

실시간 성질을 가지는 자료의 인과순서화 전달

장익현*, 도재수**

*동국대학교 정보통신공학과

**동국대학교 컴퓨터학과

E-mail:{ihjang,dojesu}@dongguk.ac.kr

Causal Order of Message Delivery for Real-Time Data

Ik-Hyeon Jang*, Jae-Su Do**

*Dept. of Info. & Comm. Eng., Dongguk Univ.

**Dept. of Computer Science, Dongguk Univ.

요약

실시간 데이터를 처리하는 응용에서는 유효시간이 지난 데이터를 사용할 수 없다. Δ -인과순서화는 데이터의 유효시간 내에 메시지를 인과순서로 전달하는 추상 통신 개념이다. 인과순서를 유지하기 위해서는 순서를 유지하기 위한 제어정보를 교환하여야 하며, 전송부하를 줄이기 위하여 제어정보의 양을 최소화하기 위한 많은 노력이 계속되어 왔다. 본 논문에서는 전송부하를 줄이기 위하여 유효한 통신패턴을 분석하여 중복으로 교환되는 제어정보가 최소화되는 Δ -인과순서화 알고리즘을 제안한다.

1. 서론

분산시스템은 여러 프로세스들이 서로 메시지를 교환하면서 공조하여 하나의 작업을 수행한다. 그러나 메시지 전송시간은 유한하지만 전달시간의 차이 때문에 송신하는 메시지와 전달되는 메시지들의 순서가 일치하지 않을 수 있다. 이러한 메시지 순서에 관련된 불일치성을 해결하기 위하여 인과순서(causal order)가 제안되었다[4]. 분산시스템에서의 인과순서화는 Lamport가 제안한 *happened before* 관계에 기초를 두고 있으며, 두 개의 메시지가 서로 관련되어 있고 같은 목적지를 가진다면 송신한 순서와 같은 순서로 목적지에 전달할 것을 요구한다[6].

멀티미디어 통신시스템에서 응용프로그램이 전달한 메시지는 전송된 후 일정 시간이 경과하면 데이터로서의 의미를 잃게 되는 특성을 갖는다. 이때 전송된 메시지의 전송제한시간을 유효시간(lifetime)이라고 하며 이 시간 안에 목적지에 전달되는 메시지는 의미있는 데이터로 응용프로그램에서 이용될 수 있다. 멀티미디어 응용프로그램을 지원하기 위하여 데이터의 유효시간을 고려한 Δ -인과순서(Δ -causal order)가 제안되었으며[10] 이는 인과순서의 확장으

로 볼 수 있다.

인과순서화와 관련된 주요 이슈로는 메시지 간의 인과관계를 파악하는 방법과 인과관계를 유지하기 위해 교환되는 제어정보의 양을 최소화하는 방법이 있다. 지금까지는 주로 논리적 시간을 사용하여 인과관계를 파악하고 있으며 제어정보의 양을 줄이기 위한 다양한 방법이 연구되고 있다[2,4,5,8]. Δ -인과순서화는 실시간적 특성을 가지는 자료를 대상으로 하기 때문에 논리적 시간이 아니라 물리적 시간을 사용하여야 한다. 실시간 특성을 가지는 멀티미디어 자료는 점점 그 응용분야가 넓어지고 있기 때문에 이와 관련된 연구도 계속되고 있다[1,3,9,10].

본 논문에서는 통신패턴을 분석하여 중복되는 제어정보의 전송을 억제하여 Δ -인과순서를 유지하기 위하여 필요한 제어정보의 양을 최소화하는 알고리즘을 제안한다. 제안하는 알고리즘은 기존에 제안하였던 인과순서화 알고리즘[4]을 실시간 데이터 처리를 위하여 논리적 시간을 물리적 시간을 처리하는 것으로 수정한 것이다.

2장에서는 Δ -인과순서화의 개념을 설명하고 3장에서 새로운 알고리즘을 제안한다. 4장에서 알고리

즘의 성능을 비교분석하며 5장에서 결론을 맺는다.

