

# 네트워크 및 웹 서버의 부하를 고려한 지능적인 검색시스템의 설계 및 구현 (Intelligent Search Engine Considering the Load of Web Servers and Networks)

김 창 근\*      탁 한 호\*\*  
(Chang-Geun Kim) (Han-Ho Tack)

## 요 약

기존의 검색엔진은 분산된 불특정 다수의 웹 서버에 대해 웹 서버 부하, 네트워크 부하를 고려하지 않고 로봇을 동작시키므로 웹서버 및 네트워크의 과부하를 초래한다. 또한 정보의 갱신 기간도 3~4주 간격으로 필요한 시기에 정보가 갱신되지 않는 경우가 많으며, 웹서버의 내용 변경이 없음에도 불구하고 무조건 웹서버의 내용을 수집하여 갱신한다. 본 논문에서는 네트워크 및 웹서버 부하를 고려한 지능적인 검색시스템을 제안하고, 이 시스템에 필요한 실시간 제어 검색엔진 및 분산되어 있는 정보를 모아서 갱신해 주는 지능적인 로봇을 설계 및 구현하였으며, 네트워크 부하 및 웹 서버의 부하를 모니터링하여 전체 시스템의 성능을 향상시킬 수 있는 로봇 제어 정책을 수립하였다.

## ABSTRACT

The existing search engine cause heavy loads on web servers and networks, because they operate robots not considering the loads of many distributed diverse web servers and networks. As the information update period is 3-4 weeks, information is not updated at proper time. Although there is no contents to be updated on web servers, the robot collects and updates the contents of the web servers. This study proposes a intelligent search engine considering the load of web servers and networks, designs a real-time controlled search engine and intelligent robots, and implements them. The intelligent robot updates distributed information real-time. Robot control policies are also proposed, which can upgrade the overall performance of the system by monitoring the loads of web servers and networks.

## 1. 서론

최근 인터넷의 보급이 급속히 확산되면서 많은 사용자들이 인터넷을 통해 유용한 정보를 손쉽게 접할 수 있게 됨에 따라 정보검색의 문제는 관련 정보 획득의 어려움뿐만 아니라 정보의 홍수라 표현될

정도로 엄청난 양의 정보들 중 어떻게 하면 원하는 정보를 정확하고 신속하게 검색해 낼 것인가에 있다고 볼 수 있다.

---

\* 정희원 : 진주산업대학교 컴퓨터공학과 부교수

\*\* 정희원 : 진주산업대학교 전자공학과 부교수

논문접수 : 2001. 7. 30.

심사완료 : 2001. 8. 13.

기술의 발달로 인해서 사용자들은 좀 더 편리하게 인터넷을 통해서 정보 얻기를 원하게 되었으며, 이러한 요구 때문에 생겨난 것이 주문형 서비스이고 사람을 대신해서 정보를 가져오는 것이 로봇 에이전트이다[1].

특히, 정보의 검색에 시간의 제약이 있는 경우 사용자는 필요한 정보를 적정시간 내에 정보를 얻어야만 한다. 시간적 제약을 가진 정보에 대한 서비스를 제공하려면 분산된 정보를 실시간으로 검색하고 검색시스템의 정보를 갱신하여 실시간으로 서비스해주는 효율적인 검색시스템이 필요하다.

현재 개발되어 사용되는 대부분의 검색시스템은 많은 문제점을 내포하고 있다.

첫째, 웹 로봇이 다녀가는 주기가 길다.

둘째, 특정 시간대에 집중된 통신, 대용량의 데이터 송수신 등으로 인한 네트워크의 과부하를 초래할 수 있다.

셋째, 웹 서버의 과부하를 초래한다.

넷째, 기존의 검색엔진은 웹서버의 내용변경이 없음에도 불구하고 무조건 웹 서버의 데이터를 가지고 가 네트워크와 웹서버의 과부하를 초래하고 있다.

본 논문에서는 네트워크 및 웹서버 부하를 고려한 지능적인 웹 로봇의 모델을 제안하고 설계 및 구현하였다.

본 논문에서 제안한 시스템은 크게 정보를 검색할 수 있는 검색엔진서버와 정보를 제공하는 다수의 웹서버로 구성되며, 검색엔진서버는 크게 질의서버(Query Server)와 서버 집중화 관리자(Server Integration Manager), 자동색인기(Automatic Index Builder), RCRA 관리자, 모니터링 에이전트(Monitoring Agent) 등으로 구성되어 서비스를 제공한다.

## 2. 지능적인 로봇의 활용

인터넷이 처음 등장할 당시에는 텍스트 위주의 정보교환이나 의사전달과 같은 커뮤니케이션 기능이 주된 역할이었지만 정보통신기술의 급격한 발전과 웹의 등장으로 오늘날에는 각종 멀티미디어 서비스와 전자상거래가 가능해졌다. 특히 웹서비스는 인터넷 이용을 확산시키는데 있어서 결정적인 기여를 하였으며, 그 결과 전세계적으로 성공적인 웹 서버를

구축하려는 많은 노력들이 진행 중에 있다[4,5].

