

분할가능 분산환경에서의 신뢰성 있는 자바 프로세스 그룹 서비스†

최혁재* · 문남두 · 김현규 · 박양수 · 이명준
울산대학교 전자계산학과

A Reliable Java Process Group Service under Partitionable Distributed Environments

*Hyukjae Choi · Nam-Doo Moon · Hyun-Gyu Kim · Yang-Su Park · Myung-Joon Lee
Dept. of Computer Science, University of Ulsan

요 약

순수자바기술을 이용한 응용서비스가 정보통신망의 발달과 함께 빠른 속도로 증대되고 있다. 이러한 자바용 서비스는 네트워크 분할(partition) 및 서버의 실패(crash)와 같은 전산망의 결함에도 불구하고 안정적이고 지속적인 서비스를 제공하는 것이 바람직하다. 이러한 요구사항은 서비스를 제공하는 분산된 서버들을 논리적인 하나의 그룹으로 동작하도록 하는 프로세스 그룹을 이용함으로써 해결될 수 있다. 본 연구에서는 네트워크의 분할로 인해 프로세스 그룹이 두 개이상의 구성요소(component)로 나뉘어 지더라도 지속적인 고가용성(high availability)의 지원을 위해 응용서비스의 특성에 따라 기존의 처리결과에 영향을 받아 다음 서비스를 처리해야 하는 *history-sensitive* 그룹과 기존의 처리결과에 영향을 받지 않는 *history-free* 그룹의 개념을 도입하여 기존의 프로세스 그룹을 확장한 신뢰성 있는 프로세스 그룹 서비스에 대하여 살펴본다.

1. 서 론

현대 산업과 사회는 정보통신망의 발달과 함께 이용 가능한 모든 정보를 신속 정확하고 편리하게 이용할 수 있는 정보서비스가 증가하면서 정보중심사회로 빠르게 변화하고 있으며 이와 함께 순수자바기술을 이용한 응용서비스도 빠른 속도로 증대되고 있다. 이와 같은 환경에서 다양한 형태의 자바용서비스는 때와 장소를 구분하지 않고, 안정적이고 지속적인 서비스의 제공이 필수적으로 요구된다. 이를 위해서는 네트워크 분할(partition) 및 서버의 실패(crash)와 같은 전산망의 결함에도 불구하고 안정적이고 지속적인 서비스를 제공할 수 있는 결함포용(fault-tolerant) 시스템을 고려하는 것이 바람직하다. 이와 같은 필요성에 의해, 분산된 여러 서버 프로세스들이 하나의 논리적인 그룹을 이루고 동작하는 프로세스 그룹의 개념이 나타났다.

이와 같은 프로세스 그룹 형태의 자바용서비스가 전산망의 결함을 포용하는 시스템으로서 각 서버 프로세스가 일관성 있는 서비스를 제공하기 위해서는 그룹내의 서버 프로세스들이 가지는 뷰(view)의 정보가 항상 일치해야 한다. 뷰는 프로세스 그룹을 이루는 서버 프로세스들에 대한 멤버십(membership)과 그 그룹의 상태라고 할 수 있다. 직관적으로 생각하면 프로세스 그룹의 멤버십은 하나의 프로세스가 그 그룹에 참여(join)하거나 빠져나갈 때(leave) 멤버십을 변화시키지만 하면 된다. 하지만 프로세스 그룹의 형태를 이루고 동작하는 시스템은 분산환경의 비동기적(asynchronous)인 시스템이기 때문에 단순하고 직관적인 방법으로 문제를 해결한다는 것은 불가능하다. 프로세스 그룹의 모든 구성원들이 항상 같은 뷰를 가지기 위해서는 전산망의 오류에도 불구하고 프로세스들이 주고받는 메시지에 대한 안전성이 보장되어야 한다[1]. 이러한 메시지의 안전성 문제를 지원하기 위한 노력은 Cornell 대학에서 개발된 ISIS를 시작으로 그룹통신서

