲ장에서는 HLR 시스템 S/W 구조의 문제점과 개선안에 대해 기술한다. 제Ⅳ장에서는 성능 향상을 고려한 효과적인 백업 등에 대하여 기술한다. 제Ⅴ장에서는 결론과 함께 앞으로의 HLR 시스템의 개선 방향을 제시한다.

Ⅱ. HLR 시스템 분석

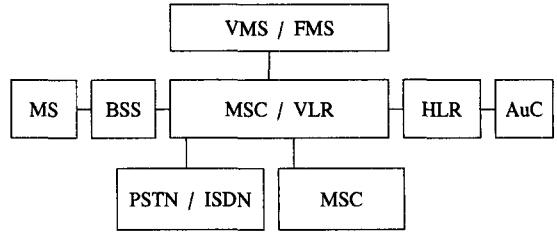
본 장에서는 논문에서 제시하는 방법들에 대한 이해를 돕기 위하여 HLR 시스템에 대해 개괄적으로 설명하고 HLR 시스템 S/W 구조에 대해 분석한 내용을 기술한다.

2.1 HLR 시스템

본 절에서는 이동통신 망에서의 HLR의 위치와 역할에 대하여 설명하고, HLR 시스템의 기능을 가장 잘 나타내는 호의 성립 과정과 위치 정보의 변경 과정의 두 가지 동작에 대하여 설명한다.

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 셀룰러 망(Cellular Network)에서 사용되고 있다.

그림 2.1은 HLR 시스템과 관련된 이동 통신 망의 구성요소를 나타낸다[7]. 그림에서 HLR 시스템은 MSC/VLR과 공통신 신호방식(CCS7:Common Channel Signalling No.7)에 의하여 연결되어 MSC/VLR로부터의 가입자에 대한 위치 정보의 갱신 및 조회 요구 등에 대한 처리 기능을 제공하며, 가입자 정보 변경의 경우에는 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.



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 2.1 이동 통신 망에서의 HLR 시스템

다음의 그림 2.2는 HLR 시스템을 포함하여 호가 성립되는 과정을 보여 준다. 가입자는 먼저 전화번호를 누름으로써 착신 단말에 대해 통화 요구를 한다. 이는 BSS를 거쳐 MSC에 전달되고, MSC에서는 루팅정보조회라는 MAP(Mobile Application Part) 동작을 발생시켜 HLR 시스템에 위치 정보를 요구한다. 이 요구를 받은 HLR 시스템은 자신의 주기억장치 데이터베이스를 검색하여 해당 가입자의 위치 정보를 내려주고, MSC는 이 정보를 이용하여 대국 MSC에 호 설정을 요구한다. 다음으로 대국 MSC는 착신 단말에 대하여 페이징(paging)을 시도하여 응답이 있으면 통화가 설정되어 통화가 이루어지게 된다. 따라서 HLR 입장에서는 루팅정보조회라는 MAP 동작을 처리하여 호의 성립에 도움을 주는 역할을 한다.

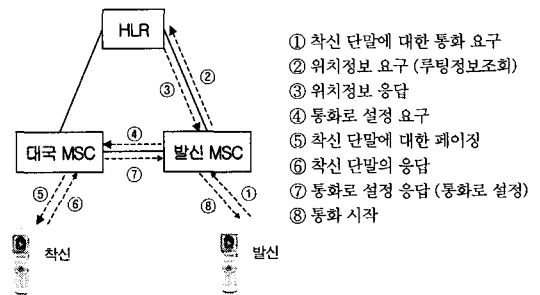


그림 2.2. 호의 성립 과정

다음의 그림 2.3은 가입자, 즉 단말의 이동에 따른 위치 정보의 변경 과정을 나타낸다. 그림과 같이 가입자가 이동하면 위치등록과 위치삭제의 두 가지 MAP 동작이 발생한다. 먼저 이동한 위치에서 위치등록 동작이 BSS를 거쳐 MSC/VLR에 전달된다. 위치등록 요구를 받은 MSC/VLR에서는 이를 HLR에 전

달하며, 그 결과를 받아 자신의 데이터베이스에 해당 가입자 정보를 저장한다. 위치등록 요구를 받은 HLR 시스템은 해당 가입자의 위치 정보를 갱신하고 위치가 변경된 경우에는 위치삭제 동작을 발생시킨다. 위치삭제 동작에 의하여 가입자가 존재하였던 이전 MSC/VLR에서는 그 정보가 삭제됨으로써 가입자의 이동이 완료된다. 따라서 HLR 시스템 입장에서는 위치등록 요구를 받아 이를 저장하고 위치가 변경되었으면 위치삭제를 사용하여 가입자의 위치를 변경하는 기능을 한다. 이렇게 변경된 새로운 위치는 다음의 호처리 과정에 사용된다.

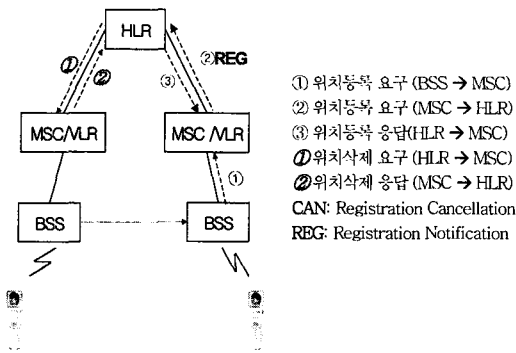


그림 2.3. 단말의 이동에 따른 위치 변경

2.2 HLR 시스템 소프트웨어 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 데이터베이스 시스템(Database 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 2.4는 HLR 시스템의 소프트웨어 구성 요소를 나타낸다.

2.2.1. 응용서비스 요소부

응용서비스 요소부는 HLR 소프트웨어 중에서 MAP 동작에 해당하는 부분으로서 위치 정보의 갱신 및 조회 등을 위하여 데이터베이스 시스템과 관련된 부분은 부분이다. 응용서비스 요소부는

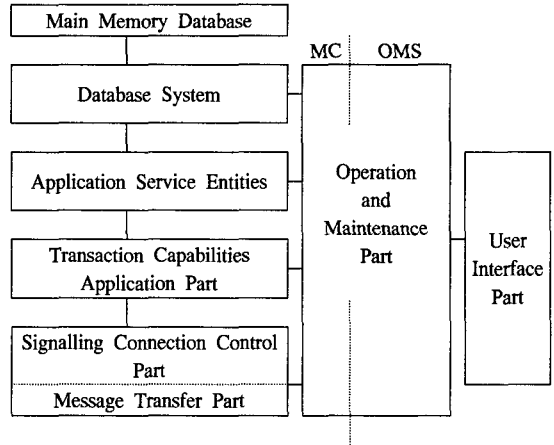


그림 2.4 HLR 시스템 소프트웨어 구조

KMAP(Korean-MAP)[7] 및 기존의 이날로그 방식 이 동통신 규격으로 사용되는 IS(Industry Standard)-41(A)[16]의 MAP을 바탕으로 구현되었으며, 각각의 동작이 하나의 프로세스로 구성되어 있다. 일반적으로 ASE와 MAP은 별도의 구분 없이 동일한 의미의 용어로 사용된다.

2.2.2 운용유지 보수부

운용유지 보수 부분의 목적은 크게 두 가지로 나눌 수 있다. 첫째는 HLR 시스템 운용에 필요한 통계, 상태 및 장애 등의 정보를 수집하고 분석하며 관리하는 기능적인 측면이고, 두 번째는 운용자가 보다 편리하게 HLR 시스템을 운용할 수 있도록 하는 운용의 용이성을 제공하는 측면이 있다.

2.2.3 문답처리 응용부

문답처리 응용부는 공통선 신호방식 프로토콜의 신호연결 제어부와 응용 서비스 요소부 사이를 연결하는 프로토콜 계층(protocol layer)이다. 이 부분은 MSC/VLR 과의 통신을 위한 프로토콜 관련 부분으로 데이터베이스 시스템과 관련된 동작이 별도로 필요로 하지 않은 부분이다. 다음의 그림 2.5는 문답처리 응용부를 포함하는 일반적인 메시지의 전달과정을

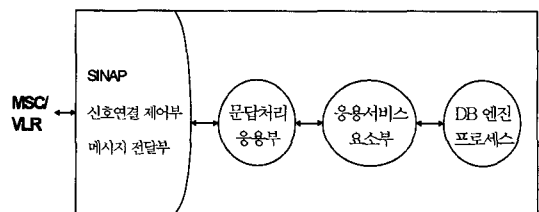


그림 2.5 MAP 메시지 흐름도

나타낸다.

