

자바를 이용한 Extended Virtual Synchrony의 지원*

문 남 두* · 최 혁 재 · 유 양 우 · 박 양 수 · 이 명 준
울산대학교 전자계산학과

Supporting Extended Virtual Synchrony using Java[†]

Nam-Doo Moon* · Hyukjae Choi · Yang-Woo Eu · Yang-Su Park · Myung-Joon Lee
Department of Computer Science, University of Ulsan

요 약

자바는 분산환경에 적합하고, 이식성이 높으며, 코드 재사용이 용이하다는 장점을 가지고 있다. 이러한 이유로 다양한 분야에서 자바로 구현된 응용서비스를 제공하고 있다. 현재와 같은 정보중심사회에서 이러한 자바응용서비스는 서버의 실패(crash)나 네트워크 분할(partition)과 같은 결함이 발생되더라도 투명하고, 안정적이며, 지속적인 서비스를 제공하는 것이 바람직하다. 이러한 요구사항은 이기종 분산환경에서 동일한 서비스를 제공하는 자바응용서비스를 그룹으로 동작하게 함으로써 해결될 수 있다. 본 논문에서는 자바응용서비스 그룹이 네트워크의 분할로 상호 통신할 수 없는 두 개 이상의 구성요소로 분리되더라도 지속적인 서비스를 제공하고, 네트워크가 복구(merge)되면 이전과 같이 하나의 그룹으로서 동작할 수 있도록 그룹구성원의 일관성을 유지하는 EVS(Extended Virtual Synchrony) 모델을 구현함으로써 자바응용서비스의 신뢰성과 고가용성(high availability)을 지원하고자 한다.

1. 서 론

자바는 객체 지향, 분산환경 지원, 높은 이식성 등의 장점을 가지고 있다. 이러한 이유로 자바로 구현된 응용서비스는 인터넷 브라우저 환경뿐 아니라 데이터베이스 연동, 전자 화폐에 이르기까지 다양한 분야에서 제공되고 있다. 오늘날 정보통신망의 발달과 이용 가능한 모든 정보를 신속, 정확, 편리하게 제공하는 자바응용서비스의 증가로 인하여 고도화정보사회로 빠르게 변화하고 있다. 정보중심사회에서 이러한 자바응용서비스는 서비스 측면에서 서버의 실패나 네트워크의 분할과 같은 결함에도 불구하고 신뢰성과 고가용성을 제공하는 것이 바람직하다. 이러한 요구사항은 이기종 분산환경에서 동일한 서비스를 제공하는 자바응용서비스를 그룹으로 동작하게 함으로써 해결된다.

응용프로세스를 그룹으로 지원하기 위한 그룹통신시스템(group communication system)으로, VS 모델을 지원하는 Cornell 대학의 ISIS가 그 시작이 되었다. ISIS와 같이 VS를 지원하는 시스템은 네트워크의 비동기적 특성으로 인하여 발생하는 메시지의 분실과 응용프로세스의 실패와 같은 결함을 처리하여 서비스의 가용성을 높인다. 하지만 응용프로세스가 실패로부터 복구될 때 이전의 구성원으로 참여하는 것이 아니라 새로이 그룹에 참여하는 구성원으로 간주되고, 네트워크의 분할로 인하여 그룹이 상호 통신할 수 없는 여러 구성요소로 분리될 때 지속적인 서비스를 제공하는 하나의 주 구성요소(primary component)와 서비스를 제공하지 못하고 정지되는 나머지 구성요소로 구분됨으로써 이러한 결함들에 대하여 견고성과 고가용성을 제공하지 못한다.

최근에는 VS의 개념을 확장하여 그룹의 구성원이 실패로부터 복구되거나 분할된 네트워크가 재결합되었을 때, 투명하게 이전과 같이 하나의 그룹으로 동작할 수 있도록 그룹구성원들간의 일관성을 유지하기 위하여 EVS[1]를 지원하는 Transis[2], Totem[3,4], 그리고 Horus[5]와 같은 시스템들이 소개되고 있다. 이와 같이 네트워크의 분할과 같은 동적인 네트워크 환경에서 전체 서비스가 중단되지 않도록 결함포용(fault-tolerant) 시스템을 고려함으로써 자바응용서비스에 대한 견고성을 지원하는 것이 바람직하다.

