

□ 특 집 □

실시간 완료 규약 프로토콜

윤 용 익*

◆ 목 차 ◆

- | | |
|---------------------|---------------|
| 1 개 요 | 4. RCP의 운영 절차 |
| 2. 지연요소 패제 사례 | 5 결 론 |
| 3 Real-Time 완료 프로토콜 | |

요 약

원자성 완료 프로토콜 (Atomicity Commitment Protocol)은 기본적으로 일반적인 분산 시스템 내에서 트랜잭션들 사이의 원자성 및 일치성을 보장하기 위한 프로토콜이다. 이러한 프로토콜은 분산 트랜잭션 처리 과정 중에서 고장이 발생된 경우에 신뢰성 있게 진행이 될 수 있는 방안들을 제시하고 있다. 즉, 원자성 완료 프로토콜은 분산 처리 과정에서 발생하는 각종 오류 발생에 대하여 신뢰성 있는 진행이 이루어지도록 지원하여 준다.

그러나 일반적인 원자성 완료 프로토콜들은 실시간 시스템에서 요구하는 시간적인 요구 조건들은 고려하지 않고 설계되었다. 예를 들어, 두 단계 완료 프로토콜에서는 트랜잭션이 지연되는 문제가 발생할 수 있다. 또한 삼단계 완료 프로토콜에서는 빠른 처리가 이루어져야 하는 실시간 시스템의 기본 조건을 무시하고 있는 프로토콜이다.

따라서, 실시간 시스템에 유용한 원자성 완료 프로토콜은 단순히 원자성과 일치성만을 보장하는 것이 아니라 시간적인 제약조건도 만족할 수

있도록 설계되어야 한다. 본 논문에서는 실시간 분산 시스템을 위한 원자성 완료 프로토콜에 대하여 기술한다.

1. 개 요

고장 허용 (Fault Tolerance)을 추구하는 분산 시스템의 지속성은 운영되는 데이터 요소들인 데이터베이스 및 응용 프로그램들을 여러 사이트에 분산 시킴으로써 향상 시킬 수 있다. 다시 말하면, 분산 시스템의 신뢰성을 증진 시키기 위하여 분산 프로그램들에 의해 수행될 요소들을 여러 사이트에 복제 시켜, 고장에 따른 지연요소를 배제하고 있다[10, 11]. 이러한 고장 허용을 추구하기 위해 여러 사이트 내에 데이터 및 기능들을 중복 시킨 분산 시스템의 운영 중에 사이트의 고장을 포함한 오류 가 발생하는 경우에 중복 및 분산된 화일의 내용에 불일치성이 발생되게 된다. 따라서, 분산 환경하에서 데이터 화일에 대한 변경 요구가 발생하는 경우에 여러 사이트에 중복된 화일의 내용들도 동시에 변경이 이루어져야 하는 문제를 내포하고 있다. 즉, 모든 분산된 화일들 간의 상호 일치성을 유지하여야 만이 분산 시스템의 지속성 및 신뢰성을 유지 할 수 있다.

* 정회원 : 숙명여자대학교 전산학과 교수

분산 환경 하에서 오류의 발생으로 인하여 위와 같은 자료의 불일치성을 배제하고자 제안되었던 방법이 원자성 보장 완료 프로토콜이고, 그 중 가장 널리 사용되고 있는 프로토콜이 두 단계 완료 프로토콜이다[1, 2, 9]. 두 단계 완료 프로토콜의 진행 과정에서 협력자 트랜잭션은 해당 참여자 트랜잭션들로부터 수행된 결과 메시지를 수신하여야 만이 현재 변경하고 있는 화일에 대한 일치성을 유지시킬 수 있다. 즉, 같은 목적을 위하여 생성된 모든 트랜잭션들이 종료할 준비가 된 경우에만 협력자 트랜잭션이 자료의 일치성을 유지하기 위한 완료 행위를 결정 할 수 있기 때문에, 참여자 트랜잭션에 오류가 발생되어 수행 결과를 해당 협력자 트랜잭션에게 보고하지 못하는 경우에는 협력자 트랜잭션이 지금까지 수행된 결과에 대한 최종 결정을 할 수 없으므로 완료 프로토콜의 진행이 지연되고, 그 경우에는 실시간 시스템의 가장 중요한 요구 조건 중의 하나인 시간적 제약조건을 위배하는 문제점이 발생된다.