스템(Group Communication System)이라는 형태로 Transis, Totem, HORUS 등의 시스템이 개발되었다. 특히, Transis, Totem, HORUS 등의 경우 프로세스 그룹이 네트워크의 분할로 인해 두 개 이상의 구성요소(component)로 나뉘어 지더라도 네트워크의 복구(merge) 이후 다시 하나의 그룹으로서 동작할 수 있도록 하면서 그룹의 구성원들이 가지는 뷰의 정보가 일관성을 유지할 수 있도록 하는 EVS(Extended Virtual Synchrony)를 지원한다[2,3]. 하지만 주 구성요소(primary component)에 속한 프로세스들은 서비스를 계속할 수 있으나 부가 구성요소(non-primary component)의 프로세스들은 네트워크가 다시 복구될 때까지 서비스를 중단할 수밖에 없으므로 그만큼 시스템의 가용성(availability)은 저하된다고 할 수 있다.

본 연구에서는 이러한 메시지의 안전성과 관련된 요구사항을 잘 지원해 줄 수 있는 EVS 지원시스템을 기반으로 한 프로세스 그룹 서비스를 개발한다. 또한, 네트워크의 분할로 인해 프로세스 그룹이 두 개이상의 구성요소로 나뉘어 졌을 때, 보다 체계적이면서 효율적인 서비스를 제공하기 위하여 기존의 프로세스 그룹의 개념을 다음과 같이 확장한다. 응용서비스의 성격상 기존의 처리결과에 영향을 받아 다음 서비스를 처리해야 하는 경우, 그룹이 생성될 때 그 속성을 *history-sensitive* 그룹으로 지정하여 주 구성요소만이 서비스를 진행하고 부가 구성요소는 시스템이 복구될 때까지 서비스를 중단하는 방법과, 기존의 처리결과에 영향을 받지 않고 다음 서비스를 처리할 수 있는 경우, *history-free* 그룹으로 지정하여 주 구성요소가 아니라도 개별적으로 서비스를 진행할 수 있도록 하는 방법이다.

본 논문의 구성은 다음과 같다. 2장과 3장에서는 EVS와 EVS를 지원하는 그룹통신시스템에 대해 각각 간략하게 살펴보고, 4장과 5장에서는 신뢰성 있는 프로세스 그룹서비스를 위한 구조와 각 구성요소에 대해 자세히 설명한다. 마지막 6장에서는 앞에서 언급한 바와 같이 확장된 개념의 프로세스 그룹에 대해 살펴본다.

† 본 연구는 '98년도 정보통신부 대학기초연구지원 사업과제의 지원으로 수행되었음.

2. Extended Virtual Synchrony(EVS)

프로세스 그룹을 이루는 구성원들은 일관된 서비스를 제공하기 위해 항상 동일한 뷰를 유지해야 한다. 이를 위해서는 그룹내의 구성원들이 상호간에 메시지를 주고받을 때 전달되는 메시지를 관련 프로세스들에게 안전하게 전달되어야 하며 이를 위하여 메시지 전달의 신뢰성 면에서 atomic delivery와 safe delivery, 메시지 전달의 순서보장 면에서 casual delivery와 agreed delivery등이 지원된다. 특히, 이러한 개념으로부터 확장된 EVS 모델은 네트워크의 분할로 인해 프로세스 그룹이 여러 개의 구성요소로 나뉘어지더라도 분할 이전의 모든 그룹 구성원들에게 모두 전달되어야 한다는 것이다[2,3].

