2002년 한국정보처리학회 춘계학술발표논문집 제9권 제1호

광역 컴퓨팅 환경에서 분산 객체의 통합 바인딩 서비스의 최적 객체 선정

전병택*, 정창원*, 주수종*
*원광대학교 컴퓨터공학과
e-mail:mate@wonkwang.ac.kr

Optimal Object Selection of Integrated Binding Service for Distributed Objects in Wide Area Computing Environments

Byung-Taek Jun*, Chang-Won Jeong*, Su-Chong Joo*
*Dept of Computer Engineering, Won-Kwang University

요약

최근 분산 컴퓨팅 환경은 인터넷 기반으로 규모가 확장됨에 따라 광역 컴퓨팅 환경으로 변화되고 있다. 이러한 환경에서 클라이언트가 원하는 객체나 자원들의 투명성을 제공하는 메커니즘은 필수적이다. 그러나 기존의 네이밍 또는 트레이딩 서비스가 다루는 객체에 대한 식별자는 이름 대 주소 그리고 속성 대 주소의 한쌍으로 이루어져 위치에 따라 객체나 자원에 대한 식별자를 변경해야 하는 문제점을 갖는다. 특히, 이동 객체와 중복객체에 대한 위치 서비스는 더더욱 제공하기 어렵다. 이러한 문제점을 해결하기 위해 식별자를 이름/속성에 따르는 객체 핸들 그리고 컨택 주소로 나누고 각각 독립적인 서비스로 관리하는 통합 바인딩 서비스를 제안하였다. 본 논문에서는 통합 바인딩 서비스에서 위치서비스가 관리하는 컨택 레코드 상에 컨택 주소가 여러 개가 등록이 되어 있을 경우, 중복된 객체를 의미한다. 이러한 중복된 객체들 중에 이들이 존재하는 서버의 부하정보를 이용하여 하나의 컨택 주소를 선택하는 과정을 보였다.

1. 서론

분산 컴퓨팅 환경은 인터넷 기술의 발전으로 인하여 광역 컴퓨팅 환경으로 확장될 전망이다. 이에 따르는 "광역 시스템"[1]에 대한 연구가 활발해졌으며, 대표적으로는 Globe[2], Legion[3], Infospheres[4], Globus[5], Berkeley's Web OS[6], Web Computer[7] 등의 프로젝트가 수행되고 있다. 이들 연구는 공유 데이터와 객체의 설계 및 구현에 주안점을 두고 있으며, 이러한 연구 기반은 분산 투명성(distributed transparency)에 기초로 두고 있다.

이러한 환경에서, 객체의 이동(migration)과 중복(replication)은 필수적인 요소 중의 하나이다. 이러한 객체들을 관리하기 위한 메커니즘으로, 대부분 네이밍 기반의 위치 서비스를 제공하고 있다. 그러나 네트워크 상의 수많은 객체들을 이름으로 식별하기에 부족하다. 이를 해결하기 위해 ISO ODP (International Standard Organization Open Distributed Processing)에서는 트레이딩 서비스를 제시하게 되었고, 이를 OMG에서 채택하여 객체지향 기반의 트레이딩 서비스 스펙을 제안하였다. 이 사양에 따르는 트레이딩 서비스를 여러 분산 컴퓨팅의 아키텍쳐에서 필수 서비스로 채택하고 있는 실정이다. 그러나 이러한 위치 서비스의 근간이 되는 객체에 대한 레퍼런스가 이름 또는 속성별 네트워크 주소가 한 쌍으로 되어 동일한 서비스를 제공하는 객체임에도 불구하고, 서로 다른 객체로 구분되는 문제점을 갖는다. 이로 인해 기존의 네이밍이나 트레이딩 기법[8]으로는 객체들의 이동 그리고 중복된 객체들에 대한 위치 투명성을 제공하지 못한다.

