지능형 경로 탐색 이동 에이전트 모델 설계

고 현* · 김광종* · 이연식*

Design of Mobile Agent Model Supporting the Intelligent Path Search

Hyun Ko* · Kwang-jong Kim* · Yon-sik Lee*

요 약

본 논문에서는 네트워크 트래픽 감지 및 이동 에이전트의 이주 노드들에 대한 최적 경로 탐색을 통해 분산 환경에서의 효율적인 작업 처리 능력을 가진 CORBA 기반의 이동 에이전트 모델(CMAM: CORBA-based Mobile Agent Model)을 설계한다. 기존 이동 에이전트 모델은 사용자로부터 다양한 작업을 부여받게 될 경우, 실행모듈 크기의 증가로 인해 네트워크 부하 및 트래픽을 가중시키고, 사용자에 의한 수동적인 호스트(이하 노드) 라우팅 스케줄 지정에 따라 노드 이주 수행 시 많은 트래픽이 발생할 경우, 트래픽으로 인한 많은 노드 순회 검색 시간 비용이 소요된다. 따라서, 본 논문에서는 능동적인 라우팅 스케줄 지정에 따라 특정한 상황에 동적으로 대처하여 에이전트의 이주 신뢰성을 보장하고 최적 경로 탐색을 통해 에이전트의 순회 작업 처리 시간을 최소화할 수 있는 새로운 이동 에이전트 모델을 설계한다. 제안된 모델은 확장된 형태의 MAFFinder를 통해 능동적으로 이주 노드들의 라우팅 스케줄을 지정하고, 에이전트 크기로 인한 네트워크 부하를 감소하기 위해 기존의 이동 에이전트 객체를 CORBA의 분산 객체 형태에 기반하여 에이전트 호출 모듈만을 포함한 이동 에이전트와 작업 실행 모듈을 가진 푸시 에이전트로 분리한다. 또한, 이주 노드의 최적 경로 탐색을 통해 이동 에이전트의 순회 작업 처리 시 소요되는 시간을 단축시킨다.

ABSTRACT

In this paper, we design the CORBA-based Mobile Agent Model (CMAM) which has the capability of the efficient work processing in distributed environment through sensing network traffic and searching the optimal path for migration nodes of mobile agent. In case existing mobile agent model is given various works from user, the network overhead and traffic are increased by increasing of execution module size. Also, if it happens a large quantity of traffics due to migration of nodes according to appointment of the passive host(below node) routing schedule by user, it needs much cost for node search time by traffic. Therefore, in this paper, we design a new mobile agent model that assures the reliability of agent' s migration through dynamic act on the specific situation according to appointment of the active routing schedule and can minimize agent's work processing time through optimal path search. The proposed model assigns routing schedule of the migration nodes actively using an extended MAFFinder. Also, for decreasing overhead of network by agent' s size, it separates the existing mobile agent object by mobile agent including only agent calling module and push agent with work executing module based on distributed object type of CORBA. Also, it reduces the required time for round works of mobile agent through the optimal path search of migration nodes.

