

로봇 에이전트 원격 관리 시스템의 설계 및 구현

김성* 박철우* 황호찬* 박규석**
경남대학교 컴퓨터공학과

Design and Implementation of Remote Control System for Robot Agent

Sung Kim*, Chul-Woo Park*, Ho-Chan Hwang*, Kyoo-Seok Park**
Dept. of Computer Science, Kyungnam University

요 약

최근 들어 급속한 인터넷 사이트의 증가와 이용자들의 증가로 인해 이용자들의 요구에 상응하는 정보를 제공하기 위한 검색 엔진들의 개발이 가속화되고 있다. 이용자들의 다양한 요구와 엄청난 정보를 가진 웹(WWW, World Wide Web)의 상호 유기적인 작용을 위해서 검색 엔진들의 역할이 지속적으로 커지고 있다. 본 연구에서는 네트워크의 부하를 고려하고, 이용자들의 요구에 부합하는 원격 관리가 가능한 로봇 에이전트를 제안하고 설계 및 구현한다.

1. 서론

인터넷의 발전과 더불어, 검색엔진의 발전 또한 커다란 변화를 가져왔다. 인터넷이 처음 등장할 당시에는 텍스트 위주의 정보교환이나 의사 전달과 같은 커뮤니케이션 기능이 주된 역할이었기 때문에 검색엔진 또한 단순히 문서를 수집해서 분류한 후 실시간으로 제공하는 것이 주된 역할이었다. 그러나 현재는 인터넷의 급격한 발전과 이용자들의 증가와 다양한 요구로 인해서 실시간 정보 제공과 더불어 이용자의 특성을 분석해서 이용자가 원하는 정보를 카테고리별로 분류해서 서비스 할 수 있는, 네트워크 부하를 최대한으로 고려한 검색엔진의 설계 요구가 발생하게 되었다.

본 논문에서는 인터넷을 이용한 사용자의 특성에 맞는 인터넷 정보 시스템 모델을 제안하고, 웹을 통해서 원격 제어가 가능한 실시간 제어 검색엔진을 설계 및 구현하였다. 또한 플랫폼에 독립적인 로봇 에이전트를 설계 및 구현하였다. 따라서 분산되어 있는 정보를 쉽게 모아서 갱신해 줄 수 있는 지능적인 로봇의 구현과 주기적으로 네트워크의 부하를 모니터링하고 모니터링한 결과를 기반으로 한 혼합 정렬 스케줄링 기법을 보완하여 전체 시스템의 성능을 향상시킬 수 있는 실시간 로봇 제어 정책이 가능하게 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 검색 기

법에 관련된 연구를 살펴보고, 3장에서는 네트워크 과부하를 고려해서 원격 제어가 가능한 실시간 제어 검색엔진에 대한 시스템을 설계하고 4장에서는 구현 및 평가, 5장에서는 결론을 내린다.

2. 관련연구

2.1 로봇 에이전트

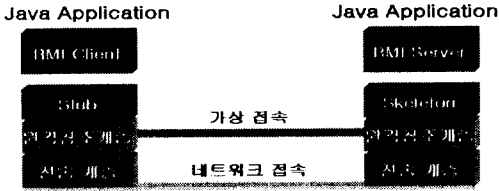
로봇 에이전트란 원하는 정보를 얻기 위해 웹상의 문서들을 검색하고 참조되는 문서들을 재귀적으로 검색하면서 웹의 하이퍼텍스트 구조를 추적하여 정보를 저장해 주는 프로그램이다.[1]

2.2 로봇 혼합 정렬 스케줄러

혼합정렬 스케줄러는 정렬스케줄러에서 나온 리스트와 최적 부하시간 탐색기의 결과로 나온 리스트를 결합하여 우선순위 테이블을 생성한다.

2.3 RMI

RMI (Remote Method Invocation : 원격 메소드 호출 기능)을 이용하여 클라이언트 컴퓨터에서 실행되는 프로그램이 원격서버 컴퓨터에 위치한 객체의 메소드를 호출할 수 있다.[2]



[그림 1] RMI 구조도

그림1의 구조도에서 Stub은 원격 객체를 대변하는 클라이언트 측 대리자 원격 객체가 구현하는 인터페이스를 정의, 프로그램 내에서 지역적인 다른 객체처럼 참조되어 서버측 객체와 접속을 유지한다.

원격 객체가 호출 되면 Stub은 파라미터 데이터를 직렬화 시킨 후 정렬 스트림(Marshal stream)형태로 원격참조계층으로 전달한 후, 원격 메소드 수행이 끝나면 다시 원격참조계층으로 부터 정렬스트림을 받아 역직렬화를 통해 값을 리턴한다.

