

이질형 데이터베이스에서 Jini와 JDBC를 이용한 검색 시스템

박 제 성, 황 부 현
전남대학교 전산학과
e-mail: jspark@sunny.chonnam.ac.kr

Retrieval System using Jini and JDBC in Heterogeneous Databases

Jesung Park, Buhyun Hwang
Dept. of Computer Science, Chonnam National University

요 약

분산되어 있는 이질형 데이터베이스들을 통합하여 단일 인터페이스를 이용하여 접근 가능하게 하는 시스템에 대한 연구가 진행되고 있다. JDBC는 하나의 응용 프로그램 내에서 여러 데이터베이스와의 연결이 불가능한 문제점을 해결하고 데이터베이스에 대한 일관된 흐름을 제공하는 자바 API이다. 그러나 새로운 데이터베이스가 추가되거나 제거되는 경우, JDBC를 이용하여 이질형 데이터베이스들을 통합하는 방법은 응용 프로그램을 정적으로 재구성해야 한다는 문제점이 발생한다. 또한 CORBA를 이용하여 이질형 데이터베이스를 통합하는 방법에서도 데이터베이스의 추가나 제거가 발생하는 경우, 미들웨어에 존재하는 구현객체의 수정이 필요하거나 추가되는 구현객체의 정보를 알아야한다. 본 논문에서는 지니 기술과 JDBC를 이용하여 사용자가 단일 인터페이스를 통하여 이질형 데이터베이스들에 저장된 정보를 접근할 수 있는 검색 시스템을 제안한다. 제안하는 시스템은 네트워크에 연결된 데이터베이스들의 연결 변화를 동적으로 감지하여 시스템을 자발적으로 구성함으로써, 기존의 이질형 데이터베이스 통합 방법의 문제점을 제거한다. 또한 본 논문에서 제안하는 지니와 JDBC를 이용한 검색 시스템의 모델을 응용하여 이질형의 데이터베이스로 운영되는 쇼핑몰에서 비교 검색 방법을 제안한다.

1. 서 론

네트워크 상에 분산되어 있는 다양한 정보 시스템들을 논리적으로 연결하고자 하는 요구가 증가하고 있다. 그리고 기업형 정보관리 시스템에서 정보는 이질형 데이터베이스들에 분산되어 존재한다. 자료의 분산은 중앙 집중된 데이터베이스 시스템에 비해 각 지역 데이터베이스 시스템의 자치성(local autonomy)을 높이고, 서로 다른 데이터베이스를 동시에 접근함으로써 성능의 향상을 가져온다. 그리고 일반적으로 정보 제공자가 제공하는 정보의 유형은 정보 제공자의 환경에 따라 다를 수 있다. 따라서 서로 다른 환경의 정보 제공자가 제공하는 정보를 클라이언트가 접근하고자 하는 경우에는 구축된 환경에 맞는 각각의 인터페이스가 필요하다. 따라서 분산되어 있는 이질형 데이터

베이스 시스템들의 자율성을 보장하고 효율적으로 정보를 공유 및 교환할 수 있는 시스템의 통합 방법에 대한 연구가 필요하다.

JDBC는 하나의 응용 프로그램 내에서 여러 데이터베이스와의 연결이 가능하고 데이터베이스에 대한 일관된 흐름을 제공하는 자바 API이다. [8,9,10]에서는 JDBC를 이용한 이질적인 데이터베이스 시스템을 제안하였다. 그러나 새로운 데이터베이스의 추가나 제거가 발생하는 경우, JDBC를 이용한 응용프로그램을 정적으로 재구성해야 한다는 문제점이 발생한다.

이질형 데이터베이스를 보다 효율적으로 통합하기 위한 CORBA와 같은 미들웨어(middle ware)를 제공하는 통합 시스템 구조가 제안되었다[7,12]. [7]에서는 CORBA를 이용한 이질적인 데이터베이스 통합 시스템을 제안하였다. 그러나 JDBC를 이용한 시스템 통합 방법과 마찬가지로 데이터베이스의 추가나 제거가 발생하는 경우, 구현객체의 수정이 필요하거나 추가되

※ 이 논문은 한국과학재단 1999년도 특정기초연구비(1999-2-30300-006-3) 지원에 의하여 연구되었음.

는 구현객체의 정보가 필요하다.

