

# 주기억장치 상주형 DBMS를 위한 실시간 LDAP Interface에 관한 연구

이 정 배\* · 차 상 균\*\* · 김 환 철\*\*\* · 박 병 관\*\*\*

## 요 약

우리는 정보통신의 발달로 인해서 정보의 호수 속에서 살고 있다. 이러한 이유로 해서 많은 사용자들은 많은 양의 정보를 빠르게 검색하기를 원한다. 본 논문에서는 이러한 요구사항을 만족하기 위해서 주기억장치 상주형 DBMS를 이용한 실시간 LDAP Interface를 제안하여 고속의 검색을 지원하고자 한다. 이를 위해서 디스크 기반의 DBMS 대신 주기억장치 상주형 DBMS를 대신하여 고속의 검색을 지원하여 응용의 변경 없이 서비스를 제공할 수 있게 해준다.

## The Study on Real-time LDAP Interface in used Main Memory Resident Database System

Jeong-Bae Lee\* · Sang-Gyun Cha\*\* · Hwan-Chul Kim\*\*\* · Byung-Kwan Park\*\*\*

## ABSTRACT

We live in the flood of information due to advancement of information communication and increase of E-mail. Managing users huge information systematically and speedy searching are needed in these social advancement. In this thesis, in order to satisfy these requirement, We suggested Real-time LDAP Interface using Main Memory Resident Database Management System which can manage a lot of information fast systematically. It is expected that system can provide advantage of performance improvement through replacing Main Memory Resident Database Management System without change of application which is required high speed process.

키워드 : 주기억장치상주형(Main Memory Resident), LDAP, 고속검색(High Speed Process)

### 1. 서 론

정보통신기술의 발달로 인해 고속의 처리를 요구하는 응용분야가 점차 확대되고 있다. 현재 널리 사용되고 있는 디스크 기반 데이터베이스 시스템에서는 데이터를 다루기 위한 디스크 액세스 오버헤드가 지나치게 크므로 빠른 처리를 요구하는 응용에는 적합하지 않다.

더구나 주기억장치의 용량이 커지고 가격이 많이 하락함에 따라 컴퓨터 시스템내의 주기억장치 용량은 점점 증가하는 추세이다. 이에 따라 데이터베이스 분야에서는 늘어나는 주기억장치의 용량을 최대한 활용하여 디스크 내에 저장된 데이터를 모두 주기억장치로 상주시켜 데이터베이스 시스템의 성능을 개선하는 주기억장치 데이터 베이스 시스템(Memory resident database system)에 관한 연구가 활발히 진행 중이다.

또한 인터넷의 급격한 사용 증가와 인터넷을 이용한 네트워크 서버 및 정보 시스템의 사용증가로 인한 다양한 정보들을 체계적으로 구축하고 네트워크에 연결되어 있는 컴퓨터를 어느 곳에서나 손쉽게 검색하여 서비스를 받을 수 있는 디렉토리 시스템과 이를 인터넷을 통해 접속할 수 있는 LDAP에 관한 관심이 고조되고 있으며, 사용자들은 더 빠른 결과를 얻고자 하는 요구사항이 급증을 하고 있는 추세이다. 이러한 요구사항을 만족하기 위해서는 기존의 디스크기반 데이터 베이스 보다 대용량 데이터의 고속처리 응용을 위한 성능향상 도구로서 활용될 수 있는 주기억장치 상주형 DBMS를 이용한 디렉토리 서비스 인터페이스를 제공하고자 한다.

주기억장치 상주형 데이터베이스 시스템이란 주기억장치에 모든 데이터를 상주시키는 데이터베이스 시스템으로서, 데이터에 대한 액세스를 디스크 입출력보다 훨씬 빠른 주기억장치 액세스만으로 해결하고자 한다. 주기억장치 상주형 데이터베이스는 다음과 같은 특성을 가지고 있다. 첫째, 주기억장치에 대한 액세스는 디스크에 대한 액세스보다 훨씬

\* 종신회원 : 선문대학교 컴퓨터정보학부 교수  
\*\* 정 회 원 : (주) 코메스타 이동통신개발팀 팀장  
\*\*\* 정 회 원 : 선문대학교 컴퓨터정보학부 교수  
논문접수 : 2004년 8월 3일, 심사완료 : 2004년 11월 9일

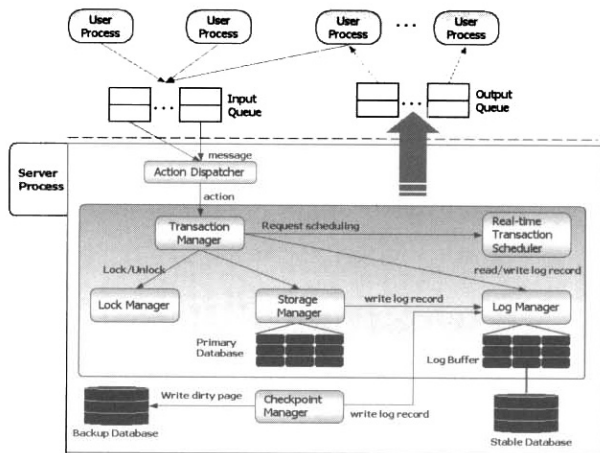
빠르다. 둘째, 대부분의 주기억장치는 휘발성을 갖는다. 비휘발성 주기억장치도 존재하지만 가격대 성능의 비가 낮다. 셋째, 주기억장치는 임의의 액세스(random access)가 가능하므로 액세스되는 데이터의 위치에 상관없이 고른 액세스 시간(access time)을 보장한다. 넷째, 주기억장치는 프로세스에 직접 액세스(direct access)가 가능하다. 주기억장치 상주형 데이터베이스는 이러한 특성을 살려 디스크기반 데이터베이스에서 제공하는 기능을 보다 효율적으로 제공한다[1].

