

CORBA/JAVA를 이용한 분산 서지정보 검색시스템 개발에 관한 연구*

A Study on the Development of Distributed Bibliographic Information Search System Using CORBA and JAVA

유재옥(Jae-Ok Yoo)**, 박우창(UChang Park)***

목 차

1 서 론	4.2 인터페이스 구성
2 기존의 정보검색방법	4.3 질의 처리과정
3 통합검색시스템	5 분산 서지정보 검색시스템 구축
3.1 중앙 검색서버 통합 데이터베이스 방식	5.1 개발환경
3.2 중앙 검색서버 분산 데이터베이스 방식	5.2 서버구현
3.3 분산 검색서버 분산 데이터베이스 방식	5.3 클라이언트 구현
4 분산 서지정보 검색시스템 설계	5.4 인터페이스 구현
4.1 사용기술	6 결 론

초 록

본 연구는 지리적으로 분산되어 있는 분산서버상태를 유지하면서 또한 분산되어 있는 데이터베이스 검색을 허용하는 통합 서지정보 검색시스템의 개발가능성을 모색하고자 하였다. 분산되어 있는 서지정보 데이터베이스와 검색시스템을 통합하기 위한 미들웨어로서 CORBA를 사용하였으며 검색시의 부하를 클라이언트에게 분산시키기 위해 JAVA를 채택하였다. 덕성여자대학교 도서관 정보검색시스템인 DISCOVER를 두 분산서버에 설치하고, 두 분산 데이터베이스를 대상으로 통합 검색할 수 있는 통합 서지정보 검색시스템 구축을 실험하였다. 시스템 구현결과 통합 인터페이스를 사용한 두 분산 서버에 대한 검색요청은 성공적으로 수행되었으며, 클라이언트는 검색결과를 성공적으로 종합하였다.

ABSTRACT

The study attempts to propose a distributed bibliographic information retrieval system using distributed retrieval servers and distributed databases. CORBA and JAVA applications have been employed to solve both the bottleneck and the problems that simultaneous searches in distributed databases cannot deal with under the CGI or JAVA applications.

The proposed distributed bibliographic retrieval system using two distributed servers with their distributed databases was successfully experimented. However, it is recommended that both heterogeneous information retrieval systems and databases are included in further studies.

* 이 논문은 1998년 한국학술진흥재단의 학술연구비에 의하여 지원되었음

** 덕성여자대학교 문헌정보학과 교수

*** 덕성여자대학교 전산학과 교수

■ 논문 접수일 : 1999년 12월 1일

1 서 론

국내 대학도서관의 온라인 열람목록(OPAC: Online Public Access Catalog)의 수는 괄목할 만 한 증가추세를 보이고 있다. 1987년에 조사한 국내 대학도서관의 자동화실태를 보면 15%인 9개 도서관만이 부분적으로 도서관 업무를 자동화하였으며, 이들 9개 도서관중 7개 도서관만이 온라인목록서비스를 제공하는 것으로 조사되었다(사공철 1987). 그 후 약 10여 년이 지난 1998년 현재 인터넷상에서 웹으로 접근 가능한 대학도서관 온라인목록의 수는 약 500여 개에 달해 폭발적인 증가추세를 나타내고 있다. 국내 대부분의 대학도서관이 온라인 목록에 대한 웹서비스를 제공하고 있는 실정이다.

그럼에도 불구하고 사용자의 입장에서는 이들 온라인목록을 통합 검색하는 작업은 매우 열악한 형편이다. 예를 들면 한 사용자가 여러 도서관을 방문해야 할 경우, 각각의 도서관의 온라인 목록에 개별적으로 접속하는 작업을 반복해야 한다. 또한 각각의 도서관을 방문하기 위해서는 각 도서관의 URL주소를 알고 있어야 하며, 또한 동일한 검색작업을 각 사이트에서 반복해야 한다. 이러한 문제점을 개선하기 위해 제한적이기는 하나 그림 1과 같은 사용자 인터페이스가 개발되어 사용자가 원하는 사이트의 초기화면까지 안내하는 서비스가 제공되고 있다. 즉 CGI기술을 이용한 웹방식으로 사용자 인터페이스를 통합하여 검색 서비스환경을 제공한다. 그러나 이와 같은 사용자 인터페이스는 실제적으로는 프론트 엔드(front-end)시스템에 보다 가깝다고 하겠다.

온라인 목록을 구축하기 위한 데이터베이스 구조는 일반적으로 정형화, 표준화되어 있을 뿐만 아니라 검색시스템도 SOLARS 그룹과

VINTAGE그룹으로 크게 대별되어 있는 상황이다. 이와 같은 맥락에서 동일한 사용자그룹뿐만 아니라 서로 다른 사용자그룹을 포함한 통합검색시스템의 개발은 시급한 과제로 보인다.

본 연구는 이와 같은 여건을 감안하여 웹 상에서 접근 가능한 국내의 모든 도서관자료를 도서관의 위치나 검색시스템의 종류, 사용하고 있는 데이터베이스 구조에 관계없이 종합적으로 검색할 수 있는 분산 서지정보 검색시스템의 개발가능성을 타진해 보고자 한다.

