

이질적인 분산시스템에서 객체공유를 위한 협력모델

윤인숙*, 양해권*, 이재완*

*군산대학교 정보통신공학과

A Cooperation Model for Object Sharing in Heterogeneous Distributed Systems

In-Suk Yun*, Hae-Kwon Yang*, Jae-Wan Lee*

* Dept. of Telecommunication Eng., Kunsan National Univ.

E-mail : jwlee@knusun1.kunsan.ac.kr

요 약

분산 객체지향 시스템에서 시스템들의 규모가 커짐에 따른 분산 객체들의 관리의 복잡성으로 인하여 분산 객체들간의 효율적인 상호협력 기법이 필요하게 되었다. 따라서 본 논문에서는 이기종 객체 시스템들간의 접속을 위한 트레이딩 서비스를 단순협력, 단순교섭, 그리고 연합으로 확장하여 트레이더간의 상호협력 기법을 제시함으로써 객체들간의 효과적인 접속과 클라이언트에게 보다 나은 서비스 속도 제공 및 선택의 폭을 넓히고자 하였다.

I. 서 론

오늘날 분산 컴퓨팅 환경의 분산 시스템을 구성하는 시스템들과 네트워크 환경 및 특성은 광범위한 네트워크를 기반으로 한 분산처리 기술에 객체 지향 기술의 도입으로 인하여 분산 시스템의 크기가 거대해지고 응용 서비스 기능들이 보다 다양해지고 있으며, 이들간에 많은 상호협력이 요구되고 있다.

기존의 프로그래머가 분산 시스템을 설계할 때, 네트워크 상의 상호작용을 위해 자신의 시스템 사양에 맞는 프로토콜과 인터페이스를 정의하여 설계하고, 개발하는데 많은 시간이 소비되었다. 그 뿐만 아니라, 서로 다른 개발자에 의해 개발된 분산 시스템들은 이식성(portable)이나 호환성(compatible) 그리고 확장성(scalability)을 보장하지 못했다. 이러한 문제점을 해결하고자, 분산 처리 기술에 객체 지향 기법을 접목시키게 됐다. 이에 따르는 대표적인 표준안으로는 분산 오브젝트 컴퓨팅의 표준안인 OMG(Object Management Group)와 차세대 정보 통신망 구조를 정립하기 위한 표준안인 TINA-C(Telecommunications Information Networking Architecture-Consortium)가 거론 되어왔다. [1,2]

본 논문에서는 분산시스템 규모의 거대성, 서비스의 다양성과 관리의 복잡성, 그리고 객체들간 상호작용의 필요성에 따른 문제점을 해결하기 위하여 각각의 영역에서 독립적으로 운용되는 트레

이더들을 연결한 트레이딩 상호협력에 목적을 두고, 서로 다른 영역에 위치한 독립적인 트레이더들의 서비스 타입, 서비스들의 수와 서비스 요청 빈도수에 따라 협력모델들을 단순협력(Light Weight Cooperation), 단순교섭(Simple Negotiation), 그리고 연합(Federation)으로 분류하고, 이 협력모델은 하나의 호스트 관리자(host manager)와 여러 개의 지역 트레이더(local trader)를 묶어 하나의 트레이딩 도메인을 형성하고 호스트 트레이더간에 협력이 이루어진다는 모델에 기초한다.[3,4,5,6,7] 제안된 협력에 따라 트레이더간의 접속 과정을 설계함으로써 분산시스템을 이루는 객체들의 접속을 효과적으로 할 수 있으며, 클라이언트에게 보다 나은 서비스 선택의 폭을 넓힐 수 있도록 하였다.

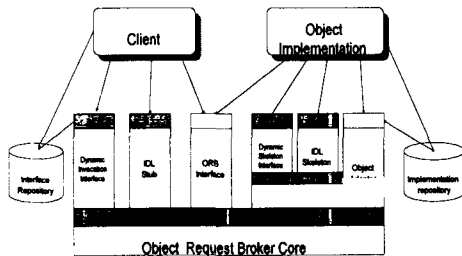
본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 CORBA(Common Object Request Broker Architecture)와 상호협력의 주체로서 운용되는 트레이더, 그리고 트레이더 설비 및 목적에 대해서 알아보고 3장에서는 다른 도메인 상의 트레이더들을 연결시켜주는 기능을 갖는 호스트 관리자의 구조와 기능을 살펴본다. 4장에서는 트레이더간의 상호협력의 필요성과 트레이딩 상호협력 모델 그리고 호스트 트레이더를 통한 서비스 수행을 설명한다. 끝으로 5장에서는 결론으로서 앞으로 연구해야 할 내용에 대해서 기술한다.

