

객체지향 분산 시스템에서 실시간성을 위한 연구

정부금*, 장형규*, 차영준*, 김은숙**, 김영만**

*ETRI 실시간 OS 팀, **국민대학교 컴퓨터학부

bgjung@etri.re.kr, ekim@cs.kookmin.ac.kr, ymkim@kmu.kookmin.ac.kr

A study on the real-time technology for the object-oriented distributed system

Boo-Geum Jung*, HyeoungKyu Chang*, Young-Jun Cha*, Eun-Suk Kim**, Young Man Kim**

*ETRI Real-time OS Team, **Department of Computer Science, Kookmin University

요약

OMG에 의해 제안된 CORBA는 현재 분산 컴퓨팅 환경의 표준으로 자리 매김을 하고 있다. 이에 따라 실시간 환경에서도 CORBA에 대한 관심이 증가하고 있다. 실시간 분산처리에서는 핵심 요소인 시간관리(time management), QoS, 스케줄링, 병행성(concurrency), 장애감내(fault tolerance), 그리고 성능(performance)에 대한 인터페이스와 시스템 사양을 명시할 수 있는 특화된 실시간 CORBA 서비스의 제공이 필요하다. 본 논문에서는 장애감내성과 병행성을 동시에 제공하는 객체 서비스 구조를 제안한다.

1. 서론

90년대에 들어와서 소프트웨어 생산성 및 재사용성을 획기적으로 개선하는 객체지향 분산 플랫폼이 등장하면서 많은 분야의 응용들이 객체지향 분산 플랫폼으로 이식되고 있거나 새로 개발되는 추세에 있다. CORBA[OMG 95]는 OMG에서 제안한 대표적인 객체지향 분산 플랫폼의 하나이다.

기존 CORBA 사양을 실시간 운영체제상에서의 미들웨어 플랫폼으로 구현하여 실시간 응용을 지원할 수도 있으나, 이 방법은 실시간 운영체제에 내재하는 실시간 특성을 충분히 활용하지 못하여 분산 실시간 객체시스템에서 요구하는 예측가능 수행(predictable execution) 요구정도를 충분히 만족시키기 어렵다. 따라서 실시간 분산처리의 핵심 요소인 시간관리, QoS, 스케줄링, 병행성, 장애감내, 그리고 성능에 대한 인터페이스와 시스템 사양을 명시할 수 있도록 특화된 실시간 CORBA 사양 및 실시간 서비스가 필요하다.

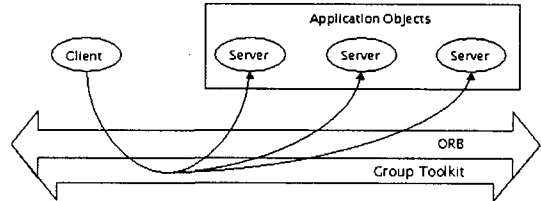
본 논문에서는 실시간 CORBA의 핵심 요소로서 제공되어야 할 장애감내와 병행 처리 서비스 방식에 대하여 제안하고자 한다. 2장에서는 현재 수행되고 있는 장애감내형 CORBA의 연구동향을 살펴보고, 3장에서는 장애감내성과 병행성을 동시에 제공하는 서비스 모델을 제시한다. 마지막으로 4장에서는 본 논문의 결론과 향후 연구과제에 대하여 기술한다.

2. 장애감내형 CORBA의 연구동향

현재의 CORBA는 점대점 통신방식에 기반을 두고 있으나, 장애감내 기능을 추가로 제공하기 위해서는 서비스 객체의 중복 기능과 신뢰성 있는 멀티캐스트가 필요하다.

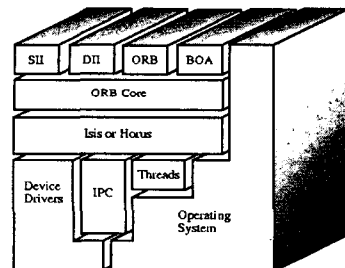
이 장에서는 최근에 발표된 장애감내 방식들을 소개한다.

2.1 통합 방식(integration approach)



[그림 1] 통합 방식

[그림 1]에서 보듯이 통합 방식은 OMG가 제안한 기존 ORB에 그룹 툴킷을 통합시킨 새로운 형태의 ORB가 중복 객체의 그룹 관리 및 신뢰성 멀티캐스트와 같이 장애감내성 제공에 필요한 기본 기능들을 제공하는 방식이다. 대표적인 사례로 Orbix+Isis와 Electra[Maffeis 99]가 있다. [그림 2]는 Orbix+Isis 구조를 나타내고 있다.