정렬 스트림은 서버측의 RRL과 클라이언트측의 RRL과의 통신상에 사용된다.

Skeleton은 서버측 RRL과 인터페이스하는 서버측의 생성체로서 클라이언트 측 RRL에게 메소드 호출요청을 받으면 서버측 RRL은 원격 메소드로 전송되어온 파라미터들을 복원시킨 후 Skeleton이 서버측에 구현된 실제 객체를 호출한다. 원격 객체에서 리턴 값을 받아 다시 정렬 스트림으로 정렬 후 서버측의 RRL에 전달한다.

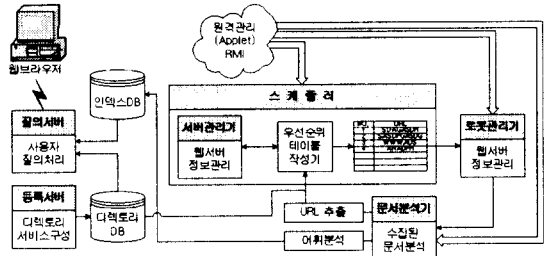
원격참조계층은 특정한 Stub 또는 Skeleton 모델에 종속적이지 않은 독립 참조 프로토콜을 관리하여 다른 두 계층에 영향을 주지 않고 RRL교체가 가능하다. 하위 운송 인터페이스와 작용하여 Stub과 Skeleton 계층에 스트림을 제공한다.

전송계층은 클라이언트와 서버사이의 접속을 만들고, 기존의 접속을 관리하고, 자신의 주소 공간에 상주하는 원격 객체를 처리한다. 전송계층이 클라이언트측 RRL에서 요청을 접수하면 원격 객체에 적합한 RMI서버를 검색하여, 그 서버에 대한 소켓을 만들고 이 접속을 클라이언트 측 RRL에 전달한후 내부 테이블에 그 원격 객체의 참조를 추가시켜 연결을 확립한다.

3. 시스템 설계

3.1 시스템 구성

정보 검색 시스템의 구성도는 그림2과 같다.



[그림 2] 정보 검색 시스템

그림 2와 같이 정보를 검색하기 위한 각 웹서버의 정보를 관리하는 서버관리기와 서버관리기에서 우선 순위별로 재구성된 검색대상 URL의 문서를 수집하는 로봇의 관리를 맡는 로봇관리기, 그리고 그 수집된 문서를 분석하는 문서분석기, 사용자의 질의를 처리하여 결과를 내는 질의 서버등으로 구성되어 통합된 서비스를 제공한다. 이 서비스는 통괄하여 RMI를 이용한 원격관리 시스템에 의해 통괄 관리되며 수시로 변하는 상황에 관리자는 민첩한 대처를 해나갈 수 있도록 서비스되어진다.

3.2 서버관리기

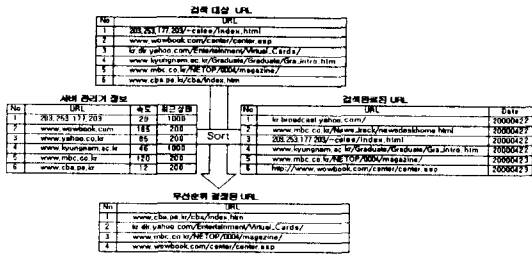
다수의 웹서버에 대한 정보를 관리하는 모듈로서 웹서버마다의 시간대별 상황 및 문서 수집 상태를 저장 관리하게 된다.

서버관리기는 로봇이 가져온 웹문서에서 분석된 모든 URL에 대해 관리항목이 아닌 웹서버는 신규로 등록을 시켜 차후 스케줄링에 의해 사용될 수 있는 유용한 정보를 창출하기 위해 사용된다.

웹서버들은 서버관리기에 의해 매시간 Ping을 통한 대상 웹서버의 시간대별 부하량을 측정하여 그 값을 매번 평균값으로 저장을 하게 된다. 이는 보다 표준화된 판단의 근거를 마련하기 위한 한 방편으로 이는 검색대상 URL의 우선 순위 결정시 로봇의 실시간 부하량과 함께 사용된다.

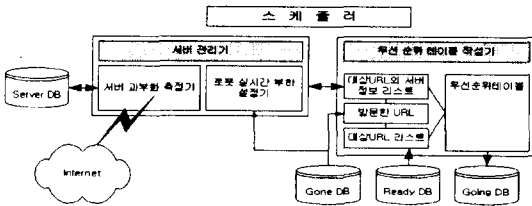
3.3 스케줄러

스케줄러는 서버관리기와 우선순위테이블 작성기를 포함하여 정의되며 서버관리기의 작동과 검색대상 URL들의 검색시점등을 결정, 보다 효율적인 검색작업이 수행되도록 스케줄링한다.



