

디렉토리 서버의 과부하 및 동기화 문제 해결을 위한 LDAP 서버 아키텍처의 재구성

Reconfiguration of LDAP Sever Architecture for Resolving Directory Server Overloads and Synchronization

김 혜 원*
Hye-Won Kim

한 지 인**
Ji-In Han

이 병 곁***
Byong-Gul Lee

요 약

많은 기업 및 기관들이 그들의 업무정보를 인터넷을 통해 공유함에 있어서 얼마나 원하는 정보를 정확하게 그리고 신속하게 검색할 수 있는가에 대한 관심이 높아지고 있다. 따라서 업무정보의 신속한 저장, 검색 및 관리를 지원하기 위한 디렉토리 서버의 표준으로 LDAP 프로토콜이 제안되었고 이미 세계적 표준으로 자리 잡아 가고 있는 실정이다. 하지만 기존의 LDAP 프로토콜만으로는 서버의 과부하 및 데이터 동기화 등의 문제에 효과적으로 대처하기 어렵다. 본 연구에서는 LDAP의 등장 배경 및 특징에 대해 살펴보고, LDAP 운용환경에서 발생할 수 있는 서버의 과부하와 데이터 동기화 문제를 해결하기 위하여 서버 독립형 방식의 아키텍처를 제안하고 그 특징을 기술한다. 특히 서버 독립형 방식은 기존의 LDAP 아키텍처 위에 설계되었기 때문에 확장성과 이식성도 함께 보장할 수 있다는 장점이 있다.

Abstract

Many companies and departments are increasingly concerned of searching their business information in correct and faster way as they share the information through the internet. LDAP has been proposed for directory server standards which supports storing, searching and managing the business information, and it has become a global standard. Current LDAP, however, cannot solve problems such as server overloads and data synchronization. This paper describes the history and characteristics of LDAP, and proposes an architecture for solving those problems. Since the proposed architecture is designed based on the existing LDAP architecture, it guarantees extensibility and portability of server configuration.

1. 서 론

많은 기업들과 기관들이 네트워크 시스템에 그들의 업무정보들을 의존하고 있으며 이들이 인터넷을 통해 공유되고 있는 시점에서 이제는 얼마나 자신이 원하는 정보를 정확하게 그리고 빨리 찾아낼 수 있는가에 대한 관심이 집중되고 있다. 현재 대부분의 기관들에서는 서로의 정보를 공유하기 위해 주로 데이터베이스 시스템을 매개체로

이용하고 있다. 하지만 공유하고자 하는 정보의 양이 방대할 뿐만 아니라 기업마다 자기들의 정보를 독립적으로 구축하고 관리하고 있기 때문에 이들 정보에 대한 통일되고 일관성 있는 접근과 공유가 어렵다는 문제가 있다. 이는 한 기업이나 기관 내의 문제뿐만 아니라, 기업 간의 상호 협력을 통해 더 큰 효과를 누리고자 하는 현대의 산업 구조에서는 더욱 심각하고 비효율적인 문제가 아닐 수 없다. 따라서 현재와 같이 분산 관리되고 있는 다양한 정보들을 얼마나 효율적으로 관리할 수 있는가가 향후 정보화시스템의 고도화 달성에 가장 중요한 이슈가 될 것임이 분명하고, 이에 대한 해결책으로 가장 주목받고 있는 기술이 디렉토리서비스 기술이다[1].