그림에서 보면 MSC/VLR로부터 전달된 메시지는 SINAP 소프트웨어의 메시지 전달부 및 신호연결 제어부를 거쳐 문답처리 응용부에 전달된다. 그리고 문답처리 응용부에서는 각 메시지의 종류에 따라 해당 응용서비스 요소 프로세스에 전달한다. 응용서비스 요소 프로세스에서는 데이터베이스 엔진 프로세스를 통하여 데이터베이스 작업을 수행하고 그 결과를 반대의 과정으로 전달한다. 그림에서 볼 수 있듯이 문답처리 응용부는 데이터베이스에 접근하는 작업을 수행하지 않는다. 현재는 문답처리 응용부와 응용서비스 요소 프로세스 사이에 ATIF 프로세스가 존재한다.

Ⅲ. HLR 시스템 S/W 구조에 대한 연구

3.1 HLR 시스템 S/W 구조의 문제점

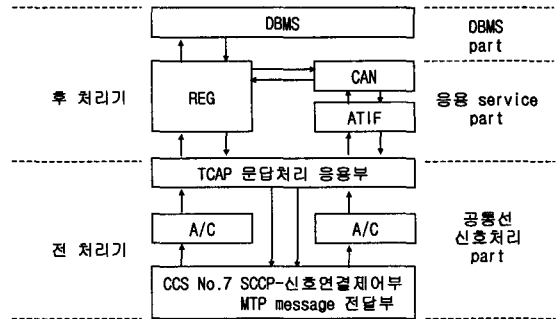
주요 문제점으로는 많은 프로세스로 인한 잦은 메시지 통신과 문맥 전환, 기존 색인 구조의 많은 메모리 양과 검색 속도의 변화, 백업 기능에 의한 성능 저하와 일관성 유지의 문제점 등이다. 이러한 문제점에 대해서 가능하면 기존의 구조를 크게 변화하지 않으면서 문제를 해결할 수 있는 방법을 연구하였으며, 필요에 따라서는 새로운 구조를 제시하기도 한다.

3.1.1. 응용서비스 요소부 라이브러리

일반적으로 여러 사람이 대규모의 소프트웨어를 개발할 때 라이브러리를 사용한다. 이러한 라이브러리를 사용하는 이유는 라이브러리 내의 함수들이 여러 프로그램에서 공통적으로 사용되거나 해당 라이브러리에 대한 재사용을 위해서다. 따라서 라이브러리를 잘 사용하면 효과적인 소프트웨어 개발이 가능하다. 현재 HLR 시스템의 응용서비스 요소부에서도 많은 라이브러리가 구현되어 각 응용서비스 요소 프로그램이 이들 라이브러리를 사용하도록 설계되어 있다. 그러나, 응용서비스 요소부의 많은 라이브러리의 각 함수는 각 응용서비스 요소 프로세스별로 사용하는 함수가 지정되어 있으므로 굳이 라이브러리로 만들지 않아도 되는 부분이다.

3.1.2 독립적인 ATIF 프로세스의 존재

ATIF(ASE/TCAP Interface) 프로세스는 응용서비스 요소(MAP) 프로세스와 문답처리 응용부(TCAP) 프로세스의 중간에 위치한다. 따라서 ATIF 프로세스는 TCAP 프로세스로부터 받은 메시지를 분석하여 그 종류에 따라 해당 MAP 프로세스에 분배해 주는 역할을 한다. 또한 이와 반대로 각 MAP 프로세스로부터 받은 메시지를 TCAP 프로세스에 전달하여 MSC/VLR로 전



A/C :Analysis/Creation CAN :Registration Cancellation ASE
 REG :RegistrationNotification ASE CCS :Common Channel Signalling
 그림 3.1 위치 등록시 HLR 내부 프로세스들의 통신

달되도록 하는 역할을 한다. ATIF 프로세스는 각 MAP 프로세스 및 TCAP 프로세스와의 통신을 위하여 UNIX 시스템의 메시지 큐를 사용한다. 이러한 프로세스 간의 잦은 통신은 UNIX 시스템 함수를 호출하게 되고, 이로 인하여 많은 문맥 전환(context switch)이 발생하므로 성능에 큰 영향을 미친다. 또한 모든 MAP 메시지는 MSC에서 HLR로 전달될 때와 HLR에서 MSC로 전달될 때 각각 ATIF 프로세스를 통과해야 하므로 더욱 많은 메시지 통신이 발생하게 된다.

3.1.3 너무 많은 응용서비스 요소 프로세스

각 응용서비스 요소 동작은 각각 별도의 프로세스로 구현되어 그 수가 매우 많다. 이러한 많은 프로세스를 수행하기 위하여 다음과 같은 문제가 발생한다. 첫째는 각 프로세스를 동작시키기 위하여 많은 메모리 양을 필요로 하는데, 이는 대용량 주기억 장치 데이터베이스를 구성하기 어렵게 한다. 다음으로 많은 프로세스와 잦은 프로세스간 통신으로 인하여 문맥 전환이 자주 일어나게 된다. 이로 인하여 시스템의 전체 성능이 저하된다. 또한 규격이 확대됨에 따라서 계속 증가하는 응용서비스 요소 동작을 매 번 별도의 프로세스로 구현할 경우, 그 수가 너무 많아져 이에 대한 관리가 어렵다.

3.1.4 고객관리 시스템과의 접속을 위한 프로세스

HLR 시스템은 가입자의 가입 및 해지, 부가서비스의 등록 및 삭제 등의 기능을 수행하기 위하여 고객관리 시스템과 연동한다. 그리고 이러한 엔진 프로세스와 고객관리 시스템의 응용 프로세스와의 통신을 지원하기 위하여 OmdSend(Omd: Operation and Maintenance data Distributing system) 및 OmdRecv 프로세스를 두어 TCP/IP 통신 기능을 수행한다. 즉, 데이터베이스 시스템의 엔진 프로세스가 직접 TCP/IP 통신을 해야 하는 오버헤드를 줄이고 이 기능을 별도의 프로세스로 구현한 것이다. 그러나, 이러한 고객관리 시스템과의 연동을

위한 부분에서 프로세스의 생성에 문제점이 있는 것으로 파악된다. 고객관리 시스템과는 송신 및 수신과 관련하여 각각 하나의 소켓 통신만이 필요하다. 그러나 송신과 수신을 각각 담당하는 OmdSend 및 OmdRecv 프로세스는 이러한 하나의 접속을 위하여 다시 자식 프로세스를 생성하도록 되어있다. 실제로 잦은 프로세스의 생성 및 소멸은 시스템의 성능 및 안정성에 좋지 않은 영향을 줄 수 있다.

3.1.5 장애 및 상태 처리의 많은 프로세스

HLR 시스템과 같은 통신 시스템은 그 중요성으로 인하여 각 구성 요소를 포함한 시스템의 장애와 상태가 매우 중요한 운용유지보수 기능이다. 그런데 현재의 장애 및 상태 처리의 경우 그 기능에 비하여 많은 프로세스가 동작하는 문제점이 있다. 현재 프로세스의 정상 동작 여부를 검사하기 위한 프로세스, 공통선 신호방식 프로토콜의 장애 정보를 처리하기 위한 프로세스, 대국 MSC/VLR에 HLR 시스템 상태를 전달하기 위한 프로세스, 그리고 하드웨어 장애 정보를 처리하는 프로세스 등이 있다. 현재와 같이 각 기능을 프로세스로 구분하면 구현은 용이하지만 프로세스간 통신이 필요하고, 이로 인하여 성능이 나빠지며 관리가 어려운 문제가 발생한다.