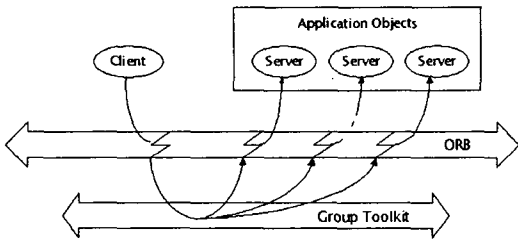


[그림 2] Orbix+Isis의 구조

**본 논문의 연구는 1999년도 한국전자통신연구원(ETRI)의 위탁과제로 수행되었음

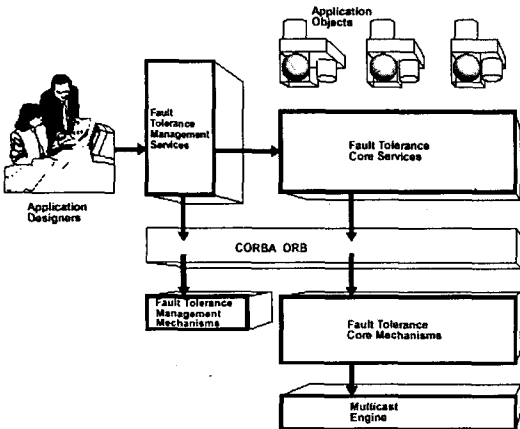
Orbix+Isis 는 Orbix™ CORBA-compliant C++ 개발 환경과 Isis Reliable runtime 기술을 통합한 것이다. Orbix 는 표준 객체지향 프로그래밍 환경, 인터페이스 정의의 추상화 등의 분산 객체지향 CORBA 환경을, Isis 는 프로세스 그룹, 뷰 관리, 상태 전송, request ordering, virtual synchrony를 제공한다. Orbix+Isis 는 클라이언트에 투명하여, 클라이언트 코드의 수정 없이 장애감내 기능을 제공하는 그룹 객체들과 함께 수행될 수 있다.

2.2 방수 방식(interception approach)



[그림 3] 방수 방식

방수 방식은 SUN사에서 제안한 방식[SUN 98]으로, 기존 ORB를 수정하지 않고 클라이언트가 보낸 메시지를 그룹 툴킷이 가로채서 고장감내 기능을 구현한다. [그림 3]은 방수 방식의 기본 구조 및 기능을 보여준다. SUN사의 장애감내 모델의 하부 구조는 객체 복제와 장애감내를 제공하기 위해, [그림 4]에서 보듯이 장애감내 관리 서비스(fault tolerance management service), 장애감내 핵심 서비스(fault tolerance core service), 장애감내 핵심 메커니즘(fault tolerance core mechanism)의 3단계 제어 방식을 채용하고 있다.

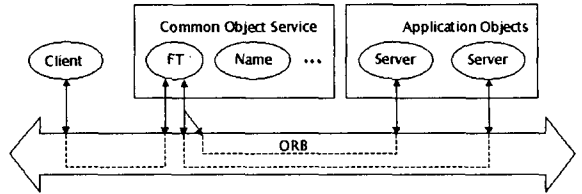


[그림 4] 장애감내 하부 구조

2.3 서비스 방식(service approach)

서비스 방식에서는 기존의 ORB를 수정하는 대신 객체 그룹을 관리하고 장애감내 기능을 제공하는 COS(Common Object Service) 객체를 별도로 정의하여 사용하고 있다. [그림 5]는 서비스 방식의 구조 및 객체간 운용 관계를 보여준다. 대표적인 예로는 OGS(Object Group Service)[Felber 96]와 DOORS(Distributed Object Oriented Reliable System)[Chung

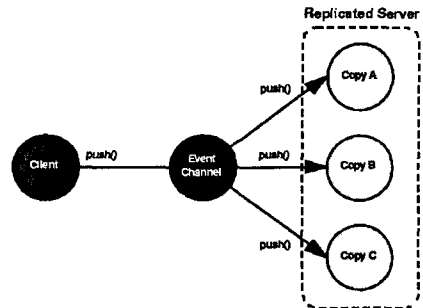
98]이 있다.