본 연구의 동기는 네트워크 자원의 빠른 접근을 하기 위해 기존의 디스크기반 데이터베이스를 대신해 주기억장치 상주형 데이터베이스를 이용하여 사용자들에게 빠른 서비스를 해줄 수 있는 시스템을 설계 및 구현을 통한 성능비교를 통해 고속처리를 위한 디렉토리 서비스 제공에 대한 연구를 하는 것을 목적으로 한다.

## 2. 관련 연구

### 2.1 주기억장치 상주형 DBMS - Kairos

Kairos 시스템은 전자통신연구원(ETRI)에서 개발된 실시간 응용을 위한 주기억장치 상주형 제정 시스템인 Mr.RT 2.5버전을 기술이전 받아 (주)리얼타임테크에서 개발한 주기억장치 상주형 실시간 데이터베이스 구축을 위한 저장 시스템이다[1].



(그림 1) Kairos 시스템의 구성요소

(그림 1)과 같이 Kairos 서버 프로세스는 동시에 여러 액션을 수행하기 위해 다중 쓰레드 구조를 갖는다. (그림 1)에서 액션 디스패처와 체크포인트 관리자는 독립적인 쓰레드로 수행되며, 다른 관리자들은 하나의 액션 수행 쓰레드(Action Processing Thread)를 구성한다.

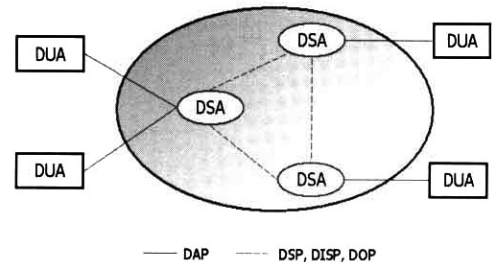
T-트리 인덱스 구조, ECBH 기법, 실시간 특성을 지원하기 위한 새로운 인덱스 Hybrid-TH를 지원한다. 동시성 제어 를 위해서는 컨테이너를 잠금 단위로 하는 2PL-PI를 사용하고, 응용에 따라서 순차 수행을 선택하는 것이 가능하다.

다. 또한, ARIES 기반 회복 기법을 지원하며 로깅 기능은 필요에 따라서 선택할 수 있는 선택 기능이 제공된다.

### 2.2 X.500

X.500 디렉토리 서비스는 정보 통신망에 필요한 정보들을 특정한 규칙에 맞게 데이터베이스로 구축할 수 있는 여러 가지 규정과 구축한 정보를 바탕으로 사용자의 요청을 처리할 수 있는 여러 가지 편리한 기능 및 서비스를 규정하여 놓은 것이다[2, 3].

X.500 디렉토리 서비스는 ITU와 ISO/IEC/JTC/SC21 협동 프로젝트로 1985년부터 개발에 착수하여 1988년도에 X.500 시리즈 권고사항으로 발표된 이래 각 국가에서도 표준화 단체 및 관련 기관에 의해 꾸준히 표준화 작업이 진행되고 있다. 1992년까지는 많은 기관들이 X.500 디렉토리 서비스 서버를 설치하여 약 300,000개의 엔트리를 가지며, 370여 기관에서 서비스가 이루어 졌다[4].



(그림 2) X.500 디렉토리 서비스 시스템

(그림 2)에서와 같이 X.500 디렉토리 서비스 시스템은 디렉토리 서비스를 요구하는 디렉토리 사용자와 디렉토리와 사용자간의 인터페이스를 담당하는 DUA(Directory User Agent), 그리고 디렉토리를 관리하며 사용자의 요구를 처리하는 DSA(Directory System Agent)로 구성된다. 디렉토리에 사용자의 요구를 전달하고, 디렉토리로부터 사용자의 요구에 대한 응답을 받는 응용 프로세스인 DUA는 디렉토리와 사용자 사이의 인터페이스 역할을 수행한다. 디렉토리 내에 존재하는 정보들은 각각의 시스템에 분산되어 관리, 운영되기 때문에 이를 디렉토리 내에서 디렉토리 프로토콜을 이용하여 디렉토리 정보를 보유하고 있는 시스템들과 상호 협력해야 한다. 분산된 디렉토리 시스템에서 사용자의 요구를 디렉토리 프로토콜에 따라 수행하는 응용 프로세스들이 DSA(Directory Server Agent)이다. DSA는 사용자가 지시한 명령이나 요구들에 대해서 디렉토리를 통해서 그것을 수행하고 그것의 결과를 사용자에게 응답하는 역할을 하며 디렉토리에는 여러 개의 DSA가 존재할 수 있다. DUA는 DSA에 있는 접근 점을 통하여 디렉토리에 접근하게 되는데 하나의 DSA는 하나 이상의 접근 점을 가지고 있다. 이들 구성요소 사이에 사용되는 프로토콜은 다음과 같다.