3.1.6 다른 시스템 간의 통신을 담당하는 프로세스

현재 HLR 시스템에서는 서로 다른 시스템간의 통신을 위하여 이를 별도로 담당하는 통신 처리 부분을 두고 있다. 이 부분은 주로 주제어컴퓨터와 운용유지 보수 시스템간의 통신을 지원하는 기능을 한다. 그리고 그 구성은 TCP/IP 송신을 담당하는 strSend 프로세스와 수신을 담당하는 strRecv 프로세스로 구성되어 있다. 이 통신 처리 부분에서도 잦은 자식 프로세스 생성이 문제점으로 생각된다. 프로세스의 생성을 위해서는 fork() 시스템 호출을 사용하는데, 이는 많은 자원을 사용하는 대표적인 시스템 호출이다. 또한 연결되는 시스템이 많아지면 관련된 자식 프로세스가 많아져 성능이 떨어지고 관리가 어렵게 되는 문제가 생긴다.

3.1.7 색인 구조의 문제점

3.1.7.1 엔진 프로세스 구동시의 초기 색인 구조의 구성에 따른 문제점

현재 구현된 T 트리와 변형된 선형 해싱 기법에서 사용하는 메모리 영역은 데이터베이스가 아닌 엔진 프로세스 내의 자료 구조로 되어 있다[11]. 따라서 엔진 프로세스 구동 시에 매번 새롭게 초기 색인 구조를 구성해야 한다. 이러한 구조를 사용하면 색인 구

조의 구성을 위하여 엔진 프로세스의 초기화 과정이 매우 길어지는 문제점이 있다.

3.1.7.2 T 트리 색인의 문제점

현재 구현된 T 트리 색인 방법에서 나타난 문제점은 사용하는 메모리 양이 많은 점과 가입자 수의 증가에 따른 검색, 삽입, 삭제 속도가 늦어진다는 점이다[12][13][14]. HLR 데이터베이스 시스템에서는 제한된 주기억 장치 용량을 고려해야 하므로 색인에 필요한 메모리 양을 최소로 하는 것이 중요한 요소가 된다. 또한 색인을 사용하는 주된 목적이 키를 사용한 검색을 빠르게 하기 위한 것이므로 검색 속도도 매우 중요한 요소이다. 여기에 튜플의 추가와 삭제에 따른 키의 추가와 삭제에 대한 오버헤드가 적을수록 보다 적합한 색인 방법으로 평가할 수 있다.

3.1.7.3 변형된 선형 해싱 기법 색인의 문제점

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[15]. 특히 엔진 프로세스가 새롭게 구동될 때마다 매번 초기 해싱 구조를 구성해야 하므로 삽입에 따른 디렉토리의 증가가 큰 오버헤드로 작용한다.

3.1.8 백업 방법의 문제점

3.1.8.1 잦은 위치 등록에 따른 성능 저하의 문제점

백업과 관련된 주된 작업은 휘발성인 주기억 장치의 내용을 안정된 디스크로 옮기는 것이다. 그러나 디스크 입출력은 주기억 장치에 대한 접근이나 일반 CPU 연산에 비하여 매우 시간이 많이 걸리는 작업이다. 그리고 이로 인하여 HLR 시스템 전체의 성능이 크게 저하될 소지가 있다. 따라서 HLR 데이터베이스 시스템에서도 정보가 손실되지 않아야 하는 작업만을 대상으로 백업을 실시하여야 한다.

HLR 데이터베이스 시스템에서 백업을 필요로 하는 작업은 가입자의 추가 및 삭제, 부가서비스의 등록 및 해지 등의 가입자 관련 운용 작업이다. 이들 정보의 신속한 백업을 위하여 기존의 방법과 같이 엔진 프로세스와 백업 프로세스가 갱신 플래그 테이블을 사용하여 즉시 백업할 필요가 있다. 그러나 실제 상황에서 블록이 갱신되는 주된 원인은 위치 등록에 의한 가입자의 위치 정보의 변경이고, 이는 신속한 백업이 필요치 않은 정보이다. 잦은 위치 등록으로 많은 메모리 블록이 갱신되고 이를 백업하기 위한 백업 프로세스의 과도한 디스크 입출력 발생으로 HLR 시스템의 성능에 매우 심각한 영향을 미치게 된다.

3.1.8.2 일관성 유지의 문제점

현재의 백업 방법은 단순히 변경된 메모리 SLD 블록을 디스크에 기록하는 기능을 한다. 이 방법은 엔진 프로세스에 의하여 변경된 내용이 백업된다는 보장도 되지 않으며, 나아가서 데이터베이스의 핵심 요구사항 중의 하나인 일관성 유지에 문제가 될 수 있다.

먼저 엔진 프로세스에 의한 변경이 백업된다는 보장이 없는 이유는 백업 프로세스가 별도로 동작하기 때문이다. 엔진 프로세스는 메모리 SLD 내용을 변경하고 해당 블록의 갱신 플래그를 1로 하고, 이렇게 변경된 블록은 백업 프로세스에 의하여 디스크에 기록된다. 그러나 이 과정에서, 즉 엔진 프로세스가 데이터를 변경했으나 백업 프로세스가 디스크에 기록하기 전에 시스템 오류가 발생하면 해당 변경 정보는 복구가 될 수가 없다. 따라서 이러한 과정에 대한 고려가 필요하다.

다음으로 현재의 방법은 근본적으로 일관성을 유지할 수 없는 문제점을 가지고 있다. 백업 프로세스가 메모리 내용을 디스크로 기록하는 과정에서 시스템 오류가 발생할 경우 백업 중이던 데이터에 대한 일관성을 유지할 수 없다. 또한 이 과정에서 카타로그 정보가 손상될 경우 해당 데이터베이스 자체가 손상되는 결과를 초래한다. 이렇게 되면 HLR 시스템 재기동 시에 최근의 백업 SLD를 사용할 수 없으므로 매우 많은 양의 정보를 손실하게 된다. 따라서 중요 정보에 대해 일관성을 유지할 수 있도록 개선할 필요가 있다.

3.2 HLR 시스템을 위해 제안하는 S/W 구조

3.2.1. 응용서비스 요소부 라이브러리

응용서비스 요소부의 많은 라이브러리의 각 함수는 각 응용서비스 요소부의 디렉토리에 분산하는 것이 바람직하다. 이렇게 필요한 함수를 각 응용서비스 요소부에 포함시킴으로써 소스의 이해가 쉽고, 또한 소스 코드의 관리를 용이하게 할 수 있다.

3.2.2 독립적인 ATIF 프로세스의 존재

ATIF 프로세스는 MAP 및 TCAP 프로세스의 기능으로 포함되어 수행될 수 있다. 따라서 프로세스 간 통신으로 인한 성능 저하를 피하기 위해서는 ATIF 프로세스의 기능을 MAP 프로세스와 TCAP 프로세스의 기능에 통합시키는 것이 바람직하다.

3.2.3 너무 많은 응용서비스 요소 프로세스

너무 많은 프로세스로 인해 발생하는 문제점을 해결하기 위해서는 응용서비스 요소 프로세스들을 각각의 기능

별 혹은 메시지 전달과 관련된 프로세스들로 묶어서 그 수를 줄이는 것이 바람직하다. 실제로 최근에 많이 사용하고 있는 쓰레드(thread)프로그래밍 기법을 사용하면 전체 구조에 큰 변화가 없이 프로세스의 통합이 가능하다.

3.2.4 고객관리 시스템과의 접속을 위한 프로세스

별도로 지식 프로세스를 생성하지 않고 OmdSend 및 OmdRecv 프로세스가 직접 접속을 관리하는 구조로의 개선이 필요하다.

3.2.5 장애 및 상태 처리의 많은 프로세스

현재의 장애 및 상태 처리 부분의 두 프로세스는 그 기능으로 볼 때 프로세스의 정상 동작 여부를 검사하기 위한 프로세스에 통합이 가능한 부분이다. 이러한 프로세스의 통합은 쓰레드 프로그래밍 기법을 사용하면 현재 프로세스 구조와 기능에 큰 변화가 없이 통합이 가능하다.