지니는 자바를 기반으로 네트워크에 접속되어 있는 다양한 기기들이나 소프트웨어들을 동적으로 연결하고 이들을 상호협동할 수 있도록 해주는 기반 기술이다[1,3,5].

본 논문에서는 지니 기술과 JDBC를 이용하여 지니 클라이언트가 단일 인터페이스를 통하여 이질형 데이터베이스들에 저장된 정보를 접근할 수 있는 검색 시스템을 제안한다. 제안하는 검색 시스템은 JDBC를 이용하여 각각의 데이터베이스를 접근하는 지니 서비스가 동적으로 등록되기 때문에 새로운 데이터베이스가 추가되거나 제거되는 경우, 자발적인 시스템 구성이 가능하다. 또한 제안하는 검색 시스템은 지니를 탑재한 PDA나 PCS로 네트워크에 연결하여 필요한 정보를 검색하는 응용 구조로 이용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로 JDBC, CORBA, 그리고 지니에 대해 기술하고, 3장에서는 지니와 JDBC를 이용한 검색 시스템을 제안하고 시스템 구현동작을 보인다. 끝으로 4장에서는 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 JDBC

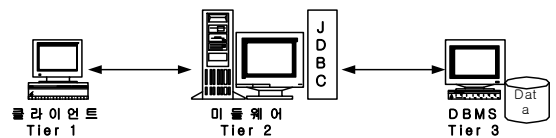
JDBC는 자바 프로그램 내에서 SQL문을 실행하기 위한 자바 API로 X/OPEN CLI(Call Level Interface)로 설계되었으며 CLI는 SQL을 사용하여 데이터베이스와 대화적으로 데이터를 주고 받는다. 그리고 하나의 응용 프로그램 내에서 여러 데이터베이스와의 연결이 불가능한 문제점을 해결하고, 데이터베이스에 대한 일관된 흐름을 제공한다[8,9,10]. 그러나 JDBC를 이용한 응용 프로그램에서는 데이터베이스가 추가되거나 제거되는 경우, 응용 프로그램을 수정이 필요하거나 데이터베이스들을 관리하는 별도의 모듈을 유지해야 한다는 문제점이 발생한다.

2.2 CORBA

CORBA는 OMG(Object Management Group)에서 정의한 분산 객체 컴퓨팅 환경의 표준으로서 이질적인 분산 환경에서 응용 프로그램을 구현하는데 필요한 프레임워크, 객체 서비스, 분야별 객체 클래스 라이브러리를 제공한다. CORBA는 분산 객체들 사이의 통신을 위해 기존의 프로그래밍 언어와는 독립적인 IDL(Interface Definition Language)을 정의하여 이용함으로써 다양한 프로그래밍 언어로 클라이언트에서

의 객체 구현이 가능하다[4].

분산 객체를 통합하는 기술은 현재 웹과 CORBA, 그리고 자바를 기반으로 하는 비즈니스 솔루션을 개발하는 것으로 관심이 모아지고 있다. 자바 언어는 가상 기계라는 개념을 이용하여 이질적 시스템 환경에서 플랫폼과 독립적으로 구현이 가능한 기능을 제공해주고 있으며, 또한 CORBA는 분산객체 환경을 위한 표준으로 자바 언어와 연계하여 응용 시스템을 구축하는 경우에 많은 비용을 줄일 수 있다. 따라서 이질적인 시스템으로 구축된 분산 객체 환경에서 자바로 개발된 응용 시스템과 데이터베이스와의 연동에 관하여 많은 연구가 진행되고 있다[11].



(그림 1) CORBA와 자바를 이용한 시스템 구조

자바와 CORBA ORB를 기반으로 한 시스템 구조는 그림 1과 같이 세 단계로 구성되어 기존의 2-tier 클라이언트/서버 구조에 비해 시스템관리나 보안, 확장성, 애플리케이션 재사용 등에서 좋은 성능을 보인다.

[7]에서는 CORBA 기반의 멀티 데이터베이스관리 시스템을 제안하였다. 제안한 시스템은 내부 통신 구조로 CORBA를 이용함으로써 확장성, 유연성, 시스템 구성의 일관성을 증대시킬 수 있지만 제안한 시스템 역시 데이터베이스 추가나 제거가 필요한 경우 구현 객체의 수정이 불가피하거나 추가되는 구현객체의 위치정보가 필요하다.