[그림 5]서비스 방식

2.4 이벤트 서비스 방식(event service approach)

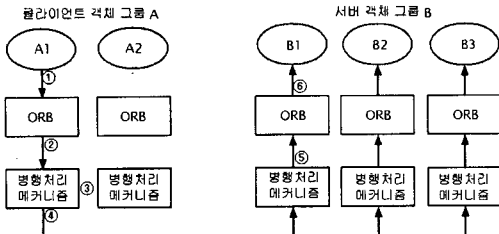
이벤트 서비스 방식에서는 공급자는 이벤트 채널을 이용해서 이벤트 데이터를 생성하고 소비자는 이벤트 채널을 통해 전달된 데이터를 처리한다. 이벤트 채널은 다수 공급자와 다수 소비자 사이에 비동기 통신을 제공하는 표준 CORBA 객체이다. 이벤트 서비스는 PUSH 모드와 PULL 모드라는 두 가지 형태의 통신모드를 제공한다. [그림 6]에서 보듯이 PUSH 모드에서는 공급자가 소비자에게 이벤트 데이터를 제공하고 추가함을 담당하며, PULL 모드에서는 소비자가 공급자에게 이벤트 데이터를 요청한다. 예로는 Fault-Tolerant CORBA Name Server[Maffeis 96]가 있다.



[그림 6] PUSH 모드에 의한 이벤트 서비스 방식

3. 병행성을 갖춘 장애감내 CORBA 서비스

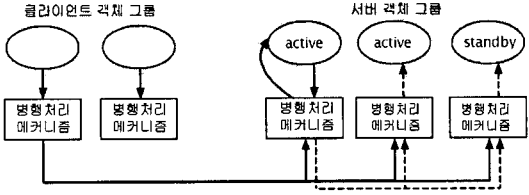
본 논문에서는 병행 및 장애감내 CORBA 서비스를 [그림 4]와 유사한 구조의 방수 방식을 통하여 구현하기로 한다. 복제 객체 그룹간의 호출/응답 메시지는 멀티캐스트 엔진을 통하여 신뢰성 있게 진행된다. 이때 멀티캐스트 엔진은 메시지 전체정렬 전달 서비스(totally ordered message delivery)를 제공한다. 다시 말하면, 모든 멀티캐스트 메시지는 그룹내의 모든 해당 객체에 동일한 순서로 입력 처리 된다. 장애감내 핵심 서비스는 각각의 복제된 객체를 위해서 동적 제어를 제공한다. 장애감내 핵심 서비스는 복제 서비스, 회복 서비스, 장애 검출 서비스, 장애 통지 서비스로 구성되어 있다. 복제 서비스는 응용에서 특별한 복제 객체를 정의할 수 있는 방법을 제공하고, 회복 서비스는 특별한 프로세서 상에서 장애회복을 위하여 복제 객체를 초기화하기 위한 방법을 제공한다. 장애 검출 서비스를 통해서 장애 통지 서비스는 응용에 장애 발생을 알려주게 된다.



[그림 7] 객체 그룹 호출

[그림 7]은 클라이언트 객체 그룹 A가 서버 객체 그룹 B에 있는 메소드를 호출하는 절차를 보여준다. 복제 객체 A1이 객체 그룹 B의 메소드를 호출할 때, 서버 객체 그룹 B의 IOR을 사용한다(1). IOR에는 객체 그룹 B에 속한 모든 객체들(B1, B2, B3)의 정보가 들어있다. ORB는 호출된 서버 객체 그룹 B의 객체 그룹 IOR과 호출한 복제 객체 A1의 복제 객체 IOR을 포함하는 GIOP 메시지를 생성한다(2). 병행처리 메커니즘은 객체 A1의 IOR을 객체 그룹 A에 대한 객체 그룹 IOR로 대체시킨다(3). 병행처리 메커니즘은 GIOP 메시지를 멀티캐스트 메시지로 요약(encapsulation)한다. 그리고 요약된 메시지를 해당 프로세서들에게 멀티캐스트 한다(4). 각 프로세서상의 병행처리 메커니즘은 호출된 서버 객체 그룹 B의 그룹 IOR을 각 프로세서상의 복제 객체의 IOR로 대체시킨다. 그리고 GIOP 메시지를 ORB에 전달한다(5). ORB는 프로세서 상의 복제 객체를 호출하기 위해서, 예를 들면, B1의 IOR을 사용한다(6). 다음 절에서는 병행처리 서비스를 제공하기 위한 두 가지 방법들을 소개한다.