3.2.6 다른 시스템 간의 통신을 담당하는 프로세스

이는 전통적인 TCP/IP 접속 프로그램을 그대로 사용하면서 발생하는 문제점으로 이를 쓰레드 프로그래밍 기법을 사용하면 쉽게 해결할 수 있다. 즉, 각 시스템과의 접속을 하나의 쓰레드가 담당함으로써 수신에 대하여 각각 프로세스 하나로 구현이 가능하다.

3.2.7 HLR 시스템을 위해 제안하는 종합적인 S/W 구조

HLR 시스템 소프트웨어는 크게 데이터베이스 시스템, 응용서비스 요소부, 문답처리 응용부, 공통선 신호방식 처리부, 운용유지 보수부, 그리고 운용자 정합부로 구성되어 있다. 기존의 소프트웨어 구조에 주요 개선안을 적용한 전체 소프트웨어 구조는 그림 3.2와 같다.

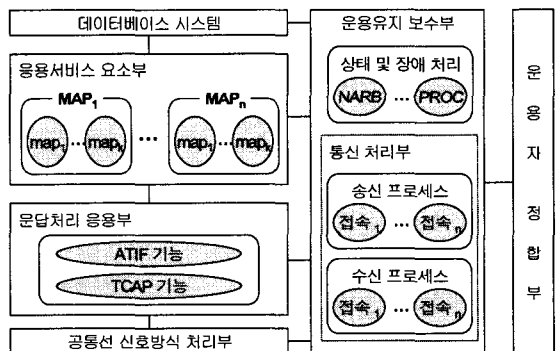


그림 3.2. 제안하는 HLR S/W 구조

그림에서 나타난 바와 같이 주요 변화 부분은 응용서비스 요소부의 프로세스 통합, 문답처리 응용부에서

ATIF 프로세스의 기능 수행, 운용유지 보수부에서 통신을 담당하는 프로세스의 개선과 상태 처리 기능을 수행하는 프로세스의 통합 등으로 가능한 전체 구조를 크게 변화하지 않고 개선하는 방법을 선택했다.

응용서비스 요소부의 문제점은 첫째, 특정 프로세스만의 기능을 수행하는 라이브러리는 해당 프로세스의 소스 코드에 포함시켜야 한다는 점과 둘째, 각 이동통신 응용부(MAP)동작이 별도의 프로세스로 구현되어 프로세스가 너무 많아진다는 점이다. 이를 해결하기 위해서는 여러 프로세스에서 공통적으로 사용되는 함수만을 라이브러리로 만들고, 응용서비스 요소 프로세스들은 기능별 혹은 메시지 전달과 관련된 프로세스들로 묶어서 그 수를 줄이면 된다. 응용서비스 요소부에서 프로세스의 수를 줄이기 위해서는 쓰레드 프로그래밍 기법을 사용하면 구현이 용이하다. 앞의 그림 3.2에서 응용서비스 요소부의 MAP_i는 여러 개의 MAP 동작이 하나의 프로세스로 통합되어 구현된 프로세스이다. 그리고 map_i은 하나의 프로세스 안에 구현된 각 MAP 동작을 나타내는 쓰레드이다. 따라서 권한자료 요청이나 권한자료 통보와 같이 그 기능이 비슷한 동작을 통합하여 하나의 프로세스로 구현하거나 위치등록과 위치삭제와 같이 하나의 동작이 다른 동작에 영향을 주는 프로세스들을 통합하여 구현할 수 있다. 이러한 통합으로 프로세스의 수를 줄임에 따라 메시지 통신과 문맥 전환이 줄어들며, 전체 프로세스가 사용하는 메모리 사용량도 줄어들게 된다. 반면에 MAP 동작이 쓰레드로 구현되므로 해당 프로세스 안에서 각 MAP 메시지의 분배 기능과 각 쓰레드별 메시지 통신 기능이 포함되어야 한다. 다음으로 응용서비스 요소부의 프로세스와 문답처리 응용부 프로세스 사이에 존재하는 ATIF 프로세스는 그 기능을 문답처리 응용부 프로세스에 포함 시킬 수 있다. 따라서 앞의 그림3.2에서 ATIF 기능이 문답처리 응용부 프로세스에 포함되어 있음을 알 수 있다. 그리고 이 기능은 응용서비스 요소부와 같이 쓰레드 개념을 사용하여 구현할 수도 있다. 이와 같이 ATIF 프로세스의 제거는 모든 MAP 메시지에 대하여 두 번의 메시지 통신을 줄이는 기능을 하므로 시스템의 성능 향상에 큰 도움이 될 것으로 예상된다. 그림에서 공통선 신호방식 프로토콜 부분도 MSC/VLR과의 통신을 담당하는 I/O 부분은 시스템의 성능에 큰 영향을 주므로 다수 개의 MAP 메시지를 하나로 묶어서 I/O 횟수를 줄이는 등에 대한 연구가 필요하다. 운용유지 보수부에서는 먼저 TCP/IP 통신을 위한 부분에 대한 개선이 필요하다. 즉, 고객관리 시스템과의 연동을 위한 통신 부분과 다른 시스템에 존재하는 프로세스 간의 통신을 지원하는 부분이 여기에

해당한다. 이 부분에서는 지식 프로세스를 생성하지 않고 송신과 수신에 대하여 각각 하나의 프로세스가 그 기능을 수행하는 구조로의 개선이 필요하다. 이를 위해서는 각 TCP/IP 접속을 지식 프로세스의 생성 대신에 쓰레드로 처리함으로써 쉽게 구현할 수 있다. 즉, 현재의 부모 프로세스는 주 쓰레드가 되어 각 TCP/IP 접속에 대하여 지식 쓰레드를 생성하면 되고, 각 지식 쓰레드는 각 시스템에 대한 TCP/IP 접속 기능을 수행하면 된다. 운용유지 보수부에서 개선되어야 하는 다음 내용은 장애 및 상태 처리 부분의 프로세스를 통합하는 것이다. 현재 이 부분의 프로세스들은 대부분 비슷한 기능을 처리하므로 쓰레드 기법을 사용하여 통합하는 것이 바람직하다. 앞의 그림 3.2에서도 장애 및 상태 처리부를 하나의 프로세스로 나타내었다.

3.3 제안하는 데이터베이스 시스템 구조

본 장에서는 제안하는 HLR 데이터베이스 시스템의 구조에 대하여 설명한다.

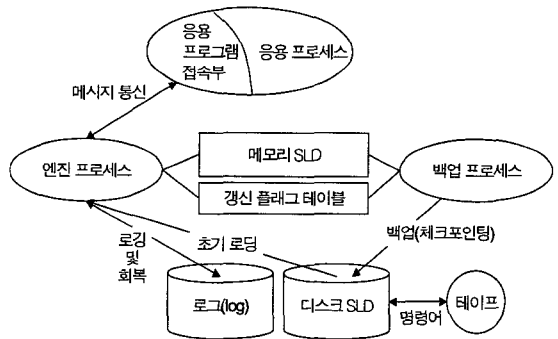


그림 3.3. 제안하는 HLR 데이터베이스 시스템 구조

그림 3.3은 이러한 데이터베이스 시스템의 구조를 나타낸다. 그림의 구조와 기존 데이터베이스 시스템과의 주된 차이점은 일관성 유지를 위한 로깅 기능이 추가되었으며, 백업 프로세스는 체크포인팅을 수행하도록 그 기능이 변경되었다는 점이다. 또한 테이프 백업 프로세스는 기존의 백업 명령어로 대체되었다.

3.3.1 엔진 프로세스

엔진 프로세스는 응용 프로세스의 데이터베이스 작업 요구를 받아 이를 수행하고 그 결과를 돌려주는 역할을 한다. 기존의 엔진 프로세스 구조는 초기화 처리 기능을 포함하여 메시지 관리부, 질의 처리부, 테이블 관리부, 그리고 색인 관리부로 구성되어 있다. 이러한 엔진 프로세스에서는 색인과 백업에 많은 문제가 있다. 본 절에서는 이러한 문제에 대한 개선안

을 수용한 엔진 프로세스의 구조와 기능에 대하여 설명한다.

