

# 분산 이형 환경에서의 이동에이전트를 이용한 정보 검색 시스템

(Information Retrieval System based on Mobile Agents in  
Distributed and Heterogeneous Environment)

박재복<sup>†</sup> 이광용<sup>\*\*</sup> 조근식<sup>\*\*\*</sup>  
(Jae-Bok Park) (Kwang-Young Lee) (Geun-Sik Jo)

**요약** 분산 이형 환경에서 대용량의 정보화 검색하는 새로운 패러다임으로 이동 에이전트가 주목받고 있다.

검색을 수행하기 위해 많은 양의 데이터를 전송하는 대신 검색을 수행하는 에이전트를 검색을 수행할 서버로 직접 전달하는 방식을 이용하는 것이다. 본 논문에서는 분산 이형 환경에 독립적으로 존재하는 정보를 효율적으로 검색할 수 있는 이동 에이전트 기반의 정보 검색 모델을 제시하고 질의응답(Q&A)을 검색하는 시스템을 디자인하고 구현한다. 제안된 모델인 이동 에이전트 기반의 질의응답 검색 시스템(QASSMA : Q&A Search System using Mobile Agents)은 이형 분산의 환경에 존재하는 질의응답 게시판 및 뉴스그룹을 검색할 수 있는 모델이다. QASSMA의 특징을 요약하면, 우선, 최적의 검색 위치로 검색 코드 자체가 이동하여 검색을 수행하므로 기존의 정적인 검색 로봇에 비하여 고정된 네트워크 거리를 극복할 수 있고 단축된 원격 서버와의 네트워크 거리만큼 검색시간을 단축시킬 수 있다. 또한 출발서버에서 검색할 위치로 검색코드가 이동하여 실행되므로 중앙 집중적인 네트워크 트래픽 형성을 막고 출발서버의 부담을 줄일 수 있다. 마지막으로, QASSMA의 이동 검색 에이전트는 검색 환경에 맞는 클래스를 능동적으로 탑재하여 효율적으로 검색 환경의 변화에 대처할 수 있고 다양한 검색 방법을 지원할 수 있다. 본 논문에서는 실험을 통해 제안된 이동 에이전트 기반의 질의응답 시스템의 검색 방법이 기존의 정적인 검색 방법에 비해 효율적임을 보인다.

**키워드** : 이동 에이전트, 질의응답 시스템, 분산 정보 검색, 분산처리, Aglet

**Abstract** We focus on the mobile agents which are considered as new paradigm to solve information retrieval of large volumes of data in the distributed and heterogeneous environment. The mobile agent moves the computation to data instead of large volumes of data to computations. In this paper, we propose an information retrieval model, which can effectively search data in the distributed and heterogeneous environment, using mobile agents. Our model is applied to the design and implementation of an Q&A(Question and Answer) retrieval system. Our Q&A retrieval system, called QASSMA(Q&A Search System using Mobile Agents), uses mobile agents to retrieve articles from Q&A boards and newsgroups that exist in the heterogeneous and distributed environment. QASSMA has the following features and advantages. First, the mobile retrieval agent moves to the destination server to retrieve articles to reduce the retrieval time by eliminating data traffics from the server to the client host. Also it can reduce the traffic that was occurred in the centralized network system, and reduce the usage of resources by sending its agent and running in the destination host. Finally, the mobile retrieval agent of QASSMA can add and update dynamically the class file according to its retrieval environment, and support other retrieval manner. In this paper, we have shown that our Q&A retrieval system using mobile agents is more efficient than the retrieval system using static agents by our experiments.

**Key words** : Mobile Agent, Q&A System, Distributed Information Retrieval, Distributed computing, Aglet

<sup>†</sup> 비회원 : (주)소프트웨어벨리 개발팀장

jbpark@eslab.inha.ac.kr

<sup>\*\*</sup> 비회원 : 인하대학교 전자계산공학과

antufire@eslab.inha.ac.kr

<sup>\*\*\*</sup> 중신회원 : 인하대학교 전자계산공학과 교수

gsjo@inha.ac.kr

논문접수 : 2000년 12월 26일

실사완료 : 2001년 10월 15일

### 1. 서론

현재 인터넷상에는 수많은 사이트가 존재하고 정보의 양은 급격히 증가하고 있다. 그러나 많아지는 정보 속에서 원하는 정보를 효율적으로 검색하기가 점점 어려워지고 있다. 이러한 양적 증가 추세에 따라 원하는 정보를 지능적이며 빠르게 얻을 수 있는 정보 검색 에이전트의 필요성이 대두되고 있다. 이렇게 분산된 환경에서의 자료 공유 문제는 그 중요성이 부각되면서 분산된 이형의 플랫폼 환경에 존재하는 정보를 검색하는 문제를 해결하기 위해 많은 시도가 이루어졌다. 최근 급속히 부각되고 있는 이동 에이전트 기반의 정보 검색 모델은 이러한 문제를 해결할 수 있는 효과적인 방안의 하나가 될 수 있다. 이동 에이전트는 자신은 스스로 이동하는 능력뿐만 아니라 상황에 따른 제어할 수 있어 보다 능동적으로 여러 사이트를 이동하며 검색할 수 있다. 또한 자기 복제를 통한 에이전트 생성을 통하여 병렬처리 작업을 수행할 수 있다. 본 논문에서는 이동 에이전트(Mobile Agent)를 이용하여 분산 이형 환경에 존재하는 자료 검색을 위한 효율적인 모델을 제시한다. 먼저 기존의 정적인 검색 모델과의 비교를 통하여 이동 에이전트 기반 검색 모델의 차이점을 보이고 기존 검색 모델이 가지고 있는 문제점을 해결하기 위해 이동 에이전트 기반의 질의응답 검색 시스템(QASSMA)을 제시한다.

인터넷상에는 수많은 사이트의 질의응답(Q&A) 게시판이 존재한다. 이러한 게시판은 독립적으로 운영이 되고 있기 때문에 질문의 중복성과 답변의 효율적 검색을 이루지 못하고 있다. 또한 게시판마다 서로 다른 플랫폼을 이용하는 경우가 많다. 이러한 이형의 분산된 질의응답 게시판의 검색 문제를 해결하기 위하여 자바(Java) 기반의 이동 에이전트를 이용한다. 본 논문에서는 질의응답 자료 검색 방법 중 기존의 서버 중심 모델(Server Oriented Model)과 클라이언트 중심 모델(Client Oriented Model)을 이용한 검색 방법과 이동 검색 에이전트를 이용한 모델의 검색 방법을 서로 비교하여 이동 검색 에이전트의 검색 코드 전송에 따른 검색 시간 감소 정도를 비교한다. 또한 뉴스그룹과 웹 게시판에 있는 HTTP(Hypertext Transfer Protocol)와 NNTP(Network News Transfer Protocol) 프로토콜 형태의 이형의 자료를 데이터베이스로 변환을 위한 로봇 모델 중 기존의 정적인 형태의 로봇 모델과 최적의 검색경로로 이동한 후에 검색하는 이동 로봇 에이전트 모델을 서로 비교하여 이동 로봇 에이전트 모델의 검색시간 단축 정도를 비교한다. 기존의 정적인 로봇 모델에 비해 이동

로봇 에이전트는 좀더 빠른 검색과 각 서버별 부하(Load)를 고려한 위치로 이동이 가능하며 검색된 원본 데이터(Raw Data)를 가공하여 정보를 추출하여 데이터의 양을 줄인 후 이동할 수 있다. 본 논문에서는 이러한 제안된 시스템 모델을 설명하고 기존 검색 시스템과의 차이를 파라미터 모델로 설명한다.

