그룹의 통합과 분리를 고려한 상호배제 알고리즘

윤재희°, 김재훈
아주대학교 정보통신전문대학원
jhyoon@dmc.ajou.ac.kr° jaikim@ajou.ac.kr

Mutual Exclusion Algorithm for Group Integration/Disintegration

Jae-Hee Yoon°, Jai-Hoon Kim
Graduate School of Information and Communication, Ajou University

요 약

컴포넌트 기반 미들웨어에서 기본 단위를 이루는 컴포넌트가 특정 동일 목적을 위하여 그룹을 생성하고 정해진 서비스를 제공할 수 있다. 이렇게 생성된 그룹은 멤버들간에 리소스를 공유하는데, 이에 따른 상호배제(mutual exclusion) 문제가 그룹 멤버들 사이에서 발생한다. 커뮤니티 컴퓨팅과 같은 그룹 서비스에서는 환경이 다양하고 변화가 심하기 때문에 그룹의 구조 또한 생성, 삭제, 통합, 분리가 빈번히 일어난다. 본 논문에서는 분산 시스템에서의 상호배제 알고리즘을 컴포넌트 기반 미들웨어에서 생성된 그룹에 적용하고, 상호배제 기법이 적용된 상태에서 그룹이 통합하거나 분할될 때 상호배제의 일치성을 유지할 수 있는 방법을 제시하였다.

1. 서론

커뮤니티 컴퓨팅을 위한 컴포넌트 기반 미들웨어에서 기본 단위를 이루는 컴포넌트가 특정 동일 목적을 위하여 그룹을 생성하고 서비스를 제공할 수 있다. 이러한 그룹의 형태는 여러 가지로 나타날 수 있다. 사회적인 관점에서 유저를 그룹으로 묶을 수 있고 [1], 서비스를 중심으로 하여 컴포넌트들을 그룹으로 묶을 수 있다. 또 다른 경우로, 유저가 사용하고 있는 디바이스를 그룹으로 묶을 수 있다. 본 논문에서는 공통된 목적을 가지고 일련의 서비스를 제공하는 컴포넌트들의 집합을 그룹이라 정의한다.

대부분 그룹에 대한 연구는 그룹 멤버쉽(Group Membership) [1][2] 과 그룹 통신(Group Communication)[3][4]에 중점을 두고 있다.

그룹의 멤버들(컴포넌트)은 서비스를 제공하기 위해 리소스를 공유하여 사용한다. 다시 말해 리소스를 공유하고 있는 컴포넌트들은 같은 그룹에 속해있다. 여기에서 리소스는 그룹에 참여한 컴포넌트 자체가 될 수도 있고 혹은 그 컴포넌트가 사용하고 있는 리소스가 될 수도 있다. 그룹 멤버들은 이러한 리소스를 공유하기 때문에 그룹 안에서 그룹 멤버들 간의 상호배제 문제가 발생한다.

그룹의 구성 단위는 컴포넌트뿐만 아니라 그룹 자체가 될 수 있다. 유비쿼터스 환경에서 커뮤니티 컴퓨팅을 위한 그룹은 환경과 서비스 상태가 다양하고 변화가 심하기 때문에 공통된 목적을 가진 그룹이 통합하여 하나의 그룹을 이룰 수 있고 반대로 공통된 목적을 상실한 그룹은 여러 개의 그룹으로 나뉘어져 각각의 그룹 업무를 수행할 수 있다. 그룹 안에서 상호배제 알고리즘이 적용된 상태에서 이러한 그룹간 통합/분할이 일어날 수 있다. 이러한 경우 그룹간 통합/분할 전후 상태에서 일관성을 유지하는 기법이 필요하다.

CORBA를 기반으로 한 그룹 통신[3]과 표준 인터페이스를 기반으로 한 그룹 통신[4]등 그룹 통신에 관하여는 많은 연구가 진행되었다. 따라서 본 논문에서는 그룹의 통신 방법에 대해서는 논하지 않는다. 본 논문에서는 커뮤니티 컴퓨팅을 위한 그룹에서의 상호배제 문제를 해결할 수 있는 기법을 제안하고 상호배제 기법이 적용된 상태에서 그룹이 통합하거나 분할한 후 일관성있게 상호배제를 보장하는 방안에 대해 제안하다. 이러한 가변적인 그룹의 통합/분할은 유비쿼터스 환경에서의 커뮤니티 컴퓨팅에서 일반적으로 발생할 수 있기 때문에 여러 응용분야에서 유용하게 사용될 것이다.