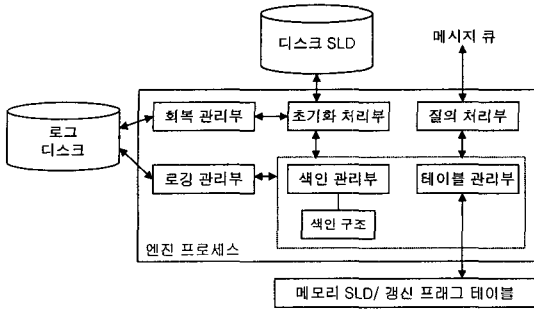


그림 3.4. 제안하는 엔진 프로세스의 구조

위의 그림 3.4는 제안하는 엔진 프로세스의 구조를 보여준다. 그림에서와 같이 새로운 엔진 프로세스에는 기존의 초기화 처리부, 질의 처리부, 테이블 관리부, 색인 관리부가 그 기능이 개선되어 포함되어 있으며, 회복 관리부와 로깅 관리부가 새롭게 추가되어 있다. 이 구조는 기존 엔진 프로세스의 골격을 유지하며 성능과 안정성을 향상시키는 방향으로 설계하였다.

그림에서 초기화 처리부는 기존의 초기화 기능을 포함하여 회복 관리부를 통한 파손시의 회복 기능과 새로운 색인 구조의 초기화 기능을 수행한다. 다음으로 질의 처리부는 기존의 데이터베이스 작업 이외에 가입자 정보 처리 작업을 별도의 메시지로 처리하는 기능을 수행한다. 테이블 관리부는 로깅 기능의 지원을 위한 기능이 포함된 것 이외에는 기존과 큰 변화가 없으며, 마지막으로 회복 관리부와 로깅 관리부는 데이터베이스 파손 시에 일관성을 유지하는 데이터베이스의 복원을 담당하는 부분이다. 본 절에서는 엔진 프로세스에 속하는 각 부분에 대하여 차례로 설명한다.

3.3.1.1 초기화 처리부

엔진 프로세스에서 수행하는 이전의 초기화 과정은 다음과 같다. 먼저 메시지 큐 식별자를 생성하고, 디스크 SLD로부터 메모리 SLD로 데이터베이스 로딩을 수행하며, 백업을 위한 갱신 플래그를 초기화 한다. 그리고 로딩된 메모리 SLD에서 테이블의 수, 튜플의 크기 등 주요한 값을 조사하고, 초기 색인 구조를 생성하며, 백업 프로세스와 테이프 백업 프로세스를 생성한다.

제안하는 엔진 프로세스에서는 기존의 기능이 변경되거나 새로운 기능이 추가된다. 먼저 기존 기능이 변경되는 부분은 초기 색인 구조를 생성하는 부분으

로 T 트리와 변형된 선형 해싱 기법의 함수를 이용하지 않고 이단계 색인과 버킷 연결 해싱 기법을 사용한다. 그리고 새로운 구조에서 테이프 백업 프로세스가 없으므로 이의 생성을 수행하지 않는다. 다음으로 추가되는 초기화 기능은 SLD를 로딩한 이후에 로그를 이용한 회복 기능을 수행하는 것이다.

변경되거나 추가되는 부분 이외의 초기화 기능은 기존 엔진 프로세스의 것을 그대로 사용한다. 또한 현재 테이블 및 속성의 카타로그 정보를 각각 관리하는 정보는 큰 변화 없이 사용이 가능하다. 다만 각 속성에 대한 색인 방법의 종류가 기존의 순차적 검색, T 트리, 변형된 선형 해싱의 세 가지에서 순차적 검색, 이단계 색인, 버킷 연결 해싱의 세 가지로 변경되는 점이 다르다.

3.3.1.2 질의 처리부

제안하는 질의 처리부는 기존 엔진 프로세스의 메시지 접속부와 질의 처리부를 통합한 기능을 수행한다. 기존의 메시지 접속부는 UNIX의 메시지 큐 함수를 사용하여 응용 프로세스에서 엔진 프로세스로의 데이터베이스 작업 요구와 엔진 프로세스에서 응용 프로세스로의 결과 전달을 하는 기능을 수행한다. 이 기능은 제안하는 엔진 프로세스의 질의 처리부에서 그대로 이용된다. 다음으로 기존 질의 처리부의 기능인 튜플의 검색, 삽입, 삭제, 변경 등의 메시지에 대한 처리 기능과 새롭게 추가된 메시지가 새로운 질의 처리부에 포함되어 처리된다. 제안하는 엔진 프로세스의 질의 처리부에서는 기존의 메시지 이외에 가입자 정보 처리와 관련된 추가 메시지를 제공한다. 이들은 실시간 처리보다는 안정된 백업 및 데이터의 일관성 유지에 초점이 맞추어진 메시지로, 각 메시지가 트랜잭션 개념으로 수행되어 실제 엔진 프로세스 내에서는 여러 개의 데이터베이스 작업이 일어난다. 이와 관련한 메시지의 종류에는 가입자 추가 메시지, 가입자 삭제 메시지, 가입자 정보 변경 메시지(부가서비스 등록 및 삭제)등이 있다. 다음은 이들 각 메시지를 트랜잭션으로 수행되는 절차의 예를 설명한 것이다.

가입자 추가 메시지

- ① 해당 가입자의 단말이 도난 가입자 테이블에 있는지 확인한다.
- ② 삽입 함수를 호출하여 가입자를 추가한다.
- ③ 갱신 함수를 호출하여 국번 테이블의 가입자 수를 증가시킨다.

가입자 삭제 메시지

- ① 삭제 함수를 호출하여 가입자 테이블에서 가입자

를 삭제한다.

- ② 갱신 함수를 호출하여 국번 테이블의 가입자 수를 감소시킨다.
- ③ 해당 가입자의 단말 번호가 도난 가입자 테이블에 있으면 이를 삭제한다.

가입자 정보 변경 메시지

- ① 갱신 함수를 호출하여 해당 가입자의 정보를 변경한다.
- ② 도난 가입자 속성이 변경되었으면, 도난 가입자 테이블에 추가하거나 삭제하는 기능을 수행한다.
위의 메시지 예에서 보았듯이 하나의 메시지는 여러 개의 데이터베이스 작업으로 구성된다. 그리고 이 작업은 원자적(atomic)성질을 갖는다. 따라서 메시지의 처리 과정에서 오류가 발생하면 메시지가 처리되기 이전의 상태로 복구되도록 구현되어야 한다. 그리고 삽입, 삭제, 갱신과 같이 데이터베이스 변경이 발생하면 그 사항은 로깅 관리부에 의하여 로그로 기록된다. 이는 데이터베이스의 트랜잭션 철회(abort)에 대해서는 질의 처리부에서 책임을 지며, 파손 시의 회복에 대해서는 로깅 관리부와 회복 관리부에서 책임을 지는 형태로 구현됨을 의미한다.

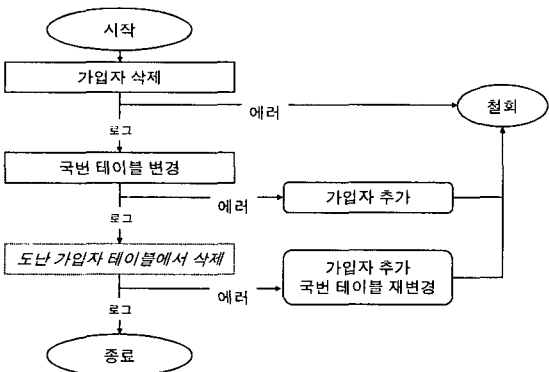


그림 3.5. 가입자 삭제 메시지의 처리 과정

위의 그림 3.5는 질의 처리부에서 가입자 삭제 메시지를 처리하는 과정을 보여준다. 그림에서 가입자 삭제 과정에서 에러가 발생하면 별다른 회복 동작 없이 바로 철회가 된다. 그러나 국번 테이블의 변경이나 도난 가입자 테이블에서의 삭제 과정에서 에러가 발생하면 회복을 위하여 가입자의 추가나 국번 테이블의 재변경의 회복 동작이 수행된 후 철회된다. 그리고 각 갱신 작업은 로그로 기록되어 파손에 대비한다. 기존의 질의 처리부에서는 검색, 삽입 등 각 메시지에 대하여 db-selectTup(), db-insertTup() 등의 함수

수를 호출하여 그 기능을 수행하였다. 따라서 새롭게 추가되는 가입자 추가 메시지, 가입자 삭제 메시지, 가입자 정보 변경 메시지에 대해서도 위에서 설명한 과정으로 기능을 수행하는 db-insertSub(), db-deleteSub(), db-updateSub()의 함수가 추가로 구현되어야 한다.