II. 본 론

1. 관련연구

1.1 CORBA(Common Object Request Broker Architecture)

분산환경에서의 시스템 구축은 이질성으로 인해 응용 소프트웨어 개발에 많은 어려움이 있었고 구현하는 언어에 대해서도 중속적이었다. 이러한 문제를 해결하고자 OMG(Object Management Group)에서는 이기종간 분산 객체 컴퓨팅을 위한 표준안으로서 CORBA(Common Object Request Broker Architecture)를 제안했다. CORBA는 서비스를 제공하는 객체와 서비스를 받고자 하는 클라이언트 사이의 중개자로서 ORB(Object Request Broker)를 이용하는 시스템으로 클라이언트는 원하는 서비스에 관한 요청(request)을 생성하고 이것을 ORB가 객체 구현(Object Implementation)에 전달한다. CORBA는 객체 기술을 가지고 이종 환경에서의 미들웨어(middleware)로서 기능을 하며 주소 공간, 하드웨어, 운영체제, 구현 언어를 넘어서 객체들이 하나의 어플리케이션으로 통합될 수 있도록 해주는 소프트웨어 버스라고 할 수 있다. CORBA의 분산객체(Distributed Object)는 네트워크상의 어느 곳에서나 존재할 수 있으며 리모트 클라이언트에 의해 액세스 될 수 있고 분산 서버 객체를 만든 언어와 컴파일러는 클라이언트에 대해 완전히 투명하며 클라이언트는 분산 객체가 어디 있고 어떤 운영체제에서 돌아가는지 알 필요 없이 ORB를 통해 원하는 객체에 링크할 수 있는데 이러한 특징들이 트레이더의 구현을 직접적으로 도와줄 수 있기 때문에 CORBA 위에서 트레이더를 구현하였다. CORBA의 구조는 다음과 같다.



(그림 1) CORBA의 기본 구성도

1.2 상호협력 주체

트레이딩 서비스는 차세대 분산처리의 기준 모델인 ISO ODP(International Standard Organization Open Distributed Processing)로부터 기본적인 트레이딩 서비스의 개념을 도입하여 객체지향 기술을 접목한 OMG(Object Management Group)의 CORBA 서비스 사양을 기반으로 하고, 트레이더의 기능은 클라이언트/서버 어플리케이션의 임포터(클라이언트)가 실시간 동안에 익스포터(서버)의

적절한 서비스를 동적으로 선택할 수 있도록 한다. 시스템의 상태, 서비스, 서비스 구성요소가 동적으로 변환되어도 적절한 서비스 제공자를 사용할 수 있도록 하는 소프트웨어가 트레이더이다.[3,4]

분산시스템에서 클라이언트들에게 좀 더 빠른 서비스 속도와 보다 넓은 서비스 폭을 제공하기 위해 상호협력하는 주체(서브젝트)로서 트레이더들이 그들의 설비와 목적에 따라 상호협력한다. 상호협력하는 방법은 그들의 설비와 목적에 의해 결정된다. 따라서 상호협력에 관한 주체(서브젝트)들의 활동은 다음과 같이 정의된다.

Subject = ((facilities, cost), (goals, weights))

이 모델에서 설비들은 다음과 같이 세가지 구성요소를 가진다.

- 동작(Actions) : 서브젝트가 수행할 수 있는 동작들
- 자원(Resources) : 동작을 수행하는 데 필요한 자원들
- 정보(Knowledge) : 자신뿐만 아니라 다른 서브젝트들의 환경에 대한 정보(지식)

1.2.1 설비(facilities)

동작(actions)은 클라이언트 요청들과 서버 오퍼들의 처리에 관계한다. 두가지 경우 예를 들어, 클라이언트 요청에서의 탐색범위와 선택기준이나 제한된 오퍼들에 대한 기준과 같은 많은 옵션들이 포함된다. 각각의 옵션이 다른 트레이더 동작을 요구하지만 트레이더는 모든 동작들을 수행할 수 없기 때문에 트레이더 간의 상호협력이 더욱 필요하다.