[그림 3] 우선 순위 결정도

스케줄링 방법은 혼합스케줄링방법을 이용 그림 3과 같이 검색대상 URL에 대해 검색대상에서 제외된 나머지 URL에 대해 서버관리가 구성해 놓은 시간대 평균 부하량 (웹서버Ping결과 + 로봇의 실시간에 체험한 부하량) 과 실시간 로봇의 체험 웹서버상황을 기준으로 검색대상 URL의 우선순위테이블을 작성한다. 스케줄러의 구성도는 그림 4와 같다.



[그림 4] 스케줄러 구성도

3.4 로봇관리기

스케줄러에 의해 생성된 우선순위테이블을 기준으로 최상위 URL부터 n개의 로봇을 생성하여 동시에 특정시간에 동시에 문서수집을 수행한다.

로봇의 개수 및 동작시간은 RMI를 이용한 원격관리 모듈에 의해 설정하며, 실시간에 관리자가 설정을 변경하여 네트워크의 과도한 부하를 방지할 수 있다.

3.5 문서분석기

로봇이 대상 URL에서 가져온 문서를 분석하여 다른곳으로 Link되어 있는 대상 URL을 추출하여 스케줄러에게 넘겨준다. 이때 검색대상 웹서버의 Depth 설정으로 무한 채귀 루프를 돌지 않도록 규제하며 검색대상 웹서버를 벗어나는 URL에 대해서는 추출시 삭제시켜 버린다. 단, 삭제 작업은 원격관리모듈에 의해 설정 변경가능하게 구현하며 이는 특성을 부가한 웹 검색을 가능하게 할 수 있다.[3]

대상 URL추출 및 HTML TAG제거후 어휘분석기를 통해 색인어와 불용어에 대한 분류와 인덱스DB를

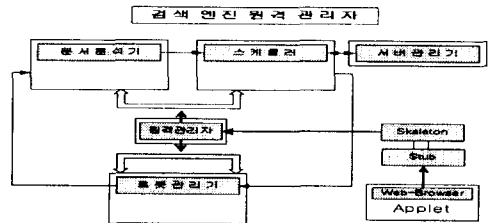
생성한다.[4]

3.6 질의서버 와 등록서버

질의서버는 일반사용자가 웹 브라우저를 통해 특정 용어에 대한 검색질의시 인덱스 DB 및 디렉토리 DB에 미리 분류 저장되어 있는 자료를 일반사용자에게 보여준다. 등록서버는 디렉토리DB생성을 담당하며 자신의 홈페이지를 홍보하고자 하는 일반사용자들의 등록품을 생성 필요한 정보를 얻어 이를 이용 디렉토리DB를 생성한다. 생성된 디렉토리정보의 사실 유무 확인을 위해 대상 URL은 스케줄러를 통해 n번의 상태확인을 거쳐 확인된 디렉토리에 대한 정보를 관리하고 불량디렉토리에 대해 사용자에게 불량여부를 통보하기위한 정보를 생성한다.

3.7 원격관리

검색시스템의 관리를 위해 시스템 제반설정사항을 RMI서버 모듈로 구성하여 모듈내 변수값의 변경을 통한 실시간으로 변경사항을 적용할 수 있도록 구성한다. 이는 시스템의 재시작이 필요없이 특수한 상황 발생시 제반설정사항을 원격으로 변경할 수 있는 이점을 제공한다. 원격관리모듈은 Applet으로 구성되어 웹 브라우저를 이용한 검색엔진 관리를 가능하게 한다. 그림 5는 원격관리의 제어 흐름을 표현한 것이다.



[그림 5] 원격 관리

자바의 RMI를 이용 관리자가 웹상의 Applet에서 설정값의 변경시 검색시스템내의 설정 모듈의 함수가 호출, 수행되어 검색시스템내의 설정값을 변경, 실시간으로 적용할 수 있다.

RMI를 이용한 원격관리 구현 코드는 다음과 같다.