본 논문은 다음과 같은 내용으로 구성된다. 2장에서 관련연구를 살펴본 후 3장에서 본 논문에서 제안한 기법을 설명한다. 마지막으로 4장에서는 결론과 향후 연구 방향에 대해 기술한다.

*본 연구는 21세기 프론티어 연구개발사업의 일환으로 추진되고 있는 정보통신부의 유비쿼터스컴퓨팅 및 네트워크원천기반기술개발사업의 지원에 의한 것임

2. 관련연구

분산 시스템에서 상호배제를 해결할 수 있는 방법은 크게 두 가지로 나뉘어 질 수 있다[5]. 코디네이터가 있는 중앙집중알고리즘과 코디네이터가 존재하지 않는 분산알고리즘이다.

중앙집중알고리즘에서는 임계영역(critical region)의 사용을 원하는 프로세스가 코디네이터에게 요청메시지(request)를 보내서 응답을 받으면 임계영역에 들어갈 수 있다. 만약에 임계영역에 이미 다른 프로세스가 들어가 있다면 요청메시지를 보낸 프로세스는 큐에 놓여지게 된다. 임계영역에 있던 프로세스가 사용을 마치면 코디네이터는 큐의 첫 번째에 놓인 프로세스에게 응답을 함으로써 큐에서 대기하고 있던 프로세스가 큐에 들어가게 된다. 이러한 방법은 구현하기 쉽고 요청을 하는 순서대로 임계영역에 들어갈 수 있어 프로세스에게 공정한 기회를 주며 영원히 기다리는 프로세스가 생기지 않는다. 하지만 코디네이터로 요청이 몰리는 병목현상이 일어날 수 있으며 코디네이터에서 장애가 발생하면 시스템 전체가 다운되어 버리는 단점이 있다.

반면 분산알고리즘에서는 코디네이터가 존재하지 않으므로 중앙집중알고리즘보다 복잡하다. 임계영역에 들어가기 원하는 프로세스는 다른 모든 프로세스에게 요청메시지를 보내고 이들로부터 grant를 받아야 임계영역에 들어갈 수 있다. 중앙집중알고리즘과는 달리 한 프로세스의 장애가 시스템 전체의 장애를 유도하지 않는다.

그룹 통합/분할을 고려한 상호배제 알고리즘은 분산알고리즘을 기반으로 한다.

3. 그룹 상호배제 문제와 그룹간의 통합/분할

3.1 시스템 구조

그룹의 상호배제에 대해 설명을 하기 전에 우리는 다음과 같이 몇 가지 시스템 구조를 가정한다.

1) 그룹에는 그룹을 관장하는 그룹 매니저(Group Manager)라는 컴포넌트가 존재한다. 그룹 매니저는 그룹을 생성, 삭제, 통합/분할을 관리하는 역할을 담당한다.
2) 그룹 멤버쉽은 중앙집중방식으로 관리된다. 그룹 매니저가 멤버의 아이디 등 멤버의 정보를 담고 있는 멤버 프로파일(Member Profile)을 관리한다.
3) 그룹의 통신에 대해서는 이 논문에서 언급하지 않는다. 대신, 그룹 매니저가 받은 유니캐스트로 받은 메시지를 멀티캐스트로 그룹 멤버 전체에게 전송을 해 주며 이 전송은 신뢰할 수 있다고 가정을 한다. 모든 멤버에게 메시지를 전송을 해야 하는 경우 그룹 매니저를 통하여 메시지를 전송한다. 그룹 매니저는 받은 메시지를 그룹 멤버에게 멀티캐스팅을 해줌으로써 멤버자신이 모든 멤버에게 메시지를 보낼 때 발생하는 오버헤드를 줄일 수 있다.
4) 각 그룹이 사용하고 있던 리소스도 그룹이 통합함에 따라 하나의 리소스로 통합될 수 있다. 예를 들어 서로 다른 파일을 사용중인 그룹 A와 B가 통합을 한다면 각 그룹이 사용하던 파일도 하나로 통합된다. 반대로 그룹이 분할할 때에는 리소스도 나뉘어 진다.

3.2 그룹 멤버들간의 상호배제

그룹의 멤버들은 리소스를 공유한다. 따라서 여러 멤버가 동시에 하나의 리소스에 접근할 수 있게 되고 이에 따라 리소스 접근 충돌이나 잘못된 데이터의 조작이 발생할 수 있다. 이러한 그룹 내에서의 상호배제문제를 분산알고리즘 방법을 사용하여 해결할 수 있다.