실험에서는 이동 검색 에이전트와 정적인 검색 에이전트를 검색 서버의 수를 늘려가면서 변화되는 검색 요청에 대한 응답 시간을 비교한다. 또한 이동 로봇 에이전트 모델의 검색 시간 차이를 비교하기 위하여 서로 다른 네트워크 환경에 존재하는 이동 에이전트 서버를 마련하여 이동 로봇 에이전트가 이들 서버로 이동한 후의 검색시간의 변화를 조사한다. 실험에 사용된 시스템은 자바(Java) 기반의 이동 에이전트 API를 제공하는 Aglet을 이용하여 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구, 3장에서는 제안된 모델, 4장에서는 실험 및 평가, 마지막으로 5장에서 결론 및 앞으로의 향후 연구에 대해 논한다.

### 2. 관련 연구

#### 2.1 웹 기반 검색 시스템의 실행위치별 분류 모델

##### 2.1.1 서버 중심 모델(Server Oriented Model)

서버 중심 모델은 노하우(Know-how)를 서버가 가지고 있는 모델이다. 즉, Thin Client-Fat Server형의 시스템이다. 서버에 검색엔진, 데이터베이스 등이 위치하고 단순히 클라이언트 쪽은 브라우저(Browser) 기능을 위주로 하는 경우이다. 이런 경우 서버 쪽에서는 Http 서비스를 가능하도록 웹 서버의 역할을 해주고 서버에 모든 네트워크 상의 정보를 저장하고 있다가 사용자의 질의가 들어오면 그 질의에 맞는 정보를 검색하여 보여주는 시스템이다. 이러한 서버 중심 모델의 문제점은 다음과 같다. 첫 번째는 모든 작업이 서버에게 집중이 되므로 클라이언트의 증가에 따른 서버의 부담이 너무 크고 많은 수행들이 서버에서 이루어지므로 상대적으로 사용자의 요구에 대한 처리속도가 늦어질 수 있다. 그러

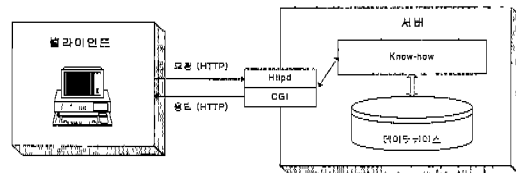


그림 1 서버 중심 모델

므로 고성능의 값비싼 서버가 필요하다. 두 번째는 네트워크 트래픽을 줄일 방법이 없다. 모든 네트워크 패킷(Network Packet)이 서버에 집중이 되므로 이를 처리하기 위한 고용량의 네트워크 증설이 필요하다. 세 번째는 검색결과에 너무 많은 정보를 가지고 있으므로 정보의 질이 떨어지고 원하는 정보를 찾기 어렵다.

2.1.2 클라이언트 중심 모델(Client Oriented Model)

클라이언트 중심 모델은 노하우(Know-how)의 전부 또는 일부를 클라이언트가 가지고 있는 모델이다. 즉, Fat Client-Thin Server형의 시스템이다. 클라이언트 쪽에 검색엔진, 데이터베이스 등의 일부 또는 전부를 위치시켜 주거나 역할의 일정부분을 할당해 주어 클라이언트 쪽에서 많은 부분을 처리하고 서버 쪽은 상대적으로 협조 정도의 일을 하는 경우이다. 이런 경우 서버 쪽에서는 Http 서비스를 이용하여 단순한 사용자 제어 등의 일을 해주고 클라이언트 쪽은 브라우저 기능 이외에 주로 사용자 정보를 위주로 검색엔진의 일부를 클라이언트 쪽에 위치하고 관리하는 기능을 지닌다. 주로 Java Applet, External Helper(MIME), Plug-In 등의 기술을 사용해서 클라이언트에서 동작하는 프로그램이 만들어지고 있다.

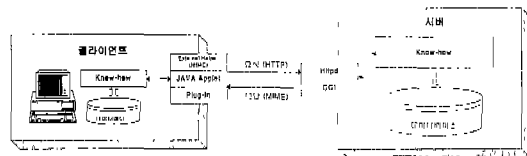


그림 2 클라이언트 중심 모델

이러한 경우의 장점은 사용자 정보를 클라이언트에서 관리할 수 있으므로 각 사용자가 자신의 기호 또는 취향에 맞는 검색을 관리하고 사용할 수 있다는 것이다. 이러한 클라이언트 중심 모델의 문제점은 클라이언트에 로드가 걸릴 수 있다는 것과 구현비용이 많이 들고 서버 쪽에서 다양한 사용자의 정보를 입수하는데 어려움이 있다는 것이다.

2.2 웹 기반 질의응답 검색 시스템

초기의 질의응답 검색 시스템은 클라이언트 중심 모델에 해당하는 클라이언트-서버(Client-Server) 구조를 채택하였으나 대부분 각 응용별로 특화되어 있었고 클라이언트의 비중이 큰 Fat-Client의 구조로 인해 비용이 많이 들고 사용자별로 클라이언트 실행(Client-Executable) 구조가 요구되어 클라이언트 프로그램을 설치해야 하는 추가부담이 필요했기 때문에 과거의 클라이언트-서버 모델은 커다란 실패를 거두지 못했다[1]. 현재의 구축 시스템들은 대부분 웹 기반 서버 중심 모델이다. 그러므로 2.1.1과 2.1.2에서 지적한 문제점이 존재한다. 표 1은 웹 기반 질의응답 검색 시스템 구축 사례이다. 본 논문에서는 이러한 문제점을 이동 에이전트 모델이라는 패러다임(Paradigm)으로 해결한다.

3. 이동 에이전트 기반 질의응답 검색 시스템 (QASSMA)

본 논문의 목표는 기존 질의응답 검색 시스템의 문제점을 해결하기 위하여 이동 검색 에이전트와 이동 로봇 에이전트를 사용하여 이형 분산의 환경에 존재하는 질의응답을 네트워크 트래픽을 줄이면서 빠른 검색 속도를

표 1 웹 기반 질의응답 검색 시스템 구축 사례

응용 및 이용기관	개발 기술	특징
Customer Support Borderbund Software	CBR과 CasePoint 웹 서버 틀은 이용한 웹 기반의 도움말 시스템	고객이 자유로운 형식으로 자신의 시스템에 발생한 문제를 적어서 이에 대한 도움말을 얻을 수 있다.
FAQ Finder, University of Chicago	자연어 인터페이스를 이용한 CBR 엔진	CBR 엔진 위에 자연어 인터페이스 틀을 제공하여 사용자의 수천 개의 FAQ 리스트에서 자신이 원하는 질문에 가장 가까운 것을 찾아주도록 제공한다.
CYLINA(Cyberspace Leveraged INtelligent Agent), GTE Laboratory	유스넷 뉴스그룹 및 특별한 참여 그룹을 위한 지능형 질의응답 시스템	GTE 연구실 사설 망에서 운영 중에 있으며 유스넷 뉴스그룹의 자료와 게시판 자료를 기초로 데이터 베이스를 구축하여 질문에 답변한다.
Netscape Help System, Netscape	Intellisystem의 SmartSite를 이용하여 사용자의 질문에 대해 웹 서버에 배치되어 있는 지능적이고 자동적으로 연결되는 웹 문서 파일을 제공하는 기술	웹 브라우저를 이용하는 온라인 고객에 대해 기술적인 도움은 도움말 시스템을 통해 제공한다.
UltraNet SelfService Support	UltraNet 고객의 문제 해결을 위한 웹 기반 도움말 시스템	UltraNet 고객에게 공통의 문제들을 스스로 해결할 수 있도록 웹을 통해 정보를 제공한다.