이용자들이 원하는 정보를 찾고자 할 때 가장 많이 사용하는 것이 정보검색시스템이다. 아무리 잘 만들어 놓은 사이트라도 이용자들이 그 사이트를 찾는 데 많은 노력과 시간을 투자해야 한다면 아무런 의미가 없다. 인터넷 웹서비스가 일반 이용자들에게 널리 확산되는데 기여한 것도 해당 정보에 대한 접근이 용이해졌기 때문이다.

정보검색시스템의 구성은 검색엔진과 웹 로봇으로 구성되는데 검색엔진은 웹에 대한 정보가 데이터베이스화되어 있을 때, 이 데이터베이스로부터 검색하고자 하는 내용을 질의어로 받아 관련된 내용을 검색하여 보여주는 것이고, 웹 로봇은 웹 상에 산재되어 있는 각 문서들의 URL 위치와 내용을 수집해 오는 역할을 한다[2,3,7,8,10].

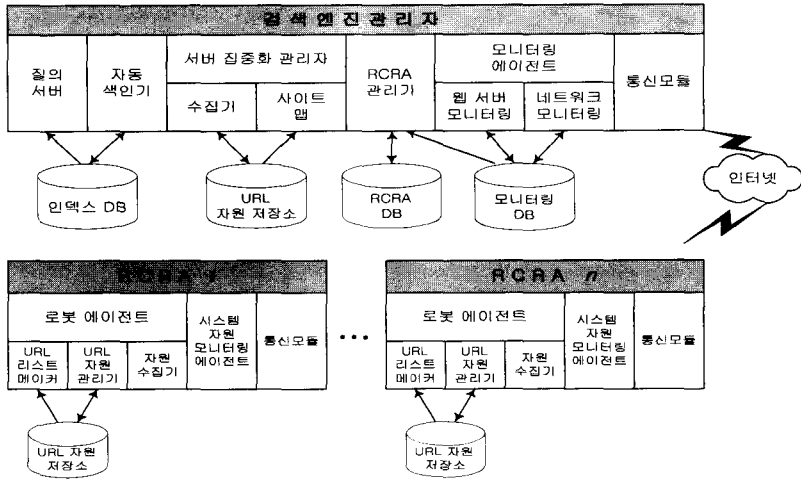
이런 정보검색시스템은 하나의 데이터베이스에 모든 홈페이지들의 URL을 저장하고 특정 키워드를 입력함으로써 원하는 정보를 찾는 단어별 검색엔진, 각 검색엔진들에게 병렬적으로 질의를 보내어 그 결과를 종합해서 이용자에게 보고해 주는 메타 검색엔진, 한번의 검색시간으로 여러 곳을 검색해 주는 통합 검색엔진으로 나뉜다[6,9].

## 3. 네트워크 및 웹서버 부하를 고려한 검색시스템의 설계

### 3.1 시스템 구성

네트워크 및 웹서버 부하를 고려한 검색시스템의 구성도는 [그림 1]과 같다.

[그림 1]에서 네트워크 및 웹서버 부하를 고려한 검색시스템은 정보를 검색할 수 있는 검색엔진서버와 정보를 제공하는 다수의 웹서버로 구성되며 검색엔진서버는 크게 질의서버와 서버 집중화 관리자, 자동색인기, RCRA 관리자, 모니터링 에이전트 등으로 구성되어 서비스를 제공한다.



[그림 1] 네트워크 및 웹서버 부하를 고려한 검색시스템의 구성도  
 [Fig. 1] Search system's structural diagram considering the load of web server and network

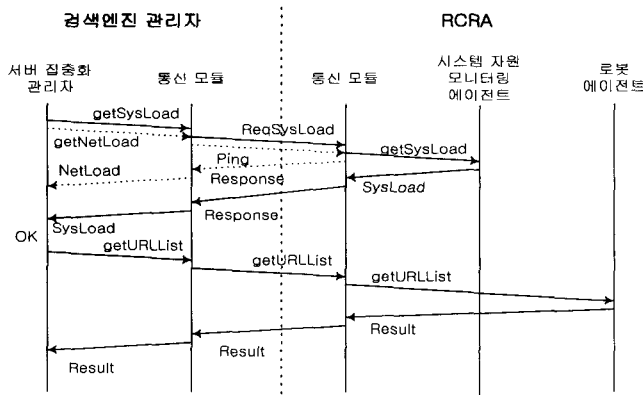
### 3.2 RCRA(Remote Control Robot Agent) 프로토콜

검색엔진은 RCRA로부터 URL 리스트에 대한 정보를 획득하기 위해 RCRA에 접속하여 필요한 정보를 가져오게 된다.