임계영역에 들어가기를 원하는 멤버들은 요청메시지에 각각이 들어가기를 원하는 임계영역의 이름과 자신의 아이디, 요청메시지를 보내는 현재 시간을 포함한다. 요청메시지를 보낸 멤버는 그룹의 모든 멤버로부터 OK 메시지를 받아야만 임계영역에 들어갈 수 있다. 요청메시지를 받은 멤버(수신자)는 메시지에 포함된 임계영역의 이름을 확인하고 그 임계영역과 관련된 자신의 상태를 확인하는데 다음 세가지 상태에 따라 수행한다.

1) 수신자가 이미 그 임계영역에 들어가 있다면 어떤 응답도 하지 않고 요청메시지를 보낸 멤버를 임계영역의 큐에 놓는다.
2) 수신자가 임계영역에 들어가지 않았고 임계영역에 들어가려고 요청메시지를 보내놓은 상태도 아니라면 요청메시지를 보낸 멤버에게 OK 메시지를 보낸다.
3) 수신자가 아직 그 임계영역에 들어가지는 않았지만 그 임계영역에 들어가려고 요청메시지를 보내고 응답을 기다리고 있는 상태라고 한다면 자신이 보낸 요청메시지의 타임스탬프와 다른 멤버로부터 받은 요청메시지의 타임스탬프를 비교하여 더 빠른 타임 스탬프를 가진 요청메시지를 보낸 멤버에게 OK 메시지를 보낸다.

그림1은 그룹의 멤버들 사이에서 상호배제가 어떻게 보장되고 있는지를 보여준다. 그림 1-(a)를 살펴보면 멤버 c4와 c5가 그룹의 모든 멤버들에게 동시에 요청메시지를 보냈다. 요청메시지의 타임스탬프를 비교해서 보면 c5가 c4보다 타임스탬프가 앞서 있으므로 c5가 임계영역에 들어가게 되고 c4의 요청메시지는 임계영역 큐에 들어가게 된다. (그림1-(b)) 나중에 c5가 임계영역에서 나오면서 임계영역 큐에 있는 메시지를 보낸 송신자에게 OK 메시지를 보낸다. (그림1-(c)) OK 메시지를 받은 c4는 임계영역에 들어갈 수 있게 된다.

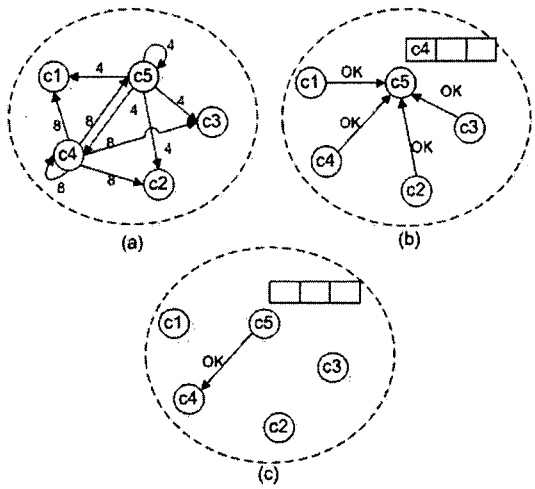

그림1. 그룹 내에서의 상호배제

3.3 상호배제와 그룹의 통합/분할

그룹의 구성(configuration)단위는 컴포넌트뿐만 아니라 그룹 자체가 될 수 있다. 따라서 하나의 공통된 서비스를 제공하기 위해 두 개 이상의 그룹이 통합하는 경우가 생긴다. 반대로 공유하고 있던 공통된 목적을 상실하거나 역할이 분담됨으로써 두 개 이상의 그룹으로 나뉘어 지는 경우도 있다. 그룹의 통합과 분할은 각 그룹의 멤버들의 통합/분할과 더불어 각 그룹이 공유하고 있던 리소스들의 통합/분할도 포함한다. 이때, 임계영역에 들어가있는 그룹 멤버는 그룹의 통합과 분할에 영향을 받지 않고 진행중인 작업을 마칠 수 있어야 하며 각 임계영역의 큐도 그룹의 통합/분할 전후 상황에 대한 일관성을 유지해야 한다.