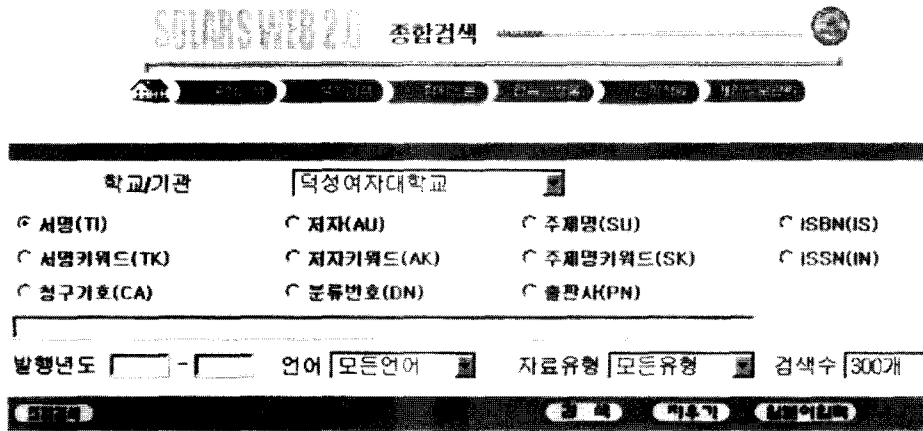
2 기존의 정보검색 방법

현재 각 도서관들이 사용하고 있는 검색방식은 웹 기반에서 CGI나 혹은 JAVA APPLICATION을 통한 데이터베이스 연동방식으로 개별적으로 검색서버에 접속하는 방식이다. 사용자가 편리하게 쓸 수 있는 통합 검색인터페이스를 사용한다는 점이 특징이라 하겠다. 그러나 여러 도서관 검색서버에 동시에 접속하여 종합적으로 검색할 수 있는 환경을 제공하지는 않는다.

〈그림 1〉은 웹의 CGI(Common Gateway Interface) 기술을 사용하여 개발한 인터페이스로 이용자가 '학교/기관'을 선택할 수 있는 드롭다운 메뉴에서 원하는 기관을 선택하면, 선택된 기관의 온라인목록 검색화면에 연결된다.

즉, 흩어져 있는 검색서버에 개별적으로 접속할 수 있도록 편리한 인터페이스 환경을 제공하고 있다. 그 대표적인 예로서 같은 종류의 도서관 온라인목록 검색시스템을 사용하는 SOLARS 사용자들은 〈그림 1〉과 같은 표준화된 검색 인터페이스를 사용하고 있다.

운영방식을 살펴보면 〈그림 2〉와 같이 중앙서버



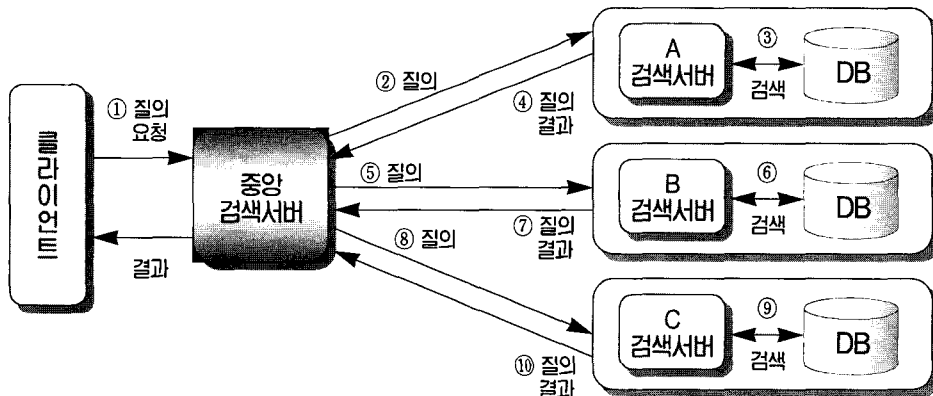
〈그림 1〉 기존 서지정보검색시스템

개별검색방식으로 운용되고 있다. 클라이언트가 중앙 검색서버에게 특정 데이터베이스에서 특정 검색을 수행해 줄 것을 요청하면 중앙서버는 해당 특정 검색서버에 검색질의를 보내고, 특정 검색서버는 탐색한 결과를 중앙서버에게 보낸다. 중앙서버는 받은 결과를 다시 클라이언트에게 전하는 것이다.

〈그림 2〉에서 볼 수 있듯이 사용자는 중앙서버에서 분산되어 있는 검색서버를 종합적으로 검색하는 것이 아니라 개별 도서관서버에 일단 접속

하여 검색을 한 후, 접속을 끊고 다른 도서관 서버에 재접속하는 작업을 반복하는 것이다. 따라서 복수의 도서관 자료를 검색하고자 할 때에는 시간이 오래 걸리고 사용자 환경이 매우 불편하다. 한편 사용자 인터페이스 개발에 CGI가 아닌 JAVA를 사용하여 개발한 국가전자도서관 사업이 있는데 마찬가지로 중앙검색서버 개별검색 방식으로 운용되고 있다.

1997년부터 시작된 국가전자도서관 사업은



〈그림 2〉 중앙검색서버 개별검색 방식

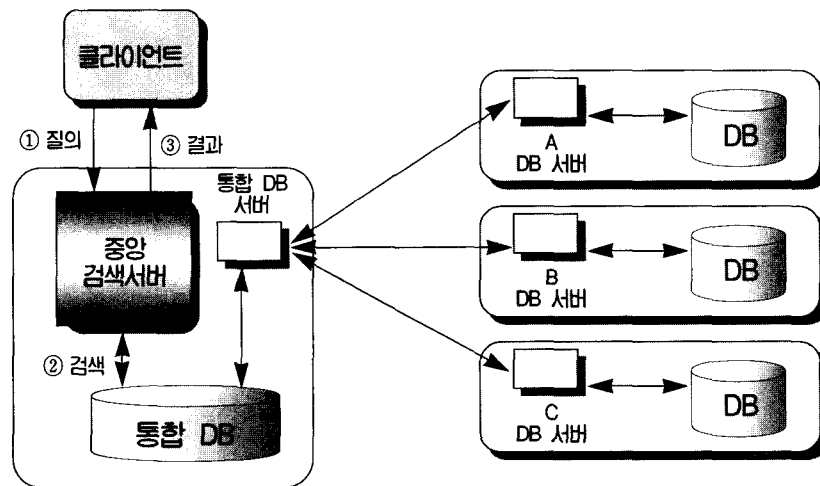
JAVA APPLICATION을 사용하여 통합 인터페이스를 제공함으로써 원하는 도서관에 접속할 수 있는 편리한 환경을 제공하고 있다. 국가전자도서관 프로젝트는 Z39.50이라는 표준화된 정보검색 프로토콜을 사용하도록 권장하고 이 방식을 채택하는 기관간에는 표준화된 인터페이스를 제공함으로써 참여기관 자료에 대한 검색이 가능하도록 하는 방안이 추진되어 왔다. 현재 국립중앙도서관, 국회도서관, 법원도서관 등 7개 기관을 웹 상에서 검색할 수 있다. 지리적으로 흩어져 있는 서지 데이터베이스에 편리하게 접속하여 검색 화면으로 직접 인도되는 환경을 제공할 뿐, 7개 기관의 자료를 종합적으로 탐색할 수 있는 서비스가 제공되는 것은 아니다(강한배 1998; 공봉석 1998; 김성혁 1998).