2. 지연 요소 배제 사례

실시간 환경 하에서 원자성 완료 프로토콜이 적용되기 위해서는 기본적으로 프로토콜의 진행 중에 협력자 트랜잭션이 참여자 트랜잭션의 결과를 기다리는 과정에서 발생 될 수 있는 지연 요소를 배제하여야 하는 상황이 가장 중요한 문제이다.

원자성 완료 프로토콜의 진행을 지연시키는 요소를 방지하기 위하여 두 단계 완료 프로토콜을 개선한 삼 단계 완료 프로토콜이 제시되었는데[1], 앞에서 기술한 것과 같이 삼단계 완료 프로토콜의 기본 전략은 두 단계 완료 프로토콜에서 협력자 트랜잭션이 참여자 트랜잭션의 결과를 기다리는 단계를 배제하기 위하여 중간에 완충 단계를 두어 참여자 트랜잭션이 오류의 발생으로 인하여 협력자

트랜잭션이 지연되는 것을 방지한 프로토콜이었다. 그러나, 이 방법은 프로토콜의 진행 과정에 완충 단계를 첨가한 형태로서 프로토콜의 진행 과정이 늘어난 형태이다. 즉, 완료 프로토콜의 진행이 종료하기 위해서 협력자 트랜잭션과 참여자 트랜잭션 사이에 주고 받는 메시지의 수가 한 단계 더 증가한 형태이다. 이와 같은 사이트 간의 통신 메시지의 증가는 시스템의 성능을 저하 시킬 수 있고, 특히 시간적인 제약조건을 만족 시켜야 하는 실시간 시스템의 경우에는 중요한 장애 요소가 된다. 따라서 이러한 삼단계 프로토콜은 일반적으로 실시간 시스템의 경우에는 적합하지 않다.

협력자 트랜잭션의 지연을 방지 시키기 위한 또 다른 방법으로 협력자 트랜잭션이 참여자 트랜잭션들과 메시지를 주고 받으면서 자료의 일치성을 보장하도록 완료 프로토콜의 진행이 이루어지는 형태가 아니라, 중간에 백업 프로세서를 새로이 설정하여 참여자 트랜잭션들은 그들의 해당 결과를 계속적으로 백업 프로세서에게 전송하여 유지 되도록 하고, 협력자 트랜잭션은 자료의 일치성을 결정하여야 하는 단계에서 백업 프로세서로부터 필요한 메시지 (해당 참여자 트랜잭션의 수행 결과 메시지)들을 추출하여 결정하는 방법이 있다[1, 4]. 이 경우에는 분산 시스템의 구성 요소 외에 또 다른 프로세서를 설정하여 분산 시스템의 운영 중에 발생하는 모든 메시지들을 유지하여야 하는 새로운 오버헤드가 발생된다.

이러한 두 가지 방법은 고장으로 인한 프로토콜의 진행 지연 배제 방법 제시가 설계의 목적으로 단순히 지연되는 문제점만을 개량한 형태였다. 따라서 두 가지 방법 모두 지연 요소를 배제하기 위하여 추가적인 완충 요소가 새로이 첨부되었고, 이러한 추가된 완충 요소와의 통신을 위한 메시지 교환 횟수가 증가 하였다. 실제 실시간 분산 시스템의 성능에 영향을 미치는 가장 큰 요소 중의 하나

는 사이트 간의 메시지 전송 횟수이며, 실시간 시스템의 가장 중요한 요소 중의 하나인 주어진 시간적인 제약 조건을 만족 시킬수 있는 가능성을 증진 시키기 위해서는 메시지 교환 횟수를 줄여야 한다 [11, 14, 15]. 따라서, 위에서 기술하였던 두가지 방법은 고장에 따른 프로토콜의 지연은 방지 시킬수 있지만, 두단계 완료 프로토콜 보다는 사이트간의 메시지의 교환 횟수가 증가한 형태 이므로 실시간 시스템을 위해서는 바람직하지 못하다. 이에 따라, 고장에 따른 완료 프로토콜의 지연 배제하면서 메시지 교환 횟수도 두단계 완료 프로토콜보다 적게 할 수 있는 새로운 프로토콜이 필요하다. 이를 위하여 제시된 방법이 실시간 완료 프로토콜(Real-Time Commit Protocol: RCP) 이다.

