

# 광역 분산 컴퓨팅 환경에서 혼합서비스(네이밍/트레이딩)를 이용한 중복객체의 관리에 관한 연구

이원중\*, 신창선\*, 주수종\*  
\* 원광대학교 컴퓨터공학과

e-mail: cosmos, csshin@gaebyok.wonkwang.ac.kr  
scjoo@wonkwang.ac.kr

## A Study on the Management of Replica Object by Using Mixed-Service(Naming/Trading) based on Wide Area Distributed Computing Environment.

Won-Jung Lee\*, Chang-Sun Shin\*, Su-Chong Joo\*  
\*Dept. of Computer Engineering, WonKwang Univ.

### 요 약

최근의 급속한 인터넷 성장에 힘입어 분산처리의 기술 및 환경은 인터넷을 기반으로 광역환경으로 확장될 전망이다. 이러한 환경은 사용자에게 보다 투명한 분산객체간의 효율적인 상호접속이 요구될 뿐 아니라, 광역환경에서 이름이나 속성에 의해 다양한 중복된 성질을 갖고 있는 객체들의 관리가 요구된다. 또한 광역 분산 환경에서 최적의 객체를 선정하는데 분산된 시스템들간의 부하분배를 고려하여 투명성을 제공하는 메카니즘이 필요하게 된다.

따라서, 본 논문에서는 광역분산 컴퓨팅 환경에서 광역 서비스를 지원할 수 있는 이름/속성기반의 중복객체들을 관리할 위한 통합 트리 구성방법을 제안한다. 그리고 이를 통한 혼합(네이밍/트레이더)된 위치서비스(Location service)를 이용하여 무수히 산재한 중복된 분산 객체들을 효율적으로 유지하고, 최적의 객체 선정을 통해 부하균형을 유지할 수 있는 방안을 제시한다.

### 1. 서론

인터넷 기반의 WWW와 전자메일과 같은 분산 서비스들이 사용자에게 제공되고 있지만 사용자에게 투명성을 제공하는데 미흡하다[1]. 이러한 환경에서 위치 투명성(location transparency)과 복제 투명성(replication transparency)에 대한 문제점을 들 수 있다. 즉, 기존의 객체를 찾는데 이름은 객체를 참조하기 위한 위치와 밀결합(tight coupled)되어 있다. 이름에 의존적인 위치는 객체의 이주성(migration)을 다루기에는 매우 어렵게 만든다. 한 객체가 이동을 하게 되면 해당 객체의 이름이 변화되어야 하며, 이들을 가리키는 포워딩 포인터의 재해석이 필요하다. 또한 위치에 의존적인 이름은 복사된 데이터를 참조할 때, 즉시 사용하지 못하는 문제점을 가지고 있다. 그러므로 복사된 객체들과 복사본의 위치에 대한 파악과 이러한 복사본간의 일관성을 유지시키는 방법에 대한 기능을 가지고 있어야 한다. 지금까지 이러한 투명성을 제공하기 위해 현재 이름 대 주소(name-to-object)의 방법을 개선한 광역 분산시스템에서 사용되는 네이밍 서비스와 컨택 어드레스를 이용한 위치 서비스가 네델란드 Vrije 대학의 A. S. Tanenbaum 연구팀에서 제안되었다[5, 6, 7]. 그러나, 현재 객체를 찾는 방법에는 있어서 이름이 아닌 다양한 속성(property)을 이용한 탐색 방법과 최적의 객체를 선정하는데 분산된 시스템들간의 부하분배를 고려하여 투명성을 제공하는 방법이 요구되고 있다.

