클라이언트기반 분산객체 메타검색엔진의 설계 및 구현

김 창근, 김 광수, 박 규석
경남대학교 컴퓨터공학과

Design and Implementation of Client Based Distributed-Object Meta-Search Engine

Chang-Gun Kim, Gwang-Soo Kim, Kyoo-Seok Park
Dept of Computer Science, Kyungnam University

요 약

전 세계적으로 산재한 검색 엔진들은 그 수가 많아짐에 따라 일관성이 없는 정보를 제공하고 있거나 어느 특정한 곳에 집중된 정보를 제공하는 경우도 있어 사용자가 원하는 정보를 얻기 위해서는 많은 검색 엔진들을 일일이 조사해야 하고, 각각의 검색 엔진들의 특성을 알아야 하는 불편한 점이 있다. 본 논문에서는 기존의 검색엔진과 메타엔진의 2단계 방식의 클라이언트서버 운용에 따르는 서버 시스템의 부하와 네트웍의 트래픽을 줄일 수 있는 Component의 할당정책을 제시하고, 또한 CORBA를 이용한 메타엔진을 설계하고 구현하였다.

1. 서론

최근 WWW(World Wide Web)이 출현한 이후로 정보의 종류와 양이 기하급수적으로 증가함에 따라 자료 검색의 효율성을 위하여 검색엔진이 출현하게 되었고, 이를 위하여 앞다투어 검색시스템을 개발 중에 있다. 현재 검색엔진에는 디렉토리방식을 지원하는 '야후', 단어별 검색방식을 지원하는 '알타비스타'나 '인포시크' 등이 있으며, 자신들의 URL을 등록 할 수 있는 검색엔진들도 있다.

이들 검색엔진들은 등록된 자료에 대한 계속적인 작업이 필요하게되며, 이를 해결하기 위하여 웹 로봇 혹은 스파이더 등으로 정보를 수집하고 있다..

인터넷에서 동작중인 웹 로봇은 대략 200여개 이상이며, 통계적 분석, 유지보수, 미러링, 자원 탐색(Resource discovery)및 종합적인 작업을 하는 로봇으로 설계되어 있다.

그러나, 웹 로봇은 래피드 파이어(Rapid Fire)와 같은 오동작이나, 버그 등으로 인한 네트웍의 트래픽과 상대방 웹서버에 대한 부하로 인해 문제를 발생시킬 수 있다.[1]

또한, 현재의 메타검색엔진은 각 서버에 질의를 보낸 후에 그 결과를 조합하여 보여주는 역활을 수행하며, 이러한 방법은 네트웍의 부하를 줄일 수 있고, 자체의 큰 저장장치가 필요 없다는 장점이 있으나, 본 논문에서는 메타엔진의 2단계방식의 클라이언트 서버운용에 따르는 서버시스템의 부하와 기존의 검색엔진의 문제점을 해결하기 위하여 분산객체인 CORBA를 이용한 메타검색엔진을 설계하고 구현하였다. 본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구, 제 3장에서는 CORBA를 이용한 메타엔진의 Components를 설계하고, 제 4장에서는 구현에 대한 설명, 제 5장에서는 결론을 맺는다.

2. 관련연구

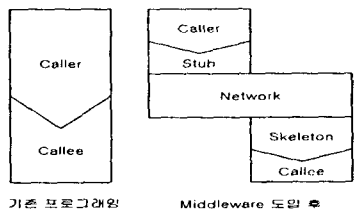
2.1 검색엔진의 원리와 메타엔진

검색엔진은 정보를 얻기 위해 문서들을 검색하고, 참조되는 문서들을 재귀적으로 검색하면서 웹의 하이퍼텍스트 구조를 자동적으로 추적하여 정보를 저장해주는 프로그램이다. 이러한 문서들을 재귀적으로 검색하기 위해서는 웹의 프로토콜인 HTTP를 준수하는 프로그램이어야 한다.[2]

메타 검색 엔진은 기존의 서비스 중인 웹 검색 엔진에 사용자의 질의어를 보낸 후에, 각 검색 엔진의 결과를 조합해서 보여주는 역할을 수행한다. 그러므로, 이들 메타 검색 엔진은 심각한 네트웍 트래픽을 유발시키지 않으며, 타 검색엔진의 데이터를 이용하기에 때문에 내부적으로 데이터를 저장하기 위한 큰 저장공간이나 별도의 데이터베이스가 필요하지 않는 잇점이 있다.[3]