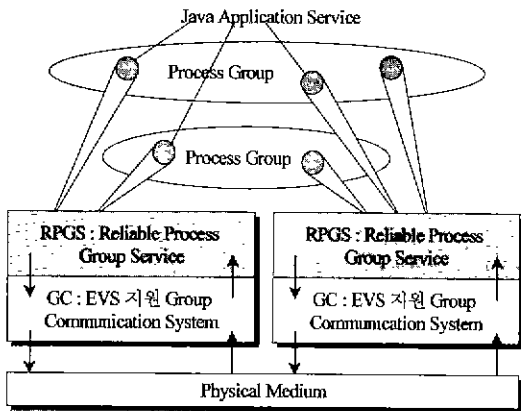
3. EVS 지원 그룹통신시스템

프로세스 그룹형태의 서비스가 네트워크의 분할이나 호스트의 실패와 같은 결함을 포용하는 시스템이 되기 위해서는 전산망의 결함에도 불구하고 그룹내의 모든 구성원들이 일관된 서비스를 제공하기 위해 항상 동일한 뷰를 가져야 하지만 분산환경의 비동기적인 특성으로 인해 이것은 그리 간단한 문제가 아니다. 이를 위한 요구사항은 다음과 같다. 첫째, 네트워크의 분할로 인해 프로세스 그룹이 두 개 이상의 구성요소로 분리된다 할지라도 네트워크의 분할 이전에 발생한 메시지는 논리적으로 그룹을 이루고 있던 모든 프로세스들에게 전달되어야 한다는 것이다. 둘째, 구성원들이 주고받는 메시지에 대한 순서가 보장되어야 한다[1]. 이러한 메시지의 신뢰성 및 안전성을 지원하기 위한 노력은 서론에서 언급한바와 같이 EVS 모델을 채용한 여러 그룹통신시스템으로 나타났다.

신뢰성 있는 프로세스 그룹 서비스의 개발을 위해서는, 호스트의 멤버십 관리와 신뢰성 있는 메시지의 멀티캐스팅, 그리고 EVS를 효과적으로 지원해 주는 그룹통신시스템을 기반으로 하며 이러한 그룹통신시스템의 개발을 위한 연구가 함께 진행되고 있다.

4. EVS를 이용한 자바 프로세스 그룹 서비스

분할가능 한 분산환경에서 견고하고 신뢰성 있는 자바 프로세스 그룹 서비스를 지원하기 위한 전체 시스템 구조는 [그림 1]과 같다.



[그림 1] 응용서비스의 견고성 지원을 위한 시스템 구조

전체 시스템은 프로세스 그룹형태의 자바응용서비스가 보다 견고하고 안정적인 신뢰성을 가질 수 있도록 지원하는 프로세스 그룹 서비스(Reliable Process Group Service, RPGS)와 EVS를 지원하기 위한 그룹통신시스템인 GC로 구성된다.

RPGS는 GC와의 통신과 GC로부터 전달되는 메시지를 포함하는 이벤트를 관리하기 위한 EVM(Event Manager)과 프로세스 그룹을 관리하는 PGM(Process Group Manager)으로 구성된다. 4장에서 이들에 관해 자세히 살펴본다.

4.1 이벤트 관리자

이벤트 관리자는 GC와의 통신과 GC로부터의 EVS를 만족하여 전달되는 모든 메시지를 포함한 이벤트를 관리한다. GC로부터 전달되는 이벤트에 포함된 메시지는 크게 다음과 같이 분류된다.

- Regular : 다른 프로세스로부터 전달되는 메시지, GC를 통해서 전달되며 이벤트 관리자는 해당 프로세스 그룹 관리자로 메시지를 전달한다.
- Membership_Change : 프로세스의 join 및 leave에 의한 프로세스 그룹의 멤버십 변화. 이벤트 관리자는 해당 프로세스 그룹 관리자로 메시지를 전달한다
- Config_Change : 네트워크의 분할이나 호스트의 결함으로 인한 호스트의 멤버십 변화. 이벤트 관리자는 현재 호스트내의 모든 프로세스 그룹 관리자로 메시지를 전달한다.

4.2 프로세스 그룹 관리자

프로세스 그룹 관리자는 하나의 논리적인 프로세스 그룹을 관리하며 하나의 RPGS내에 여러 개의 관리자가 존재할 수 있다. 각 프로세스 그룹 관리자는 다음과 같은 역할을 한다. 우선, Config_Change 메시지가 발생하여 현재 프로세스 그룹이 여러 개의 구성요소로 분리되었다는 것을 알게되었을 때 새롭게 형성된 프로세스 그룹이 주 구성요소가 될 수 있는지 여부를 결정해야 한다. 또한, 그룹이 분리되었다가 다시 복구되었을 때 각각의 분리된 그룹에서 처리되었던 메시지를 순서화해야 할 뿐 아니라 각 그룹이 다시 하나의 그룹으로 일관되게 동작하도록 상태전이(State Transfer)를 함으로써 상호간의 상태를 일치시키게 된다.