본 논문의 구성은 다음과 같다. 2장에서는 EVS 모델에 대한 기본 개념을 살펴보고, 3장에서는 자바를 이용한 EVS 지원 시스템을 설계하고, 4장에서는 3장의 설계 모델을 바탕으로 구현에 대해 소개한다. 5장에서는 결론과 향후 연구 방향에 대하여 언급한다.

2. EVS 모델

EVS는 VS의 확장된 개념이다. EVS 모델은 네트워크의 분할과 재결합, 호스트의 실패와 복구와 같이 동적인 분산환경에서도 [표 1]과 같은 메시지의 요구사항을 만족하기 위해 정규 구성요소와 멤버쉽의 변화로 새로운 정규 구성요소로 이행하기 전에 이전의 정규 구성요소의 구성원이면서 새로운 정규 구성요소의 구성원들로 멤버쉽이 형성된 과도기적 구성요소를 소개함으로써 구성원들간에 일관성을 유지하게 되어 응용서비스의 가용성을 높인다. [표 1]과 같이 메시지의 전달은 신뢰성과 순서화의 요구사항에 따라 엄격한 조건을 요구하기도 하고, atomic delivery나 causal delivery와 같이 낮은 등급의 조건을 요구하는 메시지의 전달이 있을 수 있다.

*본 연구는 '98년도 정보통신부 대학기초연구지원 사업과제의 지원으로 수행되었음

특성	요구사항
Agreed delivery	그룹에 참여하는 모든 응용프로세스에게 같은 순서로 메시지가 전달되기를 요구한다.
Safe delivery	그룹에 참여하는 응용프로세스가 실패하지 않는다면 메시지가 모두에게 전달되기를 요구한다.

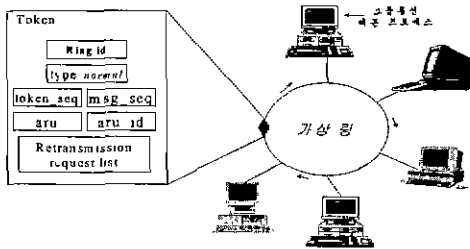
[표 1] 신뢰성 있는 그룹통신을 위한 메시지의 요구사항

3. 자바를 이용한 EVS 지원 시스템의 설계

EVS 지원 시스템은 그룹통신에 참여하는 각 호스트의 그룹통신 데몬(daemon) 프로세스들로 구성된다. 데몬 프로세스는 시스템내의 모든 물리적 통신을 처리하며, 자바용 응용 프로세스가 그룹내의 다른 응용프로세스들과 통신할 수 있도록 중개자 역할을 한다. 동일한 메시지를 그룹에 참여하는 모든 구성원들에게 신뢰성 있는 메시지 멀티캐스트를 위한 방법으로 가상 링 구조가 사용된다. 가상 링의 형성은 물리적으로 상호 통신할 수 있는 그룹통신 데몬들로 구성된다. 논리적인 토큰이 가상 링을 순환하면서 토큰을 받은 데몬만이 상위 응용프로세서로부터 전달받은 메시지를 순서화된 메시지로 멀티캐스트할 수 있고 다음 구성원에게 토큰을 전달한다.

3.1 메시지의 순서화

그룹의 멤버들간에 일관성 있는 작업을 위해서는 메시지의 순서화된 전달이 필수적이다. agreed나 safe 메시지의 순서화에 대한 기본원리는 그룹통신 데몬들로 구성된 가상 링 위에 단일의 논리적인 정규토큰을 순환시킴으로써 전체순서(total order)를 만들어내는데 있다.



[그림 1] 논리적 토큰링에 기반한 메시지 순서화

3.2 동적인 멤버십 관리

메시지의 요구사항을 만족시키기 위해서는 구성원들간의 일관된 멤버십 유지를 보장하는 멤버십 관리 기능이 필수적이다. 따라서 호스트의 실패나 네트워크의 분할, 그리고 토큰의 분실과 같은 결함이 발견되면 새 링을 형성하여 중단되었던 메시지 순서화 프로토콜의 수행을 정상화시킨다. 멤버십 관리의 목적은 상호통신 가능한 호스트들로 구성된 구성요소의 모든 구성원들이 해당 멤버십에 대해 합의를 얻어 내는데 있다. 또한 EVS를 만족시키기 위해 새로운 토큰을 생성시켜 순환시킴으로써 결함으로 인하여 누락된 메시지들에 대한 복구작업을 수행한다. 멤버십 관리를 위해 다음과 같은 유형의 특별한 메시지를 사용한다.