따라서, 본 논문에서는 먼저 광역 객체 컴퓨팅 환경에

적합한 분산 객체들에 대해 사용자들에게 분산 투명성을 보장하기 위해, 분산 객체를 비롯하여 중복된 객체들에 대한 관리를 위해 속성과 이름을 같이 사용 가능한 광역 트리의 구조의 설계 및 유지변경과 재구성, 객체 탐색 알고리즘을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서는 광역 분산 환경에 적합한 분산 객체 모델에 대해 설명하고, 3장에서는 위치 서비스의 기반이 되는 이름과 속성을 지원하는 광역 트리에 대해 기술하고, 4장에서는 위치 서비스의 요구사항과 알고리즘에 대해 설명하고, 끝으로 5장에서는 결론과 향후 연구내용에 대해 기술한다.

### 2. 광역분산객체

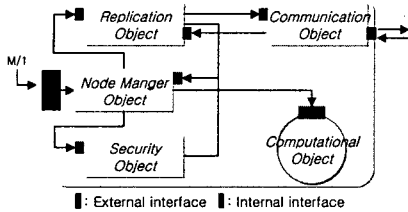
#### 2. 분산 객체

객체는 하나 이상의 인터페이스와 상태를 가지고 있으며, 다중 주소 공간에 분산된다. 이러한 분산은 분산 객체 그룹[10]을 통하여 통신하고 이들간의 상호작용에 의해 이루어진다. 분산 객체그룹은 영역(Domain)으로부터 다수의 컴퓨팅 노드에 분산되며, 구성요소들은 그룹을 포함하여 원격으로 위치할 수 있다. 단, 노드 관리자 객체는 항상 같은 노드 안에 위치한다. 또한, 객체그룹(이후, 분산 객체라 명명)은 단일 객체와 같이 데이터와 메소드, 이를 조작하기 위한 오퍼레이션을 캡슐화 하였다. 각각의 객체들은 자신을 식별할 수 있는 유일한 객체 핸들(object handle)을 갖는다.

\* 본 연구는 2000년 과학재단의 학술연구 조성비 지원에 의한 연구임.

2.1 구조

본 연구의 기반이 되는 분산 객체는 (그림 1)과 같이 5개의 객체로 구성되어 있으며, 사용자 정의하는 외부 인터페이스와 분산 객체들 내부의 내부 인터페이스로 구성된다.



(그림 1) 분산 객체

2.2 구성 요소

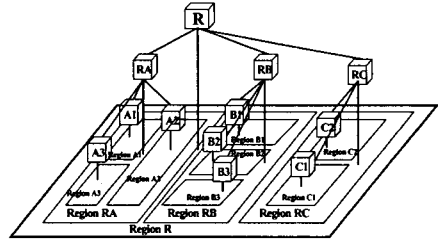
- 관리 인터페이스(Management Interface)는 분산 객체 외부에서 연산객체에 접근 할 때, 컨택 어드레스(contact address)를 통하여 노드 관리자 객체에게 서비스 요청한다. 이는 서비스에 대한 이용 가능여부를 보안 객체와 상호작용하며, 분산 객체의 생명주기 서비스 요청 및 객체에 대한 상태 정보를 위한 기능을 제공한다.
- 노드 관리자 객체(Node Manager Object) : 관리 인터페이스 객체로부터 전달된 요구에 따라 요청된 기능을 수행한다. 요청된 기능들은 분산 객체의 생명주기 서비스와 객체에 대한 정보 변경과 지정한 객체에 대한 보안 조건의 변경을 보안 객체를 이용하여 수행한다. 또한 통신 객체와 상호작용 함으로 다른 분산 객체와 통신한다.
- 보안 객체(Security Object) : 분산 객체 내 각 연산 객체들을 사용하기 위한 보안 규칙을 관리한다.
- 복사 객체(Replication Object) : 분산 객체가 포함하고 있는 복사된 객체(replica)들을 관리한다. 또한, 이들의 일관성을 유지한다.
- 연산 객체(Computational Object) : 실질적인 서비스를 제공하는 객체