프로세스 그룹 관리자를 구성하는 이러한 요소들에 대해서는 5장에서 다시 자세하게 설명한다.

5. 프로세스 그룹 관리자의 구성

프로세스 그룹 관리자는 이미 앞에서 살펴본 바와 같이 크게 세 가지로 구성되며 자세한 내용은 다음과 같다

5.1 주 구성요소의 선택

프로세스 그룹은 주 구성요소에 속해 있을 때만 서비스를 진행할 수 있다. 네트워크의 분할 및 복구로 인해 그룹의 멤버십에 변화가 있을 때 프로세스 그룹 관리자는 현재 호스트에서 관리되고 있는 프로세스들이 주 구성요소가 될 수 있는지를 확인해야한다. 이를 위한 알려진 방법은 다음과 같다[4,5].

- Monarchy : 이미 지정된 특정 서버가 속한 구성요소를 주 구성요소로 하는 방법
- Majority : 전체 프로세스 그룹의 과반수 이상이 속해 있는 구

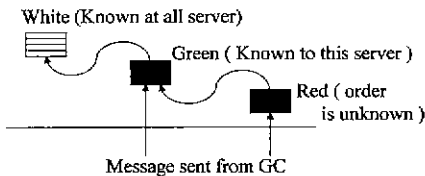
성요소를 주 구성요소로 하는 방법

- Dynamic Linear Voting : 이전의 주 구성요소에 속한 서버 프로세스들의 과반수가 속해있는 구성요소를 새로운 주 구성요소로 하는 방법

본 시스템에서는 마지막 방법을 이용하여 주 구성요소를 선택하는 알고리즘을 작성한다.

5.2 메시지의 순서화(Global Total Ordering)

GC를 통해서 각 프로세스 그룹에 전달되는 메시지는 물리적으로 연결된 하나의 구성요소 내에서 지역적인 순서(local order)만을 보장한다. 즉, 이것은 현재 구성요소 내에서만 의미가 있는 것으로 분할됐던 그룹이 다시 복구되었을 때, 새로운 구성요소 내에서의 전체적인 순서는 보장할 수 없다. 하지만 하나의 프로세스가 전달받은 메시지를 다른 프로세스가 전달받았는지 여부는 그룹내의 다른 구성원들과의 추가적인 통신이 없이는 알기 힘들게 된다. 이와 같이 전달된 메시지에 대한 불완전한 정보를 표현하기 위해 주 구성요소에서만 메시지의 순서를 결정할 수 있도록 하며, 다음과 같은 메시지의 처리 모델을 적용한다.



[그림 2] 메시지 순서화를 위한 Color 모델

프로세스 그룹 매니저가 GC로부터 전달된 메시지의 순서를 결정할 수 없을 때는 붉은색으로 표시하고 버퍼에 저장한다. GC로부터 전달된 메시지나 붉은색으로 표시된 메시지에 대해 현재 그룹 내에서만 그 순서를 결정할 수 있을 때 초록색으로 표시한다. 그리고 마지막으로, 새로 구성된 그룹내의 모든 구성원들이 존재하는 호스트의 프로세스 그룹 관리자에서 모두 그 순서를 결정할 수 있을 때는 흰색으로 표시하고 그룹 관리자가 관리하는 로컬 프로세스들에게 이 메시지를 전달할 수 있게 된다[2]. 이처럼 GC로부터 전달되는 메시지의 정보는 EVS에 의해 보장될 수 있다[3].

5.3 상태전이(state transfer)

두 개 이상의 분리되었던 그룹이 하나의 새로운 그룹으로 합쳐진 후 다시 일관성 있는 서비스를 제공하기 위해서는 두 구성요소가 다시 합쳐졌을 때 우선 새로운 구성원들간에 자신들의 상태 메시지를 교환 후 각 구성요소에서 발생했던 메시지의 교환하도록 한다.