2. Δ -인과순서

2.1 메시지의 유효시간

메시지의 유효시간(Δ)은 메시지가 전송되고 나서 목적지에 도달하여 사용될 수 있는 물리적인 시간을 의미하는 것으로 유효시간이 지나서 전달된 메시지는 사용할 수 없다. 즉 모든 메시지의 유효시간은 메시지의 출발시간 + Δ 가 되며 유효시간을 넘긴 후에 도착한 메시지는 폐기된다. 따라서 유효시간 이전에 목적지에 도착된 메시지는 유효시간이 지나기 전에 목적프로세스에 전달되어 사용되어야 한다.

멀티미디어시스템에서 메시지의 유효시간은 서비스의 질을 저하시키지 않는 최대지연시간으로 정의할 수 있으며 멀티미디어응용에 따라 차이가 있다. 예를 들어 화상회의시스템의 음성과 화상데이터는 250ms, 화상전화의 경우는 350ms, 비대화형 비디오 시스템은 1초, 카탈로그 검색시스템의 최대 지연시간은 2초 정도이다[2]. 문제를 단순화하기 위하여 모든 메시지는 동일한 유효시간을 가지는 것으로 가정한다.

2.2 전역시간

실시간 전송 제한조건을 만족시키기 위해서는 모든 프로세스들이 인과관계가 있는 사건들을 모두 다른 물리적 시간에 발생시킬 수 있을 만큼의 정확도(granularity와 precision)를 가지는 전역시간(global clock)을 가져야 한다[9]. 그러나 분산시스템에서 완전한 전역시간은 얻을 수 없다. 시간동기화에 대한 연구 결과 시간편차(clock drift)는 5-10ms 이내까지 제공되고[7], 각 응용에서 요구하는 유효시간은 그보다 충분히 크므로, 시간동기화 알고리즘이 작동하는 것을 가정하여 시간편차를 유효시간에 포함시킨다.

2.3 Δ -인과순서

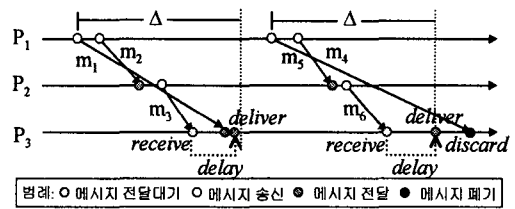
Δ -인과순서화는 Yavatkar에 의해 소개된 것으로 인과관계를 기반으로 하여 메시지가 실시간 데이터를 포함하고 있는 신뢰할 수 없는 통신시스템을 가정하며 다음과 같이 정의한다[10].

[정의 1] (인과순서화) : 메시지 M_1 과 M_2 가 같은 목적지를 가지고 있으며 $send(M_1) \rightarrow send(M_2)$ 이면 $deliver(M_1) \rightarrow deliver(M_2)$ 이다. (여기서 기호 \rightarrow 은 happened before 관계를 의미한다.)

[정의 2] (Δ -인과순서화) : 다음 두 가지 조건을 만족하면 Δ -인과순서가 유지된다고 한다.

- 1) 유효시간 내에 도착하는 모든 메시지는 유효시간 내에 전달되고 그 외의 모든 메시지는 폐기된다.
- 2) 모든 전달되는 메시지는 인과순서를 준수한다.

이는 다음 그림으로 설명된다. [그림1]에서 m_3 은 m_1 보다 먼저 P_3 에 도착하였으나 m_1 의 유효시간이 지나지 않았기 때문에 인과순서를 유지하기 위하여 m_1 이 도착할 때까지 기다렸다가 m_1 이 전달된 후에 전달된다. 그러나 m_6 의 경우 인과순서로 보면 m_1 이 전달된 후에 전달되어야 하나 m_1 이 유효시간 내에 도착하지 않았기 때문에 m_6 는 전달되나 그 후에 m_1 이 도착하면 유효시간을 넘긴 데이터이기 때문에 m_1 은 폐기된다.