따라서 우리는 이러한 문제점을 해결하기 위한 통합 바인딩 서비스 모델을 제안하였다[9, 10].

본 논문에서는 우리가 제시한 통합 바인딩 서비스에서 중복 객체의 관리와 중복된 객체들 중에 최적의 객체를 선정하는 과정에 중점을 두어 기술한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 바인딩 서비스의 문제점과 중복 객체를 관리하기 위

본 연구는 한국과학재단 목적기초연구 (2000-1-30300-010-2) 지원으로 수행되었음.

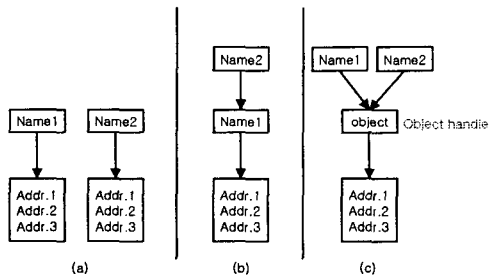
한 방법을 기술하고, 3장에서는 통합 바인딩 서비스 과정에 대한 각각의 세부 단계에 대해 살펴본다. 그리고 4장에서는 광역 통합 트리 기반의 최적 객체 선택을 위한 전략에 대해 기술한다. 끝으로 5장에서는 결론 및 향후 연구내용에 대해 기술한다.

2. 기존 바인딩 서비스의 문제점

본 장에서는 기존 네이밍 기반의 바인딩 서비스 문제점에 대해 기술한다.

2.1 이름 대 주소 매핑

이름 기반의 분산 객체 바인딩은 이름을 컨택 주소로 변환 시켜 서비스를 제공하는 분산 객체에 접속하도록 한다. 일반적으로 단일 객체는 하나의 컨택 주소를 갖지만 시스템의 성능 문제와 결부되어 복사 기술을 이용하여 여러 주소를 갖는 해결책을 제시하고 있다. 다음 (그림 1)은 단일 객체를 참조하기 위해 여러 이름으로 등록시키는 대표적인 세 가지 방법을 보이고 있다.



(그림 1) 객체 이름 대 컨택 주소의 매핑

첫 번째 (a)는 단일 객체를 중복해서 다른 이름을 갖게 하며, 컨택 주소의 집합을 복사하여 등록하는 방법으로 가장 간단한 방법이다. 두 번째 방법은 주 이름(Name1)에 컨택 주소들의 집합을 직접연결하고 한 개 이상의 부 이름(Name2)이 주 이름을 가리키도록 한다. 세 번째 방법은 간접적인 방법으로 객체들의 이름은 유일한 객체 식별자와 매핑되고, 이어서 컨택 주소들의 집합에 매핑되는 방법이다.

위 세 가지 방법들을 비교하면 다음과 같다.

- 컨택 주소들의 집합을 변경

첫 번재 방법을 이용하면 컨택 주소들의 집합을 변경하기 어렵다. 이유는 광역 시스템이기 때문이다. 단일 객체를 가리키는 네임 서비스에서 모든 객체를 탐색하기 불가능하다. 또한 단일 컨택 주소에 대한 모든 이름을 반환하는 완벽한 역 인덱스를 포함하고 있어야 한다. 이러한 관점에서는 두 번째나 세 번째 방법이 하나의 장소에 컨택 주소들을 저장하는 것이 보다 나은 방법이다.

- 객체의 이름을 생성하거나 삭제

첫 번째 방법과 세 번째 방법에서, 객체에 대한 서로 다른 이름들은 독립적이다. 첫 번째 방법은 새로운 이름은 컨택 주소들의 집합을 복사하거나 추가해야 한다. 세 번째 방법은 간단하게 객체 핸들을 참조하도록 한다.