본 논문에서는 이러한 문제점을 해결하기 위하여 지니와 JDBC를 이용한 이질형 데이터베이스 검색 시스템을 제안한다. 제안하는 시스템 모델은 데이터베이스의 추가나 제거가 유연하고 동적인 시스템 구조를 갖는다.

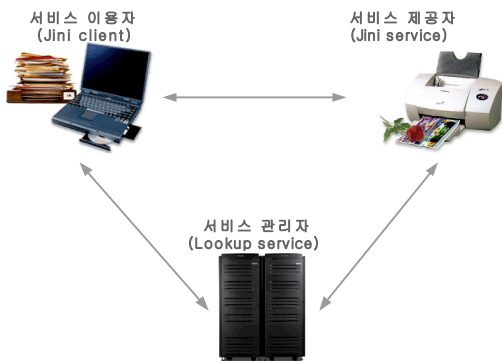
2.3 지니 기술(Jini technology)

지니는 썬 마이크로시스템즈에서 제안하고 있는 접속 기술로서 자바를 기반으로 네트워크에 접속되어 있는 지능형 기기들이나 소프트웨어들이 동적으로 상호 작용 할 수 있게 하는 런타임 인프라스트럭처 기술이다. 런타임 인프라스트럭처는 실행 시에 네트워크 상에 분산되어 있는 구성요소들을 탐지하여 동적으로

시스템을 구성하는 구조라 정의할 수 있다. 따라서 하드웨어나 소프트웨어에 상관없이 지니를 채택하고 있는 기기들에 대한 인위적인 설치나 조작과 같은 일련의 절차를 배제할 수 있는 Plug&Work가 가능하다. 그리고 지니 시스템에 참여하고 있는 모든 기기들이나 소프트웨어들은 서비스로 정의된다[1,5].

지니 시스템은 그림 2와 같이 서비스 이용자(Jini client), 서비스 제공자(Jini service), 서비스 관리자(Lookup service)로 구성된다. 그리고 각각의 구성요소들은 Discovery, Join, 그리고 Lookup 프로토콜을 이용하여 상호 통신하며 서비스 이용자와 서비스 제공자의 기본적으로 자바 RMI(Remote Method Invocation)를 이용하여 상호작용한다[1,3,5].

- ① Discovery : Discovery 프로토콜은 서비스 관리자를 찾기 위한 프로토콜이다. 서비스 제공자나 서비스 이용자는 Discovery 프로토콜의 멀티캐스트(multicast)나 유니캐스트(unicast) 방법을 이용하여 서비스 관리자에게 요청하고 응답을 받아 서비스 관리자를 찾을 수 있다.
- ② Join : 서비스 제공자는 Join 프로토콜을 이용하여 서비스 관리자에게 자신의 서비스 프록시 객체와 속성(attribute)을 등록한다.
- ③ Lookup : 서비스 이용자는 Lookup 프로토콜을 이용하여 서비스 관리자에게 등록되어 있는 서비스 프록시 객체나 속성들을 검색한다. 검색하는 과정에서 원하는 속성에 해당하는 서비스 객체를 가져올 수 있다.



(그림 2) 지니 시스템

지니 시스템의 수행과정은 다음과 같다. 먼저 서비스 제공자는 Discovery 프로토콜을 이용하여 서비스 관리자를 찾고, Join 프로토콜을 이용하여 서비스 관리자에게 서비스의 프록시 객체(proxy object)를 등록한다.

다. 프록시 객체란 서비스와 상호작용을 하는 직렬화된 자바 객체이다. 그리고 서비스 이용자는 Discovery 프로토콜을 이용하여 서비스 관리자를 찾고 Lookup 프로토콜을 이용하여 자신이 원하는 서비스를 검색하여 프록시 객체를 다운로드 받아 서비스와 상호작용한다.



(그림 3) 지니 연결 기술의 구조

지니 연결 기술의 구조는 그림 3과 같다. PDA나 Cell Phone에 지니 기술을 적용하여 네트워크 상에서 Discovery 프로토콜이나 Lookup 프로토콜을 이용하여 서비스를 찾는다. 그리고 서비스와 상호작용하여 동적인 시스템을 이룬다.