- DAP(Directory Access Protocol) : 디렉토리 서비스를 요구하는 하나의 DUA와 DSA 사이의 프로토콜
- DSP(Directory System Protocol) : 디렉토리 서비스의 연결을 지원하는 두 DSA 사이의 프로토콜
- DSIP(Directory Information Shadowing Protocol) : 복제 정보의 교환을 지원하기 위하여 shadowing 등의 선정에 관한 두 DSA 사이의 프로토콜
- DOP(Directory Operational Binding Protocol) : 두 DSA 사이의 관리 정보 교환에 관한 프로토콜

이와 같이 X.500 디렉토리에서 클라이언트와 디렉토리간의 통신을 위하여 사용되는 프로토콜인 DAP은 많은 기능이 있고, 일반적인 기업환경에 필요 없는 기능이 다수 포함되어 있으며, 프로토콜의 복잡성으로 인한 구현의 어려움이 있고, 클라이언트에 과다한 부하가 걸린다는 문제가 있다. 따라서 DAP의 복잡성을 해결하고, 일반적인 기업환경에 적합한 디렉토리 클라이언트 사이에 새로운 프로토콜인 LDAP(Lightweight Directory Access Protocol)이 개발되었다.

### 2.3 LDAP(Lightweight Directory Access Protocol)

LDAP은 인터넷 TCP/IP 사용자에 OSI 프로토콜로 구성된 X.500 디렉토리 서비스 시스템을 접근할 수 있도록 인터넷 TCP/IP 프로토콜을 기반으로 구축된 프로토콜이다. DAP이 OSI 기반으로 설계되었기 때문에 매우 복잡한 체계를 가지고 있어 접속하는 클라이언트와 디렉토리에 많은 부하를 주게 되지만, LDAP은 TCP/IP를 기반으로 하고 있으므로 클라이언트와 디렉토리에 대한 부하를 줄였다. 따라서 현재 개발되고 있는 대부분의 디렉토리 서비스 시스템은 LDAP에 기반 하여 개발되고 있다[5].

DAP에 비해 LDAP이 가지는 장점을 살펴보면 다음과 같다.

#### • TCP/IP 기반의 프로토콜

이전의 DAP이 기반으로 하고 있는 OSI 7계층과는 달라. LDAP에서 기반으로 하고 있는 TCP/IP는 4계층으로 구성되어 있어 캡슐화된 데이터의 길이가 OSI의 경우보다 짧고, 그 결과 전송률 또한 개선된다. 또한 기존에 네트워크를 사용하는 대부분의 사용자들이 TCP/IP를 기반으로 하고 있는 인터넷 사용자이므로 이들 사용자들에 대한 LDAP을 구축하기가 편리하다.

#### • 불필요한 연산자 수 감소

LDAP에서 비슷한 기능을 하는 DAP연산자들을 하나의 연산자로 표현하여 연산자 수를 줄이고 그 결과 사용자들이 고려해야 할 연산자 수가 줄어들어 서비스를 편리하게 받을 수 있다.

#### • 간단한 문자열 사용

데이터를 전송하기 위해 DAP이 복잡한 ASN.1방식의 문

자열을 사용하는 것에 비해 LDAP은 간단한 문자열을 사용한다. DAP에서는 데이터의 ASN.1방식의 표현과 DER 인코딩으로 인한 복잡성이 문제가 되었으나, LDAP에서는 데이터 원소들과 특히 DN에 대하여 간단한 스트링 인코딩을 허용함으로써 데이터 길이를 줄이고 단순화하였다.

#### • URL 사용

DAP에서는 디렉토리 서버의 주소로써 DSA의 접근점을 사용하였는데, LDAP에서는 이를 개선하여 URL을 LDAP서버의 주소로 사용하여 사용자들이 쉽게 이용할 수 있다.

LDAP은 기본적으로 사용자의 이용 편리성에 중점을 두어 개발된 디렉토리 접속 프로토콜로서, IETF에 의해 RFC 1487표준으로 제정되었다. 현재는 버전2가 RFC 1777 표준으로 정의되어 있고, 버전 3는 RFC 2251로 정의되어 대다수 디렉토리 서버 업체에서 버전 3을 지원하는 디렉토리 서버 개발을 하고 있다. RFC에서 정의된 것처럼 LDAP의 주요 목적은 클라이언트의 복잡성을 최소화시키고 디렉토리 서비스를 이용하는 응용 프로그램의 효율성을 높이는 것이다[2, 6, 7].