두 번째 방법은 부 이름들이 있을 경우 문제가 발생함으로 주 이름을 삭제한다. 이는 새로운 주 이름을 선택해야하고, 남아 있는 부 이름들을 갱신해야한다. 이렇게 하지 않으면 부 이름들은 결합되지 않은 채 남게 된다. 새로운 이름을 추가시키는데 있어 첫 번째 방법이 세 번째 방법보다 효과적이지 못하다. 이유는 새로운 이름은 이에 따르는 컨택 주소들의 집합을 복사해야만 한다.

- 이름 검색의 비용

세 번째 방법은 이름 검색하는데 비용이 더 많이 든다. 첫 번째 방법이 매우 효과적이다.

2.2 문제점

현재 대부분의 분산 시스템에서 제공되는 바인딩 서비스는 이름과 네트워크 주소를 매핑하여 관리한다. 즉, 객체의 식별인 이름이 곧 위치정보인 하나의 컨택 주소를 의미하고 있어 이동 객체나 중복된 객체들에 대한 이동위치나 중복위치정보를 나타낼 수 없다. 즉, 복사 또는 이동 객체는 동일한 객체임에도 불구하고 다른 이름과 컨택 주소를 갖는 새로운 객체로 등록된다. 또한, 트레이딩 서비스도 마찬가지로 단일 객체에 대한 속성이 여러 개 일지라도 하나의 컨택 주소를 갖기 때문에 동일한 문제점을 갖는다.

이러한 환경에서는 결국 새로운 바인딩 서비스의 도입으로 이름과 속성을 이용한 혼합서비스가 요구된다. 이러한 서비스는 중복된 동일한 객체들의 바인딩 전략을 필요로 하고 있다. 앞에서 나타난 (그림 1)의 (c)와 같이, 이동객체나 중복객체는 하나의 객체 식별자와 위치정보인 컨택 주소가 여러 개가 될 수 있으므로 이들을 독립적으로 유지해야 한다. 또한, 중복객체들에 대한 바인딩 서비스의 요청 시, 분산된 중복객체들 중 한 객체의 선정과정으로 부하분배 전략을 추가시켜야 한다.

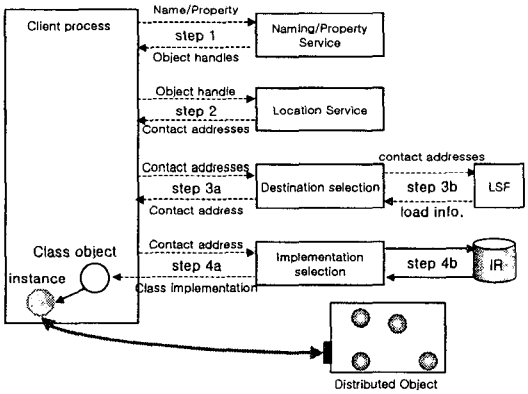
3. 통합 바인딩 서비스

본 장에서는 우리가 제시한 통합 바인딩 서비스 모델의 바인딩 과정과 세부적인 각각의 단계에 대해 기술한다.

3.1 전체 바인딩 과정

통합 바인딩 서비스를 제공하기 위한 전체 과정은 크게 두 단계로 이루어지며, 첫 번째 단계는 객체 핸들을 얻어오는 단계이며, 두 번째 단계는 얻어진 객체 핸들에 대한 컨택 주소를 얻는 과정이다. 이는 세부적으로 분류하면 4단계로 나뉜다.

다음 (그림 2)에서 나타난 바와 같이, 1단계, 2단계는 각각 혼합 서비스와 위치 서비스에서 수행되며, 3단계는 얻어진 컨택 주소들은 LSF를 통하여 해당 객체가 위치한 서버의 부하 정보를 얻어 하나의 컨택 주소를 선택하는 과정이다. 4단계에서는 구현 객체들을 관리하는 IR(Implementation Repository)에서 분산 객체에 대한 레퍼런스를 선택하는 과정을 보이고 있다.



(그림 2) 바인딩 처리과정

3.2 세부적인 단계

통합 바인딩 서비스의 세부적인 단계는 다음과 같이 4단계의 과정을 거친다.