* 준회원 : 서울여자대학교 정보통신공학부 학부과정
grease0805@hotmail.com

** 준회원 : 서울여자대학교 정보통신공학부 학부과정
omgod@unitel.co.kr

*** 정회원 : 서울여자대학교 정보통신공학부 조교수
byongl@swu.ac.kr

디렉토리 서비스는 여러 다양한 정보들을 디렉토리 서버에 저장하고, 저장된 정보들을 사용자들의 요구에 알맞게 제공해 주는 역할을 한다. 이러한 디렉토리 서버에 저장된 정보들을 사용자들이 쉽게 접근하고 검색할 수 있게 하기 위해서 LDAP (Lightweight Directory Access Protocol) 이라는 프로토콜이 제안되어있다. LDAP은 모든 형태의 디렉토리형 자료를 표준화된 방식으로 저장하고 검색하기 위한 통신규약으로서 미국 미시간 대학에서 ITU-T의 X.500을 근거로 개발되었다[2]. 하지만 현재의 LDAP 프로토콜만으로는 디렉토리 서버에 트랜잭션이 집중될 때 발생하는 서버 과부하 문제와 여러 서버에 분산 저장되어 있는 디렉토리 정보의 동기화 문제를 해결하지 못한다는 문제가 있다. 본 연구에서는 LDAP 서버 운용 환경에서 나타나는 문제점들을 서버독립형 방식의 아키텍처 재구성을 통해 해결하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 LDAP의 기본구조 및 특징에 대해 살펴보고, 3장에서는 LDAP 관련 연구 및 문제점에 대해 기술한다. 4장에서는 LDAP 서버 과부하 및 동기화 문제점들을 보완하기 위한 LDAP 운용 아키텍처의 재구성 방법에 대해 논한다. 5장에서는 문제의 해결 방안으로서 서버 독립형 방식을 소개하고 설계 알고리즘을 기술한다. 마지막으로 6장에서는 제안된 아키텍처를 바탕으로 향후 보완 및 고려해야 할 사항에 대해 설명하고 결론을 맺는다.

2. LDAP의 기본구조 및 특징

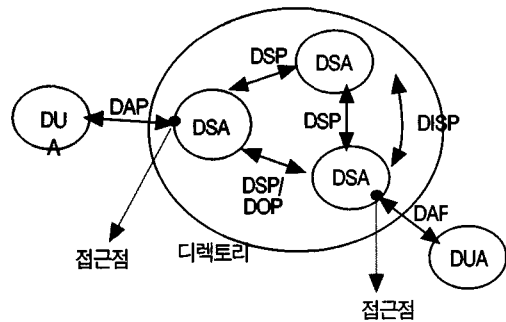
2.1 LDAP의 등장배경

LDAP은 X.500을 기반으로 개발된 통신규약이며, X.500은 인터넷 사용이 가능한 곳이라면 전 세계 어디서라도 이용이 가능하도록 나라, 기관, 사람, 기계 등과 같은 객체들을 관리하고 정보를 제공하는 디렉토리 서비스 표준이다[3]. 사용자는 X.500 디렉토리 서버로 접근하기 위해 DAP(Directory

Access Protocol)이라는 프로토콜을 이용한다. 하지만 DAP은 클라이언트에게 디렉토리 서비스를 제공하기 위해서 OSI 7계층을 모두 필요로 하기 때문에, 규모가 작은 환경일 경우 그 환경에서 제공할 수 있는 것보다 훨씬 더 많은 자원들을 필요로 하게된다. 따라서 X.500 디렉토리 서버는 자원을 많이 필요로 하지 않는 인터페이스나 경량의 프로토콜을 사용하는 인터페이스에는 적절하지 못하다는 단점이 있다. LDAP은 이러한 DAP의 문제점을 해결하기 위해 나온 프로토콜로써 DAP을 개량하여 사용자가 사용하고 구현하기에 편리하도록 경량화시킨 프로토콜이다[1]. 즉 DAP과 동일한 기능을 사용자에게 제공하면서도 TCP/IP 상에서 동작할 수 있도록 구현되어 있다.

초기에는 X.500 디렉토리 서버와 통신하기 위해서 LDAP 서버를 독립적으로 두어, X.500 서버에 대한 클라이언트 역할을 수행하면서 동시에 LDAP 클라이언트로부터의 요청도 처리하게 하였다. 하지만 이러한 방식이 중소기업의 운용 환경에서 서버 효율성 문제를 야기하게 됨에 따라, 최근에는 LDAP 서버 내에 데이터를 직접 저장 및 관리 할 수 있는 기능을 추가하여 LDAP 서버가 직접 디렉토리 정보들을 관리할 수 있도록 변경되었다[4].

그림 1은 사용자가 디렉토리 서버에 저장된 정보로 접근하기 위하여 거치게 되는 정보 전달 경로를 보여준다[5]. 어플리케이션이나 사용자는 DUA (Directory User Agent)를 통해 디렉토리의 정보를 요구하고 응답을 받으며, DSA(Directory Specification