검색엔진의 통신모듈은 TCP/IP를 통해 서버측 통신모듈과 연결(Session 설정)하게 된다. 세션이 설정

되면 검색엔진의 통신모듈은 서버측과의 네트워크 부하 및 서버의 부하를 데이터베이스에 기록하고 차후에 접속할 시간을 결정하는데 사용된다.

[그림 2]는 검색엔진관리자와 RCRA 간의 상호동작 프로토콜을 나타낸 것으로 모니터링 에이전트는 네트워크 부하 및 웹 서버의 부하를 모니터링하고 부하가 정상적일 경우 검색엔진은 서버로부터 필요한 정보를 요구하게 되며, 서버는 그 요구에 대한



[그림 2] 검색엔진관리자와 RCRA 간의 상호 동작 프로토콜  
 [Fig. 2] Mutual protocol between the search engine manage and the RCRA

정보를 전송하게 된다. 웹 서버가 과부하 상태일 경우에는 웹 서버의 부하 상태를 기록하고 다음 접속할 시간을 산출한다.

### 3.3 검색엔진관리자

#### 3.3.1 RCRA 관리기

RCRA 관리기는 웹 서버에서 전송된 RCRA 등록 정보를 이용하여 RCRA를 등록·관리하며, 모니터링 에이전트에서 기록한 각 웹 서버의 부하상태 및 네트워크의 부하상태를 이용하여 다음 RCRA에 접속할 시간을 산출하는데 사용된다.

#### 3.3.2 모니터링 에이전트

모니터링 에이전트는 웹 서버의 부하상태나 네트워크의 부하상태를 모니터링하고 그 결과를 데이터베이스에 주기적으로 기록하여 웹 서버의 부하 상태 및 네트워크의 부하 상태를 확인할 수 있도록 한다.

네트워크의 부하가 적은 시간대에 웹 서버에 접속하여 네트워크에 가중되는 부하를 줄이고 웹 서버의 부하가 적은 시간대에 접속하여 웹 서버의 부하를 줄임과 동시에 전체적인 수행 속도를 향상시킨다.

#### 3.3.3 통신 모듈

TCP/IP를 이용하여 각 서버의 정보를 얻기 위해 요구사항 정보를 서버로 보내고 웹 서버로부터 해당 요구사항의 결과를 받는다.

검색엔진의 모니터링 에이전트가 통신 모듈을 통해 웹 서버의 부하상태를 요구하게 되면 RCRA에게 그 요구사항을 전달하며, RCRA는 부하 상태를 검색엔진의 모니터링 에이전트에게 전달한다.

URL 리스트의 요청은 다음과 같이 수행된다. 검색엔진의 서버 집중화 관리자가 URL 리스트를 요청하게 되면 이 요청을 RCRA에게 보낸다.

URL 리스트의 요청을 받은 RCRA는 요청한 리스트를 파악하여 로봇 에이전트에게 해당 요구사항을 보낸다. 요구 사항을 받은 로봇 에이전트는 URL 리스트 메이커로 명령을 보내고 URL 리스트 메이커는 URL 자원 저장소로부터 결과를 추출하여 로봇

에이전트에 보내며, RCRA는 그 결과를 검색엔진에게 돌려준다.

#### 3.3.4 서버 집중화 관리자

서버 집중화 관리자는 검색엔진에 등록된 다수의 RCRA에게 각 서버의 URL 리스트나 그 외 필요한 정보를 요청하게 된다.

서버 집중화 관리자는 RCRA로부터 반환되는 결과값으로 각각의 RCRA를 관리하게 되는데, 서버 집중화 관리자가 RCRA에게 필요한 정보를 요청하는 방법에는 각 RCRA에 대하여 병렬적으로 동시에 요청을 하고 그 결과를 받는다.

웹 서버에 대한 요구 사항의 결과가 수신 완료되면 현재 데이터베이스의 내용과 비교하여 변경된 사항을 갱신한다.

### 3.4 RCRA

RCRA의 시스템 자원 모니터링 에이전트는 주기적으로 시스템 자원의 상황을 측정하며 로봇이 작동할 시간이 되면 시스템 부하를 검사하고 로봇의 작동가능 여부를 RCRA에게 전달한다.

RCRA가 로봇 에이전트에게 URL 정보를 수집하도록 지시하면 자원 수집기는 HTML 문서를 수집하고 URL 정보를 추출하여 URL 자원 관리기에 보내며, URL 자원 관리기는 URL 자원 저장소에 신규 작성, 변경, 삭제 등의 URL 정보를 저장 관리하며, URL 리스트 메이커는 검색엔진관리자로부터 요구가 들어오면 URL 자원 저장소에서 필요한 정보를 추출하여 검색엔진관리자에게 전송한다.