현재 대부분은 네트워크 이용자들은 인터넷을 광범위하게 사용자고 이러한 인터넷은 기본적으로 TCP/IP를 기반으로 하고 있다. 이에 LDAP은 기존의 DAP과 동일한 기능을 사용자들에게 제공하면서도 TCP/IP상에서 동작할 수 있도록 구현되었다. 또한 사용자들이 고려해야 할 연산자들의 수를 줄여 사용자들이 서비스를 받는데 고려해야 할 부분 중 많은 부분을 개선하였다. 결국 LDAP은 기존의 DAP과는 다른 형태를 가지게 되어 X.500디렉토리 서버와는 호환성을 가질 수 없게 되었는데, 이를 해결하기 위하여 LDAP에서는 LDAP서버를 두었다.

### 3. MMDBMS를 이용한 실시간 LDAP Interface 설계

X.500이나 LDAP등과 같은 범용 디렉토리 서비스 표준을 만족하는 많은 상용화된 시스템들은 매우 다양하고 안정된 서비스를 제공하고 있으나, 본 논문에서는 상용화된 제품들의 서비스의 질보다는 주기억장치 상주형 DBMS 기반으로 동작하는 실시간 LDAP 서비스를 제공함으로써 실시간 서비스의 활용이 가능한 핵심적인 기능과 간소화된 디렉토리 구조를 설계하여 운영될 수 있는 프로토타입을 본 장을 통하여 설계하고자 한다.

또한 본 논문에서는 이러한 서비스를 제공하기 위해서는 LDAP서버가 주기억장치 상주형 DBMS에 접근이 가능하게 하기 위해서는 DBMS의 표준 인터페이스를 제공해주어야 하는데, 이는 ODBC로 접근을 해줄 수 있게 하였으며, 본 논문에서는 이러한 기능을 제공할 수 있는 기본적인 함수를 분석하여 설계하였다.

3.1 기능

주기억장치 상주형 데이터베이스를 접근하는 다양한 종류의 어플리케이션이 존재하고, 일정한 주기 동안 정보들을 저장, 검색, 삭제 등의 기능들을 하게 된다. 이러한 여러 종류의 고속처리 응용 프로그램들을 운영하기 위해서는 특정 응용 프로그램만을 위한 접근점이 아닌 일정한 표준으로 제공함과 동시에, 필요시에 데이터에 대한 접근을 효율적으로 제공하여 주는 표준 인터페이스가 필요로 하게 된다.

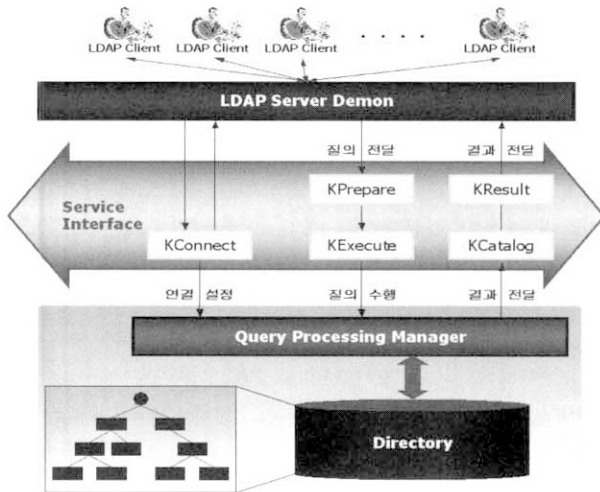
또한 기존의 X.500이나 LDAP을 기반 하는 디렉토리 시스템들은 효율적이고 경제적인 네트워크 관리를 위하여 사용되어져 왔으나, 투명한 위치정보를 제공해주어야 하는 고속처리 응용에서는 많은 한계점을 있는 단점을 가지고 있게 된다.

본 논문에서 제시하는 시스템을 구축하기 위해서는 고속의 디렉토리 서비스가 요구되는 시스템을 위한 주기억장치 상주형 DBMS 접근을 위해서 다음과 같은 기능을 제공한다.

- MMDBMS 접속 기능 : Kairos Server의 논리적인 data source 대한 접속을 형성
- MMDBMS 질의 처리 기능 : Kairos Server의 논리적인 데이터를 얻기 위한 SQL문을 입력받아 처리 그에 해당하는 질의 처리
- MMDBMS 질의 결과 처리 기능 : Kairos Server로부터 실행된 SQL문에 해당하는 데이터의 결과를 처리하는 기능을 제공
- MMDBMS 접속 해제 기능 : 접속이 형성된 Kairos Server의 논리적인 data source 대한 접속을 해제

3.2 구조

본 논문에서 주기억장치 상주형 데이터베이스를 이용한 실시간 LDAP Interface는 (그림 3)와 같이 구성된다.



(그림 3) 실시간 LDAP Interface 구조

본 논문에서 주기억장치 상주형 데이터베이스를 이용한 실시간 LDAP Interface는 (그림 3)과 같이 구성된다. 본 시스템은 단지 LDAP Server Demon과 같은 특정 응용에서만 사용할 수 있는 것이 아니라 다른 여러 응용에서도 사용될 수 있으며, 본 논문에서는 Kairos Server와 LDAP Server Demon간의 중간자 역할을 수행하게 된다.

다음은 연결을 위한 각 구성요소에 대한 설명이다.