- 1단계 : name/property lookup

이 부분은 혼합서비스에서 클라이언트의 요청에 따라 이름 또는 속성에 매핑되는 객체 핸들을 탐색하는 과정으로 이름으로 찾았을 경우에는 하나의 객체 핸들을 반환하지만 속성으로 찾았을 경우에는 여러 개의 객체 핸들을 반환한다.

- 2단계 : location lookup

이 부분은 객체 핸들은 컨택 주소의 집합으로 변환된다. 즉, 객체 핸들당 여러 개의 컨택 주소가 반환된다. 여러 개의 객체 핸들이 반환되었을 경우에는 하나의 객체 핸들을 선택하여 컨택 주소를 얻는다.

- 3단계 : destination selection

객체 핸들과 매핑되는 컨택 주소의 집합 중에 하나의 컨택 주소를 선택하는 과정으로 해당 객체가 위치한 서버에 대한 부하정보를 비교하여 하나의 주소값을 반환한다.

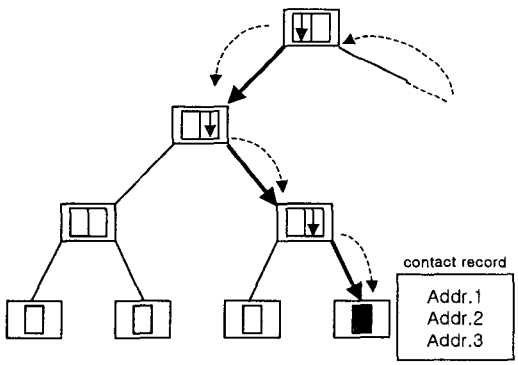
- 4단계 : Implementation selection

해당 서버의 구현 객체를 선택하는 과정으로 실제 구현 객체와 바인딩 하기 위해 클라이언트 프로세스는 새로운 지역 객체를 인스턴스화하고 분산 객체를 초기화한다.

4. 최적 객체 선택을 위한 전략

객체에 대한 탐색은 최하위에 위치한 혼합 서비스가 관리하는 영역의 잎 노드에서 수행이 시작된다.

이때, 사용자가 원하는 객체가 해당 노드에 존재하면 바로 객체에 대한 컨택 주소를 얻는다. 하지만 없을 경우에는 상위 노드로 탐색하게 된다.



(그림 3) 통합 트리로부터 객체의 탐색 과정

(그림 3)은 광역 통합 트리의 시나리오 중 최악의 탐색방법으로 근 노드까지 요청자가 원하는 객체에 대한 특히, 중복객체(이름/속성)에 대한 최적 객체 선정까지의 탐색을 나타낸 그림이다. 잎 노드에 클라이언트가 임포트 요청을 했을 때, 해당 노드에서 위치 정보를 얻지 못하였을 경우, 상위로 근 노드까지 lookup이 수행된다. 최하위 노드로부터 상위레벨의 노드에서 forwarding pointer를 따라 근 노드에 도달하고, 근 노드에서 다시 하위레벨의 노드로 컨

택 주소를 검색한다. 그래서 하나 이상의 컨택 주소를 얻게 된다. 해당 주소가 위치한 서버의 부하 정보를 비교하여 가장 부하가 적은 컨택 주소를 선정하여 하나의 컨택 주소를 반환한다.

4.1 부하 정보

LSF에서는 각 서버의 부하를 실시간으로 검사하며, 이에 대한 부하 정보의 항목은 다음 <표 1>과 같다.