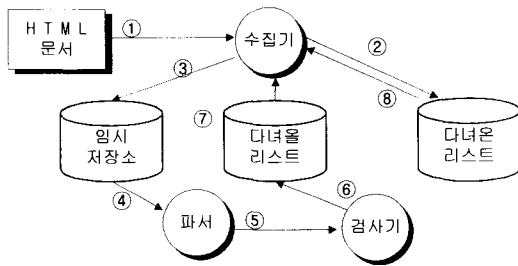
#### 3.4.1 로봇 에이전트

로봇 에이전트는 자원 수집기에서 웹 서버에 있는 URL 정보를 수집하고 URL 자원 관리기는 수집된 URL 정보를 URL 자원 저장소에 등록하며, 등록된 URL 정보는 URL 리스트 메이커에 의해 검색엔진관리자가 요구하는 형태로 가공되어 검색엔진으로 보내어진다.

자원 수집기는 수집기, 파서, 검사기로 구성되며, 자원 수집기가 HTML 문서를 수집하여 URL 리스트

를 추출하고 다녀온 리스트를 작성하는 알고리즘은 [그림 3]과 같다.

URL 자원 관리기는 자원 수집기가 수집해 온 URL 리스트에 대한 정보를 관리하는 모듈로서 URL 정보를 URL 자원 저장소에 등록 관리하는 기능을 담당한다.



[그림 3] 수집기의 수집 알고리즘

[Fig. 3] Resource collector's collection algorithm

URL 리스트 메이커는 검색엔진의 요구에 의해 필요한 정보를 URL 자원 저장소에서 추출하여 통신 모듈에게 전송하고 전송된 정보는 검색엔진의 서버 집중화 관리자에게 전달된다.

### 3.4.2 시스템 자원 모니터링 에이전트

시스템 자원 모니터링 에이전트는 웹 서버의 시스템 자원 상태를 주기적으로 모니터링하고 로봇의 가

동 시간이 되면 로봇의 가동가능 여부를 RCRA에게 전달한다. 또한 주기적으로 서버의 시스템 부하 상태를 모니터링하여 로봇의 가동 가능 시간을 결정한다.

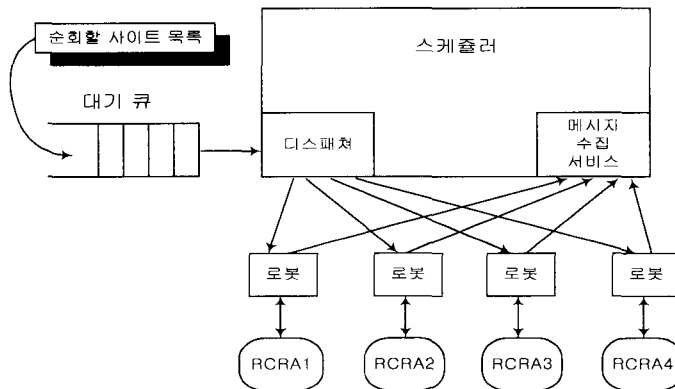
### 3.4.3 통신모듈

검색엔진관리자로부터 요구가 들어오면 메시지 이벤트 핸들링 큐가 해당 요구를 분석하여 필요한 행동을 취한 후 해당 요구사항에 맞는 결과를 전송한다. 이 모듈은 시스템 모니터링 에이전트, 로봇 에이전트와 상호 통신하여 필요한 데이터를 수집하여 검색엔진관리자에게 결과를 되돌려준다.

## 4. 네트워크 및 웹서버 부하를 고려하는 로봇 제어 정책

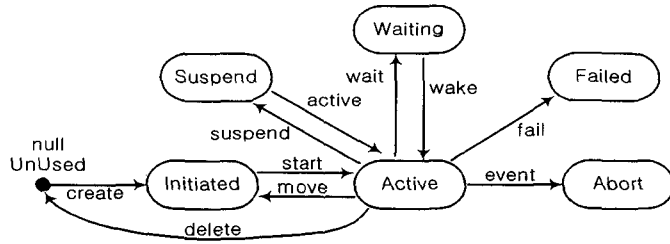
[그림 4]에서 RCRA 관리기는 네트워크 부하를 모니터링한 자료를 근거로 하여 주간별, 월별, 시간별 최우선 사이트를 선정하여 순회할 사이트 목록을 만들고 대기 큐에 삽입하게 된다.

스케줄러의 디스패처는 대기 큐에 삽입된 순회할 사이트를 가져와서 로봇을 생성 및 제어할 수 있는 기능을 가지며, 각 로봇에 대한 연결 상태를 주기적으로 체크하며 미해결 사이트를 등록하여 최적시간에 다시 방문하도록 재스케줄링한다.