키워드

CORBA 기반의 이동 에이전트 모델, 라우팅 스케줄, 노드 순회 검색 시간, MAFFinder

*한국 군산대학교

접수일자 2002. 11. 1

Ⅰ. 서 론

최근 네트워크 기술의 발달과 인터넷의 확산으로 기존의 컴퓨팅 환경이 분산 컴퓨팅 환경으로 변화함에 따라 새로운 방식의 분산 컴퓨팅 메커니즘들이 연구되고 있다. 이러한 연구들 중 분산 시스템의 새로운 패러다임인 이동 에이전트 기술은 통신망의 점유와 과부하를 효율적으로 감소시킴으로써 기존 분산 컴퓨팅 환경에서의 시스템 문제들을 해결하고 있다[1,2,3,4]. 이에 따라 최근 정보 검색 및 분산처리, 이동 컴퓨팅, 전자상거래, 네트워크 관리 등의 다양한 분야에 그 응용이 확산되고 있는 추세이다. 그러나, 기존의 이동 에이전트 시스템들은 일반적인 경우 네트워크 트래픽(Traffic) 감소 및 과부하 방지를 지원하여 분산환경에서의 한정적인 네트워크 대역폭에 대한 문제를 해결할 수 있지만, 에이전트가 사용자로부터 복잡하고 다양한 작업을 부여받거나 혹은 노드들을 이주중인 에이전트의 작업 처리 결과가 매우 많을 경우, 에이전트의 크기가 커져 네트워크 부하를 가중시키고 이로 인해 에이전트의 순회 작업 처리 시간이 많이 소요된다. 또한, 대부분의 에이전트 시스템들은 사용자가 직접 이주할 노드들의 라우팅 스케줄(Routing Schedule)을 지정함으로써 네트워크에 연결된 다수의 에이전트 시스템들로 에이전트가 이주할 경우, 통신망 결손이나 노드 장애와 같은 특정한 상황에 동적으로 대처할 수 없으며 신뢰성 있는 이주 보장을 제공하지 않는다[5]. 따라서, 에이전트 크기에 따른 네트워크 부하 문제나 수동적인 라우팅 스케줄 지정 문제들을 해결하여 에이전트의 순회 작업 처리 시간을 최소화시키고 에이전트의 이주 신뢰성을 보장할 수 있는 새로운 에이전트 모델이 필요하다. 이에 본 논문에서는 사용자에 의한 수동적 라우팅 스케줄 지정 방식에서 탈피하여 네트워크 트래픽 감지를 이용한 목적 노드까지의 최적 경로 탐색을 통해 동적인 경로 조정을 수행하여 에이전트를 이주시키는 지능형 경로 탐색 이동 에이전트 모델을 설계한다. 이는 확장된 형태의 MAFFinder를 통해 능동적으로 이주 노드의 라우팅 스케줄을 지정하고, 최적 경로 탐색을 통해 효율적인 노드 이주를 수행함으로써 이동 에이전트의 작업 처리 시 소요되는 시간 비용을 최소화 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 기존에 개발된 이동 에이전트의 이주 정책과 멀티 에이전트 모델에 대해 기술하고, 3장에서는 기존의 이주 정책들이 가진 문제점이나 미비점 등을 보완한 새로운 이주 기법을 제시한다. 4장에서는 멀티 에이전트 모델의 프레임웍(Framework)을 응용하여 이동 에이전트와 푸시 에이전트, MAFFinder로 구성된 이동 에이전트 모델을 설계하고, 설계된 모델의 통신 수행 과정을 보인다. 마지막 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

Ⅱ. 관련연구

2.1 기존 이동 에이전트의 이주 정책

기존에 개발된 대부분의 이동 에이전트 시스템들은 이동 에이전트의 이주 전략과 관련하여 특별한 이주 보장 정책을 지원하지 않거나 이를 지원한다 해도 대부분 통신망 혹은 노드 결점에 대한 이주 신뢰성만을 보장하기 위한 이주 정책들을 제공하고 있다[4,6,7,8,9,10,11]. 통신망 결손이나 노드 장애 또는 데이터베이스 등 정보 서비스의 부재 시 이동 에이전트는 무한 대기 상태나 고아(Orphan) 상태에 빠질 수 있고 혹은 파괴되어 쓰레기(Garbage)로 처리될 수 있다[5]. 이러한 이동 에이전트 이주 시의 장애 요인들에 적절하게 대처하기 위해 JAMAS[11]에서는 경로 재조정과 후위 복구 기법을 제안하였고, Mole[4]는 이동 에이전트에 대한 고아 찾기와 성공적인 종료를 위한 그림자(Shadow) 프로토콜을 제시하였다. 또한, Voyager[10]에서는 이동 에이전트의 생명 주기를 위한 5가지 정책을 지원하여 이동 에이전트의 이주를 보장할 수 있도록 하였다.

한편, 이동 에이전트의 이주 노드에 대한 라우팅 스케줄 지정과 관련하여 대부분의 이동 에이전트 시스템들은 Aglets[6,7]에서와 같이 이동 에이전트가 고정된 순서대로 노드를 방문하고 방문한 노드에서 검색한 데이터를 계속 누적시키면서 이동하는 단순 구조를 가지고 있다. 이는 복잡하고 다양한 사용자 요구를 부여받은 에이전트의 작업 수행에 있어 노드 재지정을 필요로 하는 경우나 통신망 결손 및 노드 장애와 같은 특정한 상황에 동적으로 대처할 수 없

는 문제가 있다. 이와 같이 현재에는 이동 에이전트의 이주 전략과 관련하여 통신망 혹은 노드의 장애가 발생하였을 때 이주 신뢰성을 보장하기 위한 연구가 주류를 이루고 있을 뿐 최소의 네트워크 소요시간을 갖도록 이동 에이전트의 이주 계획을 세우는 알고리즘에 대한 연구는 아직 제시되지 않았다. 따라서, 3장에서 최소의 네트워크 소요시간을 갖도록 하는 이주 기법들을 제안한다.

2.2 멀티 에이전트 모델

멀티 에이전트 모델은 독립적인 에이전트들의 능력을 초과하는 어떠한 문제를 해결하기 위해 복수의 에이전트들이 서로 협력하여 문제를 해결하는 새로운 방식의 문제 해결 시스템이다[12, 13]. 그림 1의 멀티 에이전트 모델은 이동, 푸시, 네이밍, 시스템 모니터링 에이전트와 같은 개별 에이전트들이 상호 보완적 관계를 형성하여 서로 간의 협력을 통해 효율적이고 안정적으로 작업을 처리하는 환경을 제공한다[14].