따라서 SOLARS 정보검색시스템이나 국가전자도서관 검색시스템은 인터페이스 개발에 사용한 기술만 다를 뿐, 중앙 검색서버 개별검색 방식이라는 점과 통합 인터페이스를 사용한다는 점에서는 큰 차이가 없는 중앙집중형 방식이다.

3 통합검색시스템

기존의 도서관들이 사용하고 있는 검색방식은 개별적으로 검색서버에 접속하는 중앙 검색서버 개별검색 방식이다. 복수의 도서관에 동시에 접속하여 종합적으로 검색할 수 있는 기능을 제공하지는 않는다. 더욱이 동일 사용자 그룹이 아닌 이종의 도서관의 검색시스템에 접속할 수 있는 환경은 마련되어 있지 않은 실정이다. 이러한 문제점을 개선하기 위하여 통합 검색시스템의 개발 가능성이 다양하게 모색되고 있다.

통합검색시스템의 개발환경은 약간의 차이가 있는데 세 종류로 구분할 수 있다. 첫째는 중앙 검색서버를 사용하고, 중앙검색서버에 통합 데이터베이스를 구축하여 운영하는 방식이다. 둘째는 중앙검색서버를 운영하되, 데이터베이스는 분산되어 있는 데이터베이스를 사용한다. 셋째는 분산되어 있는 각각의 검색서버와 분산되어 있는 각각의 데이터베이스를 사용하는 방법이다.



〈그림 3〉 중앙검색서버 통합데이터베이스 방식

3.1 중앙 검색서버 통합 데이터베이스 방식

미국의 OCLC와 같은 기능을 하는 국가 분담 목록시스템 구축방안이 <그림 3>과 같이 한국교육학술정보원(구 첨단학술정보센터)에 의해 추진되고 있다. 회원도서관들은 자관 데이터를 중앙 서버에 송부하고, 중앙서버는 회원도서관으로부터 받은 데이터를 종합관리 운영하며, 각 회원도서관은 중앙 서버의 정보를 공유한다. 이때 필요한 기본 시스템은 중앙서버가 관리하는 종합 목록데이터베이스와 종합목록 데이터베이스를 검색하는 검색시스템이다.

이와 같은 중앙서버 통합데이터베이스 운영방식은 중앙에서 거대한 데이터베이스를 집중 관리 운영함에 따라 중앙에서는 데이터베이스 관리업무와 검색시스템의 보완, 관리의 두 업무를 수행해야 한다. 더욱이 사용자의 검색질의(query)가 중앙 컴퓨터에 집중될 경우 사용자에 대한 응답시간이 늦어지는 단점을 안고 있다. 또한 중앙서버 고장시 다른 대안이 없는 실정이다. 색인과 실제정보의 불일치성, 색인통합의 문제, 시스템 고장과 과부하 등이 문제점으로 지적된 바 있다(김성혁 1998).

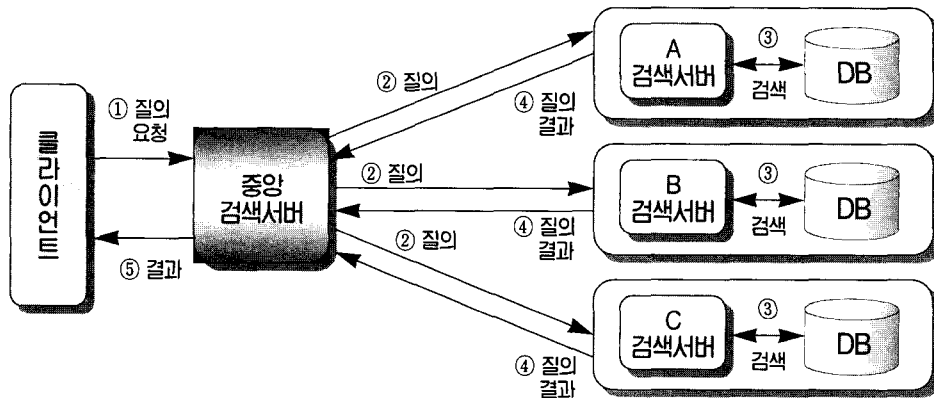
반면에 중앙집중형 방식은 한 시스템에서 작동하는 중앙서버만을 유지하기 때문에 비교적 간단한 시스템을 유지할 수 있는 것이 장점이다.

3.2 중앙 검색서버 분산 데이터베이스 방식

한편 그림4와 같이 중앙 검색서버를 유지하면서 분산되어 있는 데이터베이스를 검색할 수 있는 방안이 중앙 검색서버 분산데이터베이스 방식이다. 사용자는 중앙 검색서버에 있는 통합 인터페이스를 사용하여 검색질의를 요청한다. 중앙검색서버는 검색질의를 분산되어 있는 각각의 검색서버에게로 동시에 전달하고, 분산되어 있는 검색서버는 각각의 데이터베이스를 대상으로 검색을 수행한다. 검색결과를 중앙검색서버에게 전달한다.

분산데이터베이스 탐색에서는 각각의 데이터베이스에 대한 탐색이 동시에 수행되므로 사용자는 각기 다른 서버에 개별적으로 접속하지 않고도 거대한 데이터베이스를 한번에 탐색하는 것과 같은 효과를 내는 것이다.