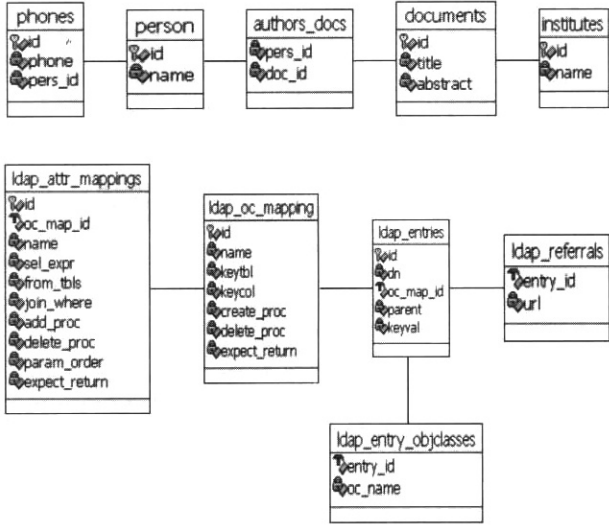
구성요소	기능
Connect	LDAP Client가 LDAP Server로 접속(bind, unbind, Abandon)을 하면, LDAP Server가 Kairos Server로의 접속과 해제를 하는 기능을 제공함(접속시 사용자 이름, 패스워드, 데이터소스 이름을 명시)
Prepare	LDAP Client로부터 요청된 질의는 Kairos Server에 맞는 질의 형태로 매핑 된 접속한 LDAP Client가 원하고자 하는 LDAP Operation(search, add, modify, delete, compare)들의 연산을 수행할 수 있도록 해주기 위한 SQL 질의 요청 준비를 하는 기능을 제공함
Execute	Kairos Server에는 디렉토리 모델에 따라 분류되어 있는 Object별로 관리하는 객체들로 구성되어 있으므로 각 객체들을 검색, 변경, 삭제 등의 기능을 수행하는 기능을 제공함
Catalog	수행된 질의의 결과를 얻기 위한 메타데이터(meta data)를 제공하는 기능을 제공함
Result	수행된 질의의 결과를 제공할 수 있는 ResultSet형태로 변환을 해서 LDAP Server에게 제공하는 기능을 제공함

X.500의 디렉토리 정보 모델에는 DIB(Directory Information Base) 구현에 대한 상세한 기술을 포함하고 있지 않는 대신, DIT(Directory Information Tree)라는 구조적인 형태를 제안하였다. 그러나 DIT는 DIB의 구조적인 형태만을 표준화한 것이고 실질적인 DIB구축에 관한 문제는 무슨 모델로 어떻게 구현하느냐에 따라 그 성능이 크게 좌우 될 수 있다[8, 9].

디렉토리 정보들을 저장하기 위한 방법으로 파일 시스템을 사용하거나, 관계형 데이터 모델, 또는 객체지향 모델에 입각한 데이터 베이스들을 사용하는 방법들이 사용된다. 각 방법들은 각각 장단점을 동시에 가지고 있기 때문에 디렉토리의 성능을 위하여 구현 도메인의 특성에 따라서 저장 방법이 선택되어야 할 것이다. 본 논문에서는 고속의 처리를 위해서 주기억장치 상주형 데이터베이스를 사용할 것이다.

본 논문에서는 주기억장치 상주형 데이터베이스 기반의 LDAP Service를 위한 디렉토리의 구축방안으로 X.500의 DIB 모델에 따라 객체 지향 모델링을 통한 구축 방안을 선택하였다. DIB의 객체지향 모델을 이루고 있는 각 Object Class들은 X.500과 LDAP에서 정의한 Object Class들 중에

서 필요한 부분만을 선택하여 간소화한 것이고, 본 논문에서는 조직(Organization) 내에서 일하는 사람들(Persons)에 대해서 정보를 저장하는 것으로 가정하였다.



(그림 4) 디렉토리 모델링

(그림 4)는 객체지향 모델링을 사용하여 조직 내에서 일하는 사람들에 대한 정보를 저장하는 객체들을 설계한 것이다.

4. MMDBMS를 이용한 실시간 LDAP Interface 구현

본 시스템은 주기억장치 상주형 데이터베이스 시스템 기반의 LDAP Service를 제공하기 위한 프로토타입 시스템으로써 기존의 디렉토리 시스템이 주기억장치 상주형 데이터베이스가 접근할 수 있는 기능을 제공해주어야 하므로 기존의 데이터베이스 표준 인터페이스에 비하여 구현 범위가 좁고, 다양한 기능들을 지원하지 못한다. 상용화된 제품들이 제공하는 기능들을 모두 지원하기에는 매우 방대한 시간과 비용을 투자하여야 한다. 따라서 본 시스템의 목표와는 부합되지 않으므로, 구현의 범위를 축소하며, 주기억장치 상주형 데이터베이스에 접근을 해서 기본적인 연산만을 수행할 수 있는 핵심적이고 기본적인 기능을 제공할 수 있도록 구현하였다.