자원(resource)은 컴퓨팅과 통신자원 특히 디스크 공간 뿐만 아니라 트레이더 동작을 수행하는데 이용되는 알고리즘으로 구성된다. 어떤 자원들은 한정된 용량을 가지고 있어서 설비들의 양을 제한한다. 이것은 특히 트레이더의 응답시간과 저장될 수 있는 서비스 오퍼들의 수에 대해서 중요한 역할을 한다.

정보(knowledge)는 클라이언트에 의한 서비스 이용내력, 저장된 서버들의 특성들과 다른 트레이더들의 특성에 관계한다. 한 트레이더는 한 시스템 내에 있는 모든 트레이더들이나 공간적으로 이웃한 트레이더들의 부집합에 대한 정보를 안다. 두 번째 경우에서 트레이더들의 고립을 피하기 위해서는 모든 트레이더들의 부집합 연합은 상호협력 하고자 하는 모든 트레이더들의 집합과 동일해야 한다.

따라서 여러 가지 설비들을 가진 다른 트레이더들이 존재할 수 있고 다른 트레이더들에게 자신의 설비를 제공할 수 있고 익스포팅하는 양은

목적에 달려있다.

1.2.2 목적(Goals)

상호협력하려는 트레이더들 사이의 목적은 클라이언트와 서버 입장에서의 목적과 트레이더의 특정한 목적으로 나뉘어질 수 있고 부분적으로 기능과 질 면에서 다시 세분할 수 있다. 클라이언트 입장에서의 목적은 각각의 클라이언트 요청에 대해서 응답하는 것과 클라이언트 기준에 따라 클라이언트와 서버사이에서 서버 응답시간 최소화 등으로 할당을 최적화하는 것이다. 서버 입장에서의 목적은 클라이언트 요청에 대한 응답 가능성을 증가시키기 위한 포텐셜 클라이언트 집합을 제공하고 서버 기준에 따라 클라이언트와 서버사이에서 최적화 할당을 하는 것이다. 그리고 트레이더의 특정한 목적은 높은 트레이더 서비스 질을 제공하고 필요하다면 다른 도메인 상의 트레이더와도 상호협력을 할 수 있도록 하는 것이다.

이러한 목적들에 가중치를 얼마나 둘 것인가는 시스템관리자에 의해 정의되어야 한다. 일반적으로 트레이더들은 제공된 목적들과 이러한 목적들의 가중치면에서 다르다.

2. 호스트 관리자(Host Manager)의 구조와 기능

2.1 호스트 관리자 구조

호스트 관리자는 트레이더간의 상호협력을 위한 중재자로서 각각 다른 트레이딩 도메인 상에서 독립적으로 운영된다. 호스트 관리자를 이용한 상호협력은 각각의 트레이딩 도메인상에 로컬 트레이더와 호스트 관리자를 두고 있기 때문에 임의의 지점에서 트레이딩이 요청될지라도 동등한 트레이더간의 정보교환이 가능하며 대규모 분산 시스템에서 효과적인 상호협력 모델이다. 호스트 관리자는 다른 도메인상의 호스트 관리자와의 상호작용시에는 자신의 트레이딩 도메인의 대표자로서 역할을 수행하며 동시에 하나의 로컬 트레이더로서의 역할을 수행한다.[8]

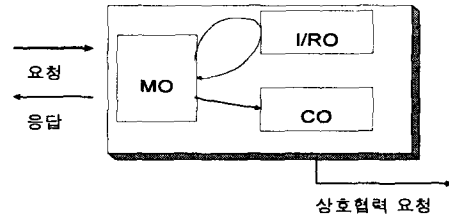
호스트 관리자는 각각의 다른 도메인상에 존재하는 호스트 관리자에 관한 설비와 목적에 관한 리스트를 가지고 있어 상호간 상호협력 импорт 및 익스포트 기능을 수행하게 된다. 그러므로 트레이더의 설비와 목적의 가중치에 따라서 적절한 상호협력 모델을 선택하여 상호작용한다.