중앙 검색서버 분산 데이터베이스 검색 방식을 기용하려고 하는 프로젝트로는 캐나다 국립도서



<그림 4> 중앙검색서버 분산데이터베이스 방식

관의 가상종합목록(Virtual Canadian Union Catalog) 사업을 들 수 있다. 캐나다 국립도서관은 분산되어 있는 개별도서관의 온라인목록을 종합하여 중앙 데이터베이스로 구축하려는 것이 아니라, 가상의 종합목록을 만들어 놓을 뿐, 기술적으로는 개별도서관의 목록을 분산 탐색하는 것이다. 즉 중앙집중식 종합목록이 아니라, 각 도서관의 자료를 분산 존재시키면서, 자료를 검색할 수 있는 분산 검색시스템이다. 이러한 분산탐색이 가능하기 위해서는 정교한 인터페이스의 개발과 성능이 우수한 네트워크가 전제조건으로 꼽히고 있다. 즉 분산탐색이 가능하기 위해서는 기술적인 지원이 요청된다는 지적이다.

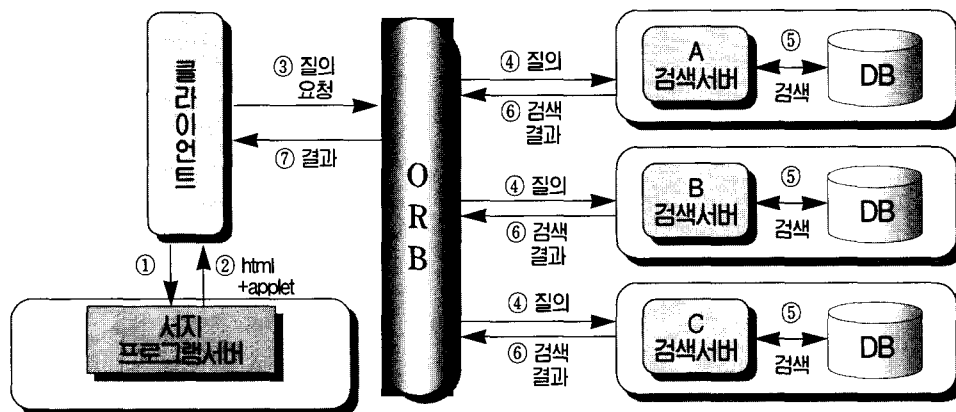
중앙검색서버를 유지하면서 분산 데이터베이스 검색방식을 실험한 연구가 최한석(1996)에 의해 이루어진 바 있다. 중앙검색 서버는 분산되어 있는 두 종류의 데이터베이스인 CD-ROM 데이터베이스인 ERL(Electronic Reference Library)과 목포대학교의 도서관 서지정보를 분산탐색하였다. 또한 이중의 검색시스템인 ERL 검색시스템인 SPIRES와 도서관 서지정보검색시스템인 SOLALIS를 통합하는 시스템을 설계하였다. 분산되어 있는 이중의 검색

서버와 이형의 데이터베이스에 접근하여 통합검색의 가능성을 보여준 바 있다.

3.3 분산 검색서버 분산 데이터베이스 방식

본 연구가 제안하는 통합검색시스템은 그림 5에서 볼 수 있듯이 분산 검색서버 분산 데이터베이스 방식이다. 궁극적으로 분산되어 있는 서버 상태를 유지하면서 분산되어 있는 데이터베이스를 검색하는 방식을 기용하고자 한다. 각 도서관의 정보검색 서버의 독립성을 유지하면서 사용자에게 웹을 통하여 복수의 도서관의 서지정보를 통합하여 검색할 수 있는 기능을 제공하고자 한다.

클라이언트는 실제적으로 검색서버의 역할을 한다. 클라이언트는 검색프로그램을 다운받아 검색시 사용할 뿐만 아니라 검색결과도 직접 받아 종합하는 역할도 하게 된다. 사용자가 웹 환경에서 정보검색을 요청하면 클라이언트 프로그램은 사용자의 질의를 여러 곳에 흩어져 있는 각각의 검색시스템에게 동시에 전달한다. 각각의 검색시스템은 각 데이터베이스에서 자료를 검색한 후 탐색한 결과를 다시 사용자의 클라이언트 프로그



(그림 5) 분산검색서버 분산 데이터베이스 방식

램에게 보내고, 분산 검색서버는 검색된 데이터들을 종합해서 사용자 웹 브라우저를 통해 최종 결과를 알려주는 방식이다.

4 분산서지정보검색시스템 설계

본 연구에서는 서로 다른 검색시스템을 효과적으로 통합하고 복수의 서버에 동시에 접근할 수 있도록 하기 위해 CORBA기반의 분산 검색시스템구조로 설계한다.

4.1 사용기술

분산 서지정보 검색서비스를 제공하는 시스템을 구축하기 위해서 다음과 같은 기술을 사용하고자 한다. 각각의 서지정보 통합은 최근 컴퓨터의 개방형 객체지향 시스템의 표준인 CORBA를 이용하여 기존의 CGI를 이용한 시스템의 정체(bottleneck)를 해소하고, 클라이언트는 JAVA를 이용함으로써 플랫폼에 독립적인 사용자 접속을 가능하게 한다.

통신을 위한 시스템 구축 기본환경으로 웹을 사용하고, 웹 환경 하에서 동적인 클라이언트 시스템을 구축하기 위한 개발언어로 JAVA를, 이종의 다양한 시스템을 통합하기 위한 중간 기술로서 CORBA를 사용하여, 자바가 제공하는 유연성과 이식성, CORBA가 제공하는 분산객체 인프라를 결합함으로써 빠르고 쉽게 원하는 정보를 찾을 수 있는 분산 서지정보 검색시스템으로 개발하고자 한다.

또한 웹 환경에서 단일 사용자 인터페이스를 사용하여 동시에 각 도서관의 데이터베이스를 검색한 후 그 결과를 종합하여 사용자에게 웹 브라우저를

통해 보여주기 위해 다음의 기술을 기용한다.

첫째, 개방형 분산객체시스템의 표준인 OMG(Object Management Group)의 CORBA(Common Object Request Broker Architecture)를 이용하여 이종(heterogeneous) 데이터베이스와 다양한 검색프로토콜을 통합하고, 둘째, 클라이언트는 JAVA의 애플릿(Applet) 기술을 사용함으로써 플랫폼에 독립적인 사용자 인터페이스를 웹 기반에서 제공하도록 한다(박남섭 외 1997; 정창균 외 1997).