보일 수 있는 효율적인 검색 모델을 제안하고자 한다.

3.1 시스템의 특징

기존의 게시판과 뉴스그룹에서 질의응답을 검색하는 방법은 웹 로봇을 이용하여 웹 로봇을 실행하는 위치에서 원격 서버에 연결을 하여 자료를 검색하는 방법이다. 검색해야 할 원격 서버의 수가 증가하게 되면 서버의 부담이 증가하고 네트워크 트래픽이 웹 로봇을 실행하는 위치에 집중된다. QASSMA는 이러한 문제점을 해결하기 위하여 이동 에이전트를 사용한다. QASSMA는 다음과 같은 특징을 가지고 있다. 첫째, QASSMA는 최적의 검색 위치로 검색 코드 자체가 이동하여 검색을 수행하므로 기존의 정적인 웹 로봇에 비하여 고정된 네트워크 거리만큼 극복할 수 있고 단축된 원격 서버와의 네트워크 거리만큼 검색시간을 단축시킬 수 있다. 즉, 질의응답을 검색할 최적의 위치로 검색 코드가 이동하여 검색을 수행 후 검색 결과를 가지고 돌아온다. 검색 환경은 검색 위치에 이동 에이전트가 실행이 될 수 있는 환경인 Context가 제공되는 경우와 제공되지 않는 경우로 나누어 질 수 있다. Context가 제공되는 경우, 직접 데이터 베이스에 접근이 허용되는 경우와 허용되지 않는 경우로 나눌 수 있다. 그림 3은 각 검색환경에 따른 이동 에이전트의 검색 유형이다.

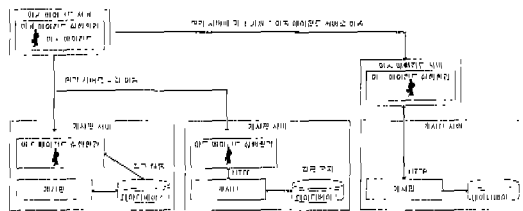


그림 3 이동 에이전트 검색 유형

QASSMA는 Context가 제공되는 경우, 검색 코드를 탑재한 이동 에이전트가 원격 서버로 이동하여 검색한다. 원격 서버에서 데이터베이스에 대한 직접 접근이 허용이 된 경우에는 데이터베이스에 직접 연결하여 검색을 수행한다. 보안상의 이유로 접근이 금지가 된 경우, 게시판 서버와는 HTTP 프로토콜을 이용하고 뉴스그룹 서버와는 NNTP 프로토콜을 이용하여 웹 문서 또는 뉴스 기사를 가져온 후에 파싱하여 검색한다. Context가 제공이 되지 않는 경우, 원격서버에 가장 가까운 이동 에이전트 서버로 이동하여 HTTP 프로토콜 또는 NNTP 프로토콜을 이용하여 검색한다. 기존의 웹 로봇은 이동하지 않고 원격 서버에 HTTP 프로토콜을 이용

하여 검색하는 방법을 사용한다. 그러므로 네트워크 거리를 극복할 방법이 없다. QASSMA는 원격 서버에 Context가 제공이 되는 경우 원격서버로 직접 이동하여 검색을 수행하므로 출발 서버와 원격 서버간의 네트워크가 필요없다. 또한 원격 서버에 Context가 제공이 되지 않는 경우에도 원격서버와 가장 가까운 이동 에이전트 서버로 이동하여 검색을 수행하므로 최적의 검색 위치로 이동함에 따라 생긴 네트워크 거리의 감소만큼 검색시간을 단축시킬 수 있다.

둘째, QASSMA는 출발서버에서 검색할 위치로 검색 코드가 이동하여 실행되므로 출발 서버의 부담을 줄일 수 있다. 그림 4는 검색코드의 이동에 따른 검색 프로세스의 이동을 나타내고 있다.

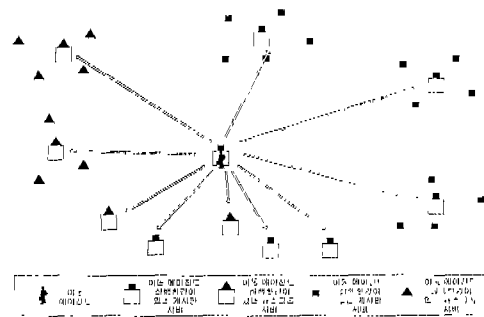


그림 4 검색 프로세스 이동

검색을 수행하는 이동 에이전트의 검색 환경은 이동 에이전트가 이동하여 검색 프로세스를 실행할 수 있는 Context가 제공되는 경우와 제공되지 않는 경우로 나뉘어진다. Context가 제공되는 경우, 이동 에이전트의 검색 코드를 직접 이동시켜 검색 프로세스를 실행한다. Context가 제공되지 않는 경우, 해당 서버에서 가장 가까운 Context가 제공되는 서버에서 검색 프로세스를 실행하고 HTTP 프로토콜과 NNTP 프로토콜을 이용하여 검색한다. 기존의 웹 로봇 서버의 경우 웹 로봇 서버에서만 검색 프로세스를 실행하여 검색할 서버의 수가 증가할수록 웹 로봇 서버의 부담이 증가한다. QASSMA는 검색 프로세스를 이동시킴으로서 서버의 부담을 줄일 수 있다. 그러므로 검색 범위가 증가함에 따라 고성능의 서버를 요구하는 기존의 검색 방식의 한계를 이동 에이전트의 분산 처리를 이용하여 극복할 수 있다.

셋째, QASSMA는 Java로 개발되어 플랫폼 독립적이며 연결 호스트의 검색 환경에 따라 능동적으로 클래스를 변경할 수 있다. 즉, 해당 서버의 Context의 제공 유

무, 서버의 사용 프로토콜에 따라 클래스를 능동적으로 변경할 수 있다. 그림 5는 이동 검색 에이전트에서 검색 환경에 따른 클래스의 추가를 나타내고 있다.

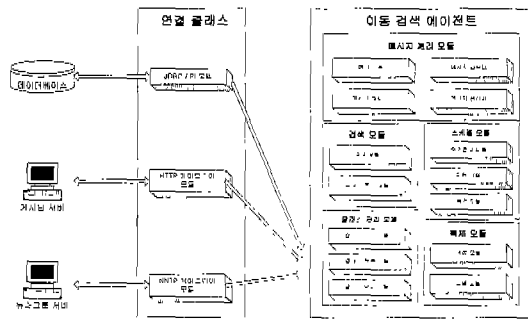


그림 5 이동 에이전트의 클래스 추가

이러한 능동적인 클래스 변경은 이동 검색 에이전트의 크기를 줄일 수 있을 뿐만 아니라 검색 정책의 변경을 능동적으로 할 수 있게 한다. 기존의 클라이언트 서버 모델의 경우, 검색 정책이 변경이 생길 때 서버의 검색 프로그램과 클라이언트의 검색 프로그램 모두를 변경해주어야 한다. QASSMA의 경우, 이동 에이전트에 추가되는 클래스의 변경으로 해결이 가능하며 능동적인 클래스 변경을 통해 실시간적인 검색 정책의 변경이 가능하다.

3.2 시스템의 구조

QASSMA는 이동 질의응답 에이전트 서버(MQAS : Mobile Q&A Agent Server), 중계기(Mediator), 이동 뉴스그룹 게이트웨이 서버(MNGS : Mobile Newsgroup Gateway Server), 이동 게시판 게이트웨이 서버(MBGS : Mobile BBS Gateway Server), 검색을 수행하기 위해 생성이 되는 이동 검색 에이전트(MSA : Mobile Search Agent)로 구성되어 있다. 그림 6은 제안된 모델의 기본구조를 나타내고 있다.