<표 1> LSF에서 제공하는 부하 정보

항 목	내 용
host_name	LSF에서 사용되는 호스트의 이름
status	호스트의 상태 ok의 경우 사용 가능
r15s	15초 평균 CPU 운행 큐 길이
r1m	1분 평균 CPU 운행 큐 길이
r15m	15분 평균 CPU 운행 큐 길이
ut	CPU 이용율
pg	메모리 페이징 비율
ls	현재 로깅한 사용자의 수
it	UNIX상의 호스트의 유휴 시간
swp	Window NT상의 swp 공간 이용량
mem	이용가능한 메모리 량
tmp	자유 공간의 양

LSF에서 제공하는 각 서버의 부하 정보에서 status와 ut에 비중을 두어, 비교한 후 가장 부하가 적은 서버에 위치한 객체의 컨택 주소를 반환한다.

4.2 최적 객체 선정

다음 (그림 4)는 위에서 탐색 결과 3 개의 컨택 주소를 얻었을 때, 각 서버의 부하 정보를 보이고 있다.

(그림 4) LSF를 이용한 각 서버의 부하 정보

(그림 4)에서 나타난 바와 같이 호스트 이름 중 yellow가 사용가능한 상태이며, cpu의 이용율이 가장 적다. mediblue는 ok 앞에 (-)부호가 있기 때문에 사용 불가능함을 나타내고, mate 서버는 현재 시스템이 power-off 상태로 부하 정보를 얻을 수 없다. 따라서, yellow 서버에 해당하는 컨택 주소를 반환하게 된다.

5. 결론

현재 인터넷 기반의 웹서비스와 전자메일과 같은 광역 분산 서비스들이 클라이언트에게 제공되고 있지만, 국부적인 분산 투명성만을 제공하고 있다. 특히, 이러한 분산 투명성을 제공하는 솔루션으로 네이밍 서비스가 대부분을 차지하고 있다. 그러나 이름만으로는 광역 환경에서는 클라이언트의 요구에 맞는 서비스를 제공하기엔 한계가 있다. 이를 해결하기 위해 속성 기반의 트레이딩 서비스가 출현하게 되었다. 이러한 서비스들이 기반이 된 광역 객체 컴퓨팅 환경에서는 수많은 객체들이 이름이나 속성에 의해 중복됨을 예측하게 한다. 따라서, 단일 객체뿐만 아니라 이름 또는 속성 기반의 중복된 객체들의 효율적인 관리와 최적객체 선정을 지원하는 통합 바인딩 서비스 모델을 제안하였다. 본 논문에서는 중복 객체들의 효율적인 관리를 위한 전략 방법과 선택하는 과정을 보였다.

향후 연구로는 분산객체간의 접속 시험을 하고, 이에 대한 최적 부하 성능 평가에 대한 검증 연구가 진행되어야 한다. 또한, 분산객체의 탐색 경로의 최적화를 위한 연구가 필요하다.

참고문헌

[1] Wide-area systems research http://ringer.cs.utsa.edu/faculty/jon/wa.html

[2] The GLOBE Project http://www.few.vu.nl/~steen/globe/

[3] A. S. Grimshaw, Wm, Wulf, and the Legion Team. "The legion vision of a worldwide virtual computer", *Communications of the ACM, 40(1), January 1997.*

[4] Infospheres http://www.cs.caltech.edu/~adam/CALTECH/infospheres.html

[5] Globus, http://www.globus.org/

[6] Berkeley's WebOS http://www.wizzo.demon.co.uk/webcom.html

[7] Web computer http://www.wizzo.demon.co.uk/webcom.html

[8] University of Helsinki - Department of Computer Science http://www.cs.helsinki.fi/TR/A.html

[9] 이원중, 신창선, 주수종 "광역 분산 컴퓨팅에서 혼합서비스(네이밍/트레이딩)기반의 중복객체 관리연구" 2001년 춘계 학술발표논문집, 2001. 4.14, 제 8권 1호, 한국정보처리학회

[10] 전병택, 정창원, 주수종 "광역 분산객체들의 바인딩 지원을 위한 연합 네이밍/트레이딩 모델", 2001년 봄 학술발표논문집(A), 2001. 4.28, 제 28권 1호, 한국정보과학회