3.3.1 그룹의 통합

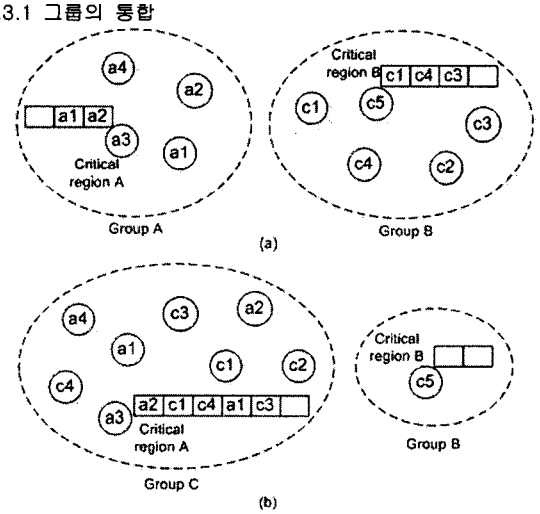

그림2. 그룹A와 그룹B가 통합되기 전과(a) 통합되는 과정(b)

그림2-(a)는 두 개의 그룹이 통합되기 전의 상태를 나타낸다. 각 그룹은 그룹 내에서의 상호배제 문제를 해결하기 위해 앞절에서 언급한 기법을 사용하고 있다. 각 그룹에서 a3와 c5가 임계영역을 사용하고 있으며 a1, a2와 c1, c3, c4가 각 그룹의

임계영역 큐에 놓여있다.

그룹의 통합은 한 그룹(통합 요청그룹)이 다른 그룹에게 통합요청을 했을 때 요청을 받은 그룹(기준그룹)이 그 요청을 수락함으로써 시작된다. 그룹이 통합할 때 상호배제의 일관성은 아래와 같이 유지되며 O(NlogN)의 복잡도를 가진다.(N은 통합된 그룹의 전체 멤버 수이다.)

1) 각 그룹에서 임계영역을 사용하고 있는 멤버가 있을 때
   A. 통합요청 그룹에서 임계영역을 사용중인 멤버를 제외한 멤버들의 통합과 임계영역 큐의 통합이 먼저 이루어 진다. 각 그룹의 임계영역 큐에 놓여있던 요청메시지의 타임스탬프를 비교하여 빠른 순서대로 큐에 놓는다. 그림 2-(b)에서 통합 요청그룹(그룹 B)에서 임계영역을 사용하고 있는 멤버(c5)를 제외한 나머지 멤버들의 통합과 임계영역 큐의 통합이 이루어 졌다. 기준그룹에서 임계영역을 사용중인 멤버(a3)와 통합 요청그룹에서 임계영역을 사용중인 멤버(c5)는 임계영역을 계속 사용한다.
   B. 통합 요청그룹에서 임계영역을 사용하고 있던 멤버(c5)가 임계영역 사용을 마치면 통합된 그룹(그룹 C)에 참여한다. 통합된 그룹에서 임계영역을 사용중인 멤버(a3)도 사용을 마치면 기존 두 그룹의 임계영역을 통합한다.(그림3) 임계영역을 통합 한 후 임계영역을 사용하고 있던 멤버는 하던 작업을 계속 진행할 수 있다.
2) 한 그룹에서는 임계영역을 사용하고 있고 다른 그룹에서는 임계영역을 사용하고 있지 않을 때
   A. 임계영역을 사용중인 멤버를 제외한 멤버들의 통합과 임계영역 큐의 통합이 이루어 진다.
   B. 임계영역을 사용하던 멤버가 임계영역의 사용을 마치면 통합된 그룹에 참여하게 되고 임계영역의 통합도 이루어 진다. 이 후 임계영역 큐에 놓여있던 첫 번째 멤버가 임계영역을 사용할 수 있다.

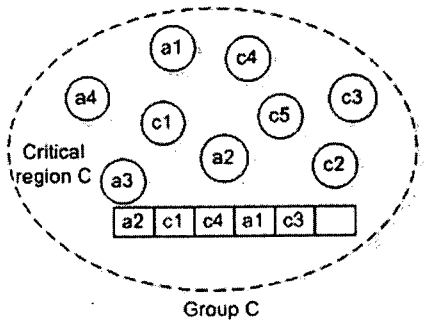

그림3. 그룹 통합 후의 상태

3.3.2 그룹의 분리

하나의 그룹이 두 개의 그룹으로 나뉘어 질 때에도 그룹의 통합과 비슷한 과정을 거치며 O(NlogN)의 복잡도를 가진다. (N은 분할될 그룹의 전체 멤버 수이다.)