[그림 1] Δ -인과순서화에 따른 메시지 전송

3. 알고리즘

3.1 인과관계의 정리

인과관계의 파악은 제어정보를 정리하는 방법에 따라 두 가지로 분류할 수 있다. 첫 번째 방법은 목적프로세스에 따라 인과관계를 정리하는 것으로 지금까지 대부분의 알고리즘은 이 부류에 속한다. 이 방법은 전달조건을 검사하기 쉽다는 장점을 가지고 있으나 인과관계를 간의 우선순위를 파악하기 위해서는 전체 제어정보를 확인해야 하는 어려움이 있다. 두 번째 방법은 송신프로세스를 기준으로 인과관계를 정리하는 것으로 전송되는 메시지와 그와 함께 전송되는 제어정보를 하나의 객체로 처리한다. 이 방법에서는 목적프로세스를 기준으로 인과관계를 검사해야 하는 전달조건을 검사하는 상대적으로 어렵다. 전달조건을 검사하기 위해서는 전체 인과관계정보를 확인해야 한다. 그러나 이 방법에서는 우선순위 관계를 파악하는 것이 쉽다.

본 논문에서는 제어정보들 간의 우선순위를 찾는 것이 중요하므로 두 번째 방법을 채택하며 전달조건을 확인을 빨리 하기 위하여 메시지 송신 시에 전달조건 확인을 위한 정보를 목적프로세스를 기준으로

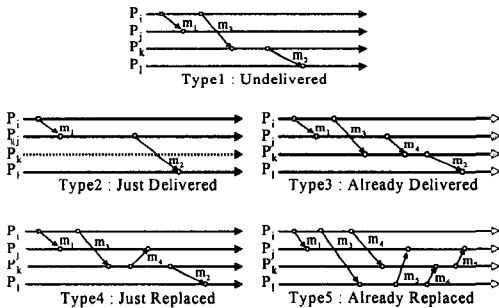
별도로 추출하도록 한다.

3.2 메시지 패턴의 분류

전송되는 메시지 m 에 대한 정보를 가지는 상태를 송신프로세스와 목적프로세스를 기준으로 분류하면 모든 프로세스는 다음 4가지 중 한 상태에 속한다.

- 1) 송신프로세스의 상태
- 2) 목적프로세스의 상태
- 3) 송신과 목적프로세스 사이의 프로세스의 상태
- 4) 목적프로세스 이후에 있는 프로세스의 상태

만약 두 개의 프로세스가 동일한 상태 정보를 가지고 있다면 그 두 프로세스는 메시지 m 에 관한 정보에 대해서는 하나의 프로세스로 볼 수 있다. 따라서 분산시스템은 4개의 프로세스로 이루어진 시스템으로 변환이 가능하며, 어떤 메시지에 관한 정보의 흐름 관점에서 보면 다중전송(multicast)도 단일전송(unicast)으로 변환되므로 인과순서화를 제공하는 유효한 모든 통신패턴은 [그림 2]에 있는 추상통신패턴 중의 하나로 변환이 가능하다.



[그림 2] 추상통신패턴 (P_k 의 m_2 전송 시점에서 보는 m_1 에 대한 정보의 흐름을 기준으로 분류)

송신프로세스와 송신과 목적프로세스 사이에 있는 프로세스들은 메시지가 목적프로세스에 전달되었는지를 알 수가 없다. 이런 프로세스들은 Type1로 취급된다. 목적프로세스는 해당 메시지가 전달되었다는 것을 바로 알 수 있으며 Type2가 된다. 목적프로세스 이후에 있는 프로세스들은 m_1 이 목적프로세스에 이미 전달되었다는 것을 알 수 있으므로 Type3가 된다.