이동 질의응답 에이전트 서버(MQAS : Mobile Q&A Agent Server)는 사용자의 질의가 게시판에 게시되면 사용자 인터페이스(User Interface)는 질의 분석기(Query Analyzer)에 전달한다. 질의는 키워드 사전(Keyword Dictionary)과 유사 키워드 사전(Similar Keyword Dictionary)을 통하여 키워드가 추출 및 확장된다. 또한 질의의 종류를 구분한다. 분석된 질의는 기록 관리자(History Manager)에 기록이 되고 이동 에이전트 관리자(Mobile Agent Manager)에 전달이 된다. 이동 에이전트 관리자는 MSA를 생성시키고 MSA를

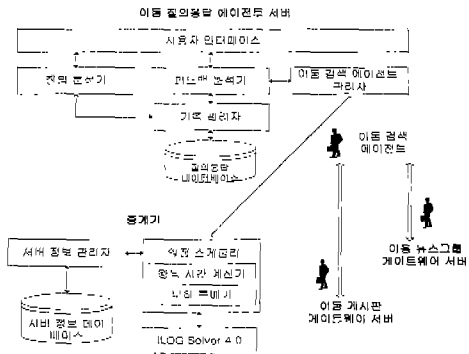


그림 6 QASSMA의 기본 구조

라우팅(Routing)시키기 위해서 중계기로 이동시킨다. 피드백 분석기(Fedcback Analyzer)는 사용자의 답변에 대한 평가를 분석하여 제 검색 시 이를 반영한다. 중계기는 해당 질의의 종류에 따라 최적의 경로로 이동 에이전트들을 라우팅(Routing)시켜주는 역할을 한다. 라우팅 시 서버에 있는 질의응답의 종류, 왕복 시간(Round Trip Time), 부하(Load)를 고려하여 이동 경로를 계산한다. 이동 검색 에이전트(MSA : Mobile Search Agent)는 원격 서버로 이동 후 로컬 데이터베이스를 검색하는 역할을 한다. 검색 수행시 중계기로 이동하여 중계기의 일정 스케줄러에 의해 지정된 서버로 전송되어 검색 작업을 수행한다. 이동 게시판 게이트웨이 서버(MBGS : Mobile BBS Gateway Server)는 질의응답 게시판 (BBS :Bulletin Board System)에 저장된 웹 문서를 읽어서 이를 파싱(Parsing)하여 원하는 기사만을 추출하여 변환하는 역할을 한다. 웹 게시물에서 다른 게시판의 기사에 연결되는 연결태그(Link Tag)를 검색하여 다른 게시물을 가져오는 자료로 참조한다. MBGS는 이동 에이전트의 코드이동 기능을 이용하여 게시판 서버 리스트를 이용하여 최적의 검색 위치로 이동하여 검색한다. MBGA는 네트워크 도메인 A에서 B로 이동하여 검색을 원하는 BBS 서버에서 자료를 검색한 뒤 검색 결과를 가지고 본래의 위치로 이동이 가능하다. 이러한 네트워크 도메인의 이동은 전체 네트워크의 트래픽을 감소시킬 뿐만 아니라 검색의 속도를 향상시킬 수 있다. 이동 뉴스그룹 게이트웨이 서버(MNGS : Mobile Newsgroup Gateway Server)는 NNTP(Network News Transfer Protocol) 프로토콜을 이용하여 뉴스그룹 서버에서 기사를 검색하고 입력한다. MNGS는 이동 뉴스그룹 게이트웨이 에이전트(MNGA :

Mobile Newsgroup Gateway Agent)를 생성하여 검색을 수행한다.

3.3 시스템의 실행 구조

검색을 수행하는 단계는 이동 검색 에이전트(MSA : Mobile Search Agent)를 생성하고 중계기로 검색 서버에 대한 라우팅 경로를 요청하는 단계, 클로닝(Cloning)을 통해 복제된 MSA를 검색위치로 이동시키는 단계, MNGS와 MBGS가 최적의 검색위치로 이동해서 검색을 수행하는 단계, 검색된 결과를 종합하여 사용자에게 전송하는 단계로 구분할 수 있다.

첫 번째 단계에서는 사용자의 질문이 입력되면 MQAS는 질문을 저장하고 해당 질문에 대한 답변을 찾기 위해 MSA를 생성하고 중계기로 MSA를 보내 검색 서버에 대한 라우팅 경로를 얻는다. 두 번째 단계에서는 MQAS는 MSA를 중계기에서 얻은 라우팅 정책에 따라 복제하고 각 검색위치로 전송한다. 세 번째 단계에서는 검색위치로 이동한 MSA가 서버의 데이터베이스를 검색하고 만약 MNGS와 MBGS일 경우에는 우선 캐쉬 데이터베이스(Cache Database)를 검색하여 자료의 유무에 따라 MNGA와 MBGA를 생성하여 검색의 최적위치로 이동하여 검색을 수행한다. 네 번째 단계에서는 검색을 마친 MSA는 다시 원래의 위치로 돌아오며 검색된 결과를 종합하여 사용자에게 검색 결과를 전달한다. 시스템의 실행 구조는 그림 7과 같다.

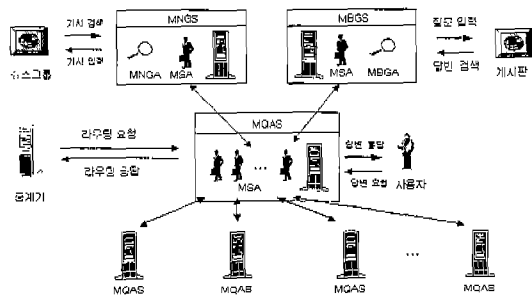


그림 7 QASSMA의 실행 구조

3.4 질의응답 정보 검색

3.4.1 검색 모드별 클래스 추가

이동 검색 에이전트는 원격 서버의 종류에 따라 클래스를 추가하고 이동 에이전트 실행환경의 존재 유무, 데이터베이스 접근 허용, 검색 자료의 종류에 따라 추가되는 클래스가 달라진다. QASSMA는 멀티 클로닝 검색 방법과 순차 검색 방법을 사용자의 우선순위의 네트워크 트래픽 상태를 고려하여 검색 모드를 변경하여 사용

한다. 멀티 클로닝 검색 방법을 사용할 때는 원격 서버에 검색 에이전트를 동시에 보내는 방법으로 원격 서버의 검색환경에 맞는 클래스만 추가한다. 순차 검색 방법은 원격 서버를 차례로 방문하는 방법으로 모든 검색 환경에 맞는 클래스가 필요하기 때문에 모든 클래스를 추가한다.

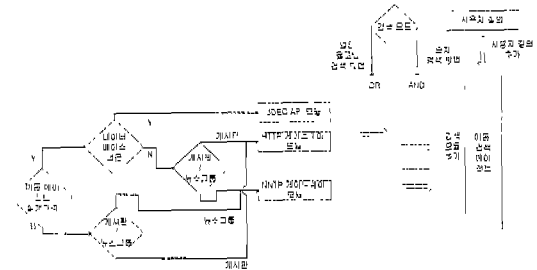


그림 8 검색 모드별 클래스 추가

3.4.2 웹 / 뉴스그룹 게이트웨이 검색 모듈

원격 서버에 이동 에이전트 실행환경이 제공되지 않는 경우 웹/뉴스그룹 모듈을 사용하여 원격 서버에 접속하여 검색을 수행한다. 웹 게이트웨이 모듈의 세부 구조는 그림 9와 같다.