1) 나뉘어지는 그룹에 따라 그룹의 멤버를 가른다.
2) 임계영역 큐에 놓여있는 멤버를 나뉘어진 멤버들과 비교하여 해당하는 그룹의 임계영역 큐로 복사한다. 이때에도 타임스탬프를 비교하여 빠른 순서대로 정렬한다.
3) 분할되기 전에 임계영역을 사용하던 멤버가 하던 작업을 그대로 진행할 수 있도록 분할 되어 생긴 그룹의 임계영역은 기존 임계영역에서 해당부분만 복사됨으로써 생성된다. 그룹의 분할로 생긴 임계영역 큐의 첫 번째에 놓여있는 멤버들은 분할되어 생긴 임계영역을 사용할 수 있다.
4) 기존 임계영역을 사용하고 있던 멤버가 사용을 마치면 기존 임계영역에서 분할되어 다른 그룹으로 이전된 부분이 삭제됨으로 임계영역의 분할이 적용된다.

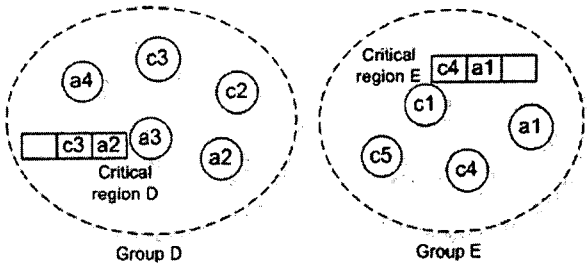

그림4. 그룹 분할 후의 상태

예를 들어, 영상서비스를 제공하는 그룹 A와 음성서비스를 제공하는 그룹 B(그림2-a)가 통합하여 동영상 서비스를 제공하는 그룹 C(그림3)가 되었다. 그룹 C가 서비스 제공을 마친 후 코덱(신호압축이나 해독)서비스를 제공하는 그룹(그림4의 그룹 D)과 필터(신호분리)서비스를 제공하는 그룹(그림4의 그룹 E)으로 나뉘어 진다고 가정을 하자. 이때, 코덱 서비스를 위한 멤버들(a2, a3, a4, c2, c3)과 필터서비스를 위한 멤버들(a3, c1, c4, c5)로 나뉘어 진다. 임계영역의 큐 또한 멤버들을 따라 나뉘어 지며 타임스탬프가 빠른 순서대로 큐에 놓이게 된다. 그룹 E의 임계영역 E가 a3가 사용하고 있던 임계영역 C에서 해당부분만 복사됨으로써 생성되며 그룹 E에서 큐에 첫 번째로 놓여있던 c1이 임계영역 E를 사용하게 된다. (그림4) a3가 임계영역의 사용을 마치면 임계영역 D로 복사된 부분이 임계영역 C에서 삭제됨으로써 임계영역의 분할이 적용되어 큐에 있던 첫 번째 멤버는 임계영역 D를 사용하게 된다.

4. 결론 및 향후 과제

본 논문에서는 컴포넌트 기반의 미들웨어에서 그룹 서비스를 사용할 때 발생할 수 있는 상호배제 문제를 해결하고자 하였다. 그룹이 통합/분할할 때 임계영역 큐를 통합/분할하면서 각 메시지의 타임스탬프를 비교하여 재정렬하고 통합/분할할 때의 임계영역 사용여부에 따라 임계영역의 통합 시기를 다르게 함으로써 그룹의 통합 전후로 상호배제 상황이 일관성을 유지하도록 하였다. 향후, 제안된 기법의 세부적인 구현과 커뮤니티 컴퓨팅을 위한 그룹 재구성에 관한 연구가 진행될 것이다.

5. 참고문헌

[1] Bin Wang, J Bodily, and SKS Gupta, "*Supporting Persistent Social Groups in Ubiquitous Computing Environments using Context-aware ephemeral group service*", PerCom 2004, Mar.2004

[2] Bottazzi D, Corradi A, Montanari R, *"AGAPE: a location-aware group membership middleware for pervasive computing environments"*, ISCC 2003, July.2003

[3] Y.Joe, D.Lee, and D.Nam, *"A Transparent Object Group Reference Management Scheme for Group Communication in CORBA"*, in Addendum to the Proceedings of Confederated International Conferences CoopIS, DOA, and ODBASE 2002, Oct. 2002

[4] Matthias Wiesmann, Xavier Defago, Andre Schiper, *"Group Communication based on Standard Interfaces"*, IEEE NCA 2003, Apr.2003

[5] Distributed System. Principles and Paradigms, Andrew S.Tanenbaum, Maarten van Steen