3. 통합트리

인터넷 기반의 광역 분산객체 환경은 객체지향 기술을 접목하면서 많은 어플리케이션은 객체의 형태로서 서비스된다. 그러나 중복된 객체에 위치 파악 및 유지관리를 위해 지정된 이름만으로는 관리의 한계를 어려움이 있다. 또한 트레이더를 활용한 속성기반의 객체탐색 및 관리서비스도 속성의 부분 중복에 대한 유지방법은 연구가 미흡한 실정이다. 그래서 광역분산환경에서의 이름과 속성의 중복 객체에 대한 새로운 관리 메커니즘이 요구하게 된다. 이를 해결하기 위한 광역분산환경에서 중복객체관리를 분류방안을 제시하고 혼합(네이밍/트레이딩)서비스를 이용 중복객체의 문제해결을 위한 네임/속성기반의 위치서비스를 위한 광역 통합 트리 구조를 제안한다.

본 논문에서 제안하고 있는 위치서비스는 (그림 2)와 같이 디렉토리 노드들의 분산 탐색 트리를 사용하여 구조화

된다. 각각의 노드는 광역 통신 시스템의 지리적, 관리적 영역으로 분할되며, 중간노드는 서브 트리에 대한 모든 지역의 결합이며, 트리의 루트 노드는 전체 세계를 나타낸다.



(그림 2) 객체들의 위치를 파악하기 위한 광역 트리

3.1 중복객체관리

광역분산환경에서 존재하는 수많은 객체들은 이름/속성에 의한 다양한 중복된 성질을 갖고 있다. 기존의 네이밍이나 트레이딩 서비스로는 중복된 객체들의 바인딩 서비스가 제한적이다. 중복객체의 이름 및 속성분류에 따른 네이밍/트레이딩 서비스 가능여부는 다음과 (표 1)과 같이 분류할 수 있다.

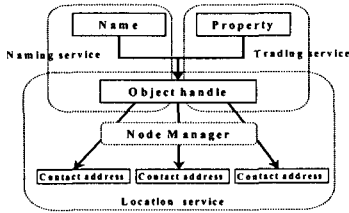
<표 1> 서비스 특징

중복 객체의 이름 및 속성 분류		재공되는 서비스 유형			
이름기반 객체1:객체2	속성기반 객체1:객체2	네이밍		트레이딩	
		비중복	중복	비중복	중복
이름=이름	전체속성=전체속성	-	x	-	x
이름=이름	전체속성≠전체속성	-	x	o	-
이름=이름	부분속성=부분속성	-	x	-	x
이름=이름	부분속성≠부분속성	-	x	o	-
이름≠이름	전체속성=전체속성	o	-	-	x
이름≠이름	전체속성≠전체속성	o	-	o	-
이름≠이름	부분속성=부분속성	o	-	-	x
이름≠이름	부분속성≠부분속성	o	-	o	-

기존의 네이밍 서비스나 트레이딩 서비스에서는 비 중복객체들에 대한 바인딩 서비스만이 이루어지고 있다(o부분). 그러나 광역환경에서는 중복객체들에 대한 바인딩이 절대적으로 필요하다(x부분). 또한 현재 네이밍 서비스나 트레이딩 서비스의 특성상 중복객체 개념을 고려할 수 없는(-부분)도 바인딩 서비스를 지원할 수 있어야 할 것이다.

3.2 네이밍/트레이딩 통합 바인딩 구조설계

(표 1)과 같이, (-, x)부분들의 지원을 위한 새로운 방안은 네이밍과 트레이딩 서비스 기술을 통합시켜, 단일객체나 중복객체에 대한 이름과 속성을 바인딩이 가능한 서비스 구조를 제시할 필요가 있다. 이러한 구조는 기존의 네이밍 또는 트레이딩 서비스에서 제공하는 단일 객체를 선정하는 과정보다 양질의 서비스를 제공하는 객체들을 선택하여 바인딩하는 이점을 갖는다. 중복객체들에 대한 바인딩 전략은 광역 분산시스템에서 기존의 비 중복객체의 바인딩을 물론, 이름/속성을 지원하는 중복객체의 바인딩까지 지원할 수 있는 서비스 구조를 (그림 3)과 같이 제시한다.