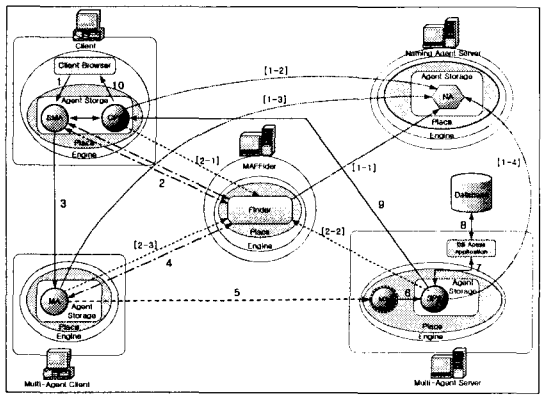

그림 1. 멀티 에이전트 모델
Fig. Multi Agent Model

그림 1에서 [1-1]~[1-4]는 각 시스템의 초기 기동 시 네이밍 에이전트에 Finder 객체 및 각 개별 에이전트(MA, CPA, SPA, SMA) 객체와 그 에이전트를 관리하는 에이전트 시스템(Agent System) 객체, 각 에이전트의 상태 정보를 저장 및 관리하기 위한 플레이스(Place) 객체를 등록하는 과정이고, [2-1]~[2-3]도 이와 유사하게 MAFFinder에 각 개별 에이전트(MA, CPA, SPA)와 에이전트 시스템, 플레이스를 등록하는 과정이다. 1~10은 각 개별 에이전트들 간의 통신 흐름을 나타내는 과정으로, SMA(System Monitoring Agent)가 사용자로부터 요구를 받은 후 MA(Mobile Agent)에 그 요구를 전달하면, MA는 해당 SPA로 이주하여 작업을 처리한다. MA에 의해 처리된 결과는 SPA(Server Push Agent)가 독자적으로 CPA(Client Push Agent) 에 이를 전송하고, CPA는 전송받은 결과를 필터링(Filtering)하여 중복 데이터를 제거한 후 사용자에게 보여준다. 이와 같이 개별 에이전트들 간 상호 협력을 통해 작업을 처리하는 멀티 에이전트 모델을 응용하여 4장에서는 에이전트 크기에 따른 네트워크 부하를 감소시켜 효율적인 이주를 수행하는 이동 에이전트 모델을 제안한다.

III. 이동 에이전트의 이주 기법

본 장에서는 2.1절의 기존 에이전트 이주 정책들의 문제점이나 미비점을 보완한 새로운 에이전트 이주 방식인 최적 이주 경로 탐색 및 조정 기법과 노드 재지정 기법을 제시한다. 제안된 기법들은 이동 에이전트의 노드 이주 시 이주 신뢰성을 보장할 뿐만 아니라 최소의 네트워크 소요시간을 갖도록 하기 위한 방법이다.

3.1 에이전트의 노드 이주 시간 최소화 방법

에이전트의 노드 간 이주 시간은 에이전트의 구조나 에이전트의 이주 정책에 따른 노드 간 이주 방법이 큰 영향을 미친다. 먼저, 이동 에이전트 구조와 이주 시간과의 관계에 있어, 에이전트의 크기는 네트워크의 부하 발생 요인으로 작용하여 이동 에이전트의 이주 시간을 많이 걸리게 하거나 적게 걸리게 한다. 기존의 이동 에이전트는 사용자로부터 복잡하고 어려운 문제를 부여받았을 경우, 작업 처리를 위한 실행 로직의 크기가 커져 에이전트의 크기를 증가시킬 수 있고, 에이전트의 작업 처리 결과가 매우 많은 경우에도 에이전트의 크기를 증가시켜 네트워크 과부하를 발생시킨다. 한편, 이동 에이전트의 크기가 일정할 경우, 이주 경로 내에 발생한 네트워크 트래픽에 따라 이동 에이전트의 노드 이주 시간이 결정

된다. 즉, 특정 경로를 통한 에이전트 이주 시 해당 경로 상에 발생한 트래픽의 양(Quantity)에 따라 에이전트의 이주 시간이 많이 걸리거나 적게 걸릴 수 있다.