6. 프로세스 그룹의 확장

EVS를 적용한 기존의 그룹통신시스템에서는 네트워크의 분할로 인한 구성요소의 분리가 발생할 때 주 구성요소에 속한 프로세스들은 지속적으로 서비스를 제공하는 반면 부가 구성요소에서는 서비스를 중단할 수밖에 없으므로 그만큼 시스템의 서비스에 대한 신뢰성 및 가용성이 저하될 수 있다. RPGS에서는 네트워크의 분할에 의해 그룹의 분리되어도 보다 체계적이면서 효율적인 서비스를 제공하기 위하여 서비스를 제공하는 응용서비스의 특성에 따라 기존 프로세스 그

룹의 개념을 history-sensitive 그룹과 history-free 그룹으로 확장한다.

6.1 history-sensitive 그룹

응용서비스의 성격상 기존의 처리결과에 영향을 받아 다음 서비스를 처리해야 하는 경우, 주 구성요소만이 서비스를 진행할 수 있도록 하고, 부가 구성요소에서는 네트워크가 복구되어 주 구성요소가 될 때까지 서비스를 중단한다.

6.2 history-free 그룹

주 구성요소뿐 아니라 부가 구성요소에서도 분리된 상태로 계속 서비스를 제공할 수 있도록 한다. 이 경우 새로 구성된 프로세스 그룹의 구성원에 대한 일관성이 문제가 될 수 있으나 응용서비스의 성격상 기존의 처리결과에 영향을 받지 않고 다음 서비스를 처리할 수 있는 성격의 서비스도 있을 수 있다는 점을 고려한 것이다. 즉, 기상정보서비스를 예를 들 수 있다. 전산망의 장애로 인해 서비스가 최신의 기상 정보를 알 수 없다 하더라도 알려진 가장 최근의 정보만을 가지고 서비스는 가능하기 때문이다.

7. 결 론

본 논문에서는 메시지의 안전성과 신뢰성을 보장하기 위한 EVS를 지원하는 그룹통신시스템을 기반으로 한 자바응용서비스의 프로세스 그룹서비스를 구성하기 위한 요소들을 살펴보고 기존의 프로세스 그룹에 관한 개념을 응용서비스의 특성에 따라 history-sensitive 그룹과 history-free 그룹으로 확장하였다. 이로 인하여 부가 구성요소에 속한 프로세스라도 응용서비스의 성격에 따라 보다 체계적이고 지속적인 서비스를 제공할 수 있도록 하였다. 그룹통신시스템을 기반으로 한 자바응용서비스의 신뢰성 있는 프로세스 그룹서비스는 금융, 국방, 의료 등과 같이 신뢰성이 절대적으로 요구되는 정보서비스를 위한 기반시스템으로서 사용될 수 있을 것으로 기대된다.

앞으로 계속 발전시켜 나가야 할 연구 내용으로는 그룹의 특성에 따른 확장에서 history-sensitive/free한 특성을 메시지 단위까지 확장함으로써 응용서비스의 신뢰도를 한층 더 높일 계획이다.

참고문헌

- [1] O.Babaoglu, R.Davoli, A.Montresor. Partitionable Group Membership : Specification and Algorithms, Technical Report UBLCS-97-1, Univ. of Bologna, Italy, 1997.
- [2] Y.Amir, D.Dolve, P.M.Melliar, L.E.Moser. Robust and Efficient Replication Using Group Communication. Technical Report CS94-20, Hebrew University of Jerusalem, November, 1994
- [3] L.E.Moser, Y.Amir et al. Extended Virtual Synchrony. In Proceedings of the 14th International Conference on Distributed Computing Systems, pp56-65, June 1994.
- [4] D.Gifford. Weighted Voting for Replicated Data. In Proc. of the ACM Symposium on Operating Systems Principles, Dec. 1979.
- [5] S.Jajodia, D.Mutchler. Dynamic Voting Algorithm for Maintaining the Consistency of Replicated Database. ACM Transaction on Database Systems, 15(2), June 1990.