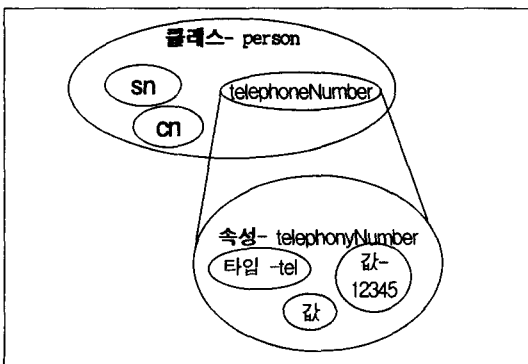
(그림 1) 디렉토리의 구성

Agent)는 사용자가 지시한 명령이나 요구들을 처리하고 그 결과를 사용자에게 응답하는 역할을 한다. DUA는 DSA에 접근할 때, 접근점을 통해서 디렉토리에 접근을 하게 되고, 하나의 DSA는 하나 이상의 접근점을 가질 수 있다. DSA 간의 관리정보 교환을 위한 통신은 DSP(Directory System Protocol)를 통해서 이루어지고, DSA 간 디렉토리 정보의 교환을 위해서는 DISP(Directory Information Shadow Protocol)가 이용된다. 그리고, DUA와 DSA간의 통신을 위해서는 DAP(Directory Access Protocol)이라는 프로토콜이 사용된다[5].

2.2 LDAP의 기본구조

LDAP 디렉토리에 정보를 저장하기 위해서는 스키마가 설정되어 있어야 한다. 디렉토리에 저장될 정보는 객체로 표현되는데 각각의 객체에 대한 특징을 정의해 놓은 것이 스키마이다. 스키마는 객체와 객체에 대한 속성으로 구성된다. 속성은 객체에 대한 특징으로 필수 속성과 옵션 속성으로 나뉜다. 각각의 속성은 속성을 표현하기 위한 데이터 타입과 값으로 나타낸다.

그림 2는 스키마에 정의된 객체 중의 하나를 나타낸 것이다. 'person'이라는 객체에는 sn(surname), cn(common name), telephoneNumber 등의 속성이 있다. 속성 중에서 telephoneNumber의 데이터 타입은 'tel'이고, 값은 '12345'이다. 하나의 속성



(그림 2) 클래스와 속성

(표 1) LDAP의 기본 연산

Search	디렉토리에 있는 데이터를 검색한다.
Modify	디렉토리에 있는 데이터 값을 변경한다
Add	디렉토리에 데이터를 추가한다.
Delete	디렉토리에 있는 데이터를 삭제한다.
Compare	입력된 값과 디렉토리에 저장된 값을 비교한다.
Abandon	명령에 대한 반응이 없을 때 그 명령을 취소한다.
Bind	디렉토리 서버와 디렉토리를 연결한다.
Unbind	디렉토리 서버와 디렉토리를 연결을 끊는다.

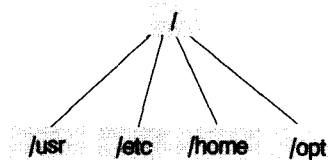
은 다중의 값을 가질 수 있다. 이러한 객체 정보를 이용하기 위해서 LDAP에서 사용하는 기본적인 연산은 표 1과 같다[6].

2.3 LDAP의 특징

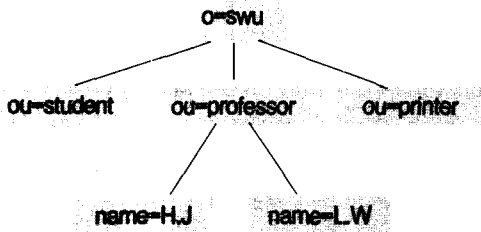
LDAP 디렉토리의 장점은 여러 가지가 있다. 첫째로, 디렉토리 시스템을 데이터베이스 시스템에서 검색기능을 최대화시킨 데이터 관리 기술로 볼 때, LDAP은 데이터의 검색이나 읽기 성능에 있어서 데이터베이스 시스템 보다 우수한 결과를 보인다. 실제로 10만 건의 데이터를 데이터베이스 시스템과 LDAP 디렉토리 시스템에 똑같은 환경을 주고 검색시켰을 때, LDAP은 0.021초가 걸린 반면 데이터베이스 시스템은 0.3초가 걸렸다는 연구 보고가 있다[6].

둘째로, LDAP은 디렉토리의 구조면에 있어서 계층적인 구조로 이루어져 있기 때문에 데이터의 관리나 검색이 편리하다.

그림 3에서와 같이 일반 파일시스템의 경우 특정 데이터를 찾기 위해서는 루트 디렉토리부터 데이



(그림 3) 파일시스템 구조



(그림 4) LDAP 디렉토리 트리 구조

터를 찾을 때까지 계속 찾아 나가야 하지만, 그림 4 처럼 LDAP 디렉토리의 경우 부모노드에서 나온 자식노드들의 타입이 모두 같은 계층적 구조로 구성되어 있어서 원하는 데이터로의 직접 접근 및 관리가 용이하다. 그림 4에서 ou=professor의 자식 노드의 타입은 모두 name이다.