(그림 3) 이름/속성 바인딩 위한 컨택 어드레스 계층구조

(그림 3)과 같이 네이밍/트레이딩의 혼합서비스와 위치 서비스는 객체에 대한 사용자 정의의 이름 및 속성을 객체들의 위치간 매핑을 관리한다. 상위 레이어는 사용자 정의, 인간이 이해할 수 있는 이름/속성 공간들을 처리한다. 하위 레이어는 사용자들에 의해 객체들을 이름부여 및 속성부여 방법에 독립적인 각각의 객체들의 위치를 저장하는 부분으로 나뉜다. 두 레이어 간의 인터페이스는 객체 핸들(object handle)에 의해 이루어진다. 노드 관리자는 이름/속성에 따라 중복되는 객체들의 유지 및 관리를 위한 객체들이다.

#### 4. 위치서비스

##### 4.1 위치서비스를 위한 요구사항

광역 분산환경에 이름/속성 기반의 중복객체에 관리 및 유지를 위한 위치서비스는 확장성(Scalability), 위치성(Locality), 안정성(Stability), 허용결함(Fault tolerance), 부하 분배(Load balancing)를 모두 고려해야 한다.

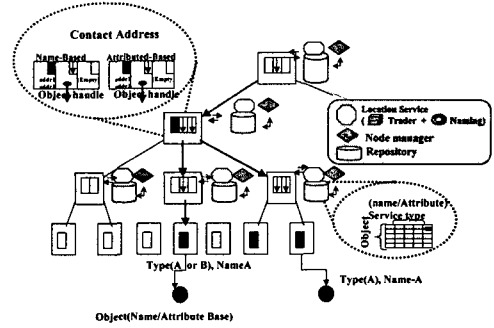
##### 4.2 위치 서비스 모델

광역환경에서 분산객체의 유지관리에 중복성과 객체들의 변환을 허용하는 이동성은 중요한 의미를 갖는다. 이름/속성기반의 혼합 서비스는 클라이언트에게 단일객체에 위치 투명성이 지원하지만 중복객체 및 중복된 객체의 이동성은 투명성이 지원하지 않는다. 이를 해결기 위한 방법으로 객체 핸들 개념을 이용한 객체의 위치투명성이 제공된다. 그러나 객체의 이름/속성을 기반으로 중복객체에 대한 위치투명성 및 이동성을 지원하지 못하고 있는 실정이다.

본 논문에서는 네이밍/트레이더의 혼합 서비스를 이용하여 광역분산환경에서 이름/속성기반의 중복객체에 대한 유지관리 및 투명성을 제공하기 위해서 위치 서비스의 기반이 되는 광역 트리 모델을 다음 (그림4)와 같이 제시한다. 제안한 모델 구조는 이름/속성기반의 각각의 객체에 대한 유일한 객체 핸들을 갖는다. 또한, 이러한 객체 핸들은 하나 이상의 컨택 어드레스(contact address)를 갖는다.

객체의 핸들은 객체의 전체생명주기를 통해 유일한 식별자(Identifier)를 이용 객체의 위치독립성을 제공하게 된다. 네이밍/트레이딩 서비스는 사용자편의의 이름/속성을 객체 핸들에 매핑 시켜준다. 각 지역노드에는 컨택 레코드(각 객체의 위치정보인 Address와 forwarding pointer)를 포함한 광역환경의 분산객체의 추적을 위한 Look\_up (Primitive Operation)이 제공하고, 노드매니저는 사용자편의

의 이름/속성 기반의 중복객체에 대한 상태의 변화 및 속성 변화에 대한 유지관리를 위한 Update(Primitive Operation)가 제공된다.



(그림 4) 이름/속성 기반 위치 서비스 모델

또한 레포지토리는 이름/속성 기반의 상태정보를 저장하고 사용자가 원하는 최적의 객체선정에 필요한 정보를 포함한다.

