

CORBA 기반의 그룹 통신 서비스의 설계와 구현

김희정, 이동만
한국정보통신대학원대학교

Design and Implementation of CORBA-based Group Communication Service

HeeJeong Kim (hjkim1@icu.ac.kr), Dongman Lee (dlee@icu.ac.kr)
Information and Communications University

요 약

현재 CORBA는 신뢰성이나 가용성을 위해 꼭 필요하다고 할수 있는 그룹 통신 메커니즘에 대한 표준을 제공하고 있지 않다. 이것이 CORBA 어플리케이션이 금융, 항공 시스템 등 중요한 부문에서 활용되지 못하는 장벽이기도 하다. 본 논문에서는 그룹 통신 메커니즘을 지원하는 CORBA 서비스의 설계와 구현에 대해 기술한다.

1 서 론

분산 시스템에서 결함 감내, 성능, 고 가용성 등의 효과를 위하여 복제(replication)를 널리 사용한다. 문제는 클라이언트에게 서버 객체 컴포넌트(component)가 복제되어 있다는 것을 숨기면서(투명성 : transparency), 복제된 프로세스들이 공유하고 있는 상태의 일관성(consistency)을 보장해야 한다는 것이다. 이러한 논의점들을 해결하는 메커니즘이 그룹 통신 메커니즘이다.[1]

CORBA는 상호운용성, 원격 객체 호출, 다양한 Common Services의 제공 등으로 많은 주목을 받고 있다. 그러나, 현재 CORBA는 신뢰성이나 가용성을 위해 꼭 필요하다고 할수 있는 그룹 통신 메커니즘에 대한 표준을 제공하고 있지 않다.

CORBA에서 그룹 통신 메커니즘을 제공하는 방법으로는 크게 두 가지로 나눌 수 있는데, ORB자체에 대해 수정하는 방법과[Electra, Eternal], 하나의 CORBA 서비스로 제공하는 방법[OGS]이 그것이다.[2,4,5,6] 클라이언트에 대한 투명성이나 효율성 등을 고려할 때, 전자의 방법이 궁극적으로 바람직하다고 할수 있지만, ORB를 수정해야하는 비용을 수반하고 더 큰 문제는 CORBA의 가장 큰 매력이라고 할수 있는 다른 CORBA 프레임워크와의 호환성 문제를 가져오므로 이 방법은 장기적인 고찰이 필요하다. 다른 CORBA Common Object Service들처럼 별도의 그룹 통신 서비스를 제공하는 후자의 방법은 그룹 통신을 원하지 않은 사용자에게 불필요한 비용을 요구하지 않으면서, 기존의 CORBA 어플리케이션과의 호환성을 유지할 수 있다.

본 논문은 그룹 통신 메커니즘을 지원하는 CORBA 서비스를 디자인하고 구현함을 목적으로 한다.

2 CORBA 기반 그룹통신 서비스의 요구사항

CORBA를 기반으로 하는 그룹 통신의 핵심은 다음과 같다.

여러 가지 이유로(신뢰성, 가용성, 성능) 그룹 서비스를 사

용하고자 하는 클라이언트는 특정 그룹을 지칭하는 그룹명을 통하여 그룹에 요청을 보낼 수 있다. 이 때, 그룹 통신 서비스는 클라이언트가 그룹으로 보낸 요청들은 해당 그룹을 구성하고 있는 하나 이상의 모든 멤버 프로세스들에게 다중 전송해야 한다.[11] 그러나, CORBA는 기본적으로 다중 전송을 지원하지 않으므로, 다중 Unicast를 사용하거나 별도의 다중 전송 프로토콜을 요구한다.

그룹의 멤버들이 공유하고 있는 그룹 상태 정보의 일관성을 위하여 그룹을 대상으로 보내진 모든 메시지는 현재 그룹을 구성하고 있는 정확한 모든 멤버 프로세스들에게 반드시 전달되어야 하므로 그룹 멤버십의 관리가 대단히 중요하다.

그룹명을 그룹 구성원들의 목록으로 확장하는 방법은, 그룹명과 그룹 구성원들의 목록을 관리하는 멤버십 서비스를 사용하는 방법과, 멤버십 정보를 각 멤버들에게 복제시키는 방법으로 나눌 수 있다.[11]. 멤버십 서비스를 사용하는 방법은 간단하게 멤버십 정보의 일관성을 보호할 수 있다는 장점이 있지만 결합지점(point of failure)이 될 가능성이 있다. 멤버십 정보를 멤버들에게 복제시키는 방법은 그룹으로의 메시지 전송 시 별도의 통신이 필요하지 않다는 장점이 있으나, 각각의 멤버 프로세스들에게 분산시켜 공유하는 멤버십 정보를 일관성있게 관리해야 하는 부담이 있다.

