

CORBA 환경에서 실시간을 지원하는 객체 그룹 서비스 설계

안계호*, 정진섭*, 이재완*
*군산대학교 전자정보공학부
e-mail:khahn@kunsan.ac.kr

A Design of Object Group Service for Supporting Real-Time in CORBA Environments

Kye-Ho Ahn*, Jin-Seop Jung*, Jae-Wan Lee*
*School of Electronics and Information, Kunsan National
University

요약

분산 이기종 시스템간에 어플리케이션을 개발하기 위해 미들웨어로 사용되는 CORBA는 객체들 간에 멀티캐스트 통신을 지원하지는 못한다. 그룹서비스는 일대일 통신만을 지원하는 CORBA에서 멀티캐스트를 지원할 뿐만 아니라 시스템의 가용성, 신뢰성 그리고 성능을 향상시킬 수 있다. 본 논문에서는 CORBA를 기반으로 실시간 스케줄링과 부하균등 알고리즘을 통하여 실시간 응용프로그램에 적합한 그룹서비스를 설계한다.

1. 서론

과거에는 일부 특수한 분야에만 사용되던 실시간 시스템이 멀티미디어의 보급과 사용자들의 실시간 욕구를 충족시키기 위해 많은 분야에 적용됨에 따라 실시간 CORBA에 대한 많은 연구들이 행해지고 OMG에서는 실시간 CORBA의 표준안을 내놓았다 [1]. 이에 따라 CORBA의 핵심이 되는 ORB를 수정하거나 기존의 CORBA 그룹 서비스를 이용하여 실시간을 지원하도록 하는 방향의 연구들이 행해지고 있다[2][3].

따라서, 본 논문에서는 분산된 객체들을 효율적으로 관리할 수 있고, 신뢰성과 병렬성을 제공하여 소프트웨어 개발과 분산시스템의 성능을 향상시킬 수 있는 CORBA를 기반으로 실시간 객체그룹서비스를 제시하며 본 논문에서 제안하는 시스템은 특정한 시스템에 의존하지 않으며 그룹서비스에서 제공되는 병렬성과 부하균등 알고리즘을 통해 처리율과 신뢰성을 향상시킬 수 있다.

2. 관련연구

2.1 객체그룹서비스

분산된 객체들을 개별적으로 관리하려면 객체들을 관리하는데 매우 복잡하다. 따라서 분산된 객체들을 하나의 객체그룹으로 관리함으로써 객체들을 관리하는데 복잡함을 줄일 수 있다. CORBA는 객체그룹의 갑 멤버들의 네트워크 식별자, 환경, 위치정보를 알고 있을 필요가 없이 객체그룹에 요구를 발생함으로써 모든 그룹멤버들에게 단일요구를 발생할 수 있다.

기존의 CORBA를 기반으로 그룹통신을 추가하는 방법은 현재 존재하는 그룹통신 툴킷을 ORB에 통합하는 통합방법(integration approach), 과 CORBA 서비스처럼 객체그룹 서비스를 구축하여 서비스하는 방법(service approach) 그리고 그룹통신 시스템의 메시지에 해당하는 IIOP 메시지를 가로채서 복제 매니저에게 전송하는 가로채기에 의한 방법(interception approach)이 있다[4][5][6].

2.2 실시간 스케줄링 알고리즘

실시간 시스템에서의 스케줄링은 시스템의 성능을 좌우하는 중요한 부분이다. 대부분의 실시간 스케줄링 알고리즘은 우선 순위를 기본으로 하고 있으며 그 시스템이나 어플리케이션에 적합한 우선순위 할당방법을 사용한다. 다중프로세서 상에서 대표적인 알고리즘으로는 LLA, EDZL, EDF 등을 들수 있다 [7][8].

LLA(Least Laxity Algorithm)은 계속 동적으로 변하는 laxity값에 의하여 스케줄링 되므로 문맥교환 오버헤드가 크다는 단점이 있지만 다중프로세서 시스템에서 좋은 스케줄링 성능을 갖고 있다고 알려져 있다[9].