- 관리자 원격관리 모듈 (Applet)

```
try {
    ControlImpl obj
        = (ControlImpl) Naming.lookup("//ControlServer");
} catch (Exception e) {
    _txtHelp.setText("원격 관리 접속에 실패하였습니다 ... " +
        e.toString());
}
```

- 검색 시스템 원격관리자

```
try {
    // RMI Registry 생성
    LocateRegistry.createRegistry(
        (new Integer(registryPort)).intValue());
    ControlServer pServer = new ControlServer();

    // bind the server to the registry
    Naming.rebind("//" + registryPort +
        "//ControlServer",pServer);
} catch (Exception e) {
    System.out.println("ERROR : " + e.getMessage());
    //e.printStackTrace();
}
```

RMRegistry를 생성하고 원격관리자를 바인딩시킨 후 원격관리 모듈의 접속이 있을때까지 대기한다. 원격관리모듈의 접속시 시스템의 자바보안관리자의 인증을 통해 원격관리모듈은 원격관리자의 내부 함수를 사용할 수 있게 된다. 내부함수의 실행을 통해 검색시스템의 제반설정사항을 변경할 수 있게 된다.

4. 구현 및 평가

본 시스템은 사용자의 질의를 받아 검색 결과를 HTML문서로 사용자에게 보내주는 질의서버, 사용자의 디렉토리입력을 위한 등록서버, 스케줄러, 로봇관리기, 문서분석기, 원격관리모듈로 구성된다.

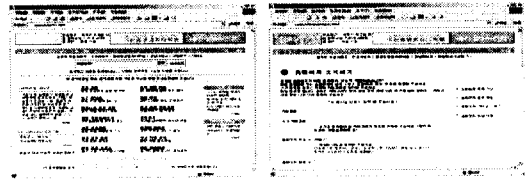
제안된 시스템의 구현환경은 다음 표 1과 같다.

[표 1] 구현 시스템

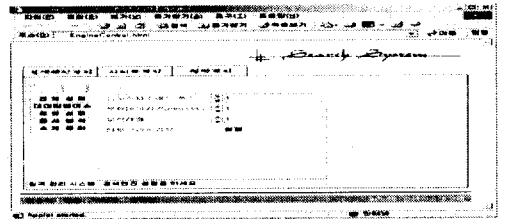
분 류	질의서버 등록서버	스케줄러, 로봇관리기	원격관리모듈
사용OS	알파 LINUX 6.0		
웹서버	아파치1.3.11		
프로세스	펜타엄 166		
개발도구	PHP 3.0.14	JAVA	RMI, Swing
DBMS	MySQL 3.22.22		

PHP언어를 이용하여 HTML을 구성하였다.

PHP언어를 사용한 이유는 타 언어와 달리 서버에 독립적인 프로세스를 생성시키지 않으므로 서버에 대한 자원부담을 줄여주고 속도가 빠른 것으로 평가되고 있기 때문이다.



[그림 6] 사용자 검색화면 및 사용자 디렉토리의 등록화면
원격관리 모듈은 JAVA Applet으로 구현하여 웹문서에 포함시켰으며, Swing을 이용한 인터페이스 구현으로 관리자가 익숙하고 간편한 조작으로 검색엔진시스템의 제반사항을 관리할 수 있다.



[그림 7] 원격 관리 구현 화면

5. 결론

기존의 혼합스케줄링기법은 특정 시간에 측정된 서버의 부하량을 기준으로 스케줄링이 이루어 짐으로서 효율적이지 못한 스케줄링이 이루어지는 단점이 있다.

이를 보완하기 위해 본 논문에서는 특정 주기를 정하여 서버의 부하량을 측정하여 매회 결과 부하량의 평균값과 실제 로봇이 대상 웹서버에 접속시 체험한 실시간 정보를 기준으로 스케줄링함으로써 보다 효율적인 문서수집이 가능하게 하였다.

제안된 검색시스템은 특정한 상황발생시 또는 특정 설정변경시 관리자의 실시간 관리 및 변경이 가능하며, 검색시스템의 재시작이 필요없는 등 능동적인 상황대처가 가능하다. 또한 본 검색시스템은 자바기반으로 개발되었기 때문에 플랫폼의 제약이 없다

앞으로의 연구방향으로 검색시스템과 원격관리모듈간의 원격관리 작업에서의 보안관련 사항에 대한 계속적인 연구가 요구된다.

[참고문헌]

[1] "A Search Robot Engine using Mixed Sorting Scheduling Method", 鄭龍勳, Kyungnam University 碩士學位, 1999, 12.
[2] "Inside Secrets Java 2 Developer's Handbook" pp.648-693, Philip Heller & Simon Roberts,

- [3] "Design and Implementation of Region Administration-Oriented Search Engine Based on Robot Control Policy", 金昌根, Kyungnam University 博士學位, 1999, 6.
- [4] "Java Networking and Communications", pp.43-88 Todd Courtois, 1997.
- [5] "Design and Implementation of Real-time Control Search Engine for Local Information Network", 金 壹, Kyungnam University 碩士學位 1998. 12.