그룹 상태 정보를 일관성 있게 관리하기 위해서는 메시지 순서화 또한 필수적이다. 멤버 프로세스들이 비동기적으로 각각 수행하는 연산들이 일관성 있는 결과를 가지도록 보장하기 위해서, 그룹으로 보내진 메시지는 모든 멤버 프로세스에게 전달되어야 하며, 각 멤버들이 수행하는 연산의 순서를 보장해야 한다는 것이다. 순서화의 간단한 방법은 그룹으로 보내진 모든 메시지에 번호를 부여하여 그 순서대로 모든 멤버 프로세스들이 수행하도록 제약하는 전순서화(Total order)이다. 그러나, 대다수의 실세계 응용 어플리케이션에서 순서화를 요구하지 않은 경우도 많다. 뿐만 아니라, 어떤 어플리케이션에서는 일부의 특정 메시지 집합에 대해서만 순서 보장을 요구하는데(Causal order), 이러한 경우에 순서화를 요구

하지 않는 다른 메시지들까지 불필요하게 전송되도록 적용한다면 매우 비효율적이다. 그러므로, 본 논문에서 개발한 그룹 통신 서비스는, 어플리케이션들의 다양한 순서화 요구들을 융통성 있게 수용할 수 있도록 하기 위하여, 사용자가 순서화 요구수준(전순서화, 인과성 순서화, 순서 지원없음)을 선택할 수 있도록 한다. 순서화의 구현은 벡터 타임스탬프를 기반으로 한다.[1,9,11]

한 그룹의 멤버 프로세스 구성은 실행 중에 변경될 수 있다. 즉, 그룹의 새로운 멤버로 새로운 프로세스가 참여하거나, 기존의 멤버 프로세스가 자신의 의지 또는 오류로 인하여 그룹으로부터 탈퇴할 수 있다. 이것을 동적 그룹 멤버십(dynamic group membership)이라고 한다.[1,9,11] 현실적으로 많은 어플리케이션에서 이러한 동적 그룹 멤버십을 요구한다. 동적 그룹 멤버십을 제공하려면 많은 노력들이 동반되어야 한다. 만약, 멤버십 변경 과정이 적절하게 처리되지 않는다면, 어떤 멤버가 수신한 메시지를 또 다른 멤버는 수신하지 않거나, 이미 멤버가 아닌 이전의 멤버 프로세스가 보낸 메시지를 헛되이 수행하는 등의 심각한 비일관성 문제를 가져올 수 있다. 따라서, 모든 그룹 멤버들이 같은 멤버십 정보를 가질 수 있도록 하는 제약이 필수적이다. 동적 그룹 멤버십을 지원하기 위한 주요 메커니즘은 멤버십 변경 메커니즘과 상태 전송 메커니즘이다.[9,11]

멤버십 변경 메커니즘은 변경되기 이전의 멤버십 상태에서 발생한 메시지를 (이전의 멤버십과 변경된 멤버십에 공통적으로 속한) 모든 멤버들에게 전달한 이후, 새로운 멤버십을 알리는 메시지를 모든 멤버 프로세스들에게 전달하도록 보장함으로써, 일관성을 유지할 수 있도록 하는 것이다. 멤버십 변경 통보 메시지는 가장 강력한 순서화인 동기 순서화(Sync-order) 과정을 거친다.

상태 전송 메커니즘은 새로운 프로세스가 이미 진행중인 어떤 그룹의 멤버 프로세스로서 참여하고자 할 때, 그 그룹의 상태 정보를 새로운 멤버에게 전달하여 다른 멤버들과 일관성을 가질 수 있도록 하는 과정을 말한다. 상태 전송은 멤버십 변경 메커니즘에서의 동기화 과정에 적용된다.

3 그룹통신 서비스 컴포넌트 설계와 구현

본 논문에서 설계한 그룹 통신 서비스는 4개의 컴포넌트(그룹 액세스 컴포넌트, 그룹 멤버 컴포넌트, 멤버 관리 컴포넌트, 멤버십 관리 컴포넌트)로 구성된다. 다음은 각 컴포넌트에 대한 설명이다.

