

분산 정보 검색의 성능 향상을 위한 이동 에이전트의 다중 복제 모델 설계 및 구현

김광종*, 김영자, 이연식
군산대학교 컴퓨터정보과학과
e-mail : kkjkim@kunsan.ac.kr

Design and Implementation of the Multi-Replication Model of Mobile Agent for Improving Performance of Distributed Information Retrieval

Kwangjong Kim, Youngja Kim, Yonsik Lee
Dept. of Computer Information Science, Kunsan National University

요 약

현재 다양한 연구들이 진행되고 있는 이동 에이전트는 통신망 점유와 과부하를 효율적으로 감소시켜 여러 문제점들을 해결하고 있으나, 이동 에이전트의 이동성 강조에 따른 지능성 결여와 다수의 이동 에이전트 시스템들로 이주시 발생하는 지연시간 및 호스트의 결점이나 장애 등으로 인하여 이동 에이전트가 무한 대기상태나 쓰레기로 처리될 수 있는 문제점들을 해결하기 위한 새로운 대안이 요구된다. 따라서 본 논문에서는 분산 환경에서 정보 검색의 성능을 향상시키기 위하여 네이밍 에이전트 기술을 이용하여 사용자의 검색 키워드에 대한 객체 참조자를 획득하고, 획득되어진 객체 참조자의 수 만큼 이동 에이전트 객체를 다중 복제하여 복제된 다수의 에이전트 객체를 여러 이동 에이전트 시스템으로 분산시킴으로써, 이주시 발생하는 지연시간을 제거하고 쓰레기로 처리될 수 있는 이동 에이전트들의 수를 줄임으로써 분산 정보 검색 서비스의 성능을 향상시킬 수 있는 모델을 설계 및 구현한다.

1. 서론

최근 컴퓨터와 통신 기술이 발달함에 따른 정보량의 증대는 사용자가 개인의 선호도와 목적에 맞는 정보를 획득하는데 어려움을 주고 있으며 사용자는 자신이 원하는 데이터에 대한 위치와 시기를 미리 알고 있어야 하고, 관련 사이트를 위해 네트워크를 검색하고 소프트웨어의 업데이트를 위해 이미 알려진 사이트들을 조사하는데 많은 시간을 보내야 한다. 또한 많은 양의 정보 서비스들을 지원하므로 발생하는 네트워크 트래픽의 과부하는 정보 서비스 중 하나의 커다란 문제로 부각되어 분산 환경의 새로운 패러다임인 이동 에이전트를 이용하여 해결하고 있으나 기존의 이동 에이전트 들은 에이전트의 특성 중 이동성만을 강조하여 지능성이 결여되었고, 이동 에이전트가 통신망에 연동된 다수의 이동 에이전트 시스템들로 이주시 발생하는 지연시간 문제와 호스트의 결점이나 장애 등으로 인하여 이동 에이전트가 무한 대기상태에 빠져 쓰레기로 처리 될 수 있는 문제점들을 가지고 있다[1,2]. 따라서 이러한 이동 에이전트의 지능성 및 노드 이동에 따른 문제점들의 새로운 대안이 요구된다. 본 논문에서는 다수의 이동 에이전트 시스템에 분산되어 있는 구현 객체가 네이밍 에이전트에 등록되어 유지 관리되므로 사용자의 요구 사항에 대한 검색 키워드를 네이밍 에이전트의 메타데이터에서 일치하는 키워드의 구현 객체 수와 위치를 확인한 후 객체 참조자 정보와 수를 획득, 획득되어진 객체 참조자의 수 만큼 이동 에이전트 객체를 다중 복제하여 복제된 다수의 에이전트 객체를 여러 이동 에이전트 시스템으로 분산시킴으로써 이주

시 지연시간을 제거하고, 쓰레기로 처리될 수 있는 이동 에이전트들의 수를 줄임으로써 분산 정보 검색 서비스의 성능을 향상시킬 수 있는 모델을 설계 구현한다.

본 논문의 구성은 2 장에서는 제안된 객체의 복제 모델을 설계하고 3 장에서는 제안된 복제 모델의 구현 및 수행과정을 보이며 마지막으로 4 장에서는 결론 및 향후 연구에 대해 기술한다.