따라서, 이러한 에이전트의 이주 시간 증가 문제를 해결하기 위해 두 가지 방법을 사용한다. 먼저, 이동 에이전트의 크기에 따른 네트워크 부하 문제는 에이전트 작업 실행 모듈의 분리 및 Locker 패턴[15]을 응용한 푸시 에이전트의 생성을 통해 해결한다. 이는 기존의 이동 에이전트를 CORBA의 분산 객체 형태에 적용시켜 작업 실행 모듈 가진 CORBA 구현 객체는 푸시 에이전트 객체로, 호출 모듈을 가진 CORBA 클라이언트 객체는 이동성 객체인 이동 에이전트 객체로 구성하고, 에이전트가 처리된 결과를 가지고 다음 호스트로 이주하는 대신 푸시 에이전트 객체가 처리 결과를 사용자에게 독자적으로 전달하도록 하여 실행 시의 에이전트 크기를 일정하게 유지시킨다. 한편, 네트워크 트래픽으로 인한 이동 에이전트의 이주 지연 문제는 Ping을 이용한 트래픽 감지를 통해 최적의 이주 경로 탐색 및 조정 기법을 적용하여 해결한다.

3.2 최적 이주 경로 탐색 및 조정 기법

최적 이주 경로 탐색 및 조정 방법은 이동 에이전트의 이주 시간을 단축시키기 위한 방법으로, 에이전트가 이주할 노드들 중 최소의 네트워크 소요 시간을 갖는 경로를 선택하는 것이다. 즉, Ping을 이용하여 네트워크의 부하 발생 요인인 트래픽을 측정한 후 과다한 트래픽이 발생하는 경로는 회피하고 최소의 네트워크 지연을 갖는 경로를 선택하여 보다 효율적인 작업 처리가 이루어지도록 지원하기 위한 방식이다. 이러한 최적 이주 경로 탐색 및 조정 방식의 수행 과정은 다음 그림 2와 같다.

(1) 이동 에이전트는 사용자로부터 입력받은 검색 키워드를 이용하여 MAFFinder에 등록된 모든 SPA(Server Push Agent) 객체들 중 해당 정보 서비스를 지원할 수 있는 SPA 객체 참조자 리스트(이하 라우팅 테이블)를 전달받는다. 이 때, SPA 객체 참조자 리스트는 검색 적중률이 높은 순으로 되어 있다.

(2) 이동 에이전트는 검색 적중률이 우선인 라우팅 테이블 내의 SPA 객체 참조자를 이용하여 모든 원격 노드에 Ping을 실행한다.

(3) Ping을 실행하여 얻은 정보들 중 각 노드 간 패킷의 평균 왕복 시간인 Average 값과 패킷의 분실 수인 Lost 값을 추출한다.

(4) 각 노드까지의 경로들 중 패킷의 평균 왕복 시간과 패킷의 분실 수가 최소인 노드 경로를 찾기 위해 추출된 값들을 비교한다.

(5) 추출된 값을 통해 얻어진 최적의 노드 경로가 라우팅 테이블에서 최상위의 SPA 객체 참조자에 해당하는 노드 경로인지 검사한다.

(6) 만약, 최상위의 SPA 객체 참조자에 해당하는 노드까지의 경로가 최적이 아닌 경우, 패킷 왕복 시간과 패킷 분실 수의 값이 최소인 노드 경로들 순으로 라우팅 테이블을 재구성한다. 이 때, 패킷의 평균 왕복 시간 및 분실된 패킷의 수를 이용한 값 비교 시 패킷 분실 수가 적은 것을 우선으로 한다.

(7) 이동 에이전트는 라우팅 테이블의 최상위 노드를 선택하여 에이전트를 이주시킨다.

그림 2. 최적 경로 탐색 및 조정 방식의 수행과정
Fig 2. Performing Procedure of the Optimal Path Search and Coordination Methods

다음 그림 3은 최적의 이주 경로 탐색 및 경로 조정 방식을 수행하는 과정에 대한 예이다.

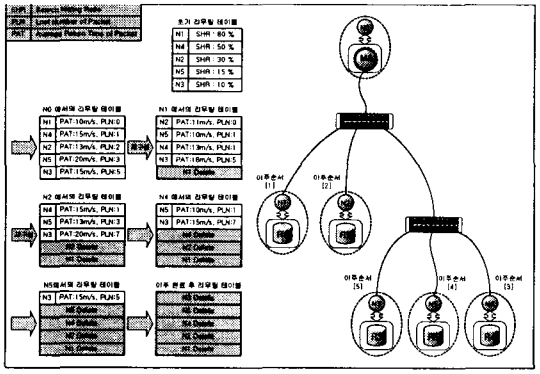

그림 3. 최적 경로 탐색 및 조정 방식의 수행과정 예
Fig 3. Performing Procedure Example of the Optimal Path Search and Coordination Methods

