에이전트의 모듈 및 서비스객체 이동 설계

이승율, 이근상, 전병국, 최영근
광운대학교 컴퓨터과학과
원주대학 사무자동학과

Mobility Design of Agent module and Service Object

Seung-Yul Lee, Keun-Sang Yi, Byung-Kook Jeon, Young-Keun Choi
Dept. of Computer Science, Kwangwoon University
Dept. of Office Automation, Wonju Nat'l College

요약

최근들어 이기종의 컴퓨터들이 네트웍으로 연결되고, 분산처리화 되면서 다양한 분산기술들이 연구되고 있다. 이런 기술들의 일종으로 이동에이전트 시스템에 대한 개념이 만들어졌으며, 최근에는 이동에이전트 시스템에 대한 연구와 개발이 증가하고 있다. 하지만 에이전트에게 많은 기능을 부여할 경우 에이전트 모듈자체가 커져서 네트웍 트래픽을 증가시키는 단점이 있다. 본 논문에서는 CORBA객체와 에이전트간의 통신을 가능하게 하여 에이전트 기능의 일부를 CORBA객체로 구현하므로써 에이전트 모듈 크기의 문제점을 해결하였다. 또한 호스트에 서비스객체가 없는 경우 에이전트의 이동경로를 토대로 서비스 객체의 위치를 확인하여 이동시키는 것을 제안한다.

1. 서론

최근 컴퓨터 환경이 분산 환경으로 변화되면서 분산처리에 대한 기술들이 많이 연구되고 있으며, 최근에는 이런 기술들 중에서 이동에이전트 시스템이 활발히 연구,개발되고 있다. 이동 에이전트시스템은 에이전트(Agent)가 작업할 실행 모듈을 가지고 네트워크를 통해 자율적으로 에이전트 시스템들을 이동하면서 작업을 수행하는 시스템이다. 하지만 이런 이동에이전트시스템의 단점으로 에이전트가 수행할 작업이 많을 경우 실행모듈크기가 커지므로써 에이전트 이동시 네트웍에 트래픽이 증가한다는 문제점이 있다. 따라서 본 논문에서는 에이전트의 모듈 크기문제를 CORBA를 이용하여서 줄이는 방법을 제안한다. 이 방법은 에이전트가 수행하여야 하는 여러 작업을 CORBA객체로 구현하여서 에이전트는 단지 CORBA객체를 호출할 모듈만 가지고 이동하므로서 에이전트 모듈 크기를 줄일 수 있다.

2.관련 연구

2.1 CORBA (Common Object Request Broker Adapter)

CORBA는 이기종간의 분산환경에서 CORBA객체들간의 상호연결을 제공하는 분산처리 기술이다. CORBA객체들은 클라이언트와 서버객체로 나누어지며 각각의 객체들이 ORB(Object Request Broker)에 등록하고, 클라이언트는 서비스 요청을 ORB에 전달하고 다시 ORB는 서버객체에 전달한다. 서버객체는 서비스를 처리하여 결과값을 ORB에 전달하고 ORB는 다시 클라이언트에게 전달하므로서 분산처리를 하는 구조를 가진다.

2.2 이동에이전트 시스템 개요

에이전트 시스템은 에이전트를 생성,수행,소멸시키는 에이전트의 생명주기를 관리하는 시스템이며[1][2], 이동 에이전트시스템은 에이전트의 이동과 보안,인증이 추가된 형태이다. 이동 에이전트 시스템의 기본적인 구성요소는 에이전트와 플레이스(Place)이며[3], 에이전트는 네트웍을 이동하면서 자율적으로 주어진 일을 수행하는 실행 모듈이고, 플레이스는 에이전트 시스템 안에 존재하며, 에이전트 시스템이 에이전트를 거주, 실행을 시키는 요소이다.이동 에이전트 시스템의 전체적인 구성 요소들은 에이전트를 구현하는 프로그래밍 언어, 에이전트를 받아들이고 수행시키는 에이전트 시스템, 에이전트와 호스트의 보호를 위한 보안측면, 에이전트를 전송하기 위한 프로토콜 등으로 분류될 수 있다.