다른 도메인 상의 트레이더들을 연결시켜주는 호스트 관리자의 구조는 다음과 같다.

■ 관리자 객체(MO : Management Object)

로컬 트레이더의 트레이딩 서비스 요청을 받아 다른 도메인 상에 있는 호스트 트레이더에 대한 위치정보를 얻기위해 정보 저장소 객체(I/RO)에게 요청한 후, 얻은 정보를 가지고 상호협력 객체

(CO)를 통해 다른 도메인 상의 호스트 관리자에게 서비스 요청을 한다. 또한 트레이더가 처리하여 반환된 레퍼런스를 요청한 로컬트레이더에게 전달한다.



(그림 2) 호스트 트레이더의 구조

■ 정보 저장소 객체(I/RO : Information Repository Object)

관리자 객체(MO)에게 연결되어 있는 로컬 트레이더 또는 호스트 관리자의 정보를 관리하여 그 위치정보를 관리자 객체(MO)에게 반환한다.

■ 상호협력 객체(CO : Cooperation Object)

다른 도메인 상의 호스트 관리자와의 연결을 위해 로컬 트레이더 또는 호스트 관리자의 위치정보를 추가, 삭제, 그리고 수정하여 다른 트레이더에게 서비스 요청을 한다.

2.2 호스트 관리자의 기능

각각의 도메인내의 호스트 관리자는 상호협력을 위해서 다른 도메인 상에 존재하는 호스트 관리자의 설비와 목적, 위치정보 등 리스트를 가지고 있으며, 호스트 관리자의 주요 기능은 세 가지로 분류된다.

- 트레이딩 서비스 요청 기능 : 트레이딩 서비스를 받고자하는 객체(임porter, 익스porter)들의 요청에 대한 오퍼레이션을 트레이더에게 전해주는 기능
- 연결에 대한 정보 관리 기능 : 다른 트레이더와 상호협력을 위해 연결된 호스트 관리자나 트레이더에 대한 정보를 관리할 수 있는 기능
- 상호협력 기능 : 다른 트레이더에게 импорт 요청을 전달하고 얻어진 결과를 처리할 수 있는 기능

3. 상호협력

분산 시스템의 동향은 광범위한 네트워크를 기반으로 한 분산처리 기술에 객체 지향 기술의 도입으로 인하여 분산 시스템의 크기가 거대해지고, 응용 서비스 기능들이 보다 다양해지고 있으며,

이들간에 많은 상호연결이 요구되고 있다.

분산처리 환경에서 분산 시스템을 이루는 객체들의 상호 접속을 효율적으로 하기 위해 트레이딩 서비스를 이용하지만 단일 트레이더가 관리해야 할 정보의 양이 증가하게 되면 트레이더의 성능 저하의 문제점이 발생하게 된다. 그래서 각각의 영역 내에 트레이더를 두고 이들간의 상호협력력이 필요하게 되었다. 이로 인하여, 트레이더는 자신의 영역 내에서 관리하는 정보 이외에 분산된 트레이딩 정보를 제공할 수 있게 되고, 클라이언트(임porter)들에게 보다 많은 서비스 선택의 폭을 넓혀 줄뿐만 아니라 보다 나은 서비스를 제공할 수 있다. 그래서, 클라이언트(임porter)가 많은 개별적인 트레이더들로부터 해당 서버들에 대한 정보를 얻을 수 있지만, 다른 위치의 트레이더와의 상호협력력을 통한 상호작용은 더욱 편리함을 가져온다.

트레이더의 상호협력력은 각각의 트레이딩 도메인 상의 트레이더가 갖는 정책과 객체들의 상호 접속을 위한 제어에 상관없이 각 객체가 가지고 있는 서비스의 공유가 이루어지므로 인해, 다른 도메인에 있는 시스템과 연동을 가능하게 한다. 그로 인해, 한 도메인은 상호협력된 다른 도메인과 정보를 공유할 수 있고, 적당한 서비스 오퍼를 찾을 수 있다. 따라서, 클라이언트는 다른 트레이딩 도메인 상에서 관리되는 객체에 대한 레퍼런스를 트레이더로부터 획득하여 서비스 요청을 할 수 있다.