셋째로, 데이터베이스는 데이터의 중복성, 일관성 등을 유지하기 위하여 각 레코드 당 하나의 필드에 하나의 필드 값만을 허용하지만 LDAP 디렉토리는 하나의 속성에 다중 값을 지원한다[6]. 예를 들어, e-메일 주소나 전화번호를 여러 개 갖고있는 사용자의 정보를 모두 저장할 수 있는 기능을 지원한다.

넷째로, LDAP 디렉토리는 프로토콜 수준에서 표준화 되어있기 때문에 다른 여러 어플리케이션들과의 연동과 확장이 쉽다. 데이터베이스의 경우에는 하나의 DB에 여러 다른 서비스들과 연동하려면 DB 이름, 테이블 이름, 필드 이름, 필드 크기까지 모두 일치시켜야 하지만 LDAP 디렉토리는 스키마라는 표준을 제공하기 때문에 서버 이름과 포트번호만 알면 연동이 쉽다.

다섯째로, LDAP 디렉토리는 참조와 복제 기능을 가지고 있다[6]. 데이터베이스의 경우 각 데이터베이스에 대한 내용을 참조하거나 복제를 하기 위해서는 데이터베이스 관리자가 직접 이를 수행해야 하지만 디렉토리 서버들 사이에는 LDAP 프로토콜이 이러한 기능을 직접 지원한다.

여섯째로, LDAP 디렉토리는 사용자 인증 시 LDAP API를 사용하여 단순히 관련 함수를 호출하는 것만으로 사용자 인증을 처리할 수 있다. 또한

트리 구조에 기반한 보안 접근 권한을 설정하기 때문에 사용자마다 다른 권한 부여가 용이하다. 물론 데이터베이스도 권한 설정을 할 수 있지만, 사용자나 필드 단위로는 권한 설정을 할 수 없다.

일곱째로, LDAP 디렉토리는 데이터베이스의 SQL 문에 비해서 데이터를 찾는 데 매우 간단한 명령식을 사용한다. 박씨 성을 가진 사람을 모두 찾는 검색문을 예로 들면, 데이터베이스의 SQL문은 [Select Name From User Where Name like '박%']이라는 복잡한 명령식을 써야하지만, LDAP의 경우는 [sn=박*]이라는 간단한 명령식만으로도 원하는 결과를 찾을 수 있다.

마지막으로 LDAP은 URL을 LDAP 서버의 주소로 사용하여 인터넷 사용자들이 쉽게 이용할 수 있다. DAP에서는 클라이언트가 서버로 접근할 때 DSA의 접근점을 사용하여 접근하도록 되어있기 때문에 일반 인터넷 사용자들이 사용하기에는 어려움이 있었으나, LDAP은 이러한 문제점을 해결 하였다[5].

3. 관련 연구

현재 LDAP 기술의 상용화는 넷스케이프, 마이크로소프트, 노벨 등이 시장을 주도하고 있다. 넷스케이프는 이미 넷스케이프 디렉토리 서버를 통해 LDAP을 지원하고 있고, 마이크로소프트 역시 Active Directory라는 이름으로 LDAP 기반의 디렉토리 기술을 사용하고 있으며, 노벨에서도 이미 자사의 디렉토리 서비스를 LDAP과 상호 연계하여 운영하고 있으며, ADSI의 가입자 인증 요청을 처리하거나 메일 계정을 발급하기 위해 LDAP을 이용하는 기술을 선보이고 있다[1]. 하지만 이들 기술의 구현에 있어 가장 큰 애로점은 일반 사용자가 인터넷과 같은 분산 환경에서 디렉토리 서비스를 이용할 때 발생할 수 있는 서버 과부하 및 동기화 문제가 발생할 수 있다는 점이다.