2.1 CORBA의 원리와 개념

CORBA(Common Object Request Broker Architecture)란 Application들끼리 어디에 위치하고 있든, 혹은 누가 만들었건 관계없이 상호간 통신을 보장함으로써 다양한 H/W, S/W가 함께 하는 오늘날의 네트웍 분산환경을 지원하는 새로운 통합기술 시스템이다. CORBA는 ORB의 기능과 상호운용 할 수 있도록 오브젝트를 만들 수 있는 IDL과 API들을 정의하고 있으며[4], IDL은 사용자가 클라이언트와 구현 객체 사이의 통신을 담당할 것인지에 대한 여부와 구현 시 사용하는 프로그램 언어에 따라 각기 다르게 작성해야하는 문제를 해결하고자 제공하는 표준언어이고, API는 프로그램언어로 작성된 클라이언트 코드와 구현객체사이의 통신에 필요한 모든 기능을 제공한다.



<그림 1> Middleware 시스템구조

그림 1에서 보는 바와 같이 CORBA에서는 기존에 클라이언트와 서버가 바로 맺어지던 프로그래밍 환경에 비해 Stub와 Skeleton이라는 개념을 통해 양자간 통신을 가능하게 해준다. 이를 통해 프로그래머는 프로토콜이나 플랫폼 등의 컴퓨팅 환경을 고려할 필요 없이 자신의 프로그램을 만들 수 있다.

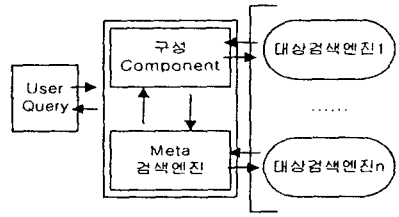
ORB는 각 오브젝트 간에 Client/Server 관계를 맺어주는 Middleware의 일종이며, 클라이언트가 사용자 눈에 보이지 않게 서버 오브젝트의 메소드를 작동시키게 해준다.

기존의 클라이언트/서버 환경에서는 프로그래머가 디바이스간의 통신을 위해 정의된 표준 프로토콜을 사용하거나 스스로 만들어야했다. 프로토콜의 정의는 프로그래밍 언어나 네트웍 환경, 기타 많은 요소에 따라 달라진다. 따라서 프로그래머는 특정 환경에 Specific된 프로토콜을 만들어야하며, 다른 환경에 대해서는 또 다른 프로토콜을 만들어야했다. 하지만 ORB는 프로토콜의 정의가 IDL이라는 독립적인 하나의 언어를 통하여 가능케해주며, 이것은 ORB가 제공하는 유연성 때문이다.

CORBA는 오브젝트 오리엔티드 표준화 및 상호운영의 발전의 한 과정이라고 할 수 있다. CORBA를 통해 사용자는 정보가 어떤 S/W, H/W 플랫폼 위에 있고 네트웍 상에 어디에 존재하는지 알 필요 없이 투명하게 접근할 수있다. 즉, CORBA는 오늘날의 컴퓨팅 환경에서 가장 적합한 공용성 및 상호 운용성을 가져다준다. [4, 5, 6]

3. 시스템의 설계

본 논문에서 개발하고자 하는 메타검색엔진을 CORBA로 구축한다.



<그림 2> 검색엔진의 구조

그림 2는 본 논문에서 설계한 메타검색엔진의 전체 구성도 이다. 본 시스템은 사용자로부터 질의를 받아 가장 적합한 Component를 찾아 각 검색엔진에 질의를 보낸 후 결과 값이 리턴되면 하나로 통합한 다음 재 순위를 부여하여 사용자에게 결과 값을 전송한다. 그리고 질의 선택으로 사용자가 검색하고자 하는 범위 즉 디렉토리를 설정하면 질의결과에 대해

서 좀더 정확한 검색 결과를 찾을 수 있도록 설정할 수 있다.

3.1 Query Analyzer

Query Analyzer는 사용자가 본 시스템으로 접속을 했을 때 사용자로부터 질의와 Boolean 연산을 받아들여 Object Manager에게 전달하고 사용자가 원하는 정보의 내용이 무엇인지를 파악한다. 또한 사용자의 요구에 적절히 동작하도록 사용자의 정보를 유지 관리하도록 한다.