3.1 상호협력 형태

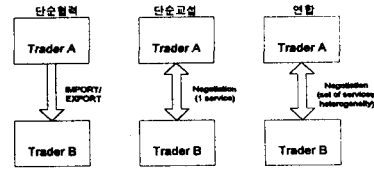
트레이더가 관리해야 할 정보의 양이 증가하게 되면 트레이더의 성능이 저하되는 상황이 초래된다. 상호협력력을 통하여 트레이더는 자신에게 속하지 않은 분산된 트레이딩 정보를 제공할 수 있게 되고 따라서 임porter는 자신의 트레이더에 등록된 서비스 이외의 많은 서버의 서비스를 이용할 수 있다.

본 논문에서는 현재 상태에 따라 합리적인 선택을 하기 위해 다음 기준에 따라 분류된 세가지 유형의 상호협력 형태를 이용한다.

- (1) 상호협력에 참여하는 트레이더들의 서비스 타입
- (2) 익스포트되거나 임포트하려는 서비스들의 수
- (3) 서비스 이용의 빈도수

위에 나타낸 기준에 따라 분류된 세가지 유형의 협력 형태는 다음과 같다.

1. 단순협력(Light Weight Cooperation) - 참여하는 트레이더들의 서비스 타입이 동일하며, 서



(그림 3) 트레이더의 상호협력 형태

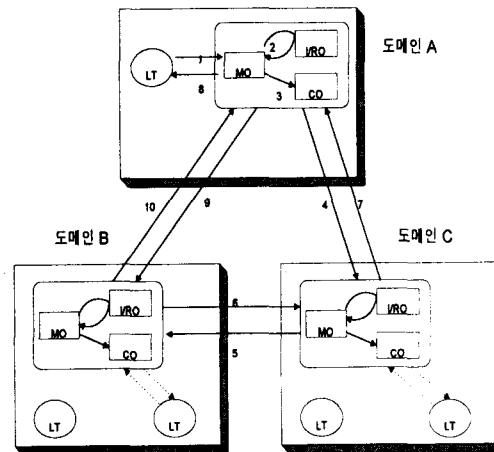
스가 한 타입일 때 적용되고 서비스 이용 빈도수가 적다.

2. 단순교섭(Simple Negotiation) - 참여하는 트레이더들의 서비스 타입이 동일하며 서비스가 한 타입일 때 적용되고 서비스 이용 빈도수가 많다.

3. 연합(Federation) - 참여하는 트레이더들의 서비스 타입에 구애받지 않고 서비스 이용 빈도수가 많다.

3.2 상호협력모델과 서비스 수행

본 논문에서 제안한 트레이더 상호협력력은 각각의 트레이딩 도메인 상에 하나의 호스트 관리자(Host Manager)와 여러개의 로컬 트레이더(Local Trader)를 묶어 하나의 트레이딩 도메인을 형성하고, 호스트 관리자를 통한 트레이딩 상호협력에 기초한다.[9,10]



(그림 4) 상호협력 서비스 수행

한 트레이딩 도메인 상에 존재하는 로컬 트레이더의 요청이 지역적으로 처리될 수 없을 때 로컬 트레이더는 같은 도메인 상에 존재하는 호스트 관리자에게 요청을 전달한다. 호스트 관리자는 먼저 자신의 구성 요소인 관리자 객체(MO)에게 서비스 요청을 하고, 관리자 객체(MO)는 트레이