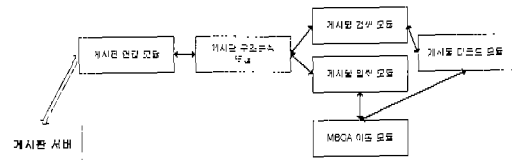


그림 9 MBGA 시스템 구조

웹 게이트웨이 검색을 수행하는 단계는 다음과 같다. 첫 번째는 게시판 연결 모듈을 통하여 검색을 원하는 게시판에 접속하고 데이터를 수신해 게시판 구조 분석 모듈에 전달한다. 게시판 연결 모듈은 일종의 HTTP 에뮬레이터(emulator)이며 원활한 연결을 위해 쿠키와 세션을 유지하고 관리한다. 두 번째는 게시판 구조 분석 모듈을 이용하여 게시판의 테이블을 분석한다. 게시판의 테이블은 번호, 제목, 작성자, 작성일, 조회 등 필드와 질의와 답변의 두 가지 타입의 레코드로 구분되어진다. 구조 분석 모듈은 필드 파서(Field Parser)와 레코드 파서(Record Parser)를 이용하여 필드와 레코드 단위의 파싱 작업을 수행한다. 세 번째는 게시물 검색 모듈을 이용하여 분석된 구조를 이용하여 연결된 링크(Link)를 추출하여 연결된 내용을 검색하고 여파한 뒤 자료를 디

코드모듈(Decode Module)을 이용하여 압축작업을 수행한다.

뉴스그룹 게이트웨이 검색을 수행하는 단계는 다음과 같다. 첫 번째는 뉴스그룹 연결 모듈을 통하여 검색을 원하는 뉴스그룹 서버에 접속한다. 이 때 사용하는 프로토콜은 NNTP로 MNGA는 뉴스그룹의 메시지 ID 범위를 구하여 분석 모듈에 전달한다. 두 번째는 분석모듈이 메시지 ID 범위 내의 기사에서 참조 ID(Reference ID)의 유무와 제목 내용 분석을 통하여 질의와 답변을 구분한다. 세 번째는 검색 모듈에서 범위내의 기사를 읽는다. 기사들은 Base64 또는 Quoted-Printable 등으로 인코딩(Encoding)되어 있으므로 디코딩(Decoding)하는 과정이 필요하다. 검색된 기사는 디코드 모듈을 이용하여 압축작업을 수행한다.

3.5 질의 검색

사용자의 질의는 제목과 내용 부분으로 나뉘어진다. 본 논문에서는 사용자의 질의와 일치하는 답변을 찾기 위해 답변이 있는 질의들 중에 질문의 제목과 내용 부분과 일치하는 질의를 찾는다. 해당 질문과 유사도를 계산하기 위해 벡터 모델을 사용한다. 벡터 모델은 사용자의 질의와 검색된 정보를 index term의 집합으로 표현한 후 각 index term에 가중치(weight)를 부여함으로써 이루어진다. 부여된 가중치를 바탕으로 사용자 질의와 검색된 정보와의 유사도를 측정하여 그 정보가 사용자 질의와 얼마나 부합되는지를 판단한다. 또한 이 유사도는 각 정보의 순위(ranking)를 통한 자료 여과에 사용되어진다. 그림 10은 벡터 모델에 의한 유사도 계산 및 정보 여과를 나타내고 있다.

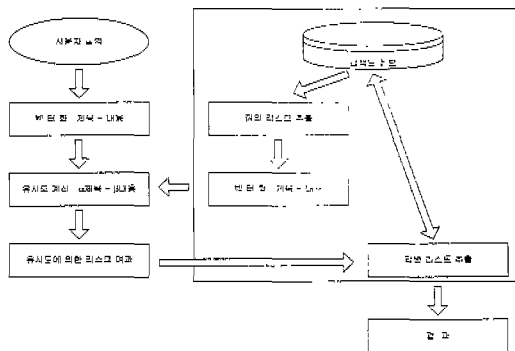


그림 10 유사도에 의한 정보 여과

사용자 질의 q의 i번째 index term에 부여된 weight를  $w_{i,q}$ ,  $t$ 를 전체 색인어 수라고 하면 질의 벡터  $\vec{q}$ 는

$\vec{q} = (w_{1,q}, w_{2,q}, \dots, w_{i,q})$ 로 정의되며 유사도 비교를 원하는 정보의 벡터  $\vec{d}_j$ 는  $\vec{d}_j = (w_{1,j}, w_{2,j}, \dots, w_{i,j})$ 로 표현된다. 사용자의 질의 벡터와 정보 벡터와의 유사도를 구하는 공식은 코사인 계수공식을 사용한다[2].

전체 유사도  $sim_{total}(d_j, q)$ 는 제목의 유사도  $sim_{title}(d', q')$ 와 내용의 유사도  $sim_{content}(d'', q'')$ 의 합으로 구한다. 제목과 내용의 유사도는 중요도에 비례하여 차등적으로 적용된다.

$$sim_{total}(d_j, q) = \alpha \times sim_{title}(d', q') + \beta \times sim_{content}(d'', q'') \quad (1)$$

$\alpha$  : 제목(title)의 유사도에 대한 weight

$\beta$  : 내용(content)의 유사도에 대한 weight ( $\alpha \geq \beta$ )

유사도를 구하기 위해서는 각 벡터의 index term에 weight를 부여해 주어야 한다. 정보의 index term에 weight를 주는 방법으로는 TF-IDF방식을 사용하여 계산하고 사용자의 질의에 index term을 주는 방법은 Salton and Buckley가 제안한 방법을 사용한다[2].

3.6 이동 에이전트 스케줄링

그림 11은 전송 파라미터를 표현한 시스템 모델을 보이고 있다. 표 2는 파라미터를 나타낸 표이다.

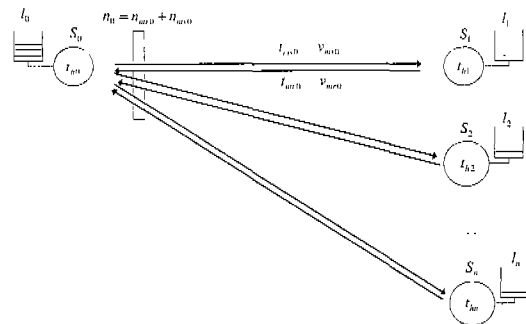


그림 11 전송 파라미터 모델

표 2 전송 파라미터

파라미터	설명
$S_i$	서버
$t_{i1}$	서버에서 소요된 처리시간
$t_{ms1}$	서버에서 전송시 걸린 시간
$t_{mr}$	서버에서 수신시 걸린 시간
$U_{msi}$	서버에서 전송한 데이터의 양
$U_{mri}$	서버에서 수신한 데이터의 양
$n_i (=n_{msi} - n_{mri})$	서버에서 데이터를 수신한 횟수(=송신+수신)
$t_w$	서버가 전송 후 응답을 대기 시간
$l_i$	서버의 로드