디렉토리 서버가 급격히 늘어나고 있는 정보들을 효율적으로 관리하기 위해서는 초기의 기업, 기관,

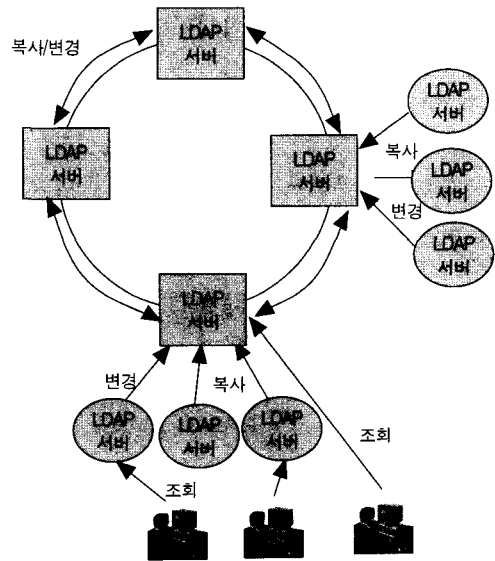
학교 등 기관이나 단체 내로 국한된 데이터베이스 차원의 서비스에서 벗어나 이들 간의 정보 공유 및 관리가 가능한 전역적 서비스 차원으로 확장되어야 한다. 따라서 많은 양의 정보를 어떻게 하면 효율적으로 분산시키고 관리할 것인가가 디렉토리 서비스 성패의 주요 요소가 될 수 있다. 예를 들어 전국에 분산되어 있는 정부 기관들이 각종 전자 문서 및 직원 인사 정보들을 관리하는 하나의 디렉토리 서비스를 이용하고자 할 때, 수십만 명이 넘는 직원들의 인적 사항과 관련 문서들에 대한 관리 서비스를 제공한다는 것은 쉬운 일이 아닐 것이다. 즉, 각 기관별로 갖고 있는 모든 정보를 하나의 디렉토리 서버에서 관리하는 중앙 집중형 방식으로는 서버가 감당해야 할 클라이언트의 수가 급격히 증가할 때 쉽게 과부하 상태에 이르게 된다.

이러한 서버의 과부하 상태를 막기 위해서, 각 기관마다 모든 기관의 디렉토리 정보를 복제하여 보유하는 DNS 형태의 서버 방식을 취할 수 있다. 하지만 이 경우, 각 서버가 감당해야 할 클라이언트의 수는 줄겠지만, 타 서버로부터 복제해야 하는 디렉토리 정보의 양은 줄어들지 않기 때문에 서버간 통신 지연의 문제가 여전히 남게 된다. 또한, 이러한 통신지연은 임의의 기관의 디렉토리 정보가 변경되었을 때 그 변경사항을 적시에 다른 서버에 반영하지 못하는 동기화의 문제를 함께 야기시킬 수 있다.

4. LDAP 아키텍처 설계

4.1 서브 서버(sub-server)형 방식

DNS 방식의 데이터의 동기화 문제와 중앙 집중형 방식의 서버 과부하 문제를 동시에 해결하기 위하여 서브 서버형 방식이 이용될 수 있다. 이 방식은 그림 5에서와 같이 각 지역의 공공기관 별로 별도의 LDAP 서버를 두게 하고, 각 지역별로 대표 디렉토리 서버(서브 서버)를 두어,



(그림 5) 서브 서버형 방식

지역별 공공기관의 디렉토리 내용은 그 지역의 대표 디렉토리 서버가 담당하게 한다. 즉 지역의 대표 디렉토리는 그 지역 공공기관의 디렉토리 내용을 모두 포함하고 있게 된다. 그런 다음 지역의 대표 디렉토리들끼리 디렉토리 내용을 복제한다. 결국 각각의 지역 대표 디렉토리는 전국의 모든 공공기관의 디렉토리 내용을 갖게 되는 것이다.

이 방식은 자료 검색 시, 같은 지역 내 기관에 대한 자료 검색은 기관 LDAP 서버로 직접 접근하여 검색하고, 다른 지역 기관에 대한 자료 검색은 자신이 속한 지역의 대표 디렉토리 서버로 접근하여 검색하는 방식이다. 또한 데이터의 변경이 일어날 경우 각 기관은 자신이 속한 기관 디렉토리 서버에 이를 알리고, 기관 디렉토리 서버는 각 지역의 대표 디렉토리 서버에게만 데이터의 변경을 알린다. 이렇게 되면 모든 기관의 디렉토리 서버로 디렉토리 정보의 변경사항을 알릴 필요가 없기 때문에 동기화 시 발생할 수 있는 서버 과부하 문제를 해결할 수 있다.