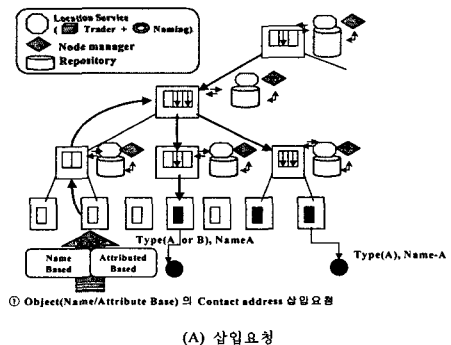
##### 4.2 알고리즘 설계

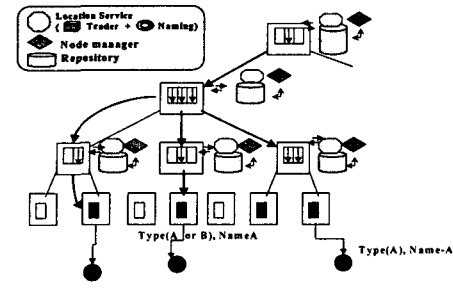
앞서 언급한 바와 같이 광역분산 객체환경에서 이름/속성 기반의 사용자가 요구하는 객체들에 대한 효과적인 컨택 어드레스들의 탐색과 갱신은 DNS의 구성과 유사한 계층적인 지리학적, 위상학적 또는 관리상의 도메인과 같은 중요 광역 네트워크로 구성하였다.

##### 가) 갱신

논리적 구성의 트리구조에서 하위레벨(앞 노드) 상에서만 발생하고 사용자 위주의 이름/속성 기반의 객체들을 추가하게 된다. 하위레벨의 객체 삽입 요청이 발생하면 앞 노드의 상위 노드에 객체 위치 정보를 표시하는 forwarding pointer를 등록 상위 노드로 반복 수행하며, 근 노드까지 위치 정보가 등록이 된다.

(그림 5)에서 나타난 바와 같이 근 노드로부터 앞 노드까지의 각 노드상의 추가를 원하는 객체에 대한 이름/속성 등록정보를 위치서비스의 레포지토리(repository)에 저장된다.

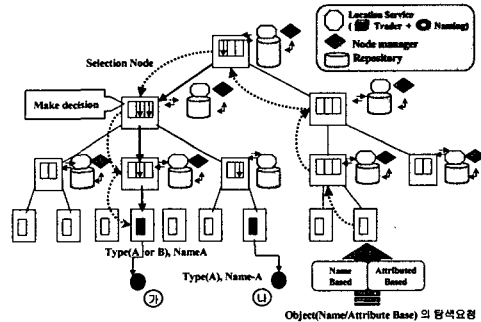




(B) 객체 삽입  
(그림 5) 객체의 삽입 과정

나) 탐색

분산환경에서 객체에 대한 탐색은 이름/속성기반으로 일 노드에서 수행된다. 사용자가 원하는 객체가 소속된 노드에 존재하면 바로 객체에 대한 컨택 어드레스를 얻어 탐색하게 된다.



(그림 6) 객체의 탐색과정

(그림 6)은 광역 분산환경에 논리적 노드 트리의 시나리오 중 최악의 탐색방법으로 근 노드까지 요청자가 원하는 객체에 대한 특히, 중복객체(이름/속성)에 대한 최적 객체 선정까지의 탐색을 나타낸 그림이다. 앞 노드에 원하는 객체에 대한 위치정보를 요청 상위노드의 위치 정보가 없을 경우 상위로 근 노드까지 look up이 수행된다. 근 노드로부터 상위레벨로 forwarding pointer를 따라 하위레벨로 찾아 중복 객체에 대한 최적 객체를 선정 하위레벨 노드로 객체를 탐색할 해당 컨택 어드레스를 넘겨준다.

4.3 부하분배

광역분산환경에서 무수히 많은 중복 객체들 중에서 최적의 객체를 선택이 요구된다. (그림 6)에서 이름/속성 기반의 객체에 탐색 요청시 속성(typeA) 또는 이름(nameA)으로 요청시 2개의 중복 객체에 대한 컨택 어드레스를 얻게 된다. 이때 최적의 객체 선정을 위한 고려사항은 다음과 같다.