[그림 4] 로봇 제어

[Fig.4] Robot control



[그림 5] 로봇의 생명주기

[Fig. 5] Each robot's life cycle

생성된 로봇은 RCRA와의 세션을 관리하게 되고 생성된 로봇의 생명주기는 [그림 5]와 같다.

create는 새로운 태스크를 할당하여 실행되는 프로세스 단계를 말하며, start는 로봇 에이전트가 서비스 제공을 위해 자신의 서비스 동작을 시작하는 과정이며 suspend는 로봇의 실행이 자체 원인, 상대방 RCRA에 의해 잠시 멈추는 과정이다.

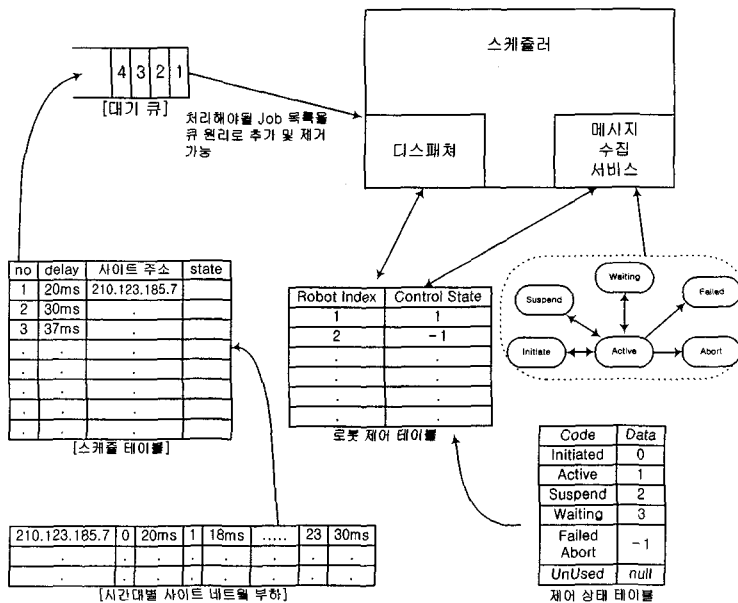
active는 suspend 상태에서 동작을 시작하는 과정이며, wait는 특정 이벤트를 위해 로봇의 상태를 Waiting 상태로 변환하는 과정이다. wake는 Waiting 상태에서부터 로봇이 활동을 재기하는 것으로 로봇 자체에 의해서가 아니라 RCRA에 의해 변환되는 과정

이며, delete는 로봇의 실행을 정지시키고 RCRA가 연결을 해제하는 과정이다.

Active 상태에서 Failed나 Abort 상태로 이전되면 관련 사이트의 연결상태가 좋지 않아 3번의 retry로 연결제어를 결정하게 된다.

[그림 6]은 로봇의 제어 상태를 나타내며, RCRA 관리기는 시간대별 사이트 네트워크 부하를 기초로 하여 스케줄 테이블을 작성한다.

작성된 스케줄 테이블에서 검색할 시간에 순회할 사이트 목록을 대기 큐에 삽입하게 된다. 이 때에 대기 큐는 환경대기 큐를 사용한다.



[그림 6] 로봇 제어 상태도

[Fig. 6] Robot control diagram

대기 큐에 삽입된 사이트는 처리해야할 Job의 목록으로 스케줄러의 디스패처는 대기 큐에 삽입된 순회할 사이트를 가져와서 로봇을 생성하게 되면 대기 큐에서 제거되고 로봇 제어 테이블에 등록되게 된다. 로봇 수행시 로봇의 상태 정보가 로봇 제어 테이블의 상태 정보에 기록되어 로봇을 관리하게 되며, [그림 7]은 로봇의 수행 알고리즘을 나타낸다.

로봇의 수행 순서는 로봇 제어 테이블에서 Initiated된 로봇을 찾아서 해당 사이트와 Ping 명령을 수행하여 사이트 간의 지연시간을 구하고 등록된 네트워크 부하와 비교를 하여 오차가 10% 이내이면 RCRA와 연결을 설정하게 되고 10% 넘어서게 되면 로봇을 Abort하여 로봇 제어 테이블에서 삭제시키고 스케줄 테이블의 상태를 retry (-2)로 설정하여 재스케줄링이 되게 한다.

로봇과 RCRA간에 세션이 설정되면 로봇은 RCRA로부터 웹 서버의 상태정보를 얻고, 웹 서버의 CPU 부하가 80% 이하가 되면 RCRA로부터 갱신된 정보가 있는지 검사하게 된다. 그러나 웹 서버의 CPU 부하가 80% 이상이 되면, 생성된 로봇을 제거하기 위해 로봇 제어 테이블에 null을 할당하여 로봇을 제거하고 스케줄 테이블에 retry (-2)를 설정하여 재스케줄링이 되게 한다.