3.6.1 기존 시스템 검색 방법

서버 중심 모델과 클라이언트 중심 모델을 이용한 검색 시스템의 경우 Know-how의 위치가 상대적으로 차이가 있다. 서버 중심 모델의 경우 출발지 서버  $S_0$ 에서 목표 서버  $S_j$ 으로 데이터를 전송시에 Know-how가 서버에 있으므로 상대적으로 클라이언트 중심 모델에 비해서 목표 서버  $S_j$ 에서 처리되는 시간  $t_{h1}$ 이 짧고 클라이언트에서 처리되는 시간  $t_{h0}$ 는 적다.  $S_0$ 에서  $S_j$ 으로 자료 요청하고 응답을 받을 때까지 기다리는 시간  $t_{wo}$ 는 순차 검색 방법과 멀티 클로닝 검색 방법이 있다. 만약 순차 검색 방법으로 요청하는 경우의  $t_{wo}$ 는  $(t_{mr1} + t_{h1} + t_{ms1}) + (t_{mr2} + t_{h2} + t_{ms2} + t_{g2}) + \dots + (t_{mrn} + t_{hn} + t_{msn} + t_{gn})$ 이다. 여기서  $t_g$ 는 Gap Time으로 다음 서버에 전송 시 걸리는 시간 간격이다.

$$t_{i0} = \sum_{i=1}^n t_{mri} + t_{hi} + t_{msi} + t_{gi} \quad (2)$$

만약 멀티 클로닝 검색 방법으로 요청하는 경우는 서버의 로드  $l_0$ 와 네트워크 트래픽이 급격히 증가한다. 또한 클라이언트는 서버에 요청한 후 응답을 기다리는 방식이 동기적(Synchronous)이므로 응답을 기다리기까지  $S_0$ 의 자원을 소모하며 기다려야 한다.

3.6.2 이동 에이전트 검색 방법

기존의 정적인 검색 시스템은 검색 요청 후 응답을 기다리는 방식으로 동기적이나 이동 검색 에이전트는 목표 서버로 이동 후에 출발지와의 연결이 필요 없는 비동기적(Asynchronous)이다. 이 것은 기존 시스템과 비교하여 서버의 로드  $l_0$ 와 네트워크 트래픽을 감소시킬 수 있다. 만약 기존 검색 시스템에서 서버의 최대 한계 로드(Maximum Limit Load)  $l_{max}$ 를 고려한 시스템의 작성한 경우  $j$ 번째 목표서버로 요청 후에  $l_{max}$ 에 도달한 경우 응답을 기다리기 위해서는 요청에 대한 응답이 오는 서버가 있어야 한다. 즉, 이동 검색 에이전트를 이용한 방법에서  $t_{hi}$ 는 상대적으로 적다. 그리고 검색을 위해 에이전트 코드를 함께 보내야 하기 때문에  $t_{mri}$ 와  $t_{msi}$ 는 상대적으로 크다. 그러므로  $t_{hi}$ 와  $t_{mri} + t_{msi}$  간에 trade-off가 존재한다. 순차 검색 방법은 기존의 서버 중심 모델이나 클라이언트 중심 모델에 비해서 전체 네트워크 트래픽과 출발지 서버의 로드를 줄일 수 있다. 순서적으로 이동하기 때문에 출발지 서버와의 데이터 수신횟수  $n_0$ 는 2이다. 정적인 검색 모델의 경우,  $n_0$ 는  $(n - 1) \times 2$ 로 출발지서버와 클라이언트간의 지속적인 데이터 송수신이 필요하다.

3.6.3 이동 하이브리드 검색 방법

QASSMA는 사용자의 우선 순위와 네트워크 트래픽

상태를 고려하여 순차 검색 방법과 멀티 클로닝 검색 방법을 혼합한 이동 하이브리드 검색 방법(Mobile Hybrid Search Method)을 사용한다. 즉, 사용자의 우선 순위, 네트워크 트래픽 상태, 서버 부담 등을 고려하여 능동적으로 검색 정책을 바꾼다. 이러한 능동적인 검색 정책의 변환은 긴급성을 요하는 경우에는 멀티 클로닝 검색 방법을 통한 빠른 검색 속도를 얻을 수 있고, 네트워크 트래픽 및 서버 부담의 최소화를 요하는 경우에는 순차 검색 방법을 사용하여 최소한의 서버와 네트워크 자원을 소비한다. 순차 검색 방법의 경우, 각 서버 노드(node)는 각 다른 네트워크 이동거리를 가지고 있다. 전체 네트워크 트래픽을 줄이기 위해서는 이 전체 이동거리를 짧게 하는 정책이 필요하다. 이 문제는 전형적인 스케줄링 문제로서  $n$ 개의 네트워크 node들을 스케줄링 할 경우  $O(n!)$ 의 매우 큰 복잡도를 갖는 NP 문제이다. 전통적으로 이러한 문제를 해결하기 위하여 OR(Operations Research), 정수형 프로그래밍 방법(Integer Programming Theory), 지식 기반 방법(Knowledge based)등 여러 가지 인공지능 기법들이 연구되어 왔다. 최근에는 이러한 기법들 중 하나인 CSP 기법이 연구되어왔다[3][4]. CSP(제약 조건 만족 문제 : Constraint Satisfaction Problems)기법은 유한 이산 도메인을 가지는 변수들의 집합과 이들 변수들 사이의 제약조건들의 집합으로 문제를 정형화한 후 일치성 검사(Consistency Checking)와 휴리스틱 검색 알고리즘(Heuristic Search Algorithm)등을 적용하여 문제를 해결하는 방법이다. QASSMA에서는 이러한 문제를 해결하기 위해 제약 조건 문제 해결기인 ILOG Solver 4.0을 사용한다.

4. 실험 및 평가

본 논문에서 실험한 환경은 다음과 같다.

- 하드웨어 (운영체제) :
  - Pentium PC 13대 (Windows 98).
  - Pentium PC 6대 (Windows NT4.0),
  - Sun Ultra Workstation 1대 (SunOS 5.6),
  - HP Workstation 1대 (SunOS 5.4)
- 웹 서버 : MS-IIS 5.0, Apache 1.3
- DBMS : MS-SQL 7.0, MYSQL 9.38

이동 에이전트 개발 틀로는 IBM Japan research group에서 개발한 Java Aglet 개발환경인 Aglet Workbench를 사용하였다. Aglet이란 Agile Applet를 뜻하며 Java로 구현된 이동 에이전트의 통칭이다. Aglet Workbench에서는 Java-AAPI(Aglet Application Programming Interface)를 제공하여 Aglet과 동



적환경(Aglet Context) 사이의 상호작용을 가능하게 하고 수많은 라이브러리를 제공할 수 있게 함으로써 이동 에이전트를 이용하는 네트워크 기반의 어플리케이션 개발을 위한 개발환경을 제공한다[5]. 실험에서는 MFC 프로그래밍 사이트의 질의응답 자료를 이용하고 있다. 그림 12는 사용자가 질문을 입력하는 인터페이스이다.