한편 목적프로세스와 송신프로세스 사이에 있는 프로세스들은 m_1 이후에 인과순서상 뒤에 있는 다른 메시지가 동일한 목적프로세스로 전송될 경우에 그들 간의 순서를 반영하여 인과관계를 판단할 수

있다. 즉 인과순서 상 뒤에 있는 메시지가 전달되었다면 그들 간의 인과순서를 유지하기 위하여 앞의 메시지는 당연히 전달되었을 것이므로 인과순서상 뒤의 메시지에 대한 정보만 유지해도 된다. Type4에서 m_1 은 m_4 에 의해 대체되었다고 한다. 만약에 어떤 메시지 m 을 프로세스 P_j 에 m_4 와 인과관계를 유지하여 전달하였다고 하자. 이 경우 m_4 가 P_j 에 전달되기 위해서는 m_1 이 전달된 후에 전달되어야 하므로 m 은 메시지 m_1 과도 인과관계를 유지하게 된다. 이런 경우의 통신패턴이 Type4가 되며 이것을 일반화하면 Type5가 된다.

3.3 중복정보의 제거

만약 Type2 프로세스들이 어떤 이유로 메시지 m_1 에 대한 정보를 가지고 있어야 한다면 그 정보는 인과순서를 유지하는데 필수적이지 않으므로 중복정보로 취급된다. 즉 중복정보는 Type2부터 Type5 사이에 해당하는 프로세스들이 인과순서를 유지하는데 필수적이지 않은 정보를 가지고 있을 경우로 이러한 정보의 파악과 제거는 인과순서를 이용한다.

동일한 프로세스에서 전송된 메시지들 간의 인과순서는 송신순서를 이용하면 쉽게 파악된다. 하나의 프로세스에는 동일한 프로세스에서 전송한 메시지에 대해서 최종 메시지에 대한 정보만 유지하면 되므로 나머지 메시지에 대한 정보는 중복정보가 된다.

중복정보는 전송된 메시지에 부착된 제어정보와 각 노드에 있는 제어정보를 비교하여 파악하며 인과순서 상 가장 뒤에 있는 메시지에 대한 정보만 유지하도록 하여 중복정보를 제거한다. 단 이 경우 모든 송신 노드에 대해 최소한 하나의 메시지에 대한 정보는 남겨 두어서 이후에 사용할 수 있도록 한다.

3.4 자료구조 및 전달조건

제어정보는 (k, τ) 의 형태로 송신프로세스 별로 분류하여 노드 i 에서는 $CI_i[j]$ 로 관리하며 메시지를 보낼 때마다 함께 보낸다. 이때 목적지에서의 전달조건을 확인하기 위한 제어정보는 분리하여 $DCM_i[j]$ 를 만들어 메시지와 함께 보낸다. 여기서 (k, τ) 는 프로세스 k 가 τ 시간에 보낸 메시지이고 $DCM_i[j]$ 는 CI 중에서 목적지가 j 인 제어정보를 별도로 추출한 것이다. 또 각 노드에는 프로세스 k 로부터 프로세스 i 에 전달된 최종 메시지의 송신시간을 나타내는 $DLV_i[k]$ 도 유지한다.

어떤 프로세스 P_i 에 도착한 메시지는 메시지와 함

게 전송된 제어정보에 포함된 인과관계 중에서 P_i 를 목적지로 하는 메시지들이 모두 P_i 에 전달되었거나 (DC2) 그들의 유효시간이 지났다는 것이 확인되면 (DC1) P_i 에 전달될 수 있다. 이 조건이 만족되지 않으면 해당 메시지는 조건이 만족될 때까지 목적프로세스에 전달되지 않고 대기하여 Δ -인과순서를 유지하도록 하며, 이는 다음 식으로 표시된다.

$$\forall (k, \tau) \in DC_m[i] :$$

$$\tau + \Delta < current_time : (DC1)$$

$$\tau \leq DLV_i[k] : (DC2)$$

3.5 알고리즘

가. 송신알고리즘 /* 프로세스 i 가 j 로 m 송신 */

- 1) $\tau_i = current_time$; /* 송신 현재 시간 */
- 2) CI_i 에서 $DC_m[j]$ 를 분리
Type5 형식의 중복정보 제거
- 3) CI_i 중 유효시간 지난 메시지 관련 정보 제거
단, 송신지별 최종 메시지에 대한 정보는 남김
- 4) τ_i, CI, DC 를 메시지와 함께 j 로 송신
- 5) (τ_i, j) 를 CI_i 에 포함시킴