있으면 로봇이 실행되어 실제적인 인덱스 작업이 이루어지게 되고, 만약 갱신된 정보가 없다면 로봇 제어 테이블에 null을 할당하여 로봇을 제거하게 된다.

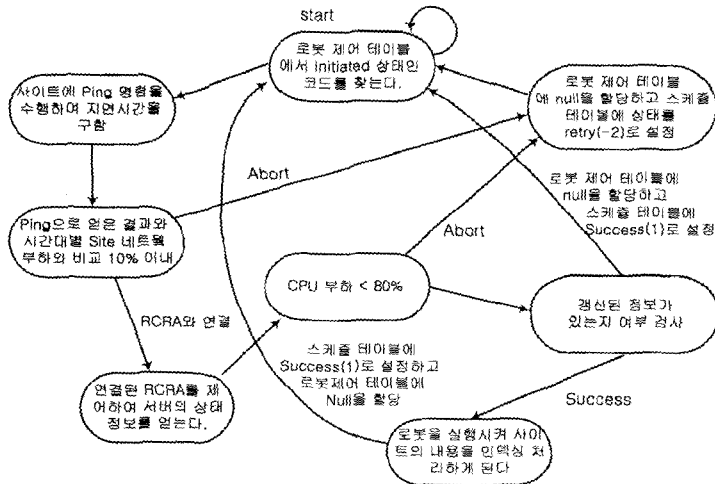
### 5. 구현 및 평가

본 논문에서 제안한 네트워크 및 웹서버 부하를 고려하는 지능적인 검색시스템을 지역정보화를 위한 검색시스템에 적용하여 구현하였다.

제안 시스템의 모듈별 데이터 흐름도는 [그림 8]과 같다.

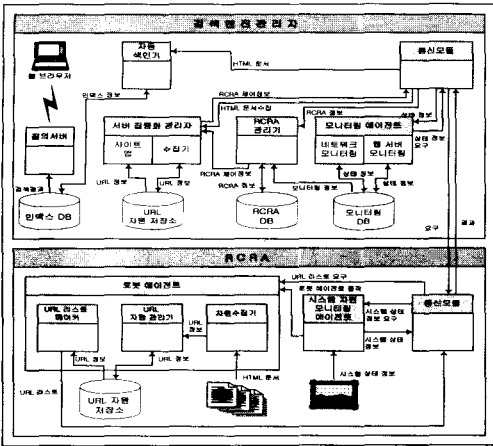
먼저, 검색엔진서버로부터 RCRA를 다운로드 받아 웹 서버에 설치한다. RCRA가 설치되면 RCRA는 설치완료정보와 등록정보를 검색엔진 관리자에게 전송하게 되고 검색엔진관리자는 RCRA를 등록한다.

웹 서버 CPU 부하가 80% 이하이고 갱신된 정보가



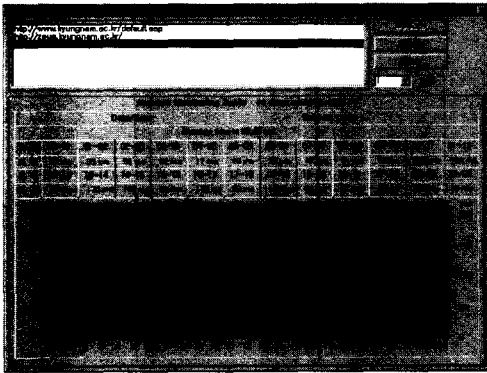
[그림 7] 로봇 수행 알고리즘

[Fig. 7] Robot performance algorithm



[그림 8] 모듈별 데이터 흐름도

[Fig. 8] Data flow chart of modules in the proposed system



[그림 9] 웹 서버에 대한 시간대별 네트워크 부하모니터링 화면

[Fig. 9] Graphic representation of monitoring hourly network loads against web servers

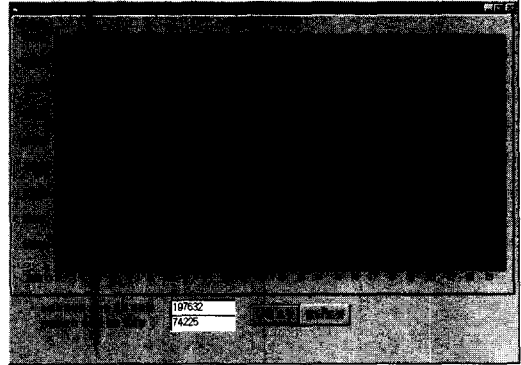
RCRA가 등록되면, 검색엔진관리자는 새로 설치된 RCRA 간의 네트워크 부하상태 및 웹 서버 부하 상태에 대한 모니터링을 실시한다.

모니터링한 데이터를 기초로 로봇 제어를 통해 정보를 수집하여 서비스를 실시한다.