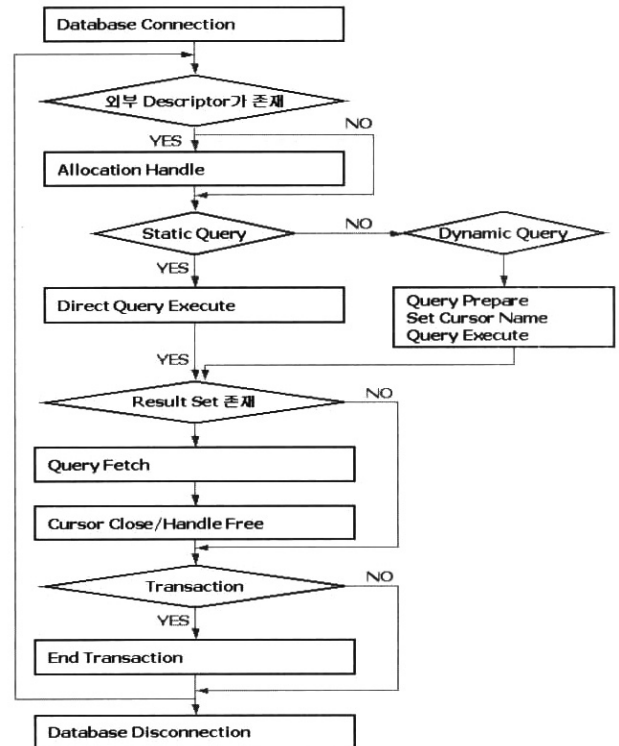
4.1 접근절차

본 논문에서 제안하는 MMSBMS를 이용한 실시간 LDAP Interface를 구현하기 위해서는 LDAP Server와 MMDBMS를 연결을 해주어야 하는데, 이를 위해서는 ODBC로 연결을 해야 한다[11, 12]. 그러나 본 논문에서 제안하는 실시간 LDAP Interface를 위해서는 ODBC에서 지원하는 모든 기능을 지원해 줄 필요는 없다. 그래서 ODBC표준을 분석하여 필요한

기능만을 발췌하였다. 이는 빠른 처리 속도를 요구로 하는 응용을 위한 주기억장치 상주형 데이터베이스 접근 인터페이스라고 할 수 있다. 이러한 기능을 제공함으로써 특정 응용만을 위한 접근만을 제공한다면 개발자의 입장에서 보면 Kairos Server의 내부의 기능을 이해하여야 하는 번거로움이 필요로 하므로 어플리케이션을 개발하는 것이 쉽지 않을 것이다.

본 논문에서는 실시간 LDAP 서비스를 제공하기 위해서는 MMDBMS에 접근을 하기위한 표준 인터페이스를 제공해야 한다. 이를 위해서 본 논문에서는 ODBC를 이용해서 MMDBMS에 접근을 해야 한다. ODBC는 DBMS 접근 표준 인터페이스를 기반으로 해서 여러 응용에 독립적인 인터페이스를 제공함으로써 빠른 처리속도를 필요로 하는 클라이언트 어플리케이션의 개발에 융통성을 부여하였다.

(그림 5)는 사용자 또는 여러 응용에서 ODBC를 통해서 Kairos Server로 접근 절차를 보여 주고 있으며 아래와 같은 Flow를 기반으로 해서 Kairos Server에 접근을 해야 한다.



(그림 5) ODBC를 이용한 MMDBMS 접근절차

<표 1>은 (그림 5)에서 제시한 MMDBMS 접근절차를 만족하여 여러 응용에서 Kairos Server로 접근을 해서 처리 결과를 얻을 수 있는 기본적인 기능을 정의해 놓은 것이다.

<표 1> ODBC 제공함수

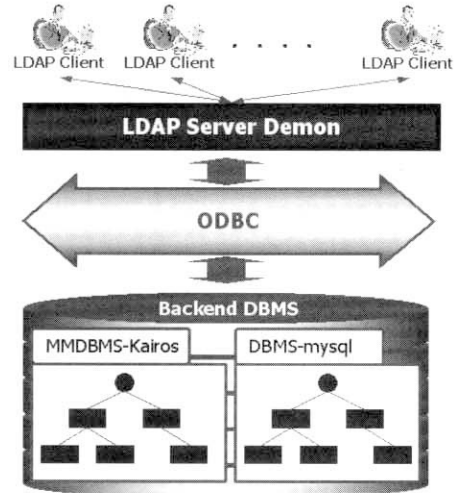
Task	Function Name	Purpose
데이터 소스 연결	SQLAllocEnv	환경 핸들을 얻음
	SQLAllocConnect	연결 핸들을 얻음
	SQLAllocStmt	Statement 핸들을 얻음
	SQLConnect	Data Source, user ID, Pwd로 특정 드라이버에 접근
SQL 요청 준비	SQLPrepare	문장 핸들 할당
	SQLBindParameter	파라미터 공간할당
	SQLGetCursorName	커서 이름을 얻음
	SQLSetCursorName	커서 이름을 할당
요청 수락	SQLExecute	준비된 문장 실행
	SQLExecDirect	문장을 실행
	SQLNumParams	파라미터 개수 리턴
질의 결과	SQLRowCount	질의에 영향을 받은 row갯수 리턴
	SQLNumResultCol	Result Set내의 column 개수 리턴
	SQLDescribeCol	Result Set내의 한 column 정보 리턴
	SQLColAttribute	Result Set내의 한 column의 Attribute 정보 리턴
	SQLColAttribute	Result Set내의 한 column의 Attribute 정보 리턴
	SQLBindCol	Result Column에 공간을 할당하고 data type 설정
	SQLFetch	여러 result row 리턴
	SQLGetData	Result Set의 한 row 내의 전부 또는 일부 column 정보 리턴
카탈 로그	SQLColumns	Column목록 리턴
	SQLStatistics	통계정보 리턴
	SQLTable	테이블 이름 리턴
연결 해제	SQLFreeStmt	Statement 자원 해제
	SQLCancel	SQL 문장 실행 취소
	SQLDisconnect	접속을 해지