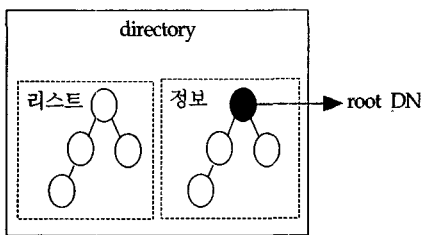
하지만 이 방법은 지역 대표 디렉토리로 같은 지역의 모든 클라이언트들이 검색을 요청했을 경우에는 서버의 과부하 문제를 해결해 주지는 못

한다. 이를 보완하기 위하여 본 연구에서는 서버 독립형 방식을 제안한다. 서버 독립형 방식은 디렉토리 구조를 본연의 디렉토리 정보를 관리하는 기능과 외부 디렉토리 서버의 주소리스트를 관리하는 기능으로 나누어 서버의 부담을 줄임과 동시에 동기화 회수를 줄이는 방식이다. 이 방식은 기존의 서버 구조 설계를 그대로 이용하면서 새로운 기능을 추가시켜 사용하기 때문에 확장성 및 이식성이 뛰어나며, 따라서 서버 유지보수에도 효과적인 구조를 갖고 있다.

4.2 서버 독립형 방식

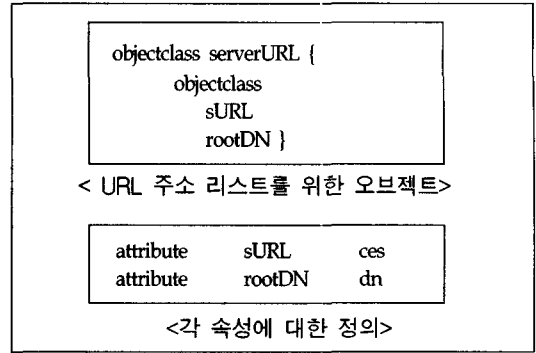
4.2.1 서버 독립형 디렉토리 구조

서버 독립형 방식의 디렉토리 구조는 그림 6에 서와 같이 두 개의 디렉토리로 나뉘게 된다. 하나의 디렉토리에는 각 서버의 URL 리스트를 저장하고, 나머지 하나의 디렉토리에는 자신의 디렉토리 정보를 저장한다. 각각의 디렉토리는 주소리스트 디렉토리와 정보 디렉토리로 나뉜다. 이때 주소리스트 디렉토리의 트리 구조는 LDAP과의 호환성을 유지하기 위하여 계층별로 구조화시키며, 웹 환경을 그대로 지원하기 위하여 URL 주소체계를 따른다.



(그림 6) 디렉토리 구조

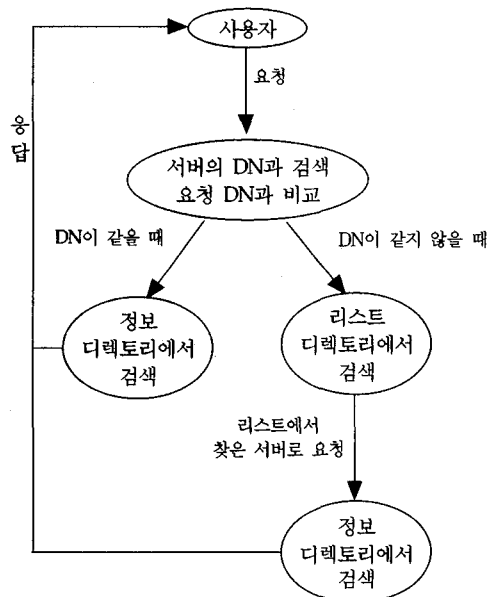
주소리스트 디렉토리의 엔트리 구성을 위해, 그림 7과 같이 서버의 URL과 정보 디렉토리의 루트 DN을 속성으로 가지는 serverURL오브젝트 스키마 및 오브젝트의 속성을 정의할 수 있다. 스키마에서 sURL과 rootDN은 각각 서버 URL과 디렉토리의 루트 정보를 나타낸다.



(그림 7) URL 오브젝트 스키마 및 속성

4.2.2 사용자 요청에 대한 설계

각각의 서버는 자신의 정보를 계층적으로 구축한 후, 사용자로부터 검색 요청이 들어왔을 때 그림 8과 같은 처리과정을 거친다. 먼저 사용자로부터 검색 요청이 들어오면 요청 데이터가 자신의 정보 디렉토리에 있는 내부 데이터인지 또는 외부 데이터인지를 구분하기 위해서 정보 디렉토리의 root DN을 사용한다. 서버는 정보 디렉토리의 rootDN과 사용자가 선택한 rootDN(사용자가 정보를 찾고자 하는 서버의 rootDN)을 비교하여, 같으면 요



(그림 8) 사용자 요청 처리 과정

청에 대한 데이터를 정보 디렉토리에서 찾고, 같지 않으면 주소 디렉토리에서 rootDN을 갖고 있는 서버를 찾아서 그 서버로 검색을 요청한다.