모니터링 에이전트는 웹 서버의 부하상태와 네트워크의 부하상태를 모니터링하고 모니터링한 결과를 데이터베이스에 주기적으로 기록하며, 모니터링 상황을 관리자가 확인할 수 있도록 [그림 9]와 같이 그

래프로 나타낼 수 있다.

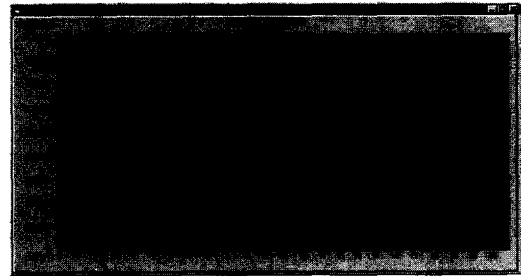
RCRA 간의 네트워크 부하상태 및 RCRA가 실행된 웹 서버의 부하를 그래프로 나타내주고 있으며 내부적으로 처리되는 모니터링을 비주얼하게 확인할 수 있다.



[그림 10] 네트워크 및 웹서버 부하를 고려한 경우의 비교(단일 로봇과 멀티 로봇)

[Fig. 10] Processing comparison based on robot control policies(a single robot vs. multiple robot's)

[그림 10]은 네트워크 및 웹서버 부하를 고려하되 단일 로봇과 멀티로봇을 비교한 그림으로 단일 로봇 보다는 멀티 로봇이 전체적인 수행측면에서 많은 시간단축을 가져옴을 알 수 있다.



[그림 11] 네트워크 및 웹서버 부하를 고려한 처리량 비교

[Fig. 11] Processing comparison between some and no robot control policies



[그림 11]은 네트워크 및 웹서버 부하를 고려한 경우와 그렇지 않은 경우를 비교하여 그래프로 나타내 주고 있다. 네트워크 및 웹서버 부하를 고려하지 않았을 경우는 시간당 처리율이 거의 일정하지만, 네트워크 및 웹서버 부하를 고려하면 네트워크 부하가 적은 시간대와 웹 서버의 부하가 적은 시간대에 시간당 처리율이 비교적 높음을 알 수가 있고, 또한 전체적인 수행완료시간을 단축할 수 있음을 알 수 있었다.

No	Delay	Site	출발	stop			
31	17	158	108	159	42	2	
44	17	158	173	108	159	28	2
12	15	230	144	176	190	29	2
46	20	150	110	131	180	26	1
27	21	230	170	129	208	44	2
15	21	158	151	111	164	41	0
91	33	102	208	142	131	41	0
53	34	144	224	173	239	46	0
17	34	240	206	185	118	38	0
30	35	160	135	137	164	23	0
14	5	132	147	164	207	9	0
5	37	222	208	209	181	44	0
21	38	152	116	117	217	27	0
82	38	216	170	207	213	39	0
96	40	220	164	220	238	13	0
30	41	145	198	114	122	45	0
11	44	180	135	218	153	45	0
55	46	170	105	191	216	31	0
47	47	160	157	230	130	17	0
47	48	155	136	230	120	17	0
41	49	168	118	150	157	8	0

[그림 12] 스케줄링 테이블  
[Fig. 12] Scheduling table

[그림 12]는 시간대별 웹 서버 간 네트워크 부하에 의해 추출된 스케줄링 테이블로서 순회할 사이트 목록과 사이트 정보를 나타낸다.

본 논문에서 제안한 시스템은 기존 정보검색시스템의 로봇이 웹 서버의 부하와 네트워크의 부하를 고려하지 않고 웹 서버에 접속하여 HTML 문서를 수집하는 방법과 달리 웹 서버의 부하와 네트워크의 부하를 고려하기 위해 로봇 제어 정책을 사용하였다. RCRA 관리는 네트워크 부하 및 웹 서버 부하의 모니터링 정보를 이용하여 로봇 제어 정책을 수립하고 제어함으로써 네트워크에 가중된 부하 및 웹 서버의 접속 시간을 조절하여 전체 시스템의 성능을 향상시킬 수 있었다.

## 6. 결론

분산된 불특정 다수의 웹 서버에 대해 웹 서버 부하, 네트워크 부하를 고려하지 않고 로봇을 동작

시키며, 3~4주 간격으로 정보를 갱신하는 기존의 검색엔진과는 달리, 본 논문에서는 네트워크 부하 및 웹서버 부하를 고려한 지능적인 로봇을 설계 및 구현하였으며, 네트워크 부하 및 웹 서버의 부하를 모니터링하여 전체 시스템의 성능을 향상시킬 수 있는 로봇 제어 정책을 수립하였다.