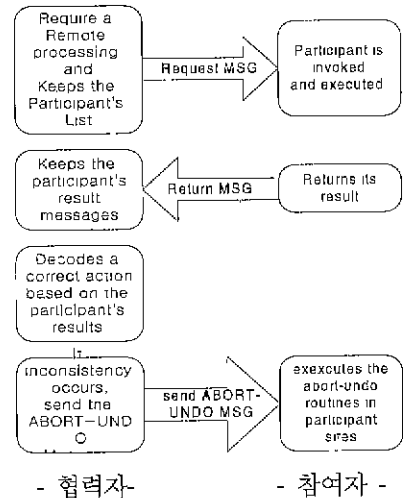
Real-time 완료 프로토콜은 실시간 시스템에서 요구하는 시간적인 제약 조건의 만족도를 높이기 위하여 완료 프로토콜의 메시지 교환 횟수를 줄였고, 분산 중복된 자료들 간의 일치성을 보장하기 위하여 제시되었다.

3. Real-Time 완료 프로토콜

분산 실시간 트랜잭션 처리과정에서 여러 단계의 협력자와 참여자 사이의 메시지 교환이 이루어진다. 이때, 협력자는 자체의 수행에 필요한 데드라인과 같은 시간적인 제약조건과 참여자가 분산처리 과정에서 요구되는 제약조건을 정의하고, 정의된 시간적인 제약조건을 참여자 트랜잭션이 생성 될 때에 전달한다. 이에 따라 참여자 트랜잭션은 주어진 시간 안에 수행이 이루어지게 되며, 수행결과를 협력자에게 보고하게 된다. 이때, 협력자는 통신상 및 사이트의 오류에도 참여자의 수행 결과에 대한 정확성을 보증하는 완료 규약 절차를 수행한다. 이러한 완료규약 절차는 참여자의 정확한 완료를 보장하게 된다. 이러한 절차는

기존의 완료규약절차와 달리 시간적인 제약 조건을 고려하여 이루어졌기 때문에 실시간 시스템에서 요구하는 시간적 정확성을 보장하게 된다.

RCP의 기본 절차는 (그림 1)에 도시 되어 있다. 협력자가 원격처리가 필요한 경우에 원격처리 메시지를 해당 사이트들에 전달하고, 이러한 메시지를 수신한 사이트에서는 메시지를 처리하기 위한 새로운 트랜잭션(참여자)을 생성시킨다. 이때 이들 협력자와 참여자는 상호 응답 메시지를 기다림이 없이 계속적으로 수행이 이루어진다. 참여자들은 수행이 종료된 후에 수행 결과에 대하여 완료 및 철회 행위를 수행하고 이에 대한 결과를 협력자에게 전달한다.

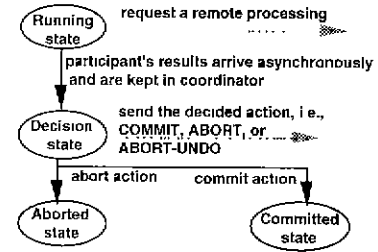


(그림 1) RCP의 흐름도

협력자는 시간적으로 참여자의 결과가 필요하다고 판단되는 시점에서 참여자들의 최종 행위 결과들을 분석하여 분산처리를 위해 수행한 결과에 대한 시간적 정확성과 수행상의 정확성을 검증하여 참여 트랜잭션들의 결과에 대한 일치성 및 원자성을 보장하게 된다. 이때 불치성을 유발 시킨 참여자가 존재하는 경우에 필요한 조치를 취하여 원자성이 보장되도록 한다.

4. RCP의 운영 절차

본 장에서 실시간 처리 성격을 요구하는 분산 시스템 내에서 신뢰성을 보장하기 위한 방법으로 제시된 실시간 완료 규약 프로토콜을 수행하는 협력자 트랜잭션과 참여자 트랜잭션의 운영 절차에 대하여 기술한다.



(그림 2) 협력자 트랜잭션의 운영 절차

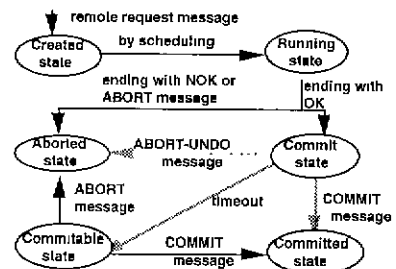
4.1 협력자 트랜잭션의 운영 절차

협력자 트랜잭션은 분산 처리를 필요로 하는 경우에 (그림 2)에 도시 되어 있는 것과 같이 참여자 트랜잭션에게 수행 명령을 전달한 후에 응답이 올 때까지 기다리지 않고, 계속적으로 다른 작업을 진행한다. 이후에 참여자의 결과를 필요로 하는 시점에서 수행 결과의 신뢰성을 검증하기 위한 완료 규약 절차를 수행한다. 이러한 절차를 수행하는 단계를 'Decision' 단계로 명명하였고, 이때에 참여자들의 수행 결과가 시간적인 제약조건 내에 정확히 수행되었는가를 분석하여 필요한 조치를 취한다.