4.1.1 CORBA/JAVA

CORBA는 객체지향 기술을 기반으로 하여 이종의 분산된 환경하에서 응용프로그램을 통합할 수 있는 표준기술로서 OMG(OMG91)에 의해서 제정되었다. CORBA는 OMG에서 제안한 OMA(Object Management Architecture)의 핵심이 되는 표준으로 객체들 사이에 위치와 구현에 관한 투명성을 보장함으로써, 이기종 객체간의 통신에 중심역할을 하며 상호운용성(interoperability)을 보장해준다(박재현 1997; 정창균 외 1997; 최한석 외 1996).

JAVA는 SUN사에서 개발하였으며 C++에서 복잡한 개념인 포인터 연산, 오버로딩, 다중 상속 등의 개념을 삭제하였으며, 자동 가비지 수거기능을 첨가한 최적화된 객체지향 프로그래밍 언어로 바이트코드에 의한 이식성이 강한 언어이다(박남섭 외 1997; 최현섭 외 1997).

분산 시스템의 표준인 CORBA를 채택함으로써 다음과 같은 장점을 얻을 수 있다.

- 인터페이스 방식의 소프트웨어 개발 방법인 CORBA를 사용함으로써 각 도서관 검색 서버는 독립적으로 시스템을 운영할 수 있다.

- 객체지향 개념을 갖는 분산표준으로 본문,

그림 등 비정형 데이터베이스를 수용하는 객체지향 데이터베이스나 객체-관계 데이터베이스와 연결이 용이하여 원문이나 멀티미디어 정보검색 시스템으로 확장이 용이하다.

- 개방형 통신 표준을 채택함으로써 가상대학 등과 같은 다른 정보 시스템과의 통합이 용이하다.

- 인터넷상에서 JAVA와 연동시킴으로서 클라이언트와 서버의 시스템간의 독립성을 구현하여 이식성을 극대화시킬 수 있다.

- 각 대학마다 각자의 플랫폼에 맞게 독립적으로 구현할 수 있어 향후 시스템확장이 쉽다.

- 기존에 구축된 검색시스템을 그대로 유지하면서 중복된 정보를 배제시킬 수 있다.

4.1.2 Z39.50/CORBA

1988년 만들어진 Z39.50은 컴퓨터간의 정보 검색을 위한 클라이언트/서버 표준 프로토콜로서 몇 번의 수정을 거쳐 현재의 정보검색시스템에 널리 쓰이고 있다(안개일, 전우직 1997; 최한석 외 1996). Z39.50은 접속서비스, 탐색서비스, 전송서비스 등 정보의 검색과 전달에 관한 일련의 서비스로 구성되어 있다.

한편 CORBA는 OMA구조의 일부로 단순 프로

토콜의 개념이 아닌 분산시스템 구조의 일부이다(박재현 1997; Siegal 1996). OMA는 CORBA를 객체통신버스로 이용하는 구조로, 사용자는 CORBA의 서비스를 만들어 이용하거나, 공통으로 쓰이는 표준화된 Common Facility(Horizontal Facility) 와 Domain-specific한 Vertical Facility로 구성)를 이용할 수 있다. CORBA의 이러한 특성으로 인해 CORBA는 기본서비스나 Facility층에서 여러 가지 사용자 서비스를 구현하도록 정의되어 있다. 따라서 Z39.50을 이러한 Facility의 하나로 구현할 수 있다. 즉 CORBA는 Z39.50이 수행하는 서비스를 구현할 수 있을 뿐만 아니라 Z39.50의 서비스를 포장(wrapping)함으로써 기존의 응용을 대신하거나 흡수할 수 있다는 것이 장점이다(최한석 1996).

4.2 인터페이스 구성

본 논문에서 제안하는 시스템은 기존 서지정보 검색시스템이 사용하는 클라이언트와 서버사이의 표준 통신 프로토콜 Z39.50을 그대로 유지하면서 다른 프로토콜과의 통합을 시도한다. 현재 사용중인 Z39.50은 일종의 CORBA Facility 서비

〈표 1〉 CORBA 인터페이스 정의

인터페이스 정의	설 명
OBR.init()	클라이언트와 서버를 초기화시킴
bind()	클라이언트와 서버를 연결
disconnect()	클라이언트와 서버의 연결을 해제
String[] Z_retrieve(in String keyword, index)	검색결과를 가져옴
void Z_sort(in String result, in int index)	검색결과를 정렬시킴
void Z_delete(in String result, in int index)	검색결과중 일부분을 삭제

스에 포함시키고, 이와는 별도로 다양한 프로토콜에 정의된 인터페이스 가운데 자주 사용되는 주요 기능에 대해서는 <표 1>과 같이 CORBA IDL을 통해 직접 재정의한다.

4.3 질의 처리과정

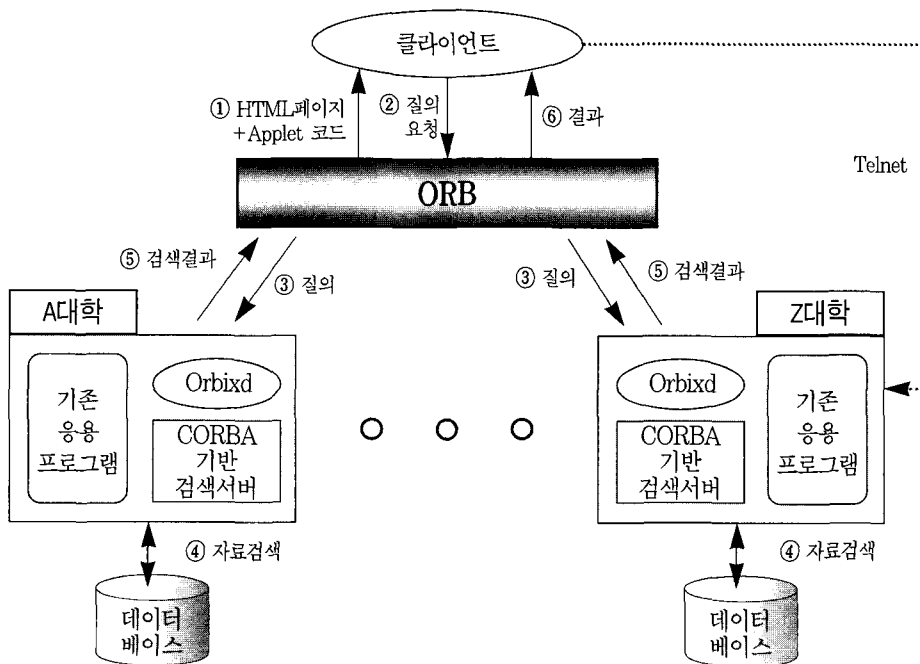
분산 서지정보 검색서버는 클라이언트가 요청한 검색 질의에 대하여, <그림 6>과 같은 과정을 통해 요청받은 질의를 처리한다.