3.1 그룹 액세스 컴포넌트(Group Access Component)

그룹 액세스 컴포넌트는 클라이언트 프로세스가 그룹 통신 서비스에 접근하여 그룹으로 메시지를 보낼 수 있도록 하는 컴포넌트로서, 클라이언트에 대해 그룹 통신 서비스를 대표한다. 본 그룹 통신 서비스는 그룹으로 보내진 메시지들의 순서화를 위하여 벡터 타임 스탬프를 사용하는데, 메시지를 보내는 클라이언트 어플리케이션은 벡터 타임 스탬프 생성하여 메시지와 함께 보내야 한다. 다음의 메소드들을 지원한다.

Connect_Group(): 그룹명을 매개변수로 받아 멤버십 관리 컴포넌트로부터 그 그룹의 멤버십 정보(멤버들의 리스트)를 얻는다.

Get_Membership(): 현재의 그룹 멤버십 정보를 알고자 할 때 호출한다.

Send_Request(): 그룹 멤버들에게 호출을 보내는 것으로, 그 요청의 순서화 요구 수준을 기술할 수 있다. 전송되는 메시지는 메시지 ID, 클라이언트 ID, 멤버십 버전 등이 포함된다.

3.2 그룹 멤버 컴포넌트(Group Member Component)

어떤 응용 서비스를 제공하는 서버 어플리케이션 프로세스가 그룹의 멤버가 될 수 있도록 하는 컴포넌트로서, 그룹 통신 서비스와 멤버 프로세스간의 인터페이스가 된다. 즉, 어플리케이션 고유의 정보, 연산을 여러 어플리케이션을 포괄적으로 지원하는 그룹 통신 서비스를 통해 주고받을 수 있도록 하는 컴포넌트이다. 논리적으로는 멤버 프로세스를 대표한다. 다음의 메소드들을 지원하는데 이들은 주로 그룹 통신 서비스를 대표하는 멤버관리 컴포넌트가 호출한다.

Get_State()/Set_State(): 그 프로세스가 그룹의 멤버로 들어올 때 상태전송을 위해, 그룹의 상태를 전달, 수집하기 위한 목적으로 사용한다. 어플리케이션 고유의 데이터를 generic type으로 변환하여 그룹 통신 서비스로 보내거나 역의 작업을 할 수 있다.

Notify_ViewChange(): 그룹의 멤버십이 변경하였을 때, 소속된 멤버들에게 알려주기 위한 메소드이다.

3.3 멤버 관리 컴포넌트(Member Administrator Component)

특정 그룹의 멤버 프로세스들을 관리하는 컴포넌트이다. 그룹 멤버 컴포넌트로부터의 그룹 참여/탈퇴 요청을 받아 실질적인 멤버십 정보를 관리하는 멤버십 관리 컴포넌트에게 전달한다. 그룹 멤버 컴포넌트에 대해 그룹 통신 서비스를 대표한다. 그룹 통신 서비스로부터의 메시지(상태정보, 멤버십 변경 등)를 멤버 프로세스에게 전달하고, 멤버 프로세스가 그룹 통신 서비스로 메시지를 보내는 인터페이스가 된다.

멤버 관리 컴포넌트의 또 다른 중요한 기능은 메시지들에 대한 순서화 작업이다. 전순서화와 인과성 순서화를 지원한다. 기본적으로 순서화를 위한 큐(Hold-back queue라고 한다.)를 관리하여, 그룹 액세스 컴포넌트에서 생성하여 메시지에 실어 보낸 벡터 타임스탬프와 현재 멤버 관리 컴포넌트 자체가 관리하고 있는 벡터 타임스탬프를 비교하여 순서화 과정을 처리한다.[1,9] 다음의 메소드들을 지원한다.

Send_Request(): 그룹 접근 컴포넌트와 같은 기능을 한다.

Join_Group(): 매개변수로 그룹명을 넘겨, 멤버십 관리 컴포넌트가 관리하는 해당 그룹에 멤버 프로세스 등록하는 과정이다. 성공적인 경우, 그룹의 멤버십 정보와 상태정보를 전송하는 과정이 동반된다.

Leave_Group(): 그룹 멤버 컴포넌트가 그룹에서 탈퇴하기 위한 기능이다.