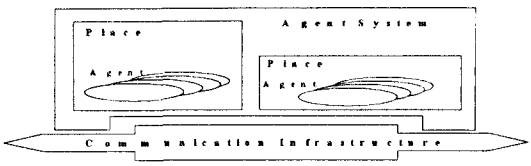

그림1. 이동에이전트 시스템 구조

3. 에이전트 모듈 및 서비스객체 이동 설계

기존의 이동 에이전트에서는 에이전트의 모듈 크기 문제 때문에 에이전트에게 많은 기능을 주지 못한다. 따라서 본 논문에서는 이동에이전트에서 에이전트 크기가 커지는 문제를 해결하기 위해서 CORBA를 사용하여 에이전트의 기능을 CORBA객체로 구현하고, 에이전트는 단지 CORBA객체의 호출 모듈만 가지게 하여 에이전트의 크기 문제를 해결하였다. 또한 에이전트 시스템이 있는 호스트 내에 에이전트에게 서비스할 객체가 없는 경우 에이전트에게 서비스를 할 CORBA객체를 찾아서 에이전트 시스템이 있는 호스트로 이동시키는 것을 제안한다.

3.1 에이전트와 서비스객체간의 통신

이동에이전트 시스템에서 여러 기능을 에이전트에게 부여할 경우 에이전트는 여러 가지 작업을 수행할 수 있으므로 유용하지만 에이전트 모듈자체의 크기가 커지므로 인해서 네트워크 트래픽이 증가하는 단점이 있다. 이에 따라 에이전트가 수행할 작업을 호스트애의 프로그램으로 구현하고 에이전트는 그 프로그램을 호출할 수 있는 모듈만 가지게 하여 에이전트의 모듈 크기를 줄일 수 있는 방법이 있을 수 있다. 하지만, 기존의 에이전트 시스템들이 에이전트와 호스트내의 프로그램간의 호출을 지원하지 않으며, 또한 표준적인 방식도 제시되지 않고 있다. 따라서 분산환경 표준인 CORBA를 사용하여 에이전트와 CORBA객체간의 호출을 제안한다. 이 경우 CORBA객체로 에이전트가 할 작업을 구현하여 에이전트가 CORBA객체와 표준적인 방법으로 호출할 수 있게 된다. 다음은 에이전트와 CORBA객체들간의 호출하는 알고리즘이다.

```
// CORBA객체와 연결한다.
CORBA_OBJECT = Agent.AgentToCORBA ( Repository
                ID,CORBAObjectName )
ObjectMethod = CORBA_OBJECT.request ( CORBA_OBJEC
          T.Method )
// CORBA객체에게 파라미터를 전달한다.
ObjectMethod.arg().string()
ObjectMethod.arg().int()
.....
// CORBA객체의 결과값 종류를 명시한다.
ObjectMethod.ReturnType(.... )
// CORBA객체에게 서비스 요구 메시지를 전다.
ObjectMethod.request()
// 원하는 타입의 리턴값형태로 CORBA객체의 결과값을 받는다.
... = ObjectMethod.getResult()
```

< 에이전트와 CORBA객체간의 통신알고리즘 >

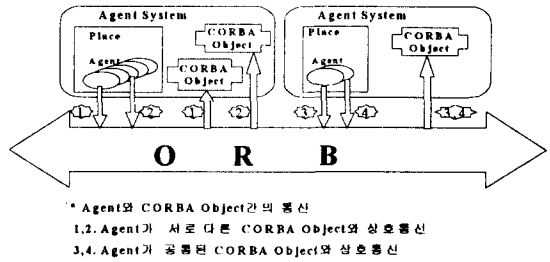

그림 2. 에이전트와 CORBA객체간의 통신

3.2 서비스객체의 이동

3.1에서 제시한 방식으로 에이전트가 할 일을 호스트안에 있는 CORBA객체로 구현하고 에이전트가 단지 CORBA객체를 호출할 모듈만 가지고 이동할 경우 다음과 같은 문제점이 있다. 첫째, 에이전트가 사용할 CORBA객체가 반드시 호스트내에 있어야 한다는 것이다. 둘째, 에이전트가 사용할 CORBA객체가 없는 경우 CORBA객체가 호스트로 이동할 시간동안 에이전트가 기다려야 한다는 것이다. 따라서 본 논문에서는 에이전트의 이동경로를 토대로 CORBA객체가 없는 에이전트시스템의 위치확인과 이동경로에 속하는 호스트에서 CORBA객체를 이동시키는 것을 제안한다.

3.2.1 Resource Manager

에이전트가 사용할 CORBA객체에 관한 관리를 하는 시스템적인 에이전트이다. 이 Resource Manager를 통해서 다른 에이전트 시스템과 통신하면서 각각의 호스트들이 가지고 있는 CORBA객체들에 관한 데이터를 얻어서 호스트들이 없는 객체에 관한 정보를 수집하며, 에이전트가 호스트로 이동하기 전에 미리 CORBA객체들을 근접한 호스트에서 이동시키도록 한다. 또한 에이전트가 이동시 다음 에이전트 시스템이 있는 호스트에 에이전트가 사용할 CORBA객체들이 있는지 체크하여 에이전트의 이동을 중지,실행시키는 역할을 한다.