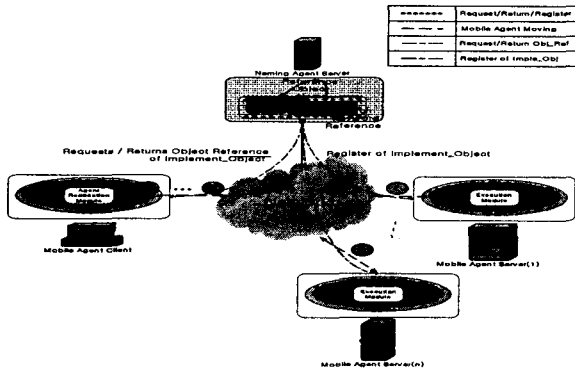
2. 이동 에이전트의 복제 모델 설계

본 연구에서는 이동 에이전트의 객체를 다중 복제하여 다수의 이동 에이전트 시스템들로 이주시 발생하는 지연시간이나, 호스트의 결점, 장애 등으로 인하여 이동 에이전트가 무한 대기상태이거나 쓰레기로 처리되는 에이전트의 수를 줄이고자 하는 것이다. 즉 CORBA 기반의 네이밍 에이전트의 기술을 이용하여 네이밍 에이전트 에 등록된 분산된 서버의 구현 객체의 객체 참조자를 획득하고 이동 에이전트 객체를 복제하여 분산된 서버로 이주시킴으로써 사용자가 원하는 결과를 검색하고 결과를 반환한다.

2.1 제안된 복제 모델의 구조

이동 에이전트 복제 모델의 구조는 [그림 1]와 같이 이동 에이전트 클라이언트, 네이밍 에이전트 서버, 분산된 서버들로 구성되어 있다. 이동 에이전트 클라이언트는 사용자의 검색 키워드를 입력 받고, 입력 받은 키워드는 네이밍 에이

전트의 메타데이터를 검색하여 등록된 구현 객체와 일치하는 키워드에 대한 객체 참조자의 정보를 획득하고 획득된 객체 참조자의 수와 정보를 기반으로 이동 에이전트를 복제한다. 네이밍 에이전트는 분산된 서버들로부터 구현 객체의 정보를 관리한다. 분산된 서버들은 네이밍 에이전트에게 자신들이 서비스할 객체의 정보를 등록하며 이동 에이전트를 클라이언트에서 복제된 이동 에이전트가 도착하면 구현 객체의 일부와 연동을 통해 사용자가 요청한 정보를 검색 후 결과를 반환한다.

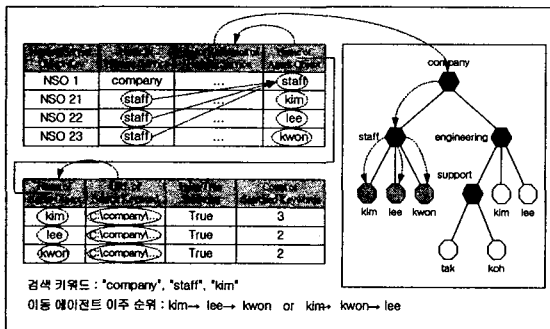


[그림 1] 이동 에이전트 복제 모델의 구조

2.2 네이밍 에이전트의 메타데이터 구조

CORBA 기반의 분산 시스템 환경에서 클라이언트가 구현 객체를 호출하기 위해서는 구현 객체의 객체 참조자(Object Reference)가 필요하다. 네이밍 서비스는 분산된 객체의 객체 참조자를 쉽게 얻을 수 있도록 객체 참조자를 한곳에 모아 이용 가능하게 해주는 CORBA의 표준 방식이다[1,2,3,4]. 이를 통해 이종의 ORB 제품간 상호 운용성을 제공하고[8], 분산 시스템 환경에서의 위치 투명성을 보장한다. 네이밍 서비스의 수행 방식은 CORBA 서버가 네이밍 서버에 구현 객체와 구현 객체의 이름을 등록하고, 클라이언트가 네이밍 서버에 접속하여 구현 객체의 이름을 통해 분산되어 존재하는 구현 객체들 중 해당 구현 객체의 객체 참조자를 얻어 구현 객체를 호출하는 방식이다.

검색 키워드를 통한 접근 방식의 메타데이터는 에이전트 이름과 검색 키워드를 포함한 파일의 URL, 검색 키워드의 서브스트링 검색 여부, 검색된 키워드로 구성된다.