3.3 노드 장애에 따른 노드 재지정 기법

이동 에이전트 시스템에서는 라우팅 테이블에 따라 이동 에이전트가 이주를 수행할 때, 임의의 목적 노드까지의 경로가 외부 요인으로 인해 결손되어 에이전트가 이주할 수 없거나 혹은 목적 노드의 장애

로 인해 이미 이주하여 실행 중인 에이전트가 무한 대기나 고아 상태, 혹은 파괴될 수 있다. 노드 재지정 방법은 노드 장애로 인해 이미 이주하여 실행중인 에이전트가 무한 대기 혹은 고아 상태에 빠지거나 쓰레기로 처리되는 경우에 적절하게 대처하기 위한 방식이다. 이는 장애가 발생한 노드의 이전 노드에서 새로운 이주 노드를 재지정하여 에이전트를 이주시킴으로써 에이전트의 이주 신뢰성을 보장한다. 다음 그림 4는 노드 재지정 방식의 수행 과정이다.

(1) 이동 에이전트는 이미 이주하여 작업을 수행한 A 노드로부터 라우팅 테이블에서 최적 이주 노드인 B 노드로 이주 한다. 이 때, B 노드로 이주하는 이동 에이전트는 자신을 복제하여 A 노드에 남겨둔다.

(2) A 노드에 복제되어 남겨진 이동 에이전트는 B 노드로부터 복제 에이전트 삭제 신호가 수신될 때까지 일정한 시간 동안 대기한다.

(3) 만약, B 노드에 장애가 발생하여 A 노드로부터 이주한 에이전트가 고아 상태에 빠지거나 파괴되었을 경우, A 노드의 복제 에이전트는 정해진 대기 시간이 지나면 B 노드에서 장애가 발생했다고 판단하고 라우팅 테이블의 우선 순위에 따라 B 노드가 아닌 다음의 최적 노드인 C 노드로 에이전트를 재이주시킨다.

(4) B 노드에서 장애가 발생하지 않고 이주된 이동 에이전트가 주어진 작업을 완전하게 처리하면, B 노드의 이동 에이전트는 다음 노드로 이주하기 전 A 노드에 남겨놓은 복제 에이전트를 삭제하도록 신호를 보낸다.

(5) A 노드는 B 노드의 이동 에이전트로부터 복제 에이전트의 삭제 신호를 수신하고 남겨진 에이전트 객체를 삭제한다.

그림 4. 노드 재지정 방식의 수행과정
Fig 4. Performing Procedure of Node Reassignment Methods

다음 그림 5는 이러한 노드 재지정 방식의 수행 과정에 대한 예이다.

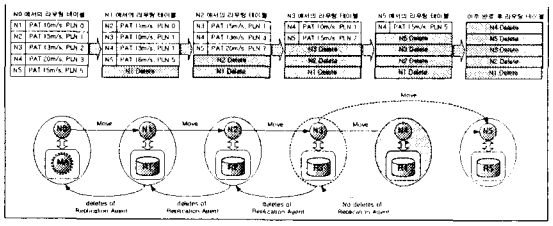

그림 5. 노드 재지정 방식의 수행과정 예
Fig 5. Performing Procedure Example of Node Reassignment Methods

IV. 지능형 경로 탐색 이동 에이전트 모델 설계

본 장에서는 2.2절의 멀티 에이전트 모델에서 변형된 MAFFinder와 이동 및 푸시 에이전트를 이용하여 효율적으로 작업을 처리하는 지능형 경로 탐색 이동 에이전트 모델을 설계한다.

4.1 이동 에이전트 모델

이동 에이전트 모델은 네트워크 트래픽 감지를 통해 현재의 노드에서 에이전트 라우팅 테이블의 각 목적 노드들 중 이주 경로가 최적인 노드를 판단하여 이주를 수행함으로써 작업 처리 시 소요되는 시간 비용을 최소화한 에이전트 프레임웍이다. 제안된 모델은 기존의 사용자에 의한 수동적 라우팅 스케줄 지정 방식에서 탈피하여 그림 6에서의 MAFFinger를 이용한 능동적 라우팅 스케줄 지정 방식을 사용함으로써 이동 에이전트가 과다한 트래픽 발생 경로를 회피하여 최적 경로를 통해 이주를 수행할 수 있도록 지원한다. 다음 그림 6에서 보인 이동 에이전트 모델의 통신 수행 과정은 4.2 절에서 설명한다.