3.2.2 Message Manager

에이전트와 Resource Manager간이나 Resource Manager과 Message Manager, 다른 이동에이전트시스템의 Message

Manager와 통신하여 메시지를 전달하는 역할을 한다.

다음은 에이전트가사용할 CORBA객체가 없을 경우 호스트로 CORBA객체를 이동시키는 알고리즘이다.

```
//Agent의 맨처음 에이전트시스템의 Resource Manager에게
   이동경로를 전달
IF FirstAgentSystem THEN
  SendMessageRM ( AgentSystemList )

// RM은 이동경로를 토대로 이동경로에 있는 에이전트 시
스템의 CORBA객체의 유무 체크
FOR AgentSystemList[ASN_MIN] TO AgentSystemList[ASN
_ MAX]
   NotObjectLists = SendMessageRMs (AgentSyst
                emList[ASN] ,ObjectNames)
//CORBA객체가 없을 경우 이동경로를 토대로 가장 가까운
호스트의 CORBA객체를 이동
ASLocation = FindCObjectLocation (NotObjectLists)
MoveCORBAObjectRM ( ASLocation, TargetAS )
ActionCORBAObject ( CORBAObjectName )
//에이전트가 이동할 호스트에 CORBA객체가 없을 경우 RM
이 에이전트에게 WAIT메시지 전달, 그후에 CORBA객체가 이
동된 경우 에이전트를 이동
WHILE ( Not exist NextRM_CORBAObjects ( ObjectsNames ) )
      {  AgentAction ( WAIT )  }
AgentMove( AgentSystemList[ASN_NEXT] )
```

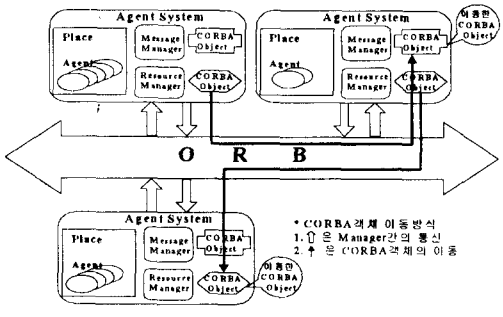

그림 3. CORBA객체의 이동방식

4. 이동에이전트 시스템의 수행과정

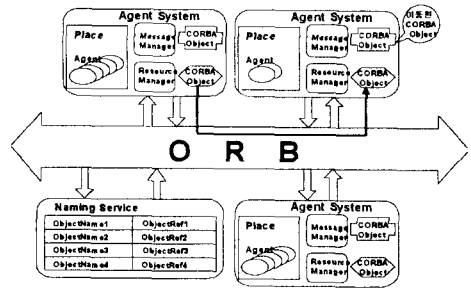

그림 4. 이동에이전트시스템의 수행과정

그림4는 본 논문에서 제시한 CMAS의 설계이다. 아래에 CMAS의 작업 수행과정을 제시한다.

1. 목적한 시스템으로 에이전트 이동
2. 에이전트의 이동경로를 RM에게 전달
3. RM이 다른 호스트의 RM을 통해서 CORBA객체에 대한 이동 여부 확인
4. RM을 통한 CORBA객체들의 이주
5. 에이전트가 모든 CORBA객체를 가지고 있는 에이전트 시스템으로 이동

5. 결론

최근 분산처리에 대한 연구가 활발해 지면서 이동에이전트에 대한 연구 및 개발이 많아 졌다. 하지만 이동에이전트에게 많은 기능을 부여시 에이전트 모듈이 커진다는 문제점이 있다. 따라서 따라서 본 논문에서는 CORBA객체와 에이전트간의 호출을 통해서 에이전트의 작업을 CORBA객체로 구현하여 에이전트 모듈 크기를 감소시켰다. 또한 CORBA객체가 호스트에 없을 경우 에이전트의 이동경로목록을 토대로 미리 CORBA객체를 이동시키는 방법을 제공한다.

6. 참고문헌

[1] 김평중, 윤석환, "이동 컴퓨팅을 위한 이동 에이전트 시스템",정보처리학회지 제4권 5호, 1997년 9월

[2] Crystaliz Inc,General Magic Inc, The Open Group, "Mobile Agent System Interoperability Facilities Specification" ,OMG TC Document, 1997년 11월

[3] Danny B.Lange, Mitsuru Oshima, "Programming and Deploying java Mobile Agents with Aglets",ADDISON-WESLEY, 1998