3.1 수동 병행처리 방식



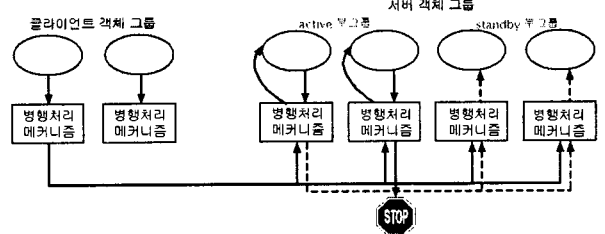
[그림 8] 수동 병행처리

서버 객체 그룹에는 복수개의 active 객체들과 standby 객체들로 이루어져 있다. 수동 병행처리 방식을 이용하는 경우, active 객체들 가운데 하나의 객체만이 요청 메시지를 받고 처리한 후 클라이언트에게 보내준다. Standby 객체들은 active 객체의 상태 정보만을 전송 받는다. 병행성을 위해 active 객체는 2개 이상 존재하도록 하고, 장애감내성을 위하여 standby 객체도 하나이상 존재하도록 한다. 또 active 객체 중 요청을 처리할 객체를 선택하는 방식은 round-robin 방식을 사용하여 실행 객체 선택에서 중복이나 불평형성이 없게 한다.

3.2 능동 병행처리 방식

능동 병행처리 방식을 이용하는 경우, [그림 9]에서와 같이 서버 객체 그룹을 나누어 active 부그룹과 standby 부그룹들을 만든다. 부 그룹 안의 모든 객체는 수동 병행처리 방식과는 다르게 동시에 요청 메시지를 받고 응답 메시지를 반환하게 되는데 병행처리 메커니즘에 의하여 첫번째로 발생한 응답 메시지

만을 전송하고 나머지 메시지는 전송을 방지한다. 실행후 active 부그룹은 standby 부그룹들에 상태 정보를 전송한다. Active 부그룹들은 수동 병행처리 방식에서와 같은 round-robin 방식으로 클라이언트 요구를 처리한다.



[그림 9] 능동 병행처리

3.3 병행처리의 최적화

수행속도를 높이기 위해, 호출 객체가 존재하는 호스트 내에 호출 대상 그룹에 속하는 지역 객체가 동시에 존재하는 경우에 그룹 IOR 대신에 지역 객체 IOR을 직접 사용하여 요청을 하게 한다. 지역 객체 IOR을 사용하여 호출을 하는 경우, 보다 적은 오버헤드를 갖게 되며, 네트워크를 통하지 않고 메시지를 처리함으로써 보다 높은 수행 속도를 얻을 수 있다.

4. 결론

본 논문에서는 실시간 객체지향 분산 플랫폼의 핵심 요소인 장애감내성과 병행성을 동시에 제공하는 CORBA 서비스를 제안하였다. 또한 방수 방식을 사용함으로써 기존 CORBA ORB의 수정 없이 사용할 수 있다. 앞으로의 연구방향은 load balancing 기능과 object migration 기능을 제공하는 서비스에 대하여 연구하고자 한다.

참고문헌

[Chung 98] P.E. Chung, et al., "DOORS: Providing Fault Tolerance for CORBA Applications", poster session of Middleware'98.
 [Felber 96] P. Felber, B. Garbinato and R. Guerraoui, "The Design of a CORBA Group Communication Service", Proceeding of the 15th IEEE symposium on Reliable Distributed Systems, pp. 150-159, October 1996.
 [Maffeis 96] Silvano Maffeis, "A Fault-Tolerant CORBA Name Server", Proceedings of the 1996 IEEE Symposium On Reliable Distributed Systems, Niagara-on-the-Lake, Canada IEEE, October 1996.
 [Maffeis 99] S. Maffeis and S. Landis, "Building Reliable Distributed System with CORBA", to appear in Theory and Practice of Object Systems, John Wiley Publisher, NY.
 [OMG 95] OMG, "The Common Object Request Broker : Architecture and Specification", July 1995.
 [SUN 98] Eternal and SUN, "Fault Tolerance for CORBA", Fault Tolerance Joint Initial Submission, OMG document orbos/98-10-08, October 1998.