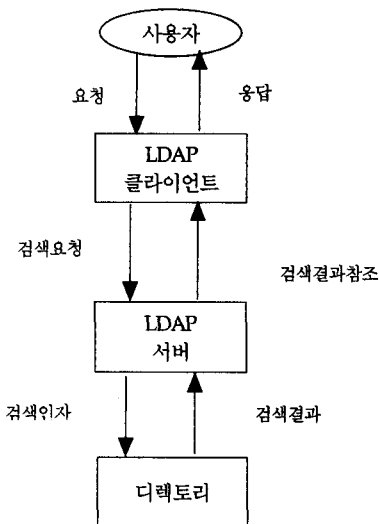
이 때, 정보 디렉토리의 rootDN과 사용자가 선택한 rootDN을 비교하기 위해서, search 연산 메시지에 baseObject형 RootDN을 다음과 같이 추가한다.

```

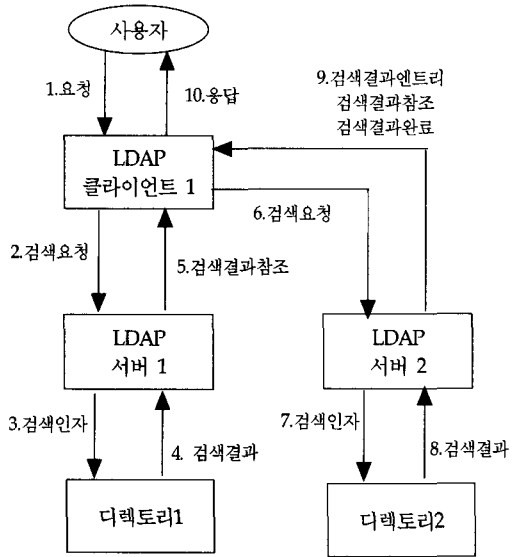
SearchRequest ::= [APPLICATION 3] SEQUENCE
{
    baseObject RootDN, → 추가
    baseObject LDAPDN,
    scope ENUMERATED {
        baseObject (0),
        singleLevel (1),
        wholeSubtree (2)
    },
    ...
    sizeLimit INTEGER (0 .. maxInt),
    timeLimit INTEGER (0 .. maxInt),
    ...
}
    
```

4.2.3 사용자 응답에 대한 설계

기존의 LDAP 서버에서 검색 요청을 받았을 때 그에 대한 응답처리는 그림 9와 같다.



(그림 9) 응답 처리 과정



(그림 10) 외부서버로의 검색 요청 처리 흐름도

사용자가 검색 요청을 하면 LDAP은 검색의 결과로 '검색결과엔트리(SearchResultEntry)', '검색결과참조(SearchResultReference)', '검색결과완료(SearchResultDone)'을 돌려준다. 기존의 LDAP 검색에서 검색결과참조(SearchResultReference)는 사용자가 서버로 검색을 요청했을 때, 요청한 DN과 유사한 DN을 갖고 있는 서버의 URL을 실어서 사용자에게 보내주는 역할을 한다.

본 연구에서는 이 기능을 수정하여 사용자에게 검색결과를 반환할 때, 주소 디렉토리에서 사용자가 요청한 URL을 찾아 그 URL을 검색결과참조 부분에 실어 보내주도록 설계하였다. 이러한 서버 재구성 방식은 기존의 설계 구조를 변형시키지 않고도 웹 환경과 LDAP 서버 기능을 모두 지원할 수 있다는 장점이 있다. 그림 10은 사용자가 자신의 서버가 아닌 다른 서버로 검색을 요청했을 때의 URL 정보처리 흐름도를 나타낸 것이다.