분산환경에서 흔히 등장하는 문제 중의 하나는 협력하는 컴포넌트중 일부가 네트워크 이상이나 시스템의 오류로 예고 없이 동작하지 않는 경우이다. 지니에서는 이러한 문제를 해결하기 위하여 특정자원에 대한 허가권을 갖는 리스(lease)라는 개념을 도입하여 분산 환경에서 보다 견고한 시스템을 제공한다. 리스는 자원을 빌려준다는 의미로 사용된다. 즉, 기억공간 등의 자원을 다른 서비스 혹은 클라이언트를 위해 할당해 놓아야 할 필요가 있을 때, 지니에서는 그 자원을 영구 할당하지 않고 단지 특정 시간동안만 빌려준다. 이런 경우 자원을 할당받은 쪽은 필요하면 리스가 끝나기 전에 재요청을 하여 기간을 연장하고, 자원을 할당하는 쪽은 리스를 점검하해서 요청된 기간이 지났으면 자원을 안전하게 회수할 수 있다.[1,5,6]

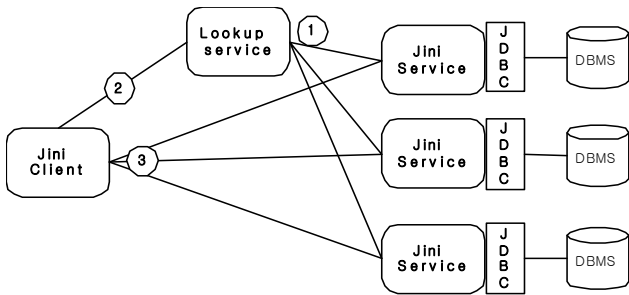
2.4 CORBA와 지니의 비교

CORBA와 지니는 분산 객체 기술이지만 이들 두 가지 기술 사이에는 차이점이 존재한다. CORBA는 이질적인 시스템의 통합을 목적으로 구현 언어에 독립적이다. 즉 CORBA 구현 객체는 C++로 구현하고 클라이언트는 자바로 구현하는 방식이 가능하다. 그러나 지니는 순수 자바 기술이기 때문에 구현 언어에 종속적이다. 그리고 CORBA는 클라이언트가 구현 객체가

존재하는 미들웨어의 위치 정보를 알아야 하지만 지니는 서비스 관리자로부터 서비스 프록시 객체를 다운로드 받아 사용하므로 서비스 정보 없이 바로 사용할 수 있다. 즉, 지니에서는 코드 자체가 이동하는 형식으로 Plug&Work가 가능하다[3].

3. 지니와 JDBC를 이용한 검색 시스템

본 논문에서 제안하는 지니와 JDBC를 이용한 검색 시스템 구조는 그림 4와 같다. 네트워크 상에는 지니 시스템의 구성요소들이 분산되어 있으며 데이터베이스에 접근하는 지니 서비스와 서비스들을 관리하는 Lookup service, 그리고 지니 클라이언트로써 지니 기술을 적용한 PDA가 존재한다.



(그림 4) 지니와 JDBC를 이용한 검색 시스템

3.1 지니 서비스

지니 서비스는 JDBC를 이용하여 각각의 데이터베이스에 접근하며 독립된 지역 호스트에서 구동된다. 지니 서비스가 접근하고자 하는 데이터베이스 종류에 따라 드라이버 로드와 연결을 설정한다. 그리고 지니 서비스는 Discovery 프로토콜을 이용하여 Lookup service를 발견하고 Join 프로토콜을 이용하여 자신의 서비스 프록시 객체를 등록한다(그림4의 ①과정). 이러한 지니 서비스들은 시스템이 운영 중에도 등록이 가능하며 등록하는 즉시 지니 클라이언트는 서비스를 사용할 수 있다. 그러므로 네트워크 상에 동적으로 등록되는 서비스들은 검색 시스템의 구성요소가 된다.

3.2 Lookup service

Lookup service는 네트워크에 연결되어 있는 독립된 호스트에 존재하며 지니 클라이언트에게 지니 서비스를 중재하는 역할을 수행한다. 그리고 지니 서비스들은 Lookup service에 자신의 서비스 객체와 자신을 설명하는 속성들을 등록한다. 또한 지니 클라이언트는 Lookup service에서 원하는 서비스 프록시 객체와 속성을 찾고 서비스 프록시 객체를 다운로드를 하여 사

용한다.