나. 수신알고리즘 /* i 가 j 로부터 m 수신 */

- 1) $\tau_m + \Delta < current_time$ 이면 유효시간이 지났으므로 메시지 삭제
- 2) 전달조건 (DC1 or DC2) 만족될 때까지 대기
전달조건이 만족되면 메시지 전달
- 3) Type2 형태의 중복정보를 CI_m 에서 제거
- 4) CI_m 과 CI_i 를 비교하여 각각에서 Type3, 5 형식의 중복정보 제거
- 5) $CI_i = CI_i \cup CI_m$
 CI_i 에서 Type4 형식의 중복정보 제거
- 6) 유효시간이 지난 메시지에 대한 정보 삭제
단, 송신지별 최종 메시지에 대한 정보는 남김

4. 결론

멀티미디어 데이터는 제한된 시간 내에 사용되지 않으면 데이터의 효용성을 상실하게 된다. 본 논문에서는 분산환경에서 사용되는 멀티미디어 응용을 위한 효율적인 Δ -인과순서화 알고리즘을 제안하였다. Δ -인과순서화에서는 기존의 인과순서화에서 사용하던 논리시간을 사용할 수 없기 때문에 동기화된 물리적 전역시간을 사용하였다.

인과관계를 목적지별로 분류, 처리하던 방식을 개선하여 송신지 별로 인과관계를 분류하면서 전달조

건을 빨리 검사하기 위하여 필요정보를 별도로 추출하여 두 가지 방식의 장점을 취하였다.

또 인과순서를 제공하는 유효한 통신 패턴을 분석하여 5가지의 추상통신패턴으로 분류하고 인과순서를 유지하는데 필수적이지 않은 정보를 중복정보로 분류하여 제거함으로써 전송되는 제어정보의 양을 최소화하였다.

참고문헌

- [1] F. Adelstein and M. Singhal, "Real-Time Causal Message Ordering in Multimedia Systems," Proc. 15th ICDCS, pp.36-43, Jun. 1995.
- [2] S. Alagar and S. Venkatesan, "Causally Ordered Message Delivery in Mobile Systems," IEEE Trans. on Computer, vol.46, no.3, pp.353-361, Mar. 1997.
- [3] R. Baldoni, R. Prakash, M. Raynal, & M. Singhal, "Efficient Δ -Causal Broadcasting," Int. Jour. of Comp. Syst. Sci. & Eng., pp.263-271, Sep. 1998.
- [4] K. Birman and T. Joseph, "Reliable Communication in the Presence of Failure," ACM Trans. Comp. Syst., pp.47-76, 1987.
- [5] I. Jang, J. Cho, and H. Yoon, "An Efficient Causal Multicast Algorithm for Distributed System," IEICE Trans. on Info. & Syst., vol.E81-D, no.1, pp.27-36, 1998.
- [6] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," Comm. of the ACM, vol.21, no.7, pp.558-564, Jul. 1978.
- [7] D. Mills, "Internet Time Synchronization: The Network Time Protocol," IEEE Trans. on Comm., vol.39, no.10, pp.1482-1493, Oct. 1991.
- [8] R. Prakash, M. Raynal, and M. Singhal, "An Efficient Causal Ordering Algorithm for Mobile Computing Environments," Proc. 16th ICDCS, pp.744-751, May 1996.
- [9] P. Verissimo, "Causal Delivery Protocols in Real-Time Systems: a Generic Model," Real-Time Systems, vol.10, no.1, pp.45-73. 1996.
- [10] R. Yavatkar, "MCP: A Protocol for Coordination and Temporal Synchronization in Multimedia Collaborative Applications," Proc. 12th ICDCS, pp.606-613, 1992.