3.3.1.2 테이블 관리부

기존의 HLR 데이터베이스 시스템은 가입자 테이블, 국번 테이블, 도난 가입자 테이블의 세 가지 테이블을 지원한다. 테이블 관리부는 이들 세 개의 테이블을 일차원 배열 형태로 UNIX의 공유 메모리인 메모리 SLD에서 관리한다. 그리고 튜플의 삽입과 삭제는 각 튜플의 첫번째 바이트인 USED 필드와 카타로그 정보의 프리 리스트를 사용하여 이루어진다.

3.3.1.3 색인 관리부

HLR 데이터베이스 시스템은 호처리를 위한 실시간 처리가 중요하고 이를 위해서는 시스템 특성에 맞는 최적의 색인 방법을 설계하여 사용하여야 한다. 기존의 T 트리와 변형된 선형 해싱 기법 대신에 이단계 색인 방법과 버킷 연결 해싱 기법을 처리한다[5][6].

3.3.1.4 로깅 관리부

시스템의 오류로 인하여 데이터베이스가 파손되었을 경우, 데이터베이스를 일관성을 유지하는 최근의 상태로 만들기 위해서는 주로 로깅이 사용된다 [17][18]. HLR 데이터베이스 시스템에서는 가입자 관련 동작에 대하여 트랜잭션 개념을 도입해야 하고, 데이터베이스가 이에 대하여 일관성을 유지해야 한다. 기존 방법은 이러한 트랜잭션 일관성을 유지할 수 없으므로 가입자 관련 작업에 대해서는 트랜잭션 및 로깅 기법을 도입할 필요가 있다. 로그의 종류에는 물리적 로그와 논리적 로그가 있다. 물리적 로그는 어느 주소에서 어떤 내용이 어떻게 바뀌었는지를 저장하며, 논리적 로그는 어떤 동작이 있었는지를 저장한다. HLR 데이터베이스 시스템의 경우 가입자 관련 동작에만 로그가 필요하므로 논리적 로깅 방법을 사용하여 구현할 수 있다. 다음의 그림 3.6은 제안하는 HLR 데이터베이스 시스템에서 사용하는 로그의 구조와 종류를 보여준다.

로그 구조 <TransactionID, OperationID, TupleContent>
Operation 종류: START, COMMIT, INSERT, DELETE, UPDATE

그림 3.6. 로그의 구조와 종류

로그의 구조는 어느 트랜잭션인지를 나타내기 위한 트랜잭션 식별자(TransactionID), 어떠한 데이터베이스 동작인지를 나타내는 동작 식별자(OperationID), 그리고 그 동작에서 사용한 튜플을 저장하는 튜플 내용(TupleContent)으로 구성되어 있다. 그리고 데이터베이스 동작에는 트랜잭션의 시작과 끝을 알리는 START와 COMMIT이 있으며, 데이터베이스 변경이 일어나는 동작인 삽입(INSERT), 삭제(DELETE), 그리고 갱신(UPDATE)이 있다.

로그의 구조에 변경되기 이전의 튜플 내용을 저장하지 않는 이유는 주기억장치 데이터베이스 시스템의 특성상 UNDO 기능은 필요로 하지 않고, REDO 기능만 필요로 하기 때문이다. 즉, 휘발성인 주기억장치의 특성으로 UNDO는 자동적으로 이루어지므로 이전의 튜플 내용은 저장할 필요가 없다. 반면에 종료(COMMIT)된 트랜잭션의 경우는 변경 이후의 튜플 내용을 사용하여 REDO를 수행해야 하기 때문에 이를 안정된 저장 장치, 즉 디스크에 저장하여야 한다. 다음의 그림 3.7은 그림 3.5에서 설명한 가입자 삭제 메시지, 즉 가입자 삭제 트랜잭션의 처리에 따른 로그의 예를 보여준다. 그림과 같이 가입자 삭제의 경우는 도난 가입자 여부에 따라서 총 4 개 혹은 5개의 엔트리로 로그가 구성된다. 따라서 시스템 파손 시에 이 로그를 사용하면 해당 가입자의 삭제 기능을 재수행할 수 있다.

```
<1001,START,null>
<1001,DELETE,(0114551644,9F48001E,.....)>
<1001,UPDATE,(011455,.....)>
<1001,DELETE,(9F48001E,.....)>
<1001,COMMIT,null>
```

그림 3.7. 가입자 삭제 트랜잭션의 로그

모든 가입자 관련 작업은 로그로 그 결과가 남는다. 따라서 시스템을 장기간 운용하게 되면 로그가 너무 많아지고, 이로 인하여 시스템 오류에 따른 회복 시에 많은 시간이 걸리게 된다. 이 문제는 백업 프로세스의 체크포인트링과의 관계에 의하여 해결된다. 즉, 백업 프로세스가 메모리 SLD를 디스크 SLD로 백업하는 체크포인트링이 완료되면, 최근의 데이터베이스가 저장되었으므로 이전의 로그들은 디스크에서 삭제된다. 이러한 체크포인트링 작업은 가입자 관련 작업이 일어나지 않는 야간이나 공휴일에 실시한다. 결국 체크포인트링 시점에서 트랜잭션이 발생하지 않으므로 자연스럽게 트랜잭션 일관성을 유지하는 데이터베이스 백업이 가능한 것이다.

3.3.1.5 회복 관리부

질의 처리부에서는 로깅 관리부의 함수를 사용하여 가입자 관련 운용 기능에 대하여 로그를 남긴다. 이러한 로그는 시스템 파손 시에 회복을 위하여 사용된다. 따라서 회복은 시스템 재가동 시에 엔진 프로세스가 디스크 SLD를 메모리 SLD로 로딩한 후에 이루어진다. 즉, 엔진 프로세스의 초기화 처리부에서 회복 관리부의 함수를 호출하여 회복 기능을 수행한다. 회복을 수행하는 과정은 다음과 같다. 먼저 디스크에 저장된 로그를 뒤에서부터 차례로 최근의 체크포인트링 시점까지 검색하면서 종료(COMMIT)된 트랜잭션들을 찾아낸다. 다음으로 찾아낸 트랜잭션들은 저장된 로그의 순서대로 REDO 기능을 수행한다. 반면에 종료 로그가 없는 트랜잭션을 그냥 무시함으로써 자연스럽게 UNDO 되도록 한다. 회복 관리부에서는 초기화 처리부에 initial_recovery() 함수 하나만을 제공한다. 즉, 초기화 처리부에서는 데이터베이스를 로딩한 후에 이 함수를 호출하여 회복을 수행한다. 이 함수에서는 종료된 트랜잭션들을 찾아서 로딩된 주기억장치 데이터베이스에 REDO 기능을 수행한다. 결과로 트랜잭션 일관성이 유지된 데이터베이스가 구성되어 정상적인 서비스를 계속 진행할 수 있다.

3.3.2 응용 프로그램 접속부

HLR 데이터베이스를 액세스하고자 하는 응용 프로세스들은 응용프로그램 접속부에서 제공하는 함수들을 사용한다. 따라서 응용 프로그램 접속부에서는 엔진 프로세스와의 메시지 통신을 통하여 데이터베이스 작업을 수행하는 튜플의 검색, 삽입, 삭제, 변경 등의 함수를 제공한다. 새로운 HLR 데이터베이스 시스템에서도 이와 동일한 구조를 사용하나 색인 및 백업 등의 구조 변화에 따라서 응용 프로그램 접속부에서 제공하는 함수가 보완되어야 한다. 새로운 응용 프로그램 접속부에 추가되는 함수는 주로 가입자 관련 운용 기능에 관련된다. 즉, 트랜잭션 및 로깅 기법의 도입으로 가입자 관련 운용 기능이 그 종류에 따라 각각 별도의 메시지로 구분되고, 이에 따라서 응용 프로그램 접속부에서 각각 별도의 함수를 제공해야 한다. 응용 프로그램 접속부에서 추가되는 함수의 종류는 가입자 관련 운용과 관련하여 추가되는 메시지의 종류와 동일하다.