Kairos 시스템은 현재 SQL-92의 주요기능을 지원하고 있다. Kairos SQL은 클라이언트 라이브러리인CLI(Call Level Interface)와 Kairos 서버의 질의 처리관리자에 의해 수행된다. CLI는 질의를 질의 처리 관리자에 전달하고 질의 수행 결과의 유지 및 상위 인터페이스에 이를 전달하는 역할을 한다. 질의 처리 관리자는 CLI로부터 전달된 질의를 수행하여 그 결과를CLI에게 전달하게 된다. ODBC는 상위 어플리케이션에서 요구하는 연산을 수행하기 위한 기능들을 제공해 주며, 질의 처리관리자와의 연결과 질의 전달, 결과 저장을 수행 할 수 있는 기능을 제공하여야 하며, 3장의 설계 내용을 기반으로 해서 구현을 하였다.

5. 성능 분석 및 평가

본 절에서는 실시간 LDAP Interface를 LDAP 서비스를 수행하기 위한 시험 환경을 구축한다. 시험 환경을 구축하

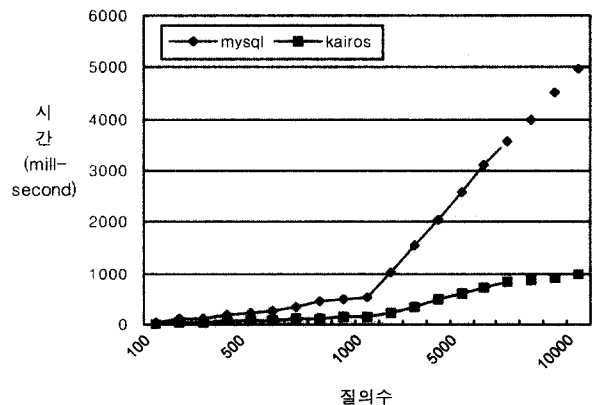
기 위해서는 디렉토리에 저장되어 있는 다양한 엔트리 정보들을 개념적인 트리로 나타낸 디렉토리 정보 트리(DIT : Directory Information Tree)를 구성하는 것이 필요하며, DBMS는 MMDBMS-Kairos와 일반적으로 많이 사용하는 RDBMS인 My-sql을 사용하여 성능 실험을 하였다.



(그림 6) 실험환경

실시간 LDAP Interface를 통해서 LDAP 서비스를 통하여 성능실험을 하였는데, 이는 엔트리의 수와 LDAP연산을 변경하면서 결과값이 나오는 시간을 통해 성능을 비교 평가하였다.

<표 1>과 (그림 7)은 실험의 결과를 나타낸다. 실험 결과는 ldap연산, 엔트리의 수에 관계없이 처리결과가 디스크 기반 데이터베이스보다 성능이 우수한 것으로 나타났다. 이러한 결과는 데이터베이스를 주기억장치에 상주시킴으로써 디스크에 접근하는 I/O가 없어 처리 속도 시간이 짧아지므로 고속의 처리를 원하는 응용에 적합하다는 결론을 내릴 수 있다.



(그림 7) ldap\_search 비교 그래프

〈표 2〉 실시간 LDAP Interface를 이용한 성능평가

DBMS	Operation	Entry #	Response Time (단위 : millisecond)
mysql	ldap_search	100	53
		1000	541
		10000	4957
	ldap_add	-	48
	ldap_delete	-	49
	ldap_modify	-	43
Kairos	ldap_search	100	15
		1000	152
		10000	983
	ldap_add	-	8.3
	ldap_delete	-	8.8
	ldap_modify	-	8.1

6. 결론 및 향후 연구방향

인터넷의 급격한 사용 증가와 인터넷을 이용한 네트워크 서버 및 정보 시스템의 사용 증가로 인한 다양한 정보들의 체계적인 구축을 기반으로 해서 고속의 처리를 필요로 하는 응용이 점차 증가하는 추세이다.

특히 이동통신과 전자 메일의 사용 증가와 컴퓨터를 통한 전자상거래 확산, 그리고 사용자가 급격히 늘어나고 있는 홈뱅킹과 같이 사용자의 거대한 정보를 관리하고 검색하기 위한 시스템이 필요한 곳에서는 더욱 디렉토리 시스템이 필요할 것으로 판단된다. 인터넷을 통해 위와 같은 디렉토리 시스템에 손쉽게 접근하여 사용할 수 있는 LDAP 프로토콜은 디렉토리 시스템의 활용과 이용을 촉진하는 활력소라고 할 수 있다.