- ① 사용자는 웹상에서 분산 검색서버에 접속하여 자바 애플릿을 다운받는다.
- ② 클라이언트(자바 애플릿)는 웹브라우저에서 서명, 저자명, 주제명에 따른 검색질의와 원하는 서버를 선택하여 탐색을 요구한다. (자바 애플릿을 실행한다.)

- ③ 클라이언트(자바 애플릿)는 분산되어 있는 각각의 서버에 검색질의를 보낸다.
- ④ 각각의 분산되어 있는 서버는 자신의 데이터베이스에서 자료를 찾아 클라이언트에게 보낸다.
- ⑤ 클라이언트는 각각의 분산 서버로부터 받은 데이터를 종합하여 사용자의 웹브라우저를 통해 사용자에게 전달한다.

5 분산 서지정보검색시스템 구축

본 연구가 실험으로 설계한 환경은 두 개의 검색서버를 분산서버로 가정하였다. 덕성여자대학교 도서관의 검색시스템인 DISCOVER시스템의 서지 데이터베이스를 두 개의 서로 다른 서버에 설치하여 이를 각 분산데이터베이스로 가정하였다.



<그림 6> 제안된 분산 서지정보검색시스템 모델의 질의처리과정

5.1 개발환경

본 시스템의 개발을 위해 CORBA의 제품으로는 Iona의 OrbixWeb3.0을 이용하였다. 클라이언트 프로그램은 Window환경하의 VisualJ++를 사용하여 JAVA로 작성하였고, 각 서버 프로그램은 Unix 환경하의 JDK 1.1.6을 사용하여 JAVA로 작성하였다. 또한 각각의 데이터베이스는 오라클7.3을 사용하였고, 응용서버와 오라클 데이터베이스와의 연동을 위해 JDBC를 이용하였다.

5.2 서버구현

서버프로그램은 ORB.init()을 통해 초기화하고 impl_is_ready() 메소드를 수행함으로써 클라이언트 질의를 받을 준비를 한다.

```
public class javaserver1{
  public static void main(String args[]) {
    library libraryImpl=null;
    org.omg.CORBA.ORB orb=
      org.omg.CORBA.ORB.init();
    try { libraryImpl=
      new LibraryImplementation("My Count");
      _CORBA.Orbix.impl_is_ready("libraryjdbc");
      orb.disconnect(libraryImpl);
    } catch (org.omg.CORBA.SystemException se) {
      System.out.println("Fail" + se.toString()); }
  } }
```

CORBA IDL에서 정의한 library 인터페이스에 해당하는 서버 객체를 생성한다. 데이터베이스 연결의 초기화로 driver_class와 connect_string 값을 설정하고 Driver Manager.getConnection() 메소드를 호출함으로써 오라클 데이터베이스와 연결을 설정하고 클라이언트로부터 넘어온 질의와 검색조건을 가지고 각 대학의 데이터베이스를 탐색한다. 각 대학 서버 프로그램과 오라클 데이터베이스 연

동은 JDBC(Java DataBase Connectivity)의 thin driver를 이용하여 구현하며 주요 코드는 다음과 같다.

```
public class LibraryImplementation extends
  _libraryImplBase {
  static final String driver_class =
    "oracle.jdbc.driver.OracleDriver";
  static final String connect_string =
    "jdbc:oracle:thin:@somenice.hufs.ac.kr:
    1521:ORA73";
  Connection conn;
  public String() find_book(String name, int separator
    _index) {
    try {
      Class.forName (driver_class);
      conn = DriverManager.getConnection
        (connect_string, "userid", "passwd");
      Statement stmt = conn.createStatement();
    } catch (Exception e) System.err.println(e);
  } }
```

5.3 클라이언트 구현

클라이언트 프로그램은 정의한 인터페이스에 해당하는 프락시(proxy) 객체를 사용자가 선택한 대학의 수만큼 생성한다. 사용자의 검색어와 검색조건이 프락시 객체를 통해 각 대학 서버에 동시에 전달된다. 또한 각 대학서버와의 연결을 위해서 호스트, 서버, 그리고 객체 이름을 매개변수로 가진다.

```
public void bindObject() {
  try { library1Ref =
    libraryHelper.bind(":library1", "namhae.
    duksung.ac.kr");
    library2Ref =
    libraryHelper.bind(":library2",
    "cs.duksung.ac.kr"); }
  catch (SystemException se) {
  displayMsg ("Fail to Connect...\n" + se.toString()); }
}
```

각 대학 서버와의 연결이 끝나면 사용자로부터 받은 검색조건을 Z_retrieve() 메소드를 이용하여 각 대학 서버에 넘긴다.

```
public void Search_Library() {
    try {result1S =
        library1Ref.find_book
            (author.seperator_index);
        result2S =
            library2Ref.find_book
                (author.seperator_index);
    } catch (SystemException se) {
        displayMsg ("Search Fail \n" + se.toString()); }
}
```

5.4 인터페이스 구현

분산 서지정보검색시스템의 인터페이스 검색 화면은 네 부분으로 나뉜다. 각 대학을 선택할 수 있는 부분, 지역을 선택할 수 있는 부분, 검색필드를 선택할 수 있는 부분, 검색어를 입력할 수 있는 검색어 입력상자로 구성되어 있다. <그림 7>은 사용자를 위한 검색 초기화면이다. <그림 8>은 저자 "이문열"을 키워드로 하여 검색한 검색 결과 화면이다. 소장처를 알 수 있다. <그림 9>는 상세 서지정보를 볼 수 있는 화면이다. <그림 8>의 검색 결과물에서 특정 항목을 클릭

하면 선택한 항목에 대한 출판사, 면수, 출판년도 등 보다 상세한 정보를 볼 수 있다.