IV. 백업 방법에 대한 연구

본 장에서는 현재 HLR 데이터베이스 시스템에서

사용하고 있는 백업 방법의 문제점을 바탕으로 새로운 효과적인 백업 방법을 제안한다.

4.1 HLR 데이터베이스 시스템을 위한 효과적인 백업 방법

HLR 시스템에서는 하드웨어를 신뢰성 있는 결합 허용 컴퓨터를 사용할 뿐 아니라 데이터베이스 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 두 가지 종류의 트랜잭션에 대하여 방법을 달리하는 백업 방법을 제안한다. 신속한 백업을 필요로 하지 않으며 빠른 응답 시간을 요구하는 위치 등록과 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 위치 정보가 다음 번의 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 위치 정보를 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 위치 등록 횟수에 비하여 대다수 가입자의 위치정보가 쉽게 변하지 않는 특성을 가지고 있다. 따라서 위치정보의 변경과 같은 데이터베이스 갱신 정보는 하루 혹은 이를 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 데이터베이스 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 가입자 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 변경하고, 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.

두 가지 종류의 트랜잭션을 구분하여 백업하는 이 방법은 기존의 구조를 크게 변경하지 않고 구현이 가능한 장점을 가지고 있다. 또한 이 방법에 의하여 잦은 위치 등록에 따른 성능 저하의 문제를 해결할 수 있을 것으로 생각한다. 그러나, 기존의 백업 프로세스의 방법을 그대로 사용할 경우에 여전히 해결되지 않은 점은 디스크에 기록하는 과정에서 오류가 발생할 경우 일관성 유지가 되지 않는다는 점이다. 이러한 문제점에 대한 개선 방법에 대해서는 다음 절에서 설명한다.

4.2 일관성 유지의 문제점을 위한 개선안

현재의 백업 방법은 단순히 변경된 메모리 SLD 블록을 디스크에 기록하는 과정으로 데이터베이스 일관성 유지에 문제가 있음을 지적하였다. 이 문제를 해결하기 위해서는 가입자 관련 운용 기능에 대해서 트랜잭션 개념의 도입과 로깅과 체크포인트를 통한 회복 기법이 필요하다. 위치등록에 의한 위치 정보 변경은 신속한 백업을 수행하지 않는다. 반면에 가입자 관련 운용 기능은 그 기능별, 즉 가입자 추가, 가입자 삭제, 부가서비스 등록, 부가서비스 삭제 등으로 그 기능을 구분하여 트랜잭션으로 만든다. 이들 트랜잭션은 각각 별도의 메시지로 구분되어 운용유지보수 시스템이나 고객 관리 센터의 응용 프로세스에서 엔진 프로세스로 전달되고, 엔진 프로세스에 의하여 그 기능이 수행되며, 수행 결과만을 다시 응용 프로세스에 전달한다.

현재의 검색 요구 메시지, 삽입 요구 메시지 등은 각각 하나의 데이터베이스 작업을 수행한다. 그러나 위의 가입자 추가 메시지의 경우는 엔진 프로세스에서 여러 개의 데이터베이스 작업을 수행해야 한다. 따라서 엔진 프로세스는 가입자 추가 메시지를 받아서 가입자 중복 및 단말 번호의 중복 방지를 위한 검색, 가입자 튜플의 추가, 해당 가입자가 속한 국번 테이블의 가입자 수 변경 등의 기능을 하나의 트랜잭션으로 취급하여 처리한다. 그리고 그 결과로는 이 트랜잭션의 성공 혹은 실패 여부와 실패인 경우 그 이유를 응답 메시지로 구성하여 응용 프로세스에 전달한다. 가입자 추가 요구 메시지 이외에 가입자의 삭제, 부가서비스 등록, 부가서비스 삭제 등의 메시지도 이와 비슷한 트랜잭션으로 구현되어야 한다.

그리고 이들 트랜잭션의 시작과 끝을 포함하여 메모리 SLD의 변경 내용은 모두 디스크에 로깅된다. 이 로그는 시스템이 오류로 인하여 재시동 될 경우 회복에 사용된다. 또한 장기간 운용에 따라 로그가 너무 많아지는 것을 고려하여 시스템의 최한 시에 메모리 SLD의 내용을 디스크로 백업하는 체크포인팅 작업을 수행한다. 이러한 체크포인팅 작업은 가입자 관련 운용 작업이 일어나지 않는 야간 혹은 공휴일에 실시한다. 이는 체크포인팅 시점에 트랜잭션이 중단되어야 하는 문제를 자연스럽게 해결하는 것이다.

HLR 시스템이 전원 차단 등의 오류로 인하여 재시동 될 경우 회복은 매우 간단한 과정에 의하여 이루어진다. 디스크 기반 데이터베이스와는 달리 주기억 장치 데이터베이스의 내용이 모두 사라지므로 별도의 UNDO 기능은 필요로 하지 않는다. 따라서 최근의 체크포인트 시점에서 저장된 디스크 SLD에 해당 시점 이후에 종료(COMMIT)된 트랜잭션의 로그들을 순차적으로 적용하여 REDO 작업을 하면 되는 것이다.

위에서 설명한 과정에 의하여 작업이 정상적으로 종료된 가입자 관련 기능은 시스템이 재시동 된 이후에도 일관성을 유지하며 복구가 된다. 또한 가입자 관련 기능에 대해서만 로깅을 위한 디스크 작업이 일어나므로 성능에도 큰 영향을 주지 않는다.

4.3 백업 프로세스

HLR 데이터베이스 시스템에서 기존의 백업 프로세스 기능은 엔진 프로세스에 의하여 변경된 주기억장치 데이터베이스, 즉 메모리 SLD의 내용을 디스크로 백업하는 기능을 한다. 기존 방법은 잦은 위치등록 동작에 의한 시스템 전체 성능 저하의 원인이 된다. 이는 이동통신 응용부의 데이터베이스 기능과 가입자

관련 운용 기능을 구분하여 수행함으로써 해결된다. 따라서 새로운 구조에서 백업 프로세스의 기능이 수정되어야 한다.

엔진 프로세스에서 이미 살펴본 바와 같이 가입자 관련 운용 기능의 결과는 트랜잭션으로 구분하여 로깅이 이루어진다. 그리고 로그의 양이 많아지고 이로 인하여 회복 시간이 길어지는 것을 막기 위하여 체크포인팅 작업이 필요하다. 새로운 데이터베이스 시스템 구조에서는 백업 프로세스가 이러한 체크포인팅 작업을 수행한다. 따라서 기존과 같이 지속적으로 변경된 메모리 SLD 내용을 디스크로 백업하는 것이 아니라 특정 체크포인트에만 메모리 SLD를 디스크로 백업하는 것이다.

백업 프로세스가 체크포인팅 작업을 수행하는 시점은 주기적인 경우와 운용자의 요구에 의한 경우의 두 가지로 나눌 수 있다. 먼저 주기적인 경우는 가입자 관련 작업이 발생하지 않는 야간이나 공휴일에 체크포인팅 작업을 수행하는 것이다. 이 기능은 UNIX의 "cron" daemon의 도움을 받아 쉽게 구현할 수 있다. 즉, cron이 지정된 특정 시각에 백업 프로세스에 시그널(signal)을 전송하고, 백업 프로세스는 이 시그널 처리 루틴에서 체크포인팅 작업을 수행하는 것이다. 다음으로 운용자의 요구인 경우는 메시지를 통하여 그 요구가 백업 프로세스에 전달되고 백업 프로세스는 체크포인팅 작업을 수행하고 그 중간 결과와 최종 결과를 운용자에게 전달한다. 이는 네트워크 장애 등의 이유로 많은 가입자 관련 작업이 한꺼번에 이루어진 경우에 로그의 양이 많아져 수행하는 기능이다.