[표 2]의 Reform 메시지는 일반 메시지와는 다르게 토큰

을 갖고 있지 않더라도 각 데몬들이 멀티캐스트할 수 있고, 메시지의 분실이 발생하더라도 재전송되지 않으며 응용 프로세스에게 전달되지도 않는다. Reform 메시지의 사용은 송신자가 생각하는 새 링의 멤버십 정보를 멀티캐스트함으로써 새 링에 대한 멤버십에 합의를 얻기 위한 것이다.

type	Reform
sender_id	송신자(host) ID
host_set	송신자가 새 링의 멤버십으로 고려하는 호스트집합
fail_set	송신자가 실패했다고 간주하는 호스트집합
ring_seq	송신자에게 알려진 링의 최대 시퀀스 번호

[표 2] Reform 메시지의 구조

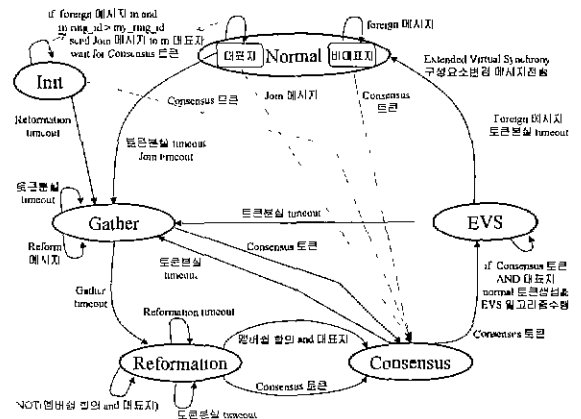
[표 3]의 구성요소 변경 메시지는 이전의 정규 구성요소로부터 과도기적 구성요소로의 변경이나 과도기적 구성요소로부터 새 정규 구성요소로의 변경을 나타낸다. 구성요소 변경 메시지는 일반 메시지와는 달리 각 호스트에서 지역적으로 생성되어 멀티캐스트되지 않고 직접적으로 응용프로세스에 전달된다.

regular_id	이전의 정규 구성요소나 메시지가 개시하는 정규 구성요소의 ID
seq_number	메시지가 정규 구성요소를 개시한다면 0, 과도기적 구성요소를 개시한다면 바로 이전 정규 구성요소에서 전달된 메시지의 최대 시퀀스 번호
transition_id	이전의 과도기적 구성요소나 메시지가 개시하는 과도기적 구성요소의 ID
membership	메시지가 개시하는 구성요소의 멤버십

[표 3] 구성요소 변경 메시지의 구조

그룹통신 데몬 프로세스의 상태는 다음과 같이 6개의 상태로 정의되며, 상태 전이도는 [그림 2]에서 설명하고 있다.

- Init state: 데몬 프로세스가 처음 구동되면 Init 상태가 된다.
- Normal state: 멤버십의 변화가 없으며 메시지 순서화 알고리즘을 수행한다
- Gather state: 새 링을 구성할 대표자들을 모은다.
- Reformation state: 새 링의 멤버십에 대한 합의를 시도한다.
- Consensus state: 새 링의 토큰 순환경로가 결정되며 새 링의 멤버십에 관한 정보가 교환된다.
- EVS state: extended virtual synchrony를 보장하기 위한 알고리즘을 수행한다



[그림 2] 멤버십 관리 프로토콜에 대한 상태 전이도

4. 자바를 이용한 EVS 지원 시스템의 구현

새로 참여하는 데몬 프로세스는 Init 상태에서 시작되며 다음의 알고리즘을 수행한다.

```

/* 새로 참여하는 그룹통신 데몬 프로세스는 외부의 멀티캐스트
되는 메시지를 받음으로써 기존 링의 존재를 알게된다 */
Foreign message m received:
  if m.ring_id ≠ my_ring_id and
    m.ring_id.seq > my_ring_id.seq then
    send Join request message to m.representative
    set Reformation timeout //Consensus 토큰 기다림
    else discard
Consensus token received:
  switch into Consensus state
/* Reformation timeout 시간이 경과될 때까지 Consensus 토큰
을 받지 못하면 자신을 대표자로 지정하고 Reform 메시지를 멀티
캐스트한다. */
Reformation timeout expired
  set self as the representative
  multicast Reform message
  switch into Gather state
    
```