더 정보 저장소 객체(I/RO)에게 다른 도메인 상의 호스트 트레이더와의 상호협력에 관해 검색을 요구하여 정보 저장소 객체(I/RO)에 저장된 다른 도메인 상의 호스트 관리자의 위치정보를 얻는다. 관리자 객체(MO)는 정보 저장소 객체(I/RO)로부터 얻은 트레이딩 도메인 상의 호스트 트레이더에 대한 레퍼런스를 상호협력 객체(CO)에게 전달한다. 상호협력 객체(CO)는 레퍼런스를 가지고 앞에서 제시된 세가지 상호협력 형태중 한 형태를 선택하여 다른 도메인 상의 호스트 관리자에게 연결한다. 로컬 트레이더의 요청에 있어서 서비스를 수행하는 트레이더들의 서비스 타입이 동일하고 요청 빈도수가 적으면 단순협력(Light Weight Cooperation) 형태를 이용하는데 이 때는 상호협력하는 트레이더들은 클라이언트와 서버로서 동작하여 한 트레이더는 임포트만하고 다른 한 트레이더는 익스포트한다. 반면에, 요청 빈도수가 많으며 서비스가 한 형태로 이루어지면 호스트 관리자는 단순교섭(Simple Negotiation)을 이용하고 이 상호협력 형태는 단순협력(Light Weight Cooperation) 형태와는 달리 대등한 수행능력을 갖는 대칭구조 형태로 서비스를 수행한다. 이기종 시스템들간의 객체통합기법으로서의 연합(Federation) 형태는 서비스 수가 많고 서비스 타입이 다를 때 이용되는 상호협력 방법이다. 연합(Federation)은 상호협력 형태중에서 가장 자원 소비가 많고 복잡하기 때문에 필요한 경우에만 이용되고 이 상호협력 형태는 서비스 수행을 하기 위해서 많은 서버들이 포함된다.

III. 결 론

분산 처리 시스템은 다양하고 많은 서비스 자원들로 구성되어 있다. 신뢰성 있고 효율적인 분산 처리 환경을 제공하는 것은 이러한 분산된 자원들에 대한 효율적인 관리에 따라 매우 중요하다. 이러한 문제점을 해결하기 위하여 클라이언트에게 서버의 위치 투명성을 제공하며, 분산된 객체간의 효과적인 상호작용을 하도록 돕는 트레이딩 서비스는 중요한 요소로 인식되고 있다.

본 논문에서는 객체들간의 상호작용에 따른 문제점을 해결하기 위하여 여러 개의 로컬 트레이더와 하나의 호스트 관리자로 묶어서 트레이딩 도메인을 형성하고 서로 다른 영역에 위치한 독립적인 트레이더들을 호스트 관리자로 하여금 관리하도록 하므로써 분산시스템을 이루는 요소 객체들간의 서비스 공유를 가능하게 하였으며 서로 다른 영역에 위치한 독립적인 트레이더들의 서비스 타입, 서비스들의 수와 서비스 요청 빈도수에 따라 협력모델을 단순협력(Light Weight Cooperation), 단순교섭(Simple Negotiation), 그리고 연합(Federation)으로 분류하여 호스트 관리자간에 상호협력이 이루어 지도록 하므로써 분산 시스템을 이루는 많

은 객체들의 접속을 효과적으로 할 수 있었고, 클라이언트에게 서비스 속도의 증가와 서비스 선택의 폭을 넓힐 수 있게 하였다.

앞으로의 연구는 트레이딩 서비스에서 비용과 보안에 관한 문제를 고려하여 연구되어야 한다.

IV. 참고문헌

- [1] OMG, "CORBA 2.0 Specification" <http://www.omg.org/corba/iiop.htm>, 1995
- [2] DEC, HP, et al. , "The Common Object Request Broker : Architecture and Specification", Revision 1.1, 91.12.1
- [3] ISO/IEC JTC1/SC21/WG7 and ITU-TS SG7. Q16. Working Document on Topic 9.1-ODP Trader, 1993
- [4] Y. Ni and A. Goscinski, "Trader cooperation to enable object sharing among users of homogenous", 1994
- [5] M. Bearman and K. Raymond. Federating Trader : An ODP Adventure. International IFIP Workshop on Open Distributed Processing, 1991
- [6] O. Spaniol, C. Popien, and B. Meyer, "Services and Service Trading in Client/Server System (in German)", 1994
- [7] C. Burger, "Cooperation policies for traders", 1994
- [8] 정창원, "이질적인 분산시스템에서 객체접속을 위한 트레이딩 연합모델", 석사학위논문, 1998
- [9] 이재승, 주용희, 홍원기, "CORBA를 이용한 ODP트레이딩 서비스", 1996
- [10] 김대중, 박성원, 안순신, "정보망에서 분산처리를 위한 투명성", 정보과학회 춘계 학술대회, 1995