3.4 멤버십 관리 컴포넌트(Membership Management Component)

멤버십 관리 컴포넌트는 그룹 통신 서비스에 등록된 각 그룹들의 현재의 멤버십 구성에 대한 정보를 가지고 있는 컴포넌트로서, 고유의 그룹명을 그룹 뷰(멤버들의 리스트)로 맵핑시킨 테이블과 멤버 프로세스들이 공유하고 있는 그룹의 상태 정보를 관리한다. 그룹 뷰는 그룹 멤버 컴포넌트들의 리스트와 그룹 뷰 버전으로 구성된다. 멤버 관리 컴포넌트로부터의 그룹 참여/탈퇴 요청을 받아 멤버십 정보를 변경시키며, 그룹 액세스 컴포넌트나 멤버 관리 컴포넌트에게 현재의 멤버십 정보를 제공한다.

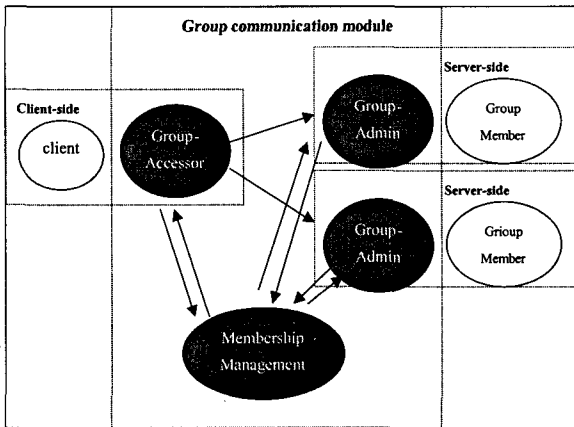
본 논문에서 개발한 그룹 통신 서비스에서, 그룹에 메시지를 보내고자(다중전송) 하는 클라이언트나 그룹의 새로운 멤버가 되고자 하는 프로세스는 이 멤버십 관리 컴포넌트를 통하여 멤버십 정보를 구할 수 있다. 일단 멤버십 관리 컴포넌트로부터 멤버십 정보를 얻은 클라이언트는 그 정보를 자신의 로컬 영역에 복사 시켜둠으로써, 그룹으로 메시지를 보내려 할 때, 더 이상 멤버십 관리 컴포넌트에 접근할 필요가 없

다. 그러나, 멤버십 구성이 런타임에 바뀔 수 있으므로 변경된 멤버십 정보를 update하는 메커니즘이 요구되는데, 현재 그 그룹에 속한 단 하나의 멤버 프로세스라도 요청을 받을 수 있다면 reply를 통해 잘못 알고 있는 멤버십 정보의 올바른 버전을 새로이 전달할 수 있다. 다만, 완전히 잘못된 멤버십 정보를 가지고 있어서, reply 조차 받지 못할 경우는 다시 멤버십 관리 컴포넌트에 접근하여 현재의 올바른 멤버십 정보를 받아올 수 있다. 멤버 프로세스에 대해서는, 멤버십이 변경될 때마다 그룹 통신 서비스에서 올바른 현재 멤버십 정보를 보내므로, 항상 일관된 멤버십 정보를 가지도록 보장한다. 다음의 메소드들을 지원한다.

Set_MbrshpChange() : 매개변수로 그룹명, 참여/탈퇴 여부를 나타내는 플래그를 전달 받아 해당 그룹의 멤버십 정보를 변경한다. 그 그룹의 멤버들에게 멤버십 변경 사실을 통보할 때, 그 통보 메시지는 동기 순서화 된다. 그룹에 새로이 참여하는 경우라면 그룹 멤버십 정보와 상태 정보를 리턴해 줘야 한다. 테이블에 기록되어 있지 않은 그룹명을 매개변수로 받았다면, 새로운 그룹을 만드는데, 그 첫번째 멤버는 master 멤버가 되어 전순서화 등에서 특별한 역할을 한다. 모든 멤버가 탈퇴한다면 그 그룹은 없어진다.

Set_MbrStateChange() : 상태 전송을 위한 상태를 수정하는 역할을 한다.

다음은 각 컴포넌트들간의 관계를 나타낸다.