3.2 Object Manager

Object Manager는 Query Analyzer로부터 질의를 전달받아 질의와 각 Query Object가 어떤 상태인지를 파악하여, 사용자의 질의에 대해 동작할 수 있는 Query Object를 선택하고 동작을 조절하는 기능을 수행한다. 그리고 대상 검색엔진의 상태를 체크해서 사용자의 질의에 적절히 응답할 수 있는 검색엔진을 조사하고 그에 대한 정보를 Query Object에 전송하여 대상 검색엔진을 결정하게된다.

3.3 Query Object

Query Object는 각 검색엔진들과의 통신을 담당하는 부분으로써 해당 검색엔진에게 질의를 보내어 결과 값을 되돌려 받게 된다. 각 검색엔진의 수행은 병렬적으로 처리되어 수행시간을 줄이며, 그리고 각각의 검색엔진에서 넘어온 결과를 통합해서 검색결과 통합자(Search Result Integrator)에 넘겨주게 된다.

3.4 Search Result Integrator

검색결과 통합자는 Query Object가 받은 결과 값을 전달받아서 하나로 통합하고 재 순위를 부여하는 일을 수행한다. 우선, 각 검색엔진으로부터 넘어온 순수한 결과 값 이외의 부수적인 데이터가 있을 경우 이를 필터링한 후, 하나의 결과 값으로 만들기 위해 중복된 값들을 제거한다. 마지막으로 재 순위를 결정하여 사용자에게 보내준다. 재 순위 결정 알고리즘은 다음과 같다.

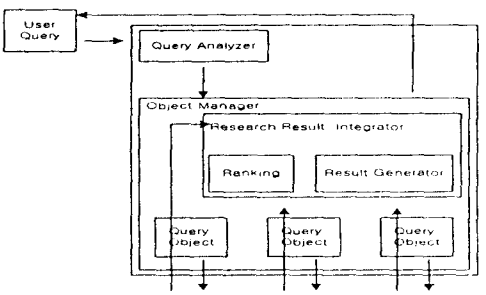
①재 순위결정은 검색엔진에서 먼저 온 데이터에게 높은 우선 순위를 부여하고 차츰 그 값을 줄여 나가는 방법을 취한다.

②중복을 제거 할 때 여러 개의 검색엔진에 공통적으로 들어 있는 데이터는 사용자의 요구에 근접했을 가능성이 높기 때문에 가중치를 증가시킨다.

③통합된 데이터를 가중치의 값에 따라 정렬한다. 그리고, 재 순위 결정을 통해 통합된 데이터를 받아서 사용자에게 전달해줄 HTML 문서의 형태로 변환하고 이를 사용자에게 전달한다.

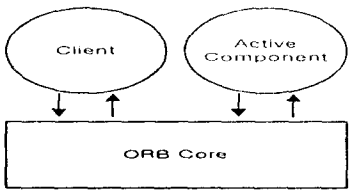
3.5 Components 구성과 동작

그림 3은 Components의 전체구성을 나타내는 그림으로써, 해당분야의 여러 가지 기능을 가지는 Component들로 구성되어있다. 예를 들면 예술과 인문에 대한 Component, 비즈니스와 경제에 관련된 Componenst, 교육에 관련된 Component 등으로 각각의 특성에 맞는 Component로 구성할 수 있다. 각 Component는 해당 분야의 내용을 검색 할 수 있도록 구성하고 이러한 Component들의 구성은 사용자의 요구가 늘어나면 확장 가능할 뿐만 아니라, 필요에 따라서는 새로운 Component들을 만들 수도 있다.



<그림 3> Component의 구성과 동작

이러한 구성은 각 검색엔진이 가지고 있는 디렉토리 서비스 기능을 하나로 통합할 수 있고, 통합에 의해 사용자의 질의에 좀더 근접한 정보를 검색할 수 있다. 이 Component들은 각각의 검색엔진의 인터페이스에 맞도록 설계되어 있으며 사용자의 질의와 선택으로 검색엔진에 전달되며 각 검색엔진으로부터 결과 값을 리턴 받게된다.



<그림 4> Components와 ORB의 통신관계

그림 4는 Components와 ORB간의 통신관계를 나타낸 것이다. Client와 Components 간의 통신 관계는 ORB에서 처리하게되고, Client 쪽에서 각각의 Component간 요구 또한 ORB가 처리하게 된다.