EDZL(Earliest Deadline Zero Laxity Algorithm)은 기본적으로 EDF에 기초하고 있고 위급한 상황이 발생시 laxity값에 의해 스케줄링한다.

EDF(earliest Deadline First)는 마감시간을 기초로 타스크들의 스케줄링이 이루어지는 방법으로 타스크 중에서 마감시간이 짧은 타스크를 선정하여 처리하므로 문맥교환이 적게 일어난다. 모든 프로세서 이용률이 1 이하이면 타스크들이 필요로 하는 시간의 총량이 프로세서 시간보다 작아지므로 마감시간을 넘기지 않는 스케줄링이 가능하다.

3. 실시간 객체그룹의 구성

실시간 객체 그룹은 그룹 관리를 위한 그룹관리자 객체, 그룹 멤버들간의 동기화를 위한 타이머객체, 팩토리객체, 서버의 상태감시와 부하조절을 위한 모니터객체, 작업분배와 시퀀스부여를 위한 분배자 객체, 클라이언트와 서버를 그룹과 연결하기 위한 프록시 객체로 구성된다, 시스템의 구성도는 다음 그림과 같다.

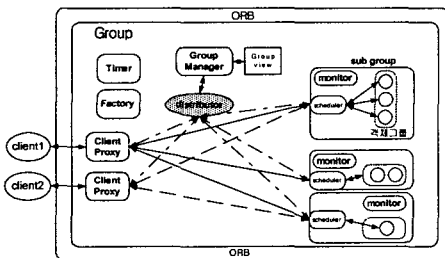


그림 1 시스템 구성

3.1 구성요소

그룹관리자(Group Manager)는 그룹을 초기화하며 객체 그룹을 관리하며 그룹관리를 효율적으로 관리하기 위해 뷰를 생성하고 관리하며 팩토리 객체와 상호작용하여 그룹의 요소들을 생성하고 삭제를 수행한다. 타이머객체는 서비스의 일관성 유지와 동기화를 위한 시간을 제공한다. 모니터객체는 서버의 부하상태를 감시하고 서버에 대한 부하조절의 역할을 하며 서버의 상태를 감시하여 그룹관리자가 뷰를 일관성있게 유지할 수 있도록 한다. 분배자는 클라이언트들의 메시지들에 시퀀스를 부여하여 그룹내 메시지들에 대한 순서성을 제공한다. 클라이언트 프록시는 클라이언트의 요청을 분배자에게 전달하는 역할을 하며 이 객체를 통해 클라이언트는 그룹을 하나의 객체처럼 인식하게 된다. 서브그룹내의 스케줄러는 실시간 스케줄러로서 그룹내에서 처리해야할 메시지들에 대한 스케줄링을 행하여 그룹 멤버들에게 전달한다. 스케줄링은 실시간 프로세스들과 비실시간 프로세스들에 대한 스케줄링을 포함한다.

3.2 스케줄링

작업이 정확하게 마감시간 전까지 수행되기 위해서는 클라이언트, 그룹, 서버간에 동기화가 이루어져야 한다. 본 논문에서 제안한 시스템에서는 클라이언트가 타임서버에 접속하여 전역시간을 요청하여 얻어오는 Cristian 알고리즘[10]을 사용한다. 그룹에 접속한 멤버는 그룹으로 메시지를 보내고 그룹의 타이머객체는 현재 서버시간 t 를 메시지에 포함하여 다시 멤버에게 전송하면 멤버는 메시지에 대한 round-trip 시간과 t 를 이용하여 서버의 시간을 추정하고 현재시간을 조정한다.

$$T = t + T_{round}/2$$

메시지를 수신한 서버 그룹에서는 메시지들을 실시간 스케줄링 알고리즘에 의해 스케줄링한다. 실시간 스케줄링 알고리즘으로는 슬랙 스틸링(slack stealing) 알고리즘을 사용한다. 즉, 실시간 프로세스들은 마감시간을 지키는 범위 안에서 최대한 수행이 연기되고 이때 생기는 슬랙(slack)시간동안에 비실시간 프로세스들을 수행함으로써 비실시간 프로세스들에 대한 반환시간(turn-around time)을 줄인다.