5. 설계 평가

본 논문이 제안한 서버 독립형 디렉토리 서버

스는 각 디렉토리 서버에서의 데이터 갱신이 있을 때에만 데이터의 추가, 삭제, 변경이 독립적으로 이루어지기 때문에 서버들간의 데이터 공유 문제가 해결된다. 서브 서버형 방식에서는 하나의 서버가 데이터를 변경했을 때 데이터를 변경한 서버는 갱신(update) 연산을 하고, 그 서버에 연결된 모든 대표 디렉토리 서버들은 복제(Replication)라는 연산을 수행해야 한다. 하지만 서버 독립형 방식은 데이터 변경 시 자신의 서버에서만 갱신(update) 연산을 수행하면 되고, 다른 서버의 데이터 변경에 전혀 신경을 쓸 필요가 없다. 또한 지역에 속한 클라이언트들은 자신이 속한 서버로만 접속하여 검색요청을 하면 되기 때문에, 중앙집중형이나 DNS형 방식의 서버가 받게되는 과부하 현상을 방지할 수 있을 뿐만 아니라, 서브 서버형 방식에서도 발생할 수 있는 대표 디렉토리의 서버 과부하 현상을 방지할 수 있다. 이러한 서버 재구성 방식은 기존의 구조에 대한 급격한 변형 없이 웹 주소체계를 지원하며 LDAP과의 호환성도 그대로 유지할 수 있다는 장점이 있다.

6. 결 론

LDAP은 기관이나 객체, 사람, 네트워크 상에 있는 파일이나 장치들과 같은 자원들의 위치를 찾아줄 수 있도록 해주는 프로토콜로, 디렉토리 내의 정보들을 쉽게 관리하고, 정보 검색의 속도와 결과에 있어서 뛰어난 성능을 보여준다. 본 논문에서는 LDAP의 개념과 그 배경 및 특징에 대해 살펴보았으며, 방대한 양의 분산된 디렉토리 서비스를 제공하는데 있어서, 서버의 과부하와 디렉토리 서버간 데이터 동기화 문제를 해결할 수 있는 아키텍처 재구성 방법을 제안하였다.

향후 LDAP을 이용하여 서버간 인증 기능을 지원하기 위한 기반 기술에 대한 연구가 필요하며, 아울러 조직간의 상이한 조직도로 인해 발생할 수 있는 데이터 구조의 불일치 문제도 함께 연구되어야 할 것이다.

Acknowledgement

본 논문은 2000년 서울여자대학교 자연과학 연구소의 연구지원비에 의해서 연구된 것임.

참 고 문 헌

- [1] 데이터베이스 웹진(database sarang), <http://database.sarang.net/database/ldap/doc/LDAP-1.htm>
- [2] W. Yeong, T. Howes, S. Kille, "Lightweight Directory Access protocol", RFC1777, 1995. 3.
- [3] W. Yeong, T. Howes, S. Kille, "X.500 Light-weight Directory Access Protocol", RFC1487, 1993. 7.
- [4] Heinz Johner, Larry Brown, Franz-Stefan Hinner, Wolfgang Reis, Johan Westman, "Understanding LDAP", IBM June 1998.
- [5] 한국 정보보호 진흥원, http://www.kisa.or.kr/K_trend/KisaNews/199904/special9.html.
- [6] 정진원, "디렉토리 서비스의 핵심 기술 'LDAP'", Network Times, 2001.
- [7] T. Howes, M. Smith, "The LDAP Application program Interface", University of Michigan, August 1995.
- [8] 최진주, "LDAP 프로토콜에 대한 고찰", 한국통신정보보호학회 3, v9, 1999.
- [9] 양원석, "LDAP을 이용한 클라이언트 시스템 개발에 관한 연구", 호서대학교 대학원, 1999.
- [10] 정권성, "공개키 인증 서비스 구축을 위한 개선된 LDAP", 성균관대학교 대학원, 1999.
- [11] Intranet Journal, <http://www.intranetjournal.com/>.
- [12] Network Working Group rfc2251, <http://sumin.donggeui.ac.kr/>.
- [13] LDAP 기술동향, <http://www.powermct.co.kr>.
- [14] Dabid Chadwick, "Understanding X.500 The Directory", Chapman & Hall, 1994.
- [15] 이용준, 오주병, 박문성, X.500 디렉토리 표준, 주간기술동향, 한국전자통신연구소, 94-35, pp. 22-41, 1994.

● 저 자 소개 ●



김 혜 원

2002년 서울여자대학교 정보통신학부 컴퓨터학부(졸업예정)
관심분야 : 분산시스템, 모바일 시스템
E-mail : grease0805@hotmail.com



한 지 인

2002년 서울여자대학교 정보통신학부 컴퓨터학부(졸업예정)
관심분야 : 분산시스템, 전자상거래
E-mail : omgod@unitel.co.kr



이 병 길

1988년 University of Bridgeport 물리학과(학사)
1996년 Auburn University 전산학과(공학 석사)
1998년 Auburn University 전산학과(공학 박사)
1998년~현재 서울여자대학교 컴퓨터학과 조교수
관심분야 : 소프트웨어 형상관리, CSCW, 분산시스템
E-mail : byongl@swu.ac.kr