3.5 구현

본 논문의 그룹 통신 CORBA 서비스는 OrbixWeb3.0과 JDK 1.1.6을 사용하여 구현되었다. 그룹 통신 서비스는 세 개의 CORBA sub-service(그룹 액세스 컴포넌트, 멤버 관리 컴포넌트, 멤버십 관리 컴포넌트)와 어플리케이션에서 상속을 받아야 하는 라이브러리(그룹 멤버 컴포넌트)로 구성되는데, 어플리케이션 개발자에게 제공되는 외부 인터페이스는 그룹 액세스 컴포넌트와 멤버 관리 컴포넌트, 그룹 멤버 컴포넌트이다. 클라이언트 어플리케이션은 CORBA의 다른 서비스를 이용할 때와 마찬가지로 바인딩을 통하여 그룹 액세스 컴포넌트와 멤버 관리 컴포넌트로 연결할 수 있다. 반면, 서버 어플리케이션은 그룹 멤버 컴포넌트 정의로부터 상속 받아 구현함으로써, 그룹 통신 서비스가 어플리케이션의 서버 프로세스를 CORBA 객체로서 다룰 수 있도록 한다. 또 하나의 CORBA sub service로 구현된 멤버십 관리 컴포넌트는 어플리케이션 프로그래머에게 공개되지는 않고 서비스 내부적으로 그룹 접근 컴포넌트나 멤버 관리 컴포넌트들이 바

인딩을 통하여 접근한다.

4 결론

본 그룹 통신 CORBA 서비스는 신뢰성, 가용성, 성능 등의 목적으로 프로세스 그룹 메카니즘의 효과를 원하는 많은 CORBA 기반 어플리케이션들을 지원하기 위해 개발되었으며, 사용자가 다양한 순서화 요구(인파 순서화, 전순서화, 순서화 없음)를 선택할 수 있도록 하였다.

차후 계획으로, 이 그룹 통신 CORBA 서비스를 발전시켜, 예고 없는 프로세스 오류(failure), 네트워크 분할(partition)를 감지하여 안전하게 멤버십으로부터 해당 프로세스를 제거함으로써, 결함 감내 능력을 가진 신뢰성 있는 서비스를 개발할 계획이다.

참고문헌

- [1] George Coulouris, Jean Dollimore, and Tim Kindberg, Distributed Systems Concepts and Design. Second edition, Addison Wesley, 1994
- [2] Isis distributes systems, inc. Object Groups : A response to the ORB 2.0 RFI, 111 South Cayuga Street, Suite 200. Ithaca, NY 14850, Apr. 23, 1993
- [3] Pascal Felber, Benoit Garbinato, Rachid Guerraoui, The Design of a CORBA Group Communication Service, Ecole Polytechnique Federale de Lausanne, Departement dInformatique, CH-1015 Lausanne(Switzerland)
- [4] Pascal Felber, The CORBA Object Group Service : A service Approach to Boject Groups in CORBA, Ecole Polytechnique Federale De Lausanne, Pour Lobtention Du Grade De Docteur Es Sciences, 1998
- [5] P. Narasimhan, L. E. Moser and P. M. Melliar-Smith, "Exploiting the Internet Inter-ORB protocol interface to provide CORBA with fault tolerance," Proceedings of the Third USENIX Conference on Object-Oriented Technologies and Systems, Portland, OR(June 1997), pp. 81-90
- [6] P. Narasimhan, L. E. Moser and P. M. Melliar-Smith, "Exploiting the Internet Inter-ORB protocol interface to provide CORBA with fault tolerance," Proceedings of the Third USENIX Conference on Object-Oriented Technologies and Systems, Portland, OR(June 1997), pp. 81-90
- [7] IONA Technologies PLC, OrbixWeb Programmers Guide, Nov. 1997
- [8] Robert Orfali, Dan Harkey, Client/Server Programming with JAVA and CORBA. Second edition, Jone Wiley & Sons, INC., 1998
- [9] Kenneth Birman, Lightweight Causal and Atomic Group Multicast. ACM Transactions on Computer Systems, Vol. 9, No. 3, Aug. 1991, pages 272-314
- [10] L. Lamport, Time, clocks, and the ordering of events in a distributed system. Commun. ACM, vol. 21, No. 7, July 1978, pages 558-565
- [11] Kenneth P. Birman, The Process Group Approach to Reliable Distributed Computing, Commun. ACM, Dec. 1993, Vol. 36, No. 12, pages 37-53
- [12] Robbert van Renesse, Kenneth p. Birman, silvano maffei, Horus : A Flexible Group Communication System, Commun. ACM, Vol. 39, No. 4, Apr. 1996, pages 76-83