첫째, 요청 횟수(Request Counter) : 사용자가 객체의 사

용빈도가 많은 객체일수록 서비스 지연에 대한 부담을 느낀다. (그림 6)에서와 같이 선택 노드(selection node)에서 노드 매니저의 중복 객체 ㉞, ㉟의 사용 요청수를 저장소에 관리하여 사용요청에 대한 빈도수가 적은 탐색된 객체의 위치 정보를 넘겨준다.

둘째, 네트워크 통신비용(network cost) : 논리적인 계층 구조의 노드는 탐색을 위한 노드 상의 많은 객체간의 통신이 발생한다. 그래서 중복객체에 대한 선정 대상의 각 객체들에 대한 ping을 이용 네트워크 통신비용을 산출하여 최소의 비용을 갖는 객체를 선정하여 이 객체로부터 서비스를 제공받게 된다. 이는 물리적인 노드내의 객체 구성과도 연관되고, 지리학적 위치가 가까운 객체는 네트워크 통신비용이 적다.

셋째, 처리성능(CPU performance) : 중복객체들의 물리적 주소 공간이 포함된 노드내에서의 시스템들에 대한 CPU 성능을 실제 이용하여 객체를 선정하는 방법이다. 광역분산 객체 환경에서는 10<sup>3</sup>개의 시스템 내에 객체를 전제로 하기 때문에 시스템 성능여부 및 유휴 상태 관리하여 최적 객체의 선택에 반영하게 된다. 요청 횟수와 마찬가지로 각 node to node매니저를 통해 시스템에 대한 노드 monitoring을 통한 시스템 성능 및 CPU의 유휴처리 능력을 고려하여 반영한다.

5. 결론

본 논문에서는 광역 분산 분산환경에서 단일 객체에 대한 이름 또는 속성 기반의 개별적인 서비스의 제한점 즉, 이름 대 객체, 속성 대 객체의 바인딩함으로써 분산 객체의 이동과 중복 객체들에 대한 관리의 어려움을 해결하기 위해 광역 트리 모델을 제안하였다. 그리고 이를 기반으로 위치 서비스의 기본 오퍼레이션에 대한 시나리오를 제시함으로써, 사용자에게 보다 효과적인 분산 투명성을 제공한다. 앞으로 논문에서 제시한 모델에 대한 성능 평가와 탐색 경로의 최적화에 대한 연구가 필요하다.

참고 문헌

- [1] F.J. Hauck, M. van Steen, and A.S. Tanenbaum. "A Location Service for Worldwide Distributed Objects." In M. Muhlhauser (ed.) Special Issues in OOP, dpunkt-Verlag, Heidelberg, 1997, pp. 384-388.
- [2] OMG. "The Common Object Request Broker: Architecture and Specification, version 1.2". T.R93.12.43, December 1993.
- [3] DCOM, Technical Overview, Microsoft Corp./ Redmond, Wash., 1996.
- [4] TINA-C. "Overall Concepts and Principles of TINA. TB\_MDC.018\_1.0\_94.
- [5] A. Baggio, G. Ballintijn, M. van Steen. "Mechanisms for Effective Caching in the Globe Location Service." Proc. 9th ACM SIGOPS European Workshop, Kolding, Denmark, September 2000, pp. 55-60.
- [6] G. Ballintijn, M. van Steen, A.S. Tanenbaum. "Scalable Naming in Global Middleware." Proc. 13th Int'l Conf. on Parallel and Distributed Computing Systems (PDCS-2000), Las Vegas, August 8-10, 2000, pp. 624-631.
- [7] M. Bearman, "ODP-Trader" in Open Distributed Processing, II, ed. J. de Meer, B. Mahr, and S. Storp, North-Holland, 1994, pp. 19-33.