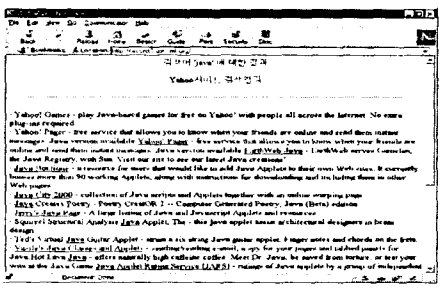
클라이언트가 ORB에게 요청을 보냈을 때 ORB는 그 요청을 만족하는 구현을 선택하고 구현을 직접 요청하기 위해 BOA(Basic Object Adapter)를 사용하게되며, BOA는 그 구현 중에 메소드들을 부르기 위해 서버 조직(Server Skeleton)들을 사용한다. 이러한 Client와 Components 간의 통신은 CORBA에서 처리하게 되고 Components 구성으로 유지복구 및 확장의 편의성도 함께 제공한다..

4. 구현

현재 Web상에서 상호작용하여 동작하는 방법에는 Java에서의 Java RMI(Remote Method Ivocation)와 DCOM(Distrubuted Component Object Model), RPC(Remote Procedure Call) 그리고 CORBA를 이용하는 방법 등이 있다.

본 연구에서는 Visigenic의 Visibroker와 JDK1.1.5, Pentinum 150, Windows NT4.0기반에서 구현하였다.

본 논문에서 개발한 검색엔진은 플랫폼에 독립적으로 실행가능한 Java와 각기 다른 언어로 만들어진 객체간의 연결통로로써 매우 뛰어난 미들웨어이며, 호환성이 뛰어 나고 개발 및 유지 보수가 쉬운 CORBA를 이용하였으며, Netscape와 Java의 통신관계는 Netscape4.5에서 지원하는 LiveConnect API를 이용하였다



<그림 5> 검색결과

그림 5는 검색결과를 나타내는 그림으로써, Yahoo에 검색 질의를 넘겨 카테고리가 결과로 넘어온 내용을 보여 주고 있다.

LiveConnect는 Java에서 Netscape의 Java Script를 호출 가능하도록 지원한다.[7, 8] 그러나 이러한 상호작용에도 불구하고 Java와 Netscape의 data 전송에서 상당히 느리다는 단점을 가지고 있다.

5. 결론

사용자가 원하는 정보를 쉽게 찾을 수 있도록 도와주기 위한 검색엔진은 늘어나고 있지만 대부분의 검색엔진이 사용자의 요구를 만족시키지 못하고 있다. 본 연구에서는 메타검색엔진을 CORBA로 설계하고 구현하여 기존의 검색엔진이 가지고 있는 서버 부하 및 네트웍 문제점을 해결하였고, 네트웍 효율성도 증가하였다. 또한 디렉토리 서비스를 통합하여 사용자의 요구에 어느 정도 만족하는 기능을 부여하였다. 향후 연구의 필요성은 완벽한 메타 검색엔진의 구현과 자체 저장장치를 통합하여 검색엔진의 효율성을 증가시키고 또한 처리속도 문제를 해결하는데 지속적인 연구가 필요하다.

참고문헌

[1] Robots in the Web: threat or treat? "http://info.webcrawler.com/mak/projects/robots/threat-or-treat.html", Martijn Koster, April 1995

[2] 까치네 로봇의 원리와 서비스형태 "http://lactt.taegu.ac.kr/~hunkim/w4/" 대구대학교 정보통신 공학부 컴퓨터 응용연구실

[3] "META 검색엔진을 이용한 웹 로봇에 대한 연구 "p.461-464

[4] Object Management Group, CORBAservices Common Object Services Specification, OMG 95-3-31, 1995

[5] CORBA를 이용한 Web 전자상거래 시스템의 설계 및 구현 "http://kitel.co.kr/~clang/commerce.htm, 정찬용

[6] Daniel D, Adele E. H, "An Information Gathering Agent for Querying Web Search Engines". Technical Report CS-96-11. Colorado State Univ., 1996

[7] Robert Orfail, Dan Harkey, "Client/Server Programming with Java and CORBA", Wiley, 1997

[8] "http://developer.netscape.com/docs/index.html", Technical Report DevEdge