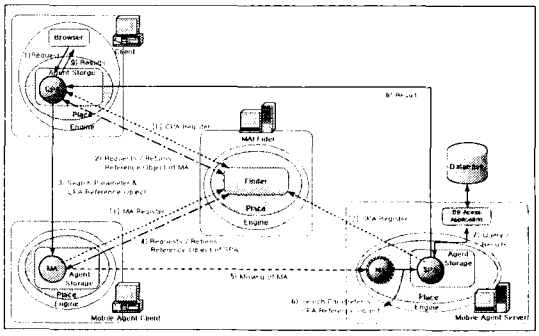

그림 6. 지능형 경로 탐색 이동 에이전트 모델과 통신 과정
Fig 6. Mobile Agent Model Supporting the Intelligent Path Search and It's Communicating Procedure

4.2 이동 에이전트 모델의 구성 요소

4.2.1 이동 에이전트 플랫폼

이동 에이전트 플랫폼(Mobile Agent Platform)은 이동 에이전트 모델내 이동 에이전트와 푸시 에이전트들이 개별적인 작업을 수행할 수 있도록 기본 환

경을 제공하는 것으로, 그림 7과 같이 각 에이전트들의 관리 및 제어를 위해 통신 관리자(Communication Manager), 에이전트 관리자(Agent Manager), 플레이스(Place), 에이전트 모니터(Agent Monitor), 에이전트 저장소(Agent Repository)로 구성된다.

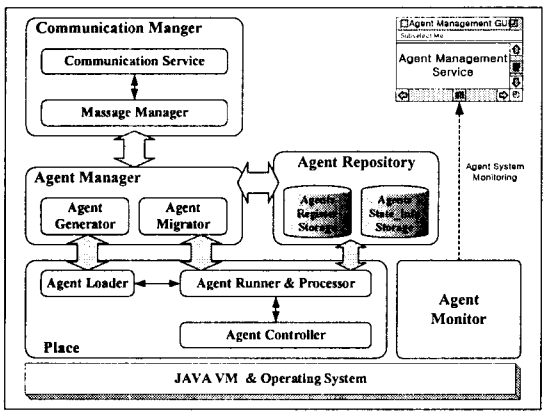

그림 7. 이동 에이전트 플랫폼
Fig 7. Mobile Agent Platform

4.2.2 이동 에이전트

이동 에이전트는 사용자로부터 부여받은 작업을 실행하고 다음 대상 목적 노드로 이주하는 이동성 객체이다. 이동 에이전트 플랫폼의 에이전트 생성기(Agent Generator)에 의해 복제된 이동 에이전트는 목적 노드로의 이주를 위해 MAFFinder에 등록된 SPA들 중 사용자의 요구를 처리할 수 있는 SPA에 대한 객체 참조자 리스트를 획득하여 이에 따라 노드 간을 이주하게 된다. 이 때, 최적의 이주 경로를 탐색하기 위해 SPA 객체 참조자 리스트에 포함된 각 노드까지의 경로에 대한 네트워크 트래픽을 검사하여 최소의 네트워크 지연시간을 갖는 경로를 선택한다. 만약, 선택된 경로로 이주한 이동 에이전트가 노드 장애로 인해 무한 대기에 빠졌거나 파괴되었을 경우, 원래의 이동 에이전트는 노드 재지정 모듈을 사용하여 장애가 발생한 노드의 다음 우선 순위 노드로 에이전트를 이주시킨다. 다음 그림 8은 이러한 이동 에이전트의 구조이다.

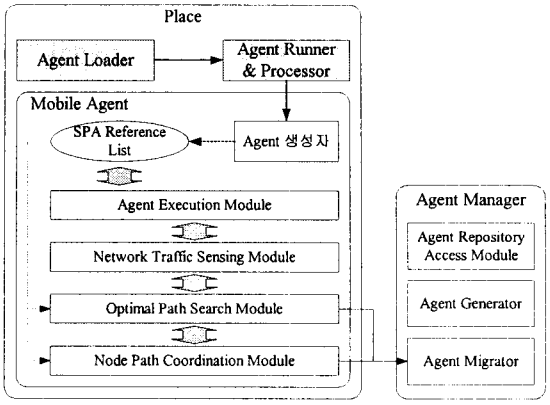

그림 8. 이동 에이전트 구조
Fig 8. The Structure of Mobile Agent

4.2.3 푸시 에이전트

제안된 푸시 에이전트는 기존의 푸시 에이전트가 클라이언트에 지속적인 정보를 제공함으로써 네트워크 트래픽을 가중시키는 문제를 해결하기 위해서 그림 9에서 보여진 것과 같이 네트워크 트래픽 감지를 이용하여 데이터 전송을 지연시키거나 최적 전송 경로 탐색을 통해 선택된 경로로 데이터를 전송하는 방식을 사용한다. 즉, 네트워크 트래픽을 감지하여 특정 시점에서 트래픽이 증가했을 경우 실시간적 정보 서비스를 지연시키고, 트래픽이 감소했을 경우 이를 사용자에 서비스하도록 함으로써, 한정된 대역폭의 네트워크 환경에서 안정적이고 신뢰할 수 있는 정보 서비스를 제공할 수 있도록 한다.