3.3 지니 클라이언트

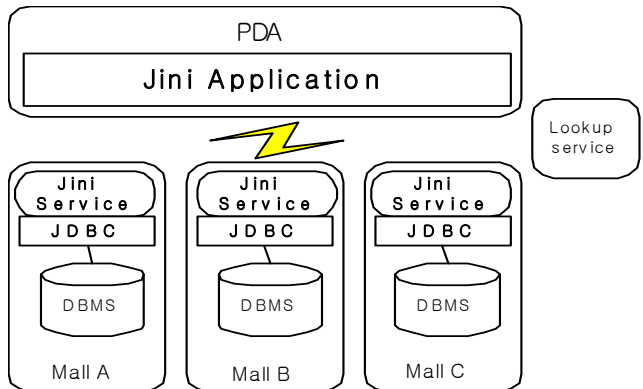
지니 클라이언트는 Discovery 프로토콜을 이용하여 Lookup service를 찾고 Lookup 프로토콜을 이용하여 원하는 서비스 객체를 찾고 다운로드를 한다(그림4의 ②과정). 그리고 지니 서비스와 상호작용하여 데이터를 검색한다(그림4의 ③과정). 그러므로 클라이언트는 JDBC 드라이버나 연결 설정 없이 이질형의 데이터베이스에서 데이터를 검색할 수 있다. 이때 지니 클라이언트는 지니 기술을 장착한 PDA가 된다.

3.4 비교 검색에 응용

본 연구에서 제안한 이질형 데이터베이스에서 지니와 JDBC를 이용한 검색 시스템은 이질적인 데이터베이스로 구성된 전자 상거래 시스템에서의 비교 검색 시스템에 적용할 수 있다.

전자상거래 규모의 확산과 함께 쇼핑몰에서 제공하는 상품의 종류가 다양해지고 거래되는 양 또한 급격히 증가하고 있다. 따라서 보다 효율적인 전자 상거래를 위해서는 비교 분석을 가능케 하는 상위 정보들(meta level information)이 필요하다. 그러나 인터넷 상의 쇼핑몰들은 각기 다른 구현 기술, 플랫폼, 데이터베이스를 사용하고 있으며 이러한 시스템을 통합한 비교 검색 시스템을 구현하기에는 많은 문제점이 있다. [2]에서는 다양한 쇼핑몰들을 통합하는 과정에서 발생할 수 있는 문제점들을 다음과 같이 제시하였다.

- ① 이질형 데이터베이스로 개발된 쇼핑몰들을 어떻게 연결한 것인가?
- ② 각각 쇼핑몰마다 다른 데이터 속성 이름을 어떻게 통합할 것인가?



(그림 5) 지니를 이용한 비교 검색 구조

이러한 문제점들을 해결할 수 있는 비교 검색 시스템 구조는 그림 5와 같다. 먼저 지니 서비스는 각 쇼핑몰

측에서 관리되며 자치적으로 실행되어 서비스 관리자에게 등록된다. 지니 서비스는 각각의 데이터베이스에 맞는 드라이버 로드와 연결을 설정하고 데이터베이스에 접근한다. 그리고 각 데이터베이스의 다른 데이터 속성 이름은 지니 서비스에 스키마 클래스를 설정하여 데이터베이스에 질의가 발생하면 참조한다. 스키마 클래스의 구조는 그림 6과 같다.

```
class Schema
{
    String Table_Name = "A_Table_Name";
    String Model_Code = "A_Model_Code";
    String Model_Name = "A_Model_Name";
    String Maker = "A_Maker";
    String Price = "A_Price";
    String Size = "A_Size";
    .....
}
```

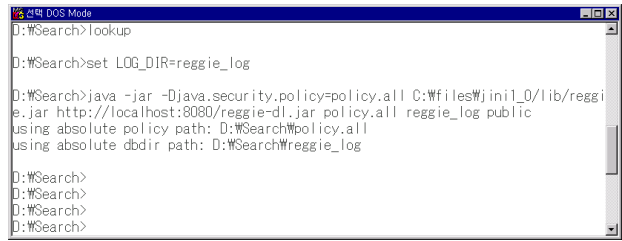
(그림 6) 스키마 클래스의 구조

각 쇼핑몰에서는 지니 서비스의 등록과 해지가 자율적으로 이루어져 하나의 동적인 시스템을 이룬다. 지니가 장착되어 있는 PDA에서 지니 클라이언트가 구동되어 서비스 관리자에게 서비스들을 취득한 후 지니 서비스들과 상호작용하여 각각의 쇼핑몰의 데이터베이스에서 데이터를 검색하고 원하는 결과를 반환받는다. 기존의 구현 방법으로 비교 검색 시스템을 제공하기 위해서는 각각의 쇼핑몰 관리자와 상호 협력하며 주어진 규칙으로 시스템을 운영해야 한다. 그리고 다양하고 많은 쇼핑몰들의 관리가 어렵다. 그러나 이와 같이 지니를 이용한 시스템을 구성하면 분산환경에서 동적이고 자발적인 시스템이 구성되어 시스템 관리나 시스템 비용이 절감될 것이다.