시간 내에 정확히 수행된 경우에는 참여자 트랜잭션과의 수행을 종료하고, 만약에 그릇된 수행을 발생 시킨 참여자 트랜잭션이 존재하는 경우에는 참여자들 간의 일치성 및 원자성을 유지 하기 위한 절차를 수행한다. 즉, 협력자는 불일치성이 발생하였던 참여자 사이트에 일치성이 유지되도록 새로운 원격처리 메시지를 전송한다. (그림 2)에 도시 되어 있는 것과 같이 불일치성이 발생된 것으로 판단된 경우에는 이를 해결하기 위하여 ABORT-UNDO 메시지를 해당 참여자 사이트에게 전송한다. 이러한 메시지를 수신한 참여자 사이트는 불일치성을 해결하기 위한 새로운 트랜잭션이 생성된다. 이와 달리, 정상적으로 수행된 경우에는 협력자는 분산처리 과정을 종료한다.

4.2 참여자 트랜잭션의 운영 절차

참여자 사이트는 협력자로부터 원격처리 메시지를 수신하면, 이를 담당하는 새로운 참여자 트랜잭션이 생성된다. 즉, 참여자 트랜잭션은 협력자로부터 수행 명령을 받는 순간에 생성이 되며, 스케줄러의 제어를 받아 수행이 이루어진다. 참여자는 그 자체의 주어진 기능이 종료된 후에 일반적인 완료 규약 절차와 달리 독자적으로 자신의 수행 결과에 대한 완료 및 철회 행위를 판단하여 필요한 행위를 수행한다. 즉, 1차적으로 참여자들이 자신의 수행 결과에 대하여 독자적인 결정권을 갖는다. 이후 참여자들은 결정된 결과를 협력자에게 전달하여 2 차적인 판단을 의뢰한다. 이후 참여자들은 두 가지의 상태로 천이하게 된다. 하나는 완료를 결정한 후에 협력자로부터 최종 결정을 기다리는 상태인 (그림 3)의 Committable state와 철회를 결정하여 수행을 종료한 상태인 Aborted State로 천이한다.



(그림 3) 참여자 트랜잭션의 운영 절차

이러한 *Committable* 상태는 참여자의 수행결과가 완전하게 종료된 상태가 아니라 협력자의 최종 결정을 기다리고 있는 상태이다. 이 상태에서 협력자가 철회를 결정한 경우에는 협력자의 결정과 참여자의 수행 결과 사이에는 불일치성이 발생된다. 따라서, 이때에는 협력자는 참여자에게 이미 수행된 결과를 철회하도록 *ABORT-UNDO* 메시지를 전송하게 된다((그림 2)를 참조). 참여자 사이트는 이러한 메시지를 수신하면 불일치성을 해결하기 위한 보상 트랜잭션이 생성되어 필요한 조치를 취하게 된다. 이와 같은 보상 트랜잭션의 수행에 의하여 협력자와 참여자들 간의 일치성은 유지하게 된다.

5. 결 론

본 논문에서는 실시간 분산 트랜잭션 시스템 내에서 신뢰성 보장을 위한 실시간 완료규약 프로토콜의 기본적인 개념만을 제시하였다. 제시한 완료 규약 프로토콜은 시간적 제약 조건을 최대한 반영한 구조를 갖도록 설계되었다. 특히, 참여자 트랜잭션들이 독자적으로 수행 결과에 대한 완료 및 철회 행위를 결정할 수 있는 권한을 부여하여 실시간성을 최대한 보장하도록 하였다. 참여자에게 완료 행위를 허용함으로써 불필요한 메시지 전송의 횟수를 줄일 수 있게 되어 실시간성을 요구하는 시스템의 시간적 정확성을 향상시키는 결과를 얻을 수 있다. 그러나, 제시된 프로토콜에서 협력자가 참여자가 완료 행위를 취한 것에 대한 철회 행위를 재수행 하도록 하는 경우가 빈번하게 발생하는 경우에는 참여자 트랜잭션들에 대한 보상 트랜잭션이 수행되어야 하는 부담이 증가될 수 있다. 이 부분에 대해서는 추후에 계속적으로 연구되어야 할 부분이다. 특히, 신뢰성을 보장하기 위한 절차들에 대하여 보다 심도

깊은 연구가 필요하다.