백업 프로세스는 위의 두 가지 경우에 의하여 다음과 같이 체크포인팅 작업을 수행한다. 먼저 체크포인팅 작업 도중의 오류에 대비하여 기존의 디스크 SLD 파일을 다른 파일로 복사한다. 다음에는 백업이 시스템에 미치는 성능 저하를 고려하여 전체의 일정 비율(예: 10%)로 메모리 SLD 내용을 디스크로 백업하고 일정 시간을 작업을 중지(sleep)한다. 그리고 일정 비율씩 백업되었을 때마다 상태 메시지를 사용하여 작업의 진행 과정을 운용자에게 알린다. 마지막으로 작업이 완료되었으면 이를 운용자에게 알리고 로그의 내용을 삭제한다. 백업 프로세스가 체크포인팅 작업을 수행하는데 있어서 갱신 플래그 테이블을 사용하면 디스크 관련 입출력을 줄일 수 있다. 특히 체크포인팅 작업이 자주 일어날 경우 메모리 SLD의 많은 데이터 블록이 변경되지 않은 상태를 유지할 수 있다. 이때 갱신 플래그 테이블을 참조하여 변경된 메모리 블록만을 디스크에 기록함으로써 백업 시간을 줄일 수 있다. 이 방법을 사용할 경우에는 디스크에 기록

후에 해당하는 블록의 갱신 플래그를 OFF하는 기능이 포함되어야 한다.

4.4 테이프 백업 기능

기존의 HLR 데이터베이스 시스템에서는 SLD 파일의 중요성을 고려하여 디스크 SLD를 테이프로 백업하는 별도의 테이프 백업 프로세스를 두고 있다. 테이프 백업 프로세스는 디스크의 파손에 대비한 것으로 운영자의 요구를 메시지로 받아서 테이프 백업 기능을 수행한다. 그러나 이 기능은 UNIX에서 제공하는 안정된 명령어로 수행이 가능하다. 따라서 제안하는 HLR 데이터베이스 시스템 구조에서는 테이프 백업을 위하여 별도의 프로세스를 두지 않고, 이러한 UNIX 명령어를 사용한다.

테이프 백업 기능을 수행하는 UNIX 명령어에는 “tar”와 “cpio”가 있다. 이들 명령어를 사용하면 디스크에서 테이프로의 백업, 테이프에서 디스크로의 복원 이외에 테이프 내용의 확인, 여러 파일의 백업 및 복원 등의 다양한 작업을 수행할 수 있다. 또한 이러한 명령어는 디바이스의 종류에 관계없이 그 사용 방법이 동일하므로 디바이스 변화에 따른 기능 구현이 불필요하다. 따라서 테이프 백업 기능을 위하여 기존과 같이 별도의 프로세스를 구현하는 것보다 이러한 UNIX 명령어를 사용하는 것이 바람직하다.

V. 결론

본 논문에서는 현재 이동통신 망에서 운용중인 HLR 시스템에서 발견된 문제점을 바탕으로 성능을 고려한 효과적인 S/W구조 및 백업 방법 등을 제안하였다. 이러한 개선점은 향후 Ubiquitous Network 및 위성통신에서의 HLR 시스템 개발에도 도움이 될 것이다.

한편 기존의 백업 방법이 위치등록에 의한 위치정보의 변경과 가입자 관련 운용 기능에 의한 정보 변경을 구분하지 않고 백업함으로써 시스템 성능 저하의 문제가 발생한다. 이 논문에서는 이 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.

기존 기법에서는 초당 약 800번 이상 발생하는 위치 등록으로 인한 실시간 백업이 과부하로 작용하였

으나, 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 서비스로의 진화에 기반한 이동 상거래(mobile commerce)에 있어서 가장 중요한 요구 사항은 보안 기능일 것이다. 이러한 기능 수행을 위해 현재의 HLR DB 스키마(Scheme)와 관련하여 변화되어야 할 최소한의 속성(attribute)은 다음과 같다. 세션 키(session key) 설정 및 상호 인증 프로토콜 수행을 위해서, 상호 인증 알고리즘 수행의 결과로 얻어지는 결과값 및 세션 키, 또 이 알고리즘에 사용된 난수(random number) 등을 표현할 수 있는 속성들을 추가하여야 한다. 향후의 IMT2000 서비스로의 진화와 함께 이질적 네트워크들과의 로밍이 본격화되기 전에 HLR과 VLR간에 교환되는 세션 키 및 인증 관련 변수들에 대한 효과적인 보안 기능 개발이 절실하다.

그 밖의 향후 연구 방향은 다음과 같다. 개인 이동성(personal mobility) 지원에 따른 신호(signalling) 트래픽의 증가를 최소화시키기 위해 단말의 상태 정보를 구분하여 관리하기 위한 속성 추가도 필요하다. 또한 새로운 부가서비스 지원을 위하여 동적인 스키마 변경을 지원하기 위한 연구가 요구된다. 가입자 수가 증가함에 따라 발생하는 HLR 시스템간의 가입자 이동에 대한 고려가 필요하며, 가입자 테이블과는 별도로 관리가 필요한 특수 부가서비스에 대한 연구가 있어야 한다.

참고 문헌

- [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 Storage Manager," *Proc. 20th Intl. Conf. on Very Large Data Bases*, Santiago, Chile, 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 Engineering*, Vol. 4, No 6, Dec. 1992.
- [5] 김장환, 이충세, "이동 통신 HLR 시스템에서의

효과적인 색인 및 백업기법”, 한국정보과학회 논문지 Feb. 2003.

- [6] Knuth, D., *The Art of Programming, Vol. 3, Sorting and Searching*, Addison-Wesley, 1973.
- [7] 한국통신기술협회, *MSC/VLR-HLR/AC간 MAP 신호방식 표준*, 1997.7; *IMT-2000 3GPP2-TIA/EIA-41 개정 A 판에 순응하는 무선 시스템의 국제적 제공*, 2002.10.28
- [8] Rhee, M. Y., *CDMA Cellular Mobile Communications and Network Security*, Prentice Hall, 1998.
- [9] Kim, J. H. and Rhee, C. S., "Traffic Estimation Algorithms in Mobile Networks," *Proc. 4th Intl. Conf. on Advanced Communication Technology*, pp. 139-141, Feb. 2002.
- [10] Son, S. H., *Advances In Real-Time Systems*, Prentice Hall, 1995.
- [11] Lehman, T.J. and Carey, M.J., "A Study of Index Structures for Main Memory Database Management Systems," *Proc. 12th Intl. Conf. on Very Large Data Bases*, Kyoto, Japan, pp. 294-303, Aug. 1986.
- [12] Bayer, R. and McCreight, C., "Organization and Maintenance of Large Ordered Indexes," *Acta Infomatica*, Vol. 1, No. 3, pp. 290-306, 1972.
- [13] Bayer, R., "Symmetric Binary B-Trees: Data Structure and Maintenance Algorithms," *Acta Infomatica*, Vol. 1, No. 3, pp. 290-306, 1972.
- [14] Comer, D., "The Ubiquitous B-Tree," *ACM Computing Surveys*, Vol. 1, No. 2, pp. 121-137, 1979
- [15] Litwin, W., "Linear Hashing: A New Tool For File and Table Addressing," *Proc. 6th Intl. Conf. on Very Large Data Bases*, Montreal, Canada, pp. 212-223, Oct. 1980.
- [16] TIA/EIA, *IS-41(A)*, Initial Version. Jan. 1991.; *IMT-2000 3GPP2-International Implementation of Wireless Telecommunication Systems Compliant with TIA/EIA-41 Revision A*, Oct. 2002.
- [17] 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.
- [18] 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 관리자.

<관심분야> Mobile & Wireless Communication, Performance Analysis of Networks, Database System, Information Security, Mobile Multimedia, Mobility Managements, Mobile Embedded System, Ubiquitous Computing, 알고리즘 및 계산이론, 정보통신 경제 예측