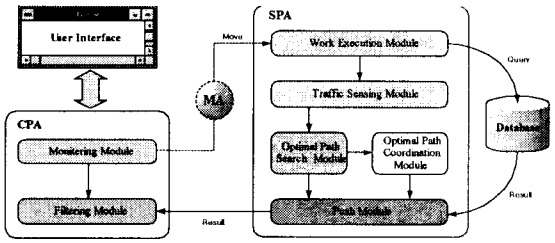

그림 9. 푸시 에이전트 구조
Fig 9. The Structure of Push Agent

4.2.4 MAFFinder

MAFFinder는 에이전트 이주 노드의 위치 투명성 및 초기 라우팅 스케줄을 지정하기 위해 각 에이전

트 시스템 관련 객체들과 사용자가 요구하는 검색 키워드를 관리하는 데몬(Deamon) 객체이다. MAFFinder는 그림 10과 같이 각 객체 정보들을 관리하기 위해 네이밍 서비스(Naming Service)별로 스레드(Thread)를 할당하여 각 스레드별 네임 스페이스(Name Space)를 생성한 후, 네임 스페이스 내의 메타 데이터 테이블에 이동 및 푸시 에이전트, 플레이스, 이동 에이전트 시스템 등의 객체를 등록한다. 또한, 정보 검색에 사용되는 키워드들을 저장하기 위해 별도의 네임 스페이스를 생성한 후 등록된 각 에이전트 이름과 객체 참조자, 계층적 검색 키워드 등의 정보를 저장 및 관리한다.

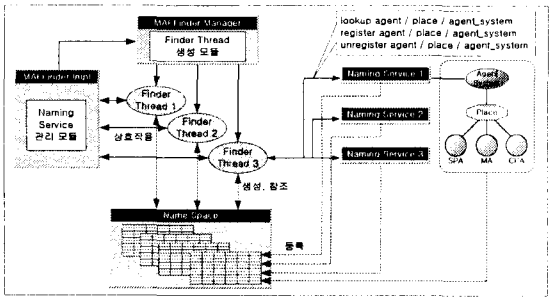

그림 10. MAFFinder 구조
Fig 10. The Structure of MAFFinder

4.3 이동 에이전트 모델의 통신 수행 과정

4.1절의 그림 6은 이동 에이전트 모델에서의 각 개별 에이전트들 간의 통신 과정이다. 그림 6에서 CPA는 클라이언트 브라우저(Client Browser)를 통해 사용자로부터 정보 검색을 위한 키워드를 입력받게 되면, 모니터링 모듈에서 이를 감지하여 MAFFinder로부터 MA의 객체 참조자(Object Reference)를 획득한다. 획득한 MA의 객체 참조자를 통해 MA에 검색 키워드와 CPA 객체 참조자가 전달되면, 이동 에이전트 클라이언트(Mobile Agent Client)는 해당 작업에 대한 MA를 생성(or 복제)하고, 검색 키워드를 이용해 해당 작업을 처리할 수 있는 SPA의 객체 참조자를 획득한 후 생성된 MA를 이주시킨다. 이 때, MA는 네트워크 트래픽 감지를 이용한 최적 이주 경로 탐색을 통해 이주 시간을 최소화하게 된다. 이주한 MA가 SPA의 작업 실행 모듈에 검색 키워드와 CPA 객체 참조자를 전달하게 되면, SPA는 데이터베이스를 검색하여 결과를 반환 받는다. 한편, SPA를 호출한 MA는 또다른 SPA로 이주하게 된다. SPA는 또한 검색 키워드를 통해 데이터베이스를 검색하는 동시에 네트워크 트래픽을 감지하여 검색 정보의 전송을 위한 최적 전송 경로를 탐색한다. 데이터베이스로부터 반환된 결과를 SPA는 탐색에 의해 선택된 전송 경로를 통해 CPA 객체 참조자를 이용하여 이를 전달하게 된다. CPA는 SPA로부터 검색 정보가 전달되면, 필터링을 수행하여 중복 데이터를 제거한 후 이를 사용자에게 보여준다.