인터페이스 상에서 질의가 처리되는 과정은 다음과 같다.

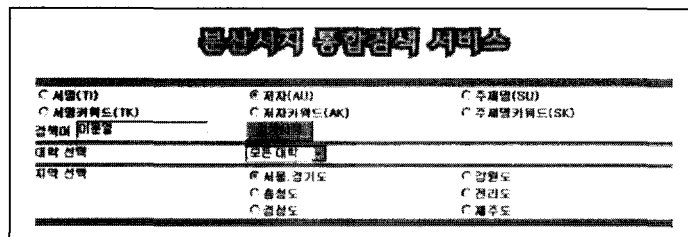
① <그림 7>과 같은 검색 초기화면에서 사용자는 저자, 서명, 혹은 주제명 등의 검색필드 중에서 특정 탐색필드를 선택한다. 검색어 상자에 특정 검색어를 입력한다. 동시에 검색 대상 기관을 한정하여 검색을 요청한다.

② 입력된 질의는 사용자 인터페이스의 애플릿을 통하여 SQL 언어로 검색 대상이 되는 두 개의 분산 서버에 동시에 전달한다.

③ 검색질의를 받은 분산환경의 검색서버는 자신의 데이터베이스를 대상으로 질의를 처리하고 그 결과를 사용자 컴퓨터로 보낸다.

④ 사용자 인터페이스의 애플릿은 검색 결과들을 받아 병합한다. 병합시킨 결과물에 대한 서지사항과 소장처에 관한 정보를 화면으로 출력한다.

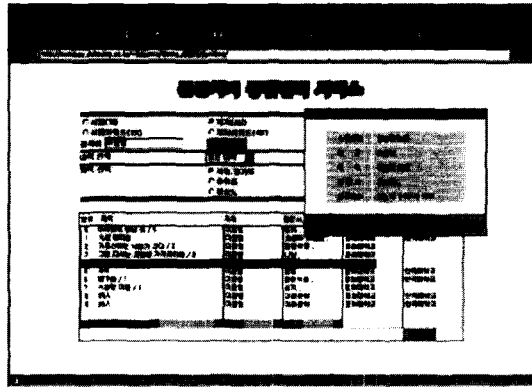
⑤ 검색된 문헌 중 특정항목을 클릭하면 보다 상세한 서지정보를 화면으로 출력한다.



<그림 7> 검색 초기 화면

번호	제목	직지	출판사	소장 위치
1	대우증권 연보 1997 / 5	대우증권	대우증권	충청대학교
2	대우증권 연보 1998 / 2	대우증권	대우증권	충청대학교
3	대우증권 연보 1999 / 5	대우증권	대우증권	충청대학교
4	대우증권 연보 2000 / 1	대우증권	대우증권	충청대학교
5	대우증권 연보 2001 / 1	대우증권	대우증권	충청대학교
6	대우증권 연보 2002 / 1	대우증권	대우증권	충청대학교
7	대우증권 연보 2003 / 1	대우증권	대우증권	충청대학교
8	대우증권 연보 2004 / 1	대우증권	대우증권	충청대학교
9	대우증권 연보 2005 / 1	대우증권	대우증권	충청대학교

〈그림 8〉 검색 결과 화면



〈그림 9〉 검색 결과 화면-상세서지 정보

6 결 론

기존의 서지정보 검색시스템 구현에 사용되고 있는 텔넷 접속방식이나 웹에서의 CGI 기용방식, Z39.50에 기반한 JAVA Application 방식 등은 기술적으로는 중앙 검색서버에서 흩어져 있는 각각의 데이터베이스를 개별적으로 탐색하는 중앙서버 개별검색 방식으로 운용되고 있다. 이러한 중앙서버 개별검색 방식의 제한점은 사용자가 다수의 서버에 접속하고자 할 때 접속을 연결하고 해제하는 과정을 반복해야 하며, 또한 동일한 검색을 여러 서버를 대상으로 수행해야 한다는 점이다.

따라서 본 연구에서는 기존 검색방식의 문제점을 개선하고, 기술적으로는 지리적으로 분산되어 있는 분산 서버상태를 유지하면서 동시에 분산되

어 있는 데이터베이스 검색을 허용하는 통합검색시스템의 개발 가능성을 타진해 보고자 한다.

분산 검색서버 분산 데이터베이스 방식을 채택한 분산 서지정보 검색시스템의 구축을 위해 덕성여자대학교의 정보검색시스템인 DISCOVER와 서지 데이터베이스를 두 개의 서버에 설치하여 실험하였다.

본 시스템구축을 위해 사용된 기술은 CORBA와 JAVA로서, 기존의 검색시스템이 분산 서지정보를 통합 검색할 수 없다는 문제점을 해결하기 위해 CORBA가 사용되었다. CORBA는 개방형 객체지향 분산시스템의 표준으로, 분산되어 있는 서지정보 데이터베이스와 검색시스템을 통합하기 위한 미들웨어(middleware)로 사용되었다.

CORBA는 기존의 CGI 기술의 시스템 정체

(bottleneck)를 해소하고, 클라이언트는 JAVA를 이용함으로써 플랫폼에 독립적인 사용자 접속을 가능하도록 하였다. JAVA를 채택함으로써 검색시의 부하를 클라이언트에 분산시켜 검색속도를 향상시킬 수 있고, 인터넷이 가능한 어느 컴퓨터에서도 원격지에 있는 서지정보를 검색할 수 있다.