참고문헌

- [1] Stefano Ceri and Giuseppe Pelagatti, *Distributed Database: Principles and Systems*, McGraw-Hill, 1984.
- [2] W. W. Chu, "Resilient Commit Protocol," *IEEE, Proceedings of 5th Real-Time Systems Symposium*, 1985, pp. 25-29.
- [3] Susan Davidson, Insup Lee, and Victor Wolfe, "Timed Atomic Commitment," MS-CIS-88-80, GRASP Lab 156, Dept. of Computer and Information Science, University of Pennsylvania, October 1989.
- [4] Hector Garcia-Molina and Robert K. Abbott, "Reliable Distributed Database Management," *IEEE Proceedings*, May 1987, pp. 601-620.
- [5] A.K.Elmagarmid, *Database Transaction Models for Advanced Applications*, Morgan Kaufmann Publishers, 1992.
- [6] Wolfgang A. Halang and Alexander D. Stoyenko, "Comparative Evaluation of High-Level Real-Time Programming Languages," *Real-Time Systems*, Vol. 2, No. 4, 1990, pp. 365-382.
- [7] H.F.Korth, E.Levy, and A.Silberschatz, "A Formal Approach to Recovery by Compensating Transactions," *Proceedings of the 16th VLDB Conference*, pp. 95-106, 1990.
- [8] Kwei-Jay Lin, "Consistency Issues in Real-Time Database Systems," *Proceedings of 22nd Hawaii International Conference on Systems Sciences*, January 1989, pp. 654-661.
- [9] C. Mohan and B. Lindsay, "Efficient Commit Protocols for the Tree of Process Model of

Distributed Transactions," IBM Research Report, RJ 3881, June 1983

[10] K.J.Lin, "Consistency Issues in Real-Time Database Systems," Proceedings of 22nd Hawaii International Conference on Systems Sciences, pp. 654-661, January 1989.

[11] K. G. Shin, "Introduction to the Special Issue on Real-Time Systems," IEEE Trans. on Computers, August 1987, pp. 901-902.

[12] J.A.Stankovic, "Misconceptions About Real-Time Computing," IEEE Computer, pp. 10-19, October 1988.

[13] M.Hsu and A.Silberschatz, "Unilateral Commit: A New Paradigm for Reliable Distributed Transaction Processing," IEEE 7th International Conference on Data Engineering, pp. 286-293, 1991.

[14] Mukesh Singhal, "Issues and Applications to Design of Real-Time Database Systems," SIGMOD Record, Vol. 17, No. 1, pp. 19-33, March 1988.

[15] Sang H. Son, "Real-Time Database Systems: A New Challenge," IEEE 6th International Conference on Data Engineering, Vol. 13, No. 4, December 1990, pp. 51-57.



윤 옹 익

1983년 동국대학교 통제학과 (이학사)
 1985년 한국과학기술원 전산학과 (공학석사)
 1994년 한국과학기술원 전산학과 (공학박사)

1998년-현재 숙명여자대학교 전산학과 교수
 관심분야 : 정보통신, 멀티미디어통신, 분산시스템, 실시간 처리 시스템, 분산 마들웨어 시스템, 분산 데이터베이스 시스템, 실시간 OS/DBMS

지로용지 작성시 유의사항

종전에는 지로용지 원본이 금융결제원에서 학회로 전달되었으나 현재는 지로용지 원본이 이미지출력으로 접수 되는 관계로 지로용지 작성시의 유의사항을 다음과 같이 안내드리오니, 보다 정확한 관리를 위하여 협조하여 주시기 바랍니다.

- ◆ 납부자 관련사항(회원번호, 성명)과 금액을 진하게 기재하여 주십시오.
- ◆ 금액을 정확하게 정자로 기재하여 주시고, 금액앞에 '₩' 표시는 가급적 사용하지 말아 주십시오.
- ◆ 금액, 납부자 성명과 같이 중요한 사항 기재시 또는 기재사항 변경시에는 적색(붉은색)을 사용하지 마시고 가급적 흑색을 사용하여 주십시오.