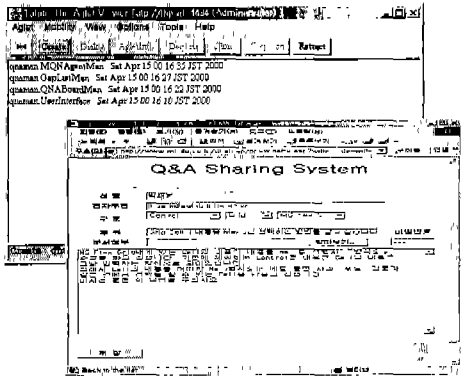


그림 12 질의 입력

질문이 입력이 되면 이동 질의응답 에이전트 서버는 이동 검색 에이전트를 생성한다. 이동 검색 에이전트는 중계기로 이동하여 스케줄 정보를 얻는다. 그림 13은 중계기에서 검색할 서버의 목록에 따라 네트워크 거리 차트를 나타내고 있다. 중계기는 네트워크 거리 측정 에이전트를 생성하여 일정한 주기로 전체 네트워크 거리 및 서버 부담을 측정한다. 이 네트워크 거리 및 서버 부담을 이용하여 ILOG Solver는 스케줄을 세운다. 이동 검색 에이전트는 스케줄에 따라 검색을 수행한다. 그림 14와 15는 이동 검색 에이전트의 게시판과 뉴스그룹의 검색 화면을 나타내고 있다. 그림 16은 검색 결과에 대한

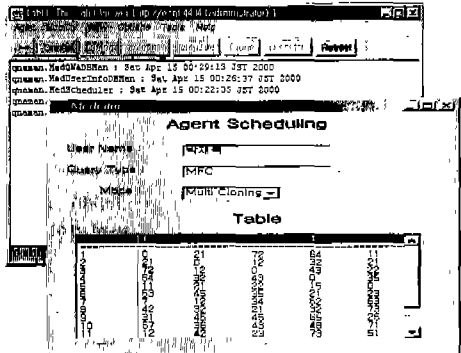


그림 13 중계기 시간 테이블 차트

유사도를 측정하는 화면을 나타내고 있다. 제목과 내용의 유사도가 계산되며 계산된 전체 유사도에 따라 순위가 결정이 되며 한계 유사도 이상의 답변이 선택된다.

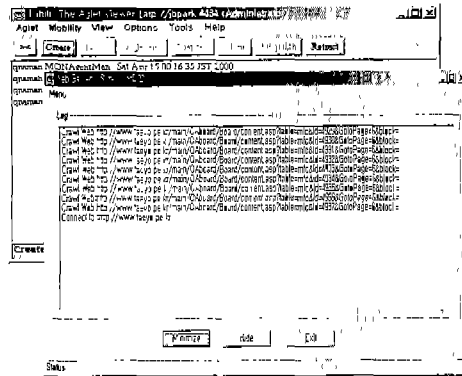


그림 14 게시판 검색

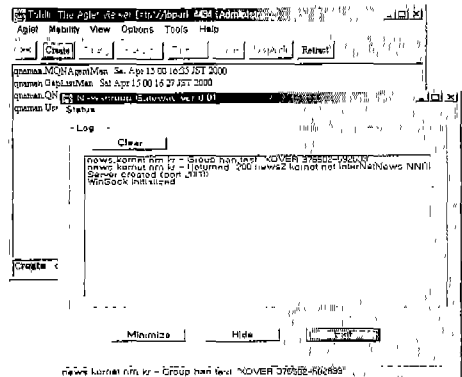


그림 15 뉴스그룹 검색

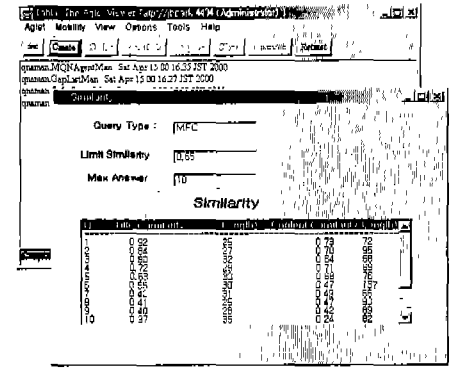


그림 16 제목과 내용의 유사도 검사

검색결과를 관리자가 정한 최대 답변 수만큼을 검색하고 자동 질문 검색으로 인한 잘못된 답변의 검색을 보완하기 위하여 답변에 대한 사용자의 피드백을 조사한다. 피드백은 추가적인 답변의 필요성 유·무와 사용자의 답변에 대한 정확성의 평가로 구성된다. 정확성의 평가는 답변의 삭제 유·무 및 답변의 적절성을 평가하는 자료로 이용된다. 다음은 답변에 대한 사용자의 피드백을 입력할 수 있는 답변의 확인에 대한 인터페이스 화면이다.

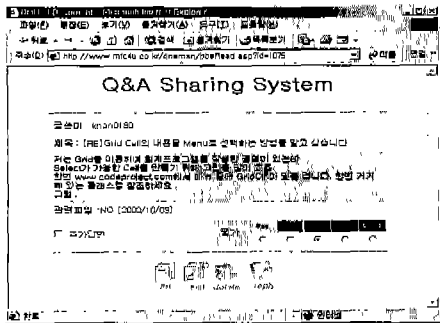


그림 17 사용자 피드백 입력

Aglet으로 구현된 이동 에이전트간의 통신은 ATP (Agent Transfer Protocol)를 이용한 통신 레이어에 의해 이루어진다. ATP는 HTTP 프로토콜(Protocol)위에 구현된 어플리케이션 레벨 인터넷 프로토콜(Application-level Internet Protocol)이다. Aglet Server인 Tahiti는 Aglet의 실행 환경을 제공한다. 모든 이동 에이전트는 이 실행환경에서만 이동할 수 있기 때문에 MQAS, MBGS, MNGS, Mediator 모두 Tahiti와 함께 실행된다. 실험에서 이동 검색 에이전트와 기존 시스템을 비교하기 위해 각 서버에 Port 번호를 다르게 하여 5개의 Tahiti 서버를 설치하여 총 20대의 서버에 100개의 Tahiti 서버를 가정한다. 출발지 서버에는 1개의 Tahiti 서버를 설치한다. 검색 엔진 모듈은 서버 중심 모델과 MSA 모두 같은 것을 사용한다.

출발지 서버가 전송 후 대기 시간  $t_{w0}$  는 순차적으로 검색하는 경우 서버 중심 모델과 MSA 모두 비슷한 검색 시간을 보였다. MSA의 경우는 검색을 위해 에이전트 코드를 전송해야 하기 때문에 상대적으로 시간이 많이 걸리지만 코드 전송 시간은 서버에서의 검색 시간에 비교하여 상대적으로 작은 시간이기 때문에 전체적인 성능은 비슷했다. 그러나 서버 중심 모델의 멀티 쓰레드를 생성하는 방법과 MSA의 멀티 클로닝 검색 방법의

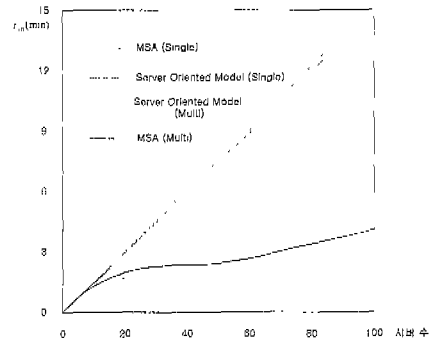


그림 18 서버 중심 모델과 MSA의  $t_{w0}$

경우 많은 차이가 났다. 서버 중심 모델의 멀티 쓰레드를 생성하는 방법은 동기적으로 처리되는 방식이기 때문에 서버의 로드가 상대적으로 MSA에 비하여 많다. 그래서 서버가 일정한 수 이상의 쓰레드를 생성했을 때 성능이 급격히 감소하는 현상이 발생하고 MSA의 멀티 클로닝 검색 방법의 경우 검색 코드 전송 후 연결이 필요 없으므로 상대적으로 성능의 감소가 적다. 이러한 성능의 급격한 감소는 시스템의 성능과 운영체제에 많은 영향을 받는다.