구현결과 통합 인터페이스를 통한 두 분산서버에 대한 검색요청은 성공적으로 수행되었으며 각각의 분산 데이터베이스를 탐색한 결과를 클라이언트는 종합할 수 있었다.

분산되어 있는 서지정보가 이와 같이 분산 데이터베이스 방식으로 통합된다면 도서관간의 협력이 수서, 목록, 상호대차 등으로 확대될 수 있는 하부구조가 보다 효율적으로 조성될 수 있을 것으로 예상된다. 더욱이 인터넷상에서의 가상도서관 구축 등이 획기적으로 발전할 수 있는 계기가 될 것으로 전망된다.

분산되어 있는 서지정보가 이와 같이 분산 데이터베이스 방식으로 통합된다면 도서관간의 협력이 수서, 목록, 상호대차 등으로 확대될 수 있는 하부구조가 보다 효율적으로 조성될 수 있을 것으로 예상된다. 더욱이 인터넷상에서의 가상도서관 구축 등이 획기적으로 발전할 수 있는 계기가 될 것으로 전망된다.

참 고 문 헌

- 강한배. 1998. 국가전자도서관 구축 기본계획. 『98 한국데이터베이스학회 가을 국제컨퍼런스-국가경쟁력향상을 위한 디지털도서관 구축방안-자료집』. 서울: 한국데이터베이스학회.
- 공봉석. 1998. 국가전자도서관 사례. 『98 한국데이터베이스학회 가을 국제 컨퍼런스-국가경쟁력향상을 위한 디지털도서관 구축방안-자료집』. 서울: 한국데이터베이스학회.
- 김동규, 이상구, 전중훈. 1997. 분산자원검색을 위한 색인기법 연구. 『정보과학회지』, 15(2): 38-46.
- 김성혁. 1998. DL관련 세계기술동향. 『98 한국데이터베이스학회 가을 국제 컨퍼런스-국가경쟁력 향상을 위한 디지털도서관 구축방안-자료집』. 서울: 한국데이터베이스학회.
- 박남섭 외. 1997. 실시간 Java. 『정보처리학회지』, 5(4): 82-87.
- 박재현. 1997. 『코아 코바』. 서울: 영한출판사.
- 사공철, 최석두, 한두완. 1987. 대형컴퓨터를 이용한 대학도서관 업무의 토탈시스템 설계. 『정보처리학회지』, 4(2): 3-30.
- 신영석 외. 1997. 분산처리환경에서 클라이언트/서버 기반 객체관리시스템 설계. 『정보처리』, 4(6): 89-95.
- 안개일, 전우직. 1997. 전자도서관 표준 프로토콜 개요 및 구현방안. 『정보과학회지』, 15(2): 20-28.
- 오정훈, 정준민. 1997. Z39.50(정보검색프로토콜)을 응용한 공동편목 게이트웨이 구축방안 연구. 『정보관리학회지』, 14(1): 183-206.
- 정영미. 1998. 도서관 네트워크에서의 OSI 프로토콜 응용에 관한 연구. 『정보관리학회지』, 14(1): 47-76.
- 최두헌, 백혜정, 박영택. 1997. 전자도서관을 위한 정보검색 지능형 에이전트. 『정보과학회지』, 15(2): 29-37.
- 최한석 외. 1996. CORBA기능을 이용한 정보검색시스템 통합에 관한 연구. 『정보관리학회지』, 13(2): 223-242.
- 최희운. 1998. 디지털 정보서비스를 통한 전문정보센터의 이용자 중심 정보서비스 구현에 관한 연구. 『정보관리학회지』, 15(1): 63-81.

- 현순주. 1998. DL요소 기술과 정보인프라. 『98 한국 데이터베이스학회 가을 국제 컨퍼런스-국가 경쟁력향상을 위한 디지털도서관 구축방안-자료집』. 서울: 한국데이터베이스학회.
- 홍기형, 박치향. 1997. 전자도서관의 요소기술: DBMS와 정보검색. 『정보과학회지』, 15(2): 13-19.
- ANSI/NISO Z39.50-1995. 1995. *Information Retrieval(Z39.50): Application Service Definition and Protocol Specification*. Bethesda, Maryland: NISO Press.
- Baker, Sean. 1998. *CORBA Distributed Objects Using Orbix*. Addison-Wesley.
- Bright, M.W., A.R. Hurson, and Simon H. Pakzad. 1992. "A Taxonomy and Current Issues in Multidatabases Systems." *IEEE Computer*, 25(3): 50-59.
- Kim, Won, Jungyun Seo. 1991. "Classifying Schematic and Data Heterogeneity in Multidatabase Systems." *Computer*, 24(12): 12-18.
- Kim, Won, Injun Choi, Sunit K. Gala, and Mark Scheevel. "On Resolving Schematic Heterogeneity in Multidatabase Systems." *Distributed and Parallel Database*, 4: 251-279.
- Object Management Group, 1991. *The Common Object Request Broker : Architecture and Specification*. New York: John Willey & Sons.
- Object Management Group, 1996. *The Common Object Request Broker : Architecture and Specification*. ftp.omg.org/pub/CORBA.
- Orbix at <http://www.iona.com>.
- Orbix 2 *Programming Guide*. 1997. IONA Technologies PLC.
- Orfail, Robert. 1996. *The Essential Distributed Objects Survival Guide*. New York: John Wiley & Sons.
- Pitoura, Evaggelia, Omran Bukhres, and Ahmed Elmagarmid. 1995. "Object-Oriented in Multidatabase Systems." *ACM Computing Survey*, 27(2): 141-195.
- Ram, Sudha. 1991. "Heterogeneous Distributed Database System." *Computer*, 24(12): 7-10.
- Schatz, B., H. Chen. 1996. "Building Large-Scale Digital Libraries." *IEEE Computer*, 29(5): 22-26.
- Siegal, John. 1996. *CORBA: Fundamentals and Programming*. New York: John Wiley & Sons.