3.5 시스템 구현 동작

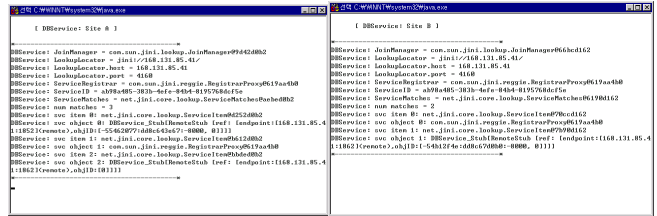
다음은 동적으로 이루어지는 지니와 JDBC를 이용한 검색 시스템의 구현 동작을 보여주고 있다. 네트워크 상의 자바 가상머신에 지니 시스템의 구성 요소들이 있다. 가장먼저 코드의 다운로드를 위해 웹서버가 구동된다. 웹서버는 지니 시스템에 필요한 코드의 다운로드를 제공하며 인터넷상에 별도의 호스트에서 구동된다. 다음은 Lookup service를 구동하기 위해 별도의 호스트에서 RMID(RMI Activation Deamon)와 Lookup service가 구동된다. RMID와 Lookup service는 같은 호스트에서 구동되어야 하며 서비스들을 주

선하는 역할을 한다[1,3].



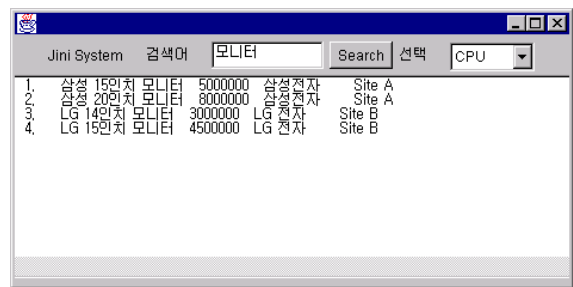
(그림 7) Lookup service 구동 화면

다음에는 각 데이터베이스를 접근하는 지니 서비스가 구동된다. 지니 서비스는 Lookup service를 발견하고 자신의 서비스 프록시 객체를 등록한다. 그림 8은 독립된 호스트 Site A와 Site B에서 구동되는 지니 서비스 화면이다.



(그림 8) Site A와 Site B에서 지니 서비스 구동

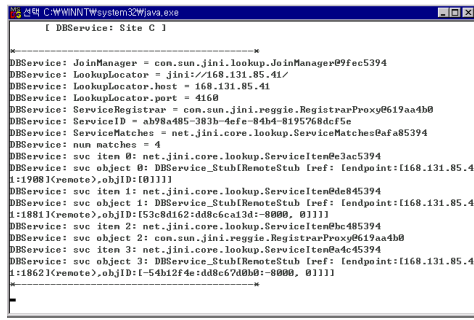
Site A와 Site B의 지니 서비스가 구동된 후 지니 클라이언트가 구동된다. 지니 클라이언트는 지니 서비스들과 동적으로 검색 시스템을 이루어 검색조건을 주어 원하는 정보를 얻는다. 그림 9에서 지니 클라이언트가 모니터라는 검색어를 Site A와 Site B에 있는 지니 서비스에 전달하여 검색된 결과이다.



(그림 9) 지니 클라이언트 검색 결과

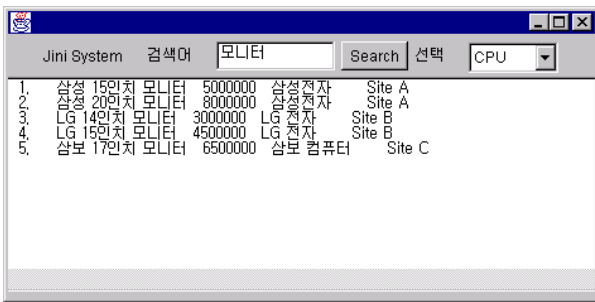
이와 같이 지니를 이용한 검색 시스템이 구동된 상태에서 Site C의 지니 서비스가 실행되어 Lookup service에 등록된다. 지니와 JDBC를 이용한 검색 시스템은 Site C의 지니 서비스가 추가되더라도 시스템의 재구성이 필요없이 자발적으로 확장되어 구동된다.