[Init 상태의 그룹통신 데몬 프로세스의 수행 알고리즘]
 Normal 상태에서 토큰분실과 같은 결함에 발견되면 멤버쉽 알고리즘이 활성화된다.

```

Normal token received:
  if token.rind_id = my_rung_id and
    token.token_seq > my_token_seq then
    check how many messages in new_message_queue
    multicast requested retransmission messages
    .....
    else discard token
Join message from processor q received:
  if I am representative then
    switch into Consensus state
    // 새로이 참여하는 데몬 프로세스를 포함하기 위함
    Create a New logical token Ring R
    and set self as representative for a new ring
    else ignore
Consensus token received:
  if I am not representative then
    switch into Consensus state
    else discard
Foreign message m received:
  if m.ring_id ≠ my_ring_id and I am representative
    m.ring_id.seq > my_ring_id.seq then
    set I am not representative
    send Join request message to m.representative
    set Join retransmission timeout
    set Join timeout //Consensus 토큰 기다림
    else discard
Regular message received:
  cancel Token Retransmission timeout if set
  add message to receive_message_queue
  update retransmission request list
  update my_aru
    
```

```

// 메시지의 요구사항에 따라 agreed나 safe 순서로
// 응용 어플리케이션에게 전달한다.
deliver messages that satisfy their delivery criteria
Token Loss timeout expired:
  switch into Gather state
Token Retransmission timeout expired:
  retransmit token
  reset Token Retransmission timeout
Reformation timeout expired:
  switch into Gather
    
```

[Normal 상태의 그룹통신 데몬 프로세스의 수행 알고리즘]

5. 결론 및 향후 연구

본 논문에서는 호스트의 실패나 네트워크의 분할과 같은 동적인 네트워크 환경에서 그룹통신에 참여하는 모든 구성원들에게 일관성 있는 메시지의 전달을 보장하기 위해 EVS 지원 시스템을 설계하고 구현하였다. 이를 위하여 가상 링과 논리적인 토큰을 사용함으로써 동일한 메시지를 그룹의 모든 구성원들에게 신뢰성 있게 전달하는 메시지 멀티캐스트와 메시지의 순서화, 그리고 물리적 호스트 그룹을 관리하기 위한 멤버십 관리 서비스를 제공하였다. 본 논문에서 구현한 EVS 지원 시스템을 기반으로 공동작업지원 환경을 개발한다면 전산망의 결함에도 불구하고 지속적이고 효율적인 공동작업이 예상된다.

추후 연구과제로서 응용 서비스들이 LAN상에 지역적으로 밀집되어 그룹형성이 이루어질 때와 WAN상에 산재되어 형성되는 각 상황에 따라 효율적인 메시지의 전송과 멤버십을 관리하는 기법에 관한 연구가 이루어져야 할 것이다.

참고문헌

- [1] L. E. Moser, Y. Amir, P.M. Melliar-Smith and D. A. Agarwal. "Extended Virtual Synchrony". In *Proceeding of the 14th International Conference on Distributed Computing Systems*, pages 58-65, June 1994. IEEE. A detailed version appears as ECE Technical Report #93-22, University of California, Santa Barbara, December 1993.
- [2] D. Dolev and D. Malki, "The Transis Approach to High Availability Cluster Communication", *Communications of the ACM*, Vol. 39, No. 4, p.64-70, 1996.
- [3] Agarwal, D.A. "Tolem: A reliable ordered delivery protocol for interconnected local-area networks" Ph.D. dissertation. Department of Electrical and Computer Engineering University of California, Santa Barbara (Aug. 1994)
- [4] Y. Amir, L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, and P. Ciarella. "Fast Message Ordering and Membership Using a Logical Token-Passing Ring". In *Intl. Conference on Distributed Computing Systems*, pages 551-560, May 1993
- [5] Van Renesse R., Birman K. P., and Maffels S, "Horus:A Flexible Group Communication System", *Communications of the ACM*, Vol 39, No 4, p75-83, 1996
- [6] WEIJIA JIA, "Implementation of a Reliable Multicast Protocol", *SOFTWARE-PRACTICE AND EXPERIENCE*, VOL. 27(7), 813-850 (JULY 1997).