[그림 2] 검색 키워드를 통한 객체 참조자 획득 과정

이 방법은 [그림 2]과 같이 먼저, 네이밍 에이전트에 사용자가 입력한 검색 키워드만을 전달하고, 메타데이터 테이블 내에 이미 등록된 모든 검색 키워드 URL을 이용하여 전달된 검색 키워드와 비교한 후 일치하는 키워드를 가진 모든 객체 참조자의 정보를 추출한다. 이렇게 추출된 객체 참조자의 정보는 클라이언트로 전달되며 전달된 객체 참조자는 수가 체크되며 체크된 후에는 전달된 객체 참조자의 수만큼 에이전트 객체의 복제를 시작하게 된다.

2.2 제안된 복제 모델의 알고리즘

아래의 복제 알고리즘은 두 부분으로 나뉘어진다. 1 - 10은 네이밍 에이전트로부터 사용자의 키워드와 일치하는 객체 참조자 정보와 수를 획득하여 원본 에이전트 객체를 생성하고 각각의 원본 에이전트의 속성 값을 True로 설정하며 또한 원본 에이전트 객체의 컬렉션을 생성한다. 11 - 18은 생성된 원본 에이전트 객체의 컬렉션과 정보를 얻어와 복제 에이전트를 생성하며 생성된 객체의 수가 원본 에이전트 객체의 수와 일치하면 객체의 복제를 중지하고 복제된 에이전트는 분산된 서버로 이주를 시작한다.

```

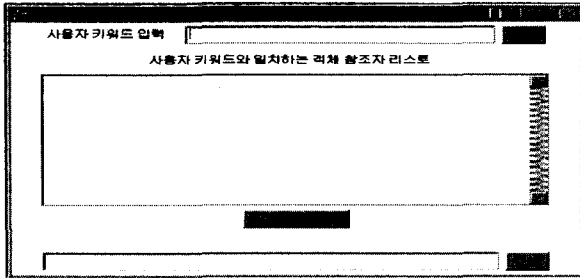
1: receive to the migration target object reference
   information
2: object reference information_num check
3: object_total_num = object reference information_num
4: original object creation_num = 0
5: Do{
6:   original agent object creation
7:   original agent object attribute = true
8:   collection creation for agent object replication
9:   original object creation_num = original object-
       creation_
       num + 1
10: } While( original object creation_num =
       object_total_num)
11: num_of_replicated = 0
12: While ( num_of_replicated < original object-
       total_num
       ) {
13:   replication agent object creation
14:   replication agent object attribute = false
15:   collection object write = replicated agent
16:   num_of_replicated = num_of_replicated + 1
       }
17: replication_stop
18: each replicated agent migration
    
```

3. 제안된 복제 모델 구현 및 수행과정

본 구현 및 실험은 네이밍 에이전트의 메타데이터에 분산된 서버들의 구현 객체가 저장되어 있다고 가정하고 사용자 키워드에 대한 객체 참조자 정보를 얻어 이동 에이전트 객체를 복제하여 복제된 이동 에이전트 객체를 각 분산된 서버로 전송하여 사용자의 키워드와 일치하는 결과를 검색한다.

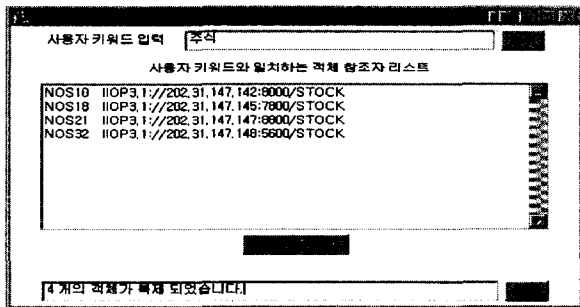
3.1 제안된 복제 모델의 사용자 인터페이스

다음 [그림 3]은 사용자의 키워드를 입력 받아 검색 버튼을 누르면 네이밍 에이전트에게 사용자의 키워드가 전달되어 네이밍 에이전트 메타데이터를 검색하여 사용자의 키워드와 일치하는 객체 참조자 정보의 결과를 보여준다.



[그림 3] 객체 복제를 위한 사용자 인터페이스

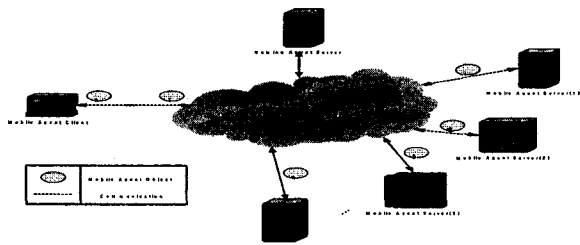
다음 [그림 4]는 사용자 키워드에 대한 네이밍 에이전트의 메타데이터를 검색한 결과이며, 객체 복제 실행 버튼을 누르면 객체 참조자의 수와 정보에 맞는 이동 에이전트 객체의 원본을 생성하고 복제하게 된다.