본 논문에서 제안한 검색시스템은 검색시스템의 성능향상을 위하여 웹서버에 대문으로 있는 RCRA를 검색시스템에서 원격제어 함으로써 네트워크의 부하와 검색시스템 및 웹서버 부하를 줄였다. 또한 영역의 특성화를 위하여 2~3일 정도의 짧은 기간에 모든 정보를 최신의 정보로 갱신하며, 주기적으로 네트워크의 부하를 모니터링하고 그 모니터링한 데이터를 기반으로 웹 서버의 RCRA에게 접속을 시도 함으로서 네트워크 부하를 줄일 수 있다.

또한, 웹 서버의 부하를 모니터링하여 웹서버의 부하가 80% 이상인 경우 네트워크 부하가 적은 상황이라도 접속을 하지 않고 다음에 재 접속하는 정책을 사용함으로써 웹 서버 부하를 고려하였다. 또한 본 논문에서 구현한 지능적인 검색시스템은 웹 서버에 변경된 HTML 문서가 없어도 HTML 문서를 수집하는 기존의 검색엔진과 달리 변경된 데이터가 있을 경우에만 웹 서버의 HTML 문서를 수집함으로써 네트워크 부하 및 웹 서버 부하를 고려했다. 또한 네트워크 및 웹서버 부하를 고려하는 로봇 제어 정책을 이용하지 않았을 경우 시간당 평균 처리율이 거의 일정한 결과를 보인 반면, 네트워크 및 웹서버 부하를 고려한 로봇 제어 정책을 이용할 경우에는 네트워크 부하가 적은 시간대를 우선하여 처리함으로써 초반기 시간대에는 상대적으로 처리율이 높았으며 시간이 지날수록 처리율이 떨어져 나중에는 네트워크 및 웹서버 부하를 고려하지 않았을 경우와 별다른 차이가 없었다. 그렇지만 전체적인 수행 측면에서는 네트워크 및 웹서버 부하를 고려하지 않았을 경우 보다 네트워크 및 웹서버 부하를 고려할 경우가 시간단축을 가져왔으며 단일 로봇보다는 멀티 로봇의 경우가 더 많은 시간 단축을 가져왔다.

앞으로의 연구 내용은 웹서버 모니터링 결과실시간 갱신에 대한 연구와 플랫폼에 독립적인 RCRA가 되도록 하는 지속적인 연구가 요구된다.

※ 참고문헌

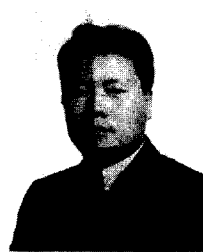
- [1] 김 창근, “로봇 제어 정책을 기반으로한 지역 관리지향 검색시스템의 설계 및 구현”, 경남대학교 박사학위 논문, 1999.8.
- [2] 여 찬기, 송 관호, 유 용석, “한국 인터넷 디렉토리 시스템 최종개발완료보고서”, 한국전산원, 1997
- [3] 심 해청, 김 병만, 김 태남, “효율적인 웹 로봇의 설계 및 구현에 관한 연구”, 1997, 한국정보과학회 가을
- [4] 박 민우, “대화형 인덱스 로봇 에이전트의 설계 및 구현”, 계명대학교 석사학위 논문, 1997.2.
- [5] 김 일, “지역정보망을 위한 실시간 제어 검색 엔진의 설계 및 구현”, 경남대학교 석사학위 논문, 1999.2.
- [6] 박 영몽, 김 민구, 이 정태, “지식 기반의 정보 검색 시스템”, 한국정보과학회 논문지, pp.2090-2098. 1994.11, 제 21권 제 11호.
- [7] Martijn Koster, ConneXions, “Robots in the Web: threat or treat?”, Volume 9, No. 4, April 1995
- [8] <http://info.webcrawler.com/mak/projects/robots/robots.html>
- [9] <http://astro.chungbuk.ac.kr/~kimhy /inf/sear.htm>
- [10] Martijn Koster, “WWW Robots, Wanderers and Spiders”, <http://info.webcrawler.com/mak/projects/robots/robots.html>

김 창 근



1985년: 경상대학교  
전산통계학과(이학사)  
1991년: 경남대학교  
컴퓨터공학과(공학석사)  
1999년: 경남대학교  
컴퓨터공학과(공학박사)  
1995년 - 현재 :  
진주산업대학교  
컴퓨터공학과 부교수  
주관심분야: 멀티미디어 통신,  
정보통신 시스템,  
컴퓨터 네트워크 등

탁 한 호



1987년 : 부경대학교  
전자공학과(공학사)  
1992년 : 동아대학교 대학원  
전자공학과(공학석사)  
1998년 : 한국해양대학교  
대학원 전자통신공학과  
(공학박사)  
1987년 1월 ~ 1989년 2월  
(주)홍창 부설연구소 연구원  
1989년 ~ 현재 :  
진주산업대학교  
전자공학과 부교수  
주관심분야 :  
멀티미디어시스템,  
퍼지-신경망시스템,  
로봇틱스, 트랜스포메이션 등