그림 10은 Site C에서 구동되어지는 지니 서비스 화면이다.



(그림 10) Site C에서 지니 서비스 구동

그림 11과 같이 지니 클라이언트에서 다시 모니터라는 검색어로 검색을 하면 Site C에 있는 정보가 추가되어 검색 결과를 볼 수 있다.



(그림 11) Site C가 정보가 추가된 지니 클라이언트 검색 결과

3.6 시스템 장점

3.5절에서 검색 시스템의 구동 과정을 보았듯이 새로운 지니 서비스가 추가되더라도 시스템의 수정이나 변경 없이 시스템이 동적으로 확장된다. 지니 서비스들은 마치 하나의 컴포넌트처럼 동적으로 조립되어 시스템을 구성할 수 있다. 따라서 지니 시스템에서는 다른 분산 객체 기술과는 달리 분산 객체를 제공하는 서비스의 위치 정보를 알지 않고도 분산 환경에서 상호 작용할 수 있다. 또한 지니 기술을 장착한 PDA가 지니 클라이언트가 되어 검색에 필요한 최소의 코드만을 유지하여 네트워크로 연결되어 다양한 지니 서비스와 상호작용이 이루어진다. 즉 검색 시스템에 필요한 구성요소와 기능을 네트워크 상에 분산하여 필요할 때 지니 서비스들을 취합하여 원하는 검색 결과를 얻을 수 있다. 지니 시스템은 분산된 시스템을 강건하고 유연하게 시스템을 구성한다.

4. 결론 및 향후 연구 방향

지니는 네트워크에 접속되어 있는 기기들을

Plug&Work로 제어하기 위한 기술일 뿐만 아니라 본문에서 제시한 이질형 데이터베이스들을 통합한 검색 시스템 같은 여러 가지 시스템 기술의 솔루션을 제공한다. 따라서 지니 기술을 이용하면 분산 환경에서 보다 적은 개발비용과 동적이고 강건한 시스템을 만들 수가 있다.

향후 연구방향으로는 지니 기술을 기반으로 사용자에게 더욱 편의를 제공하는 검색 에이전트 시스템에 관한 연구가 필요하다.

참고 문헌

- [1] Core JINI, W.KEITH EDWARDS, Prentice Hall <http://www.phptr.com>
- [2] Jae Kyu Lee, Uk Song, Jae Won Lee, "A Comparison Shopping Architecture over Multiple Malls : The Meta-Malls Architecture", ICEC '98, April 6, 1998.
- [3] JINI Tutorial Jan Newmatch <http://pandonia.canberra.edu.au/java/JINI/tutorial/>
- [4] Robert Orafali, Dan Harkey, and Jeri Edwards, "Instant CORBA", Wiley Computer Publishing.
- [5] SUN Microsystem JINI Connection Technology <http://www.sun.com/JINI/>
- [6] 구건, 전진수, 김정선, "Jini 기반의 자바 분산 일괄처리 시스템(J-JDBS)의 설계", 한국정보처리학회 추계 학술발표논문집 제6권 제2호, 1999.
- [7] 김지운, 김보경, 이근철, 문강식, 이진영, "CORBA 기반 멀티 데이터베이스 관리 시스템의 설계", 한국정보과학회 봄 학술발표논문집, 1997.
- [8] 서정민, 진은숙, 윤수영, 송원주 "다중 연결을 지원하는 JDBC 드라이버의 구조", 한국정보과학회 가을 학술발표논문집, 1998.
- [9] 박정인, 전순미, "Java RMI 기반의 3-tier 클라이언트/서버 JDBC 드라이버 설계", 한국정보과학회 가을 학술발표논문집, 1998.
- [10] 안길수, 강민규, 주수중, "JDBC를 이용한 멀티-데이터베이스 시스템 설계 및 구현", 한국정보처리학회 춘계 학술발표논문집, 1999.
- [11] 이진용, 전순미, "분산객체에서의 JDBC 3-tier 모델의 성능확대와 CORBA의 이용", 한국정보처리학회 논문지, 1998.
- [12] 이현숙, 진은숙, 송원주, "다중 데이터베이스 미들웨어 시스템의 구조 설계", 한국정보처리학회 춘계 학술발표논문집, 1999.