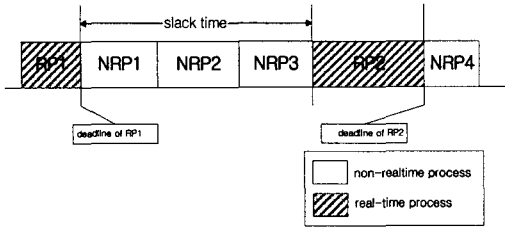


그림 2 slack stealing algorithm

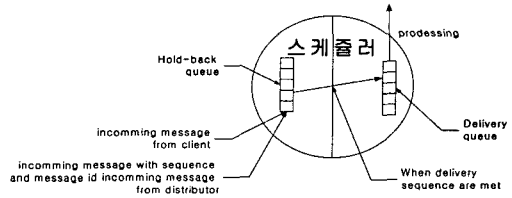


그림 4 스케줄러의 구조

```

algorithm for scheduler :
    if (exist arrived task);
        if (task is real-time process) {
            computation slack time;
            queuing task in queue;
        } else {
            queuing task;
        }
    }

algorithm for group member p:
    while ( exist tasks) {
        if ( exist slack time) {
            while(until end of current slack-time) {
                processing non-realtime process;
                if(slack time size is not enough)
                    current slack-time = END;
            }
            processing realtime processing;
        } else {
            processing non-realtime process;
        }
    }
    }
    
```

그림 3 스케줄러와 그룹멤버에서 슬랙스틸링 알고리즘

```

1. Algorithm for group member p
On initialization:  $r_g := 0$ ;
To multicast message m to group g
    multicast( $g \cup \{sequencer(g)\}, \langle m, d, i \rangle$ );
On deliver( $\langle m, i \rangle$ ) with  $g = group(m)$ 
    place  $\langle m, i \rangle$  in hold-back queue;
On deliver( $\langle i, S, IDs \rangle$ ) with  $g = group(m)$ 
    wait until  $\langle m, d, i \rangle$  in hold-back queue
    and  $S = r_g + 1$ ;
    if ( $IDs = MyID$  or  $IDs = NULL$ ) then deliver m;
    else delete( $\langle m, d, i \rangle$ );
     $r_g = S$ ;
On deliver( $\langle m, i \rangle$ ) with group member p
    deliver m;

2. algorithm for sequencer of g
On initialization:  $s_g := 0$ ;
On deliver( $\langle m, i \rangle$ ) with  $g = group(m)$ 
    Multicast sequence and server's identifier:
        multicast( $g, \langle i, s_g \rangle$ );
    Increase of sequence:
         $s_g := s_g + 1$ ;
    
```

표 3 total ordering algorithm

4. 결론 및 향후 연구과제

3.3 메시지 ordering

메시지 순서화를 위해서 시퀀서(sequencer)와 홀드백(hold-back) 큐 기법을 사용한다. 메시지를 수신한 스케줄러들은 시퀀서 객체로부터 메시지에 대한 시퀀스가 도착할 때까지 메시지를 홀드백 큐에 보관한다. 시퀀서 객체에서는 메시지를 수신한 후 메시지에 대한 시퀀스(s_g)를 부여하고 멤버들에게 메시지 식별자(i)와 메시지의 시퀀스(s_g)를 멀티캐스트를 한다. 각 서브그룹의 스케줄러 객체들이 이 메시지를 수신하게 되면 순신된 메시지 식별자를 갖는 메시지를 수신큐로 옮기고 메시지를 스케줄링한다.