[그림 4] 메타데이터 검색 결과 및 객체 복제 실행

3.2 제안된 복제 모델의 수행과정

제안된 복제 모델의 수행과정은 위 [그림 4]에서와 같이 8 개의 객체 참조자가 검색되었으며, 총 8 개의 이동 에이전트 객체가 복제되었다. 이렇게 복제된 객체는 전송 버튼을 누름으로써 각기 분산된 서버로 이주하게 된다. 아래 [그림 5]는 복제된 이동 에이전트의 이주의 수행과정을 나타낸다.



[그림 5] 복제된 이동 에이전트 객체의 이주

4. 결론 및 향후 연구

본 논문에서는 이동 에이전트 시스템들로 이주시 발생하는 지연시간 문제, 혹은 호스트의 결점, 장애 등으로 인하여 이

동 에이전트가 무한 대기상태에 빠져 쓰레기로 처리될 수 있기에 새로운 대안이 요구 되었다. 그러므로 이동 에이전트 객체를 다중 복제하여 복제된 다수의 에이전트 객체가 여러 이동 에이전트 시스템으로 분산되어 이주시 지연시간을 제거할 수 있고, 쓰레기로 처리될 수 있는 이동 에이전트들의 오버헤드를 줄이는 모델을 CORBA 를 기반으로 네이밍 에이전트 기술을 이용하여 설계 및 구현하였다. 따라서 분산 정보 검색 시 원하는 위치에 바로 접근하여 사용자의 요구를 수용하기 때문에 보다 빠르고 효율적으로 서비스할 수 있다.

향후 연구로는 네이밍 에이전트에 등록되는 객체 참조자가 증가함으로써 1- n 개의 에이전트 객체를 복제하는데 있어서의 부하 및 메모리 문제와 분산된 정보의 수집에서 오는 필터링 방법 그리고 기존의 이동 에이전트를 이용한 정보 검색 시스템과의 성능평가가 요구된다.

참고문헌

- [1] K.A Bajarat, L. Cardelli, " Migratory Application", Proc. of the 8th Annual ACM Symposium UIS Tech., Nov. 1995.
- [2] J. Baumann, " A Protocol for Orphan Detection and Termination in Mobile Agent Systems", TR-1997-09, Stuttgart Univ., July, 1997.
- [3] ObjectSpace Voyager <http://www.objectspace.com>, 2000.
- [4] Bellavista, Antonio Corradi, Cesare Stefanelli, " A Mobile Agent Infrastructure for the Mobility Support", Proc. of the 2000 ACM symposium, ACM Press USA, pp.539-545, 2000.
- [5] OMG, " Agent Technology Green Paper", Agent Platform Special Interest Group, <http://www.objs.com/agent/index.html>, 2000.
- [6] M.Hansson, " Push Technology-The Next Big Thing?", " <http://www.tcm.hut.fi/Opinnot/Tik-10.350/Tehtavat/essays/push.html>", 1999.
- [7] Vn Anh Pham and Ahmed Karmouch, " Mobile of software Agents: An Overview," IEEE Communication Magazine, pp.26-37, 1998.
- [8] 전병국, 이근상, 최영국, " Java언어를 이용한 객체 이동 시스템 설계 및 구현", 정보처리논문지, 6 권, 1 호, Jan., 1999.
- [9] 전병국, 최영국, " 인트라넷상에서 자바 객체의 이동 시스템 설계 및 구현", 정보과학회논문지(C), April, 1999
- [10] 이상호, " 웹과 CORBA 의 연동을 위한 Naming Agent", 숭실대학교 대학원, 석사학위논문, 1998.
- [11] 이승율, " CORBA 환경에서 이동 에이전트 시스템의 설계 및 구현", 광운대학교 대학원, 석사학위 논문, 1999.
- [12] 김광중, 고현, 이연식, " 분산 정보 서비스를 위한 CORBA 기반의 멀티 에이전트 모델 설계", 정보처리학회추계학술발표논문집, 2002.
- [13] 김광중, 최은실, 이연식, "효율적 이동 에이전트를 위한 객체 다중 복제 기법", 멀티미디어학회추계학술발표논문집, 2002.