또한 현재 많이 사용되고 있는 디스크 기반 데이터베이스 시스템에서는 데이터 처리를 위한 디스크 입출력 시간 때문에 빠른 처리를 요구하는 응용분야에는 적합하지 않다. 더욱이 기억 장치의 용량이 커지고 가격이 하락함에 따라서 대용량 메모리의 사용이 가능해졌다. 따라서 이러한 빠른 처리를 요구하는 응용을 효율적으로 지원하기 위해서는 주기억장치 데이터베이스 시스템이 필요하다. 따라서 본 논문에서는 기존의 디스크 상주형 데이터베이스 기반의 디렉토리 서비스와는 달리 주기억장치 상주형 데이터베이스 기반으로 디렉토리 서비스를 해줄 수 있는 서비스 API를 설계 및 구현하여 고속의 처리를 요하는 응용에서 운영될 수 있는 시스템의 프로토타입을 제시하였다.

본 연구는 앞으로 ODBC API의 기능을 수정 및 보완하는 형태로 발전시켜 나갈 것이다. 첫 번째로 데이터베이스 접근 표준에 의거하여 개발된 다른 시스템과의 상호 연동도 중요한 과제이다. 두 번째로는 현재는 특정 응용에서 시험을 해 보았으므로 다른 응용과의 상호 운용을 함으로써 시

스템의 검증도 필요할 것이다.

본 연구로 인해 고속처리를 원하는 응용의 변화 없이 기존의 데이터베이스를 주기억장치 상주형 데이터 베이스로의 교체만으로 시스템의 성능향상을 할 수 있는 장점을 제공할 수 있을 것으로 기대된다.

참 고 문 헌

- [1] "Implementation and Performance Evaluation of Multi-thread S/W Model for Customized Service in Advanced Intelligent Network," *Journal of Mathematical Modeling and Science Computing*, Vol.6, April, 1996.
- [2] 윤성순, 탁성우, "X.500과 LDAP의 비교 및 LDAP 프로그래밍", *한국정보과학회 가을 학술발표 논문집*, Vol.24, No.2, pp.555-558, 1997.
- [3] ITU-T Recommendation Q.511, "Information Technology-Open System Interconnections The Directory : Abstract Service Definition," 1995.
- [4] 진성일, 이규철, 김명호 외, "분산이종 DB 연동 기술 연구", 국방과학연구소, Dec., 1999.
- [5] T. Howes, S. Kille, "Lightweight Directory Access Protocol v3," RFC 2251, Dec., 1997.
- [6] Kong-Seon Lee, Eunjung Lee, Kisong Yoon and Myung-Joon Kim, "Development of a Reconfigurable Java Client for LDAP Interface," *ICACT 2000*, pp.31-38, Muju, Feb., pp.16-18, 2000.



이 정 배

e-mail : jblee@sunmoon.ac.kr  
 1981년 경북대학교 전자공학과 전자전공 공학사  
 1983년 경북대학교 대학원 전산 전공 공학석사  
 1995년 한양대학교 대학원 전자공학과 공학박사  
 1982년~1991년 한국전자통신연구원 선임연구원  
 1991년~2002년 부산외국어대학교 컴퓨터공학과 부교수  
 2002년~현재 선문대학교 컴퓨터정보학부 부교수  
 관심분야 : 임베디드 시스템, 실시간 운영체제, 임베디드 프로토타이핑, 이동멀티미디어방송, 디지털방송, 데이터방송, 등



차 상 균

e-mail : kkandol@comesta.com  
 2000년 충남대학교 컴퓨터공학과(석사)  
 2002년 충남대학교 컴퓨터공학과 대학원  
 2002~현재 (주)코메스타 이동통신개발팀 팀장  
 관심분야 : MIMDBMS(Main Memory DBMS), 분산객체 기술, 디지털 방송, 위성통신



**김 환 철**

e-mail : hkim@comesta.com

1984년 경북대학교 전자공학과 전자계산기  
전공 공학사

1986년 한국과학기술원 전산학과 공학석사

1986년~2000년 한국전자통신연구원 책임  
연구원

2000년~현재 (주)코메스타 대표이사

2003년~현재 선문대학교 컴퓨터정보학과 겸임교수

관심분야 : 소프트웨어공학, 이동멀티미디어방송, 위성통신방송,  
디지털방송, 데이터방송, 광대역 이동 고속무선방송,  
SDR(Software Defined Radio) 등



**박 병 관**

e-mail : bkpark@sunmoon.ac.kr

1982년 한양대학교 전자공학과 학사

1990년 KAIST 전산학과 석사

1982년~1992년 ETRI 컴퓨터구조연구실  
선임연구원

1993년~2000년 (주)유니크테크놀로지 대표  
이사

2000년~2002년 (주)유니와이드테크놀로지 이사

2003년~현재 선문대학교 컴퓨터정보학부 객원교수

관심분야 : 컴퓨터구조, 스토리지 시스템, Computer Aided En-  
gineering