분산 컴퓨팅 환경은 통신망 관리의 복잡성을 해소하고 자원을 효율적으로 관리하며 어플리케이션 개발자가 분산 어플리케이션을 설계하고 개발할 때 편의를 제공하기 위해 TINA 와 CORBA 등의 표준화된 방법을 사용하고 있다. 이러한 분산 환경 하에서 복잡한 분산 소프트웨어의 개발 및 관리의 복잡성을 줄이고 분산된 객체들을 효율적으로 관리하면서 실시간 서비스의 요구사항을 충족시키기 위한 실시간 객체그룹의 정의가 필요하다. 따라서 본 논문에서는 객체그룹을 CORBA를 기반으로 설계하였으며 실시간 서비스 수행을 위해 시간 제약조건을 만족시킬 수 있는 실시간 스케줄링과 부하분산 알고리즘을 사용하여 분산시스템의 신뢰성, 안정성 그리고 성능을 향상하고자 하였다. 본 논문에서 제안된 그룹 서비스 시스템은 특정한 운영체제에 제약받지 않고 서비스 형태로 이루어지기 때문에 분산된 객체

들의 효율적인 관리와 어플리케이션 개발을 용이하도록 하고 분산시스템의 성능을 향상시킬 수 있다. 향후 연구과제로는 제안한 시스템의 구현과 다양한 실시간 알고리즘과 부하균등 알고리즘을 적용하여 보다 나은 성능을 갖는 시스템을 구축하고 실시간 멀티미디어 서비스를 제공할 수 있는 어플리케이션으로 응용하겠다.

5. 참고문헌

- [1] OMG, "The Common Object Request Broker : Architecture and Specification v2.4.1", October 2000, <ftp://ftp.omg.org/pub/docs/formal/00-11-03.pdf>
- [2] Victor Fay Wolfe, Lisa C. DiPippo, Gregory Cooper, Russell Johnston, Peter Kortmann, Vhavani Thuraisingham, "Real-Time CORBA", IEEE Transactions on Parallel and Distributed Systems, VOL. 11, NO. 10, Oct 2000
- [3] L. DiPippo, V.F. Wolfe, R. Johnston, R. Ginis, M. Squadrito, S. Wohlever, I. Zyxh, "Expressing and Enforcing Timing Constraints in a Dynamic Real-Time CORBA System", Real-Time Systems, vol. 16, nos.2/3, May 1999
- [4] Silvano Maffeis, "Adding Group Communication and Fault-Tolerance to CORBA", Proceedings of the USENIX Conference on Object Oriented Technologies (COOTS) Monterey, California, June 1995.
- [5] P. Narasimhan, L. E. Moser, 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, June 1997
- [6] P. Felber, B.Garbinato, R.Guerraoui, "The Design of a CORBA Group Communication Service", Proceedings of the 15th IEEE Symp. on Reliable Distributed Systems, Oct. 1996, pp.150-159.
- [7] J.P. Lehoczky and S. Ramos-Thuel, "An Optimal Algorithm for Scheduling Soft-Aperiodic Task in Fixed-Priority Preemptive Systems", Proceeding of the IEEE Real-Time System Symposium, Dec. 1992, pp.110-123.
- [8] L. Sha, B. Sprunt and J. P. Lehoczkey, "Aperiodic Task Scheduling for Hard Real-Time System", The Journal of Real-Time System 1989, pp.27-69.
- [9] 이석균, 임준택, "실시간 온라인 스케줄링 알고리즘들의 분석 및 성능비교", '95 한국 정보과학회 봄 학술 발표 논문집, Vol.11, No.1, 1995, pp.383-386.
- [10] George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems Concept and Design" third edition, Addison-Wesley Publishing Company Inc., 2001., pp 389-393
- [16] George Coulouris, Jean Dollimore, Tim Kindberg, "Distributed Systems Concept and Design" third edition, Addison-Wesley Publishing Company Inc., 2001., pp 419-450