V. 결론 및 향후 연구방향

본 논문에서는 네트워크 트래픽 감지를 이용한 이주 노드들에 대한 최적 경로 탐색을 통해 사용자 요구 처리에 있어 최소의 순회 이주 시간을 가지는 CORBA 기반의 이동 에이전트 모델을 설계하였다. 제안된 모델은 기존 이동 에이전트 시스템들의 에이전트 크기 증가로 인한 네트워크 부하 문제를 해결하기 위해, CORBA의 분산 객체에 기반하여 기존 이동 에이전트를 새로운 이동 에이전트와 푸시 에이전트로 분리하여 설계하였고, 사용자에 의한 수동적 라우팅 스케줄 지정이 아닌 능동적 라우팅 스케줄 지정이 가능하도록 MAFFinder를 제시하였다. 또한, 에이전트 이주 시 최소의 네트워크 소요 시간과 이주 신뢰성을 보장하도록 이동 에이전트의 이주정책으로서, 네트워크 트래픽 감지를 통한 최적 이주 경로 탐색 및 조정 기법과 노드 재지정 기법을 제시하였다. 이러한 노드 간 이주 기법들을 통해 보다 빠르고 안정적인 이동 에이전트의 이주를 수행함으로써 효과적으로 사용자 요구 사항을 처리하여 이를 서비스할 수 있다.

향후 연구방향은 제안된 최적 이주 경로 탐색 및 조정 기법과 노드 재지정 기법들에 대한 알고리즘 개발이 요구되며, 이러한 노드 간 이주 기법을 이용하여 보다 효율적으로 작업을 처리하는 이동 에이전트 모델의 개발이 필요하다.

참고문헌

[1] K.A. Baharat, L. Cardelli, "Migratory Applications", Proceedings of the 8th annual ACM symposium on User interface and software technology, November 1995.

[2] J. Vitek and Christian Tschudin, "Mobile Object Systems : Towards the Programmable Internet", Springer-Verlag, April 1997.

[3] OMG, "Agent Technology Green Paper", Agent Platform Special Interest Group, http://www.objs.com/agent/index.html, 2000.

[4] Robert S.G, "Agent Tcl : A flexible and secure mobile-agent system", TR98-327, Dartmouth Col. June 1997.

[5] J. Baumann, "A Protocol for Orphan Detection and Termination in Mobile Agent Systems", TR-1997-09, Stuttgart Univ. 1997.

[6] D.B.Lange, M.Oshima, "Programming and deploying Java Mobile Agents with Aglets", Addison Wesley Press, 1998.

[7] B.Venners, "The Architecture of Aglets", JavaWorld, http:// www.javaworld.com/javaworld/jw-04-1997/jw-04-ood.html, 1997.

[8] S. Choy, T.Magedanz, "Grasshopper Technical Overview", IKV++ GmbH, 1999.

[9] IKV++, "A CORBA environment supporting Mobile Agent", IKV++ GmbH, 1999.

[10] ObjectSpace, "ObjectSpace Voyager, GeneralMagic Odyssey, IBM Aglets : A Comparison", VoyagerTM, 1997.

[11] 전병국, 최영근, "이동 에이전트를 위한 효율적인 이주 정책 설계 및 구현", 한국정보처리학회 논문지, 제6권, 제7호, pp.1770- 1776, 1999.

[12] Durfee, E. H., Lesser, V. R. and Corkill, D., "Coherent Cooperation among Communicating Problem Solvers", IEEE Transactions on Computers C-36(11), pp.1275-1291, 1987.

[13] Gasser, L. and Bond, A. H., "Reading in Distributed Artificial Intelligence", San Mateo, CA:Morgan Kaufmann, 1988.

[14] 김광종, 고현, 이연식, "분산 정보 서비스를 위한 CORBA 기반의 멀티 에이전트 모델 설계", 한국정보처리학회 학술발표논문집(상), 제9권, 제1호, pp.327-330, 2002.

[15] A. Yariv, D. B. Lange, Agent Design Pat-terns : Elements of Agent Application Design, Second International Conference on Autonomous Agents(Agents 98), 1998.

저자소개

고 현(Hyun Ko)

2001년 군산대학교 컴퓨터정보과학과 이학사

2001년~현재 군산대학교 컴퓨터정보과학과 석사과정

※관심분야: 에이전트, 분산객체시스템, 멀티미디어 데이터베이스,

김광종(Kwang-Jong Kim)

1993년 군산대학교 컴퓨터정보과학과 이학사

1999년 군산대학교 컴퓨터정보과학과 이학석사

2001년~현재 군산대학교 컴퓨터정보과학과 박사과정

※관심분야: 에이전트, 분산객체시스템, 능동 데이터베이스

이연식(Yon-Sik Lee)

1982년 전남대학교 전자계산학과 이학사

1984년 전남대학교 전자계산학 과 이학석사

1994년 전북대학교 전산응용공학과 공학박사

1995년~1997년 군산대학교 교무처장

1997년~1998년 University of Missouri 교환교 수

1999년~2001년 군산대학교 전자계산 소장

1998년~현재 군산대학교 컴퓨터정보과학과 교수

※관심분야: 번역기 이론, 객체지향시스템, 능동시스템, 지능형 에이전트