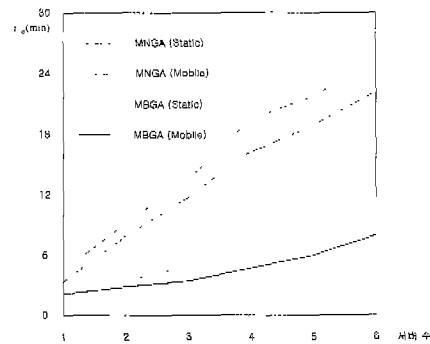


그림 19 MNGA와 MBGA의 이동 유무에 의한  $t_{w0}$

MBGA와 MNGA는 이동성 유무에 따른 전송 후 대기 시간  $t_{w0}$ 는 많은 차이가 났다. MNGA의 경우, 뉴스 그룹 기사의 양은 상대적으로 웹 서버의 게시판에 비하여 자료가 적고 게시판 자료의 변환에는 추가적인 파싱 작업이 필요하므로 서버의 로드가 크다. MNGA가 고정적으로 위치하고 있는 경우와 최적의 위치로 이동하는 경우의  $t_{w0}$ 는 뉴스 그룹 서버의 수가 적을 때는 고정형이 이동형에 비해 적은  $t_{w0}$ 를 보이지만 서버의 수가 증가함에 따라 점진적인 감소를 보였다. MBGA의 경우,

최적 위치로 이동하는 이동형이 고정형에 비해 훨씬 적은  $t_{w0}$  를 보였다. 왜냐하면, 웹 서버의 게시판 자료는 상대적으로 많기 때문이다. 즉, 게시물의 자료가 많으면 많을수록 이동한 위치의 서버와 검색을 원하는 웹 서버의 네트워크 거리가 가까울수록 이동형과 고정형의  $t_{w0}$  는 많은 차이를 보인다.

### 5. 결론 및 향후 연구

본 논문에서는 이형 분산의 환경에 존재하는 질의응답 게시판 및 뉴스그룹을 검색할 수 있는 모델을 제시하고 성능 비교를 통하여 기존 검색 모델과의 차이점을 확인할 수 있다. 첫 번째, QASSMA는 능동적인 검색 정책의 변경을 통하여 네트워크 트래픽, 서버 부담 및 검색 속도를 효율적으로 조절할 수 있다. 순차 검색 방법을 통하여 네트워크 트래픽 및 서버 부담을 최소화할 수 있다. 검색 속도의 경우 순차 검색 방법은 서버 중심 모델과 비슷한 성능을 보인다. 이것은 서버 중심 모델의 중앙 집중적인 네트워크 트래픽의 형성과 서버 부담의 증가에 비해 상대적으로 적은 네트워크 트래픽의 형성과 서버 부담의 분산을 가져온다. 또한 멀티 클로닝 검색 방법을 통하여 빠른 검색 속도를 얻을 수 있다. 성능 분석의 결과 서버의 수가 일정 수 이상으로 증가하면 서버 중심 모델은 급격한 성능의 감소를 보이지만 멀티 클로닝 검색 방법은 상대적인 성능의 감소가 적다. 뉴스그룹과 게시판 게이트웨이를 이용한 검색의 경우에는 기존의 고정적인 검색 로봇에 비하여 이동을 통해 얻은 네트워크 거리의 감소에 해당하는 시간적인 검색 시간의 감소를 확인할 수 있다. 두 번째, QASSMA는 검색 환경에 따라 능동적으로 검색 클래스를 추가 및 변경이 가능하다. 변경된 검색 환경에 맞는 클래스를 탑재하여 효율적으로 검색 환경의 변화에 대처할 수 있고 다양한 검색 방법을 지원할 수 있다.

본 논문에서의 성능 분석 결과는 비교적 간단한 환경 조건을 사용하여 기술되었기 때문에, 실제적인 검색 시스템과 검색 엔진에 대한 연구가 필요하다고 생각한다. 또한 자바 기반 무선 PDA 또는 CDMA에서 무선 인터넷을 통한 이동 에이전트의 이동에 대한 네트워크 트래픽의 분석과 접속량 등의 좀더 활발한 연구가 필요하다.

### 참 고 문 헌

- [1] Behrouz H. Far and K. Zenya, EX-W-Pert System : A Web-Based Distributed Expert System for Groupware Design, *Expert System With Applications*, Vol 11, No.4, pp.475-480, 1996.
- [2] Ricardo Baeza-Yates and R. N. Berthier, *Modern Information Retrieval*, Addison-Wesley Publish, pp.27-30, 1999.
- [3] Pascal Van Hentenryck, *Constraint Satisfaction in Logic Programming*, MIT Press, U.S.A, 1989.
- [4] Tsang, Edward, *Foundation of Constraint Satisfaction*, Academic Press, London, 1993.
- [5] Danny D. Lange and O. Mitsuru, *Programming and Deploying Java Mobile Agents with Aglets*, Addison-Wesley, pp.1-5, 1998.
- [6] Edmund H. Durfee and R. S. Jeffrey, *Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples*, In *Proceedings of the Thirteenth International Distributed Artificial Intelligence Workshop*, pp.94-104, July, 1994.
- [7] Robin D. Burke, H. J. Kristian and K. A. Vladimir, *Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder Systems*, Technical Report 97-05, Chicago University, 1997.
- [8] Steven D. Whitehead, *Auto-FAQ : an experiment in cyberspace leveraging*, *Computer Networks and ISDN Systems* 28, pp.137-146, 1995.
- [9] Burke R., K. Hammond and E. Cooper, *Knowledge-based information retrieval from semi-structured text*. In *AAAI Workshop on Internet-based information Systems*, AAAI, pp.9-15, 1996
- [10] Buckley C., *Implementation of the SMART Information Retrieval [sic] System*, Technical Report 85-686, Cornell University, 1985.
- [11] Cutting D., J. Kupiec and J. Pederson and P. Sibun, *A Practical Part-of Speech Tagger*, In *Proceedings of the Third Conference on Applied Natural Language Processing*, ACL, 1992.
- [12] Salton G. and C. Buckley, *Global Text Matching for Information Retrieval*, *Science* 253, pp.1012-1015, 1991.
- [13] Yezdi Lashkari, M. Max and M. Pattie, *Collaborative Interface Agent*, *Readings In Agents*, Morgan Kaufmann Publishers, pp.111-116, 1998.
- [14] Nathaniel Good, S. Ben and K. A. Joseph and B. Al and S. Badrul and H. Jon and R. John, *Combining Collaborative Filtering with Personal Agents for Better Recommendations*, AAAI, 1999.
- [15] Schafer J. B., J. Konstan and J. Riedl, *Recommender Systems in E-Commerce*, *Proceedings of the ACM Conference on Electronic Commerce*, pp.3-5, November, 1999.



**박재복**

1995년 경희대학교 유전공학/전자계산공학  
학과(부전공) (학사). 2001년 인하대학교  
전자계산공학과 (석사). 2000년 ~ 현재  
(주) 소프트웨어벨리 개발팀장. 관심분야  
는 이동 에이전트, 무선 인터넷, 지능형  
전자상거래



**이광용**

2000년 인하대학교 전자계산공학과 (학  
사). 2001년 인하대학교 전자계산공학과  
석사과정. 관심분야는 이동 에이전트, 지  
능형 전자상거래, 전문가 시스템



**조근식**

1982년 인하대학교 전자계산공학과 (학  
사). 1985년 Queens College/CUNY  
(City University of New York) 전자계  
산공학과 (석사). 1991년 CUNY 전자계  
산공학과 (박사). 1991년 Logic Based  
Systems Lab, 연구원. 1998년 인하대학  
교 전자계산공학과 학과장. 1998년 ~ 2001년 인하대학교  
창업지원 연구센터 소장. 1992년 ~ 현재 인하대학교 전자  
계산공학과 교수 (정교수). 관심분야는 전문가 시스템, 지능  
형 에이전트, 귀납기계학습