

# 분산 데이터베이스 시스템에서 효율적인 EPC 규약

안인순\*, 김경창\*\*

\*안동과학대학 정보처리학과

\*\*홍익대학교 전자계산학과

## Efficient EPC(Enhanced Presumed Commit) Protocol in Distributed Database System

Ihn-Soon Ahn\*, Kyung Chang Kim\*\*

\* Dept. of Computer Engineering, Andong Science College

\*\* Dept. of Computer Science, Hongik Univ.

### 요약

분산 데이터베이스 시스템에서 원자성 완료 규약으로 2PC(Two Phase Commit) 규약을 사용한다. 2PC 규약의 단점은 규약 수행시 조정자의 고장으로 인하여 참여자들이 블록킹이 될 수 있다. 참여자들이 블록킹되는 것은 자신들의 작업을 풀지 못하고 조정자가 고장에서 복구될 때까지 기다려야 한다. 블록킹을 해결하기 위해 제안된 방법은 3PC(Three Phase Commit Protocol) 규약이 대표적이다. 이 규약은 블록킹은 해결하지만 2PC 규약보다 많은 메시지 교환과 로그 기록을 하게 되고 규약을 수행하는 데 걸리는 시간이 증가된다. 본 논문에서는 2PC 규약을 향상시킨 EPC(Enhanced Presumed Commit) 규약에 블록킹을 해결할 수 있는 기법을 제안하므로써 기존의 3PC 규약보다 수행 시간을 향상시킨다

### 1. 서론

분산 데이터베이스 시스템에서는 사이트와 통신 고장에도 불구하고, 분산 트랜잭션의 원자성(Atomicity)을 확신하기 위해 완료 규약(Commit Protocol)을 수행한다[1,2]. 분산 트랜잭션의 수행은 트랜잭션이 제출된 사이트를 조정자(Coordinator)라 한다. 트랜잭션의 처리를 위해서 부트랜잭션으로 나누어 다른 사이트에 부트랜잭션의 수행을 요청할 수 있는 데 이 사이트들을 참여자(Participant)라 한다.

2PC 규약은 조정자는 선출단계(Voting Phase)에서 주기억장치에 유지하는 규약 테이블에 트랜잭션을 등록하고, 트랜잭션의 완료 준비가 되어있는 지 여부를 묻는 Prepare 메시지를 모든 참여자에게 보낸다.

참여자는 Prepare 메시지를 받으면 트랜잭션을 완료할 것인지 이ნი치 여부를 결정하여 조정자에게 Yes 또는 No 메시지를 보낸다. 참여자들이 보내는 Yes 또는 No 메시지에 따라 완료 또는 철회할 것인지 결정한다. 참여자는 마지막 결정 메시지(Commit 또는 Abort)를 받으면, 조정자에게 Ack 메시지를 보내고 결정을 승인한다. 조정자는 모든 참여자로부터 Ack 메시지를 받으면, 트랜잭션에 관련된 정보들을 주기억장치에 있는 규약 테이블에서 삭제하고 트랜잭션을 잇는다. 메시지 교환이 일어나는 동안 조정자와 참여자는 고장의 복구를 위하여 메시지 교환에 관한

내용을 로그 레코드를 하게된다. 2PC 규약은 조정자 사이트 또는 통신 고장이 발생할 경우 이를 복구하는 동안 모든 참여자는 트랜잭션의 수행을 더 이상 진행하지 못하고 정지해야 하는 블록킹이 발생한다. 2PC 규약의 가장 큰 문제점은 고장난 조정자 사이트가 복구될때까지 기다려야 하는 데 있다[1]. 이것은 참여자들이 Yes 메시지를 보내고 조정자에게서 전역적인 결정(독단적으로 완료 또는 철회)을 받지 못하고 조정자의 결정을 기다려야만 한다. 2PC 규약의 단점인 참여자의 블록킹을 해결하기 위해 제안된 방법이 3PC 규약이다[3]. 이 규약은 2PC에 Pre-commit 단계를 추가하여 블록킹 문제를 해결했다. 조정자 사이트에 고장이 발생하면 참여자들은 규약을 진행할 수 있는 새로운 조정자를 선출하는 선출 규약(election protocol)과 규약을 마감하기 위한 종료 규약(terminate protocol)를 행하여 전역적인 철회와 완료를 할 수 있도록 하였다. 그러나 3PC 규약은 3 단계로 규약이 수행되기 때문에 2PC 규약보다 더 많은 메시지 교환과 로그 레코드가 발생하고 따라서 규약 수행시간을 늦어지게 한다. 따라서 본 논문에서는 2PC 규약처럼 2단계로 완료 규약을 수행하고 3PC 규약보다 규약을 수행하는 데 걸리는 시간을 줄이면서 블록킹을 해결한 EPC(non-blocking Enhanced Presumed Commit NB-EPC) 규약을 제안한다.

본 연구의 구성은 2장에서는 EPC 규약에 대해서 설명하고 3장에서는 NB-EPC 규약 알고리즘에 대하여 설명한다. 4장에서는

NB-EPC 규약과 3PC 규약을 비교하고 5장에서는 결론 및 향후 연구 방향을 살펴본다.

## 2. EPC 규약

EPC 규약은 읽기전용 트랜잭션을 최적화하고 규약 수행시 메시지 교환 횟수와 로그 레코드를 줄인 완료규약이다. 규약의 수행은 다음과 같다. 조정자는 모든 부트랜잭션들을 Update로 규약 테이블에 등록한다. 참여자들은 연산을 수행하고 연산 결과를 일릴 때 읽기 전용인 경우 조정자에게 자신이 읽기 전용이라는 것을 알리기 위해 Ack 메시지에 Read-only를 첨가하여 보낸다. 이 메시지를 받은 조정자는 규약 테이블에 있는 참여자의 Update 등록을 Read-only로 변경한다. 이때 조정자는 참여자가 읽기 전용이라는 것을 알 수 있다. 이것은 읽기 전용 참여자는 복구가 가능하고, 트랜잭션의 직렬화 가능성이 보장되기 때문에 Initiation 레코드 하지 않는다

따라서 완료 규약이 수행되면 조정자는 규약 테이블을 검사하여 읽기 전용 트랜잭션인 경우는 Initiation 레코드 할 필요 없이 Read-only 메시지를 보낸다. 갱신 트랜잭션인 경우는 Initiation 레코드를 하고, Prepare 메시지를 보낸다. Read-only 메시지를 받은 참여자는 어떤 로그 레코드 할 필요 없이 자신이 보유한 모든 자원을 풀어주고 트랜잭션을 잇는다. 이것은 읽기 전용 트랜잭션의 완료 또는 철회와는 관계없이 선출 단계(Voting Phase)에서 읽기 전용 트랜잭션의 종료를 마감하기 위함이다. Initiation 레코드에는 읽기 전용 트랜잭션의 정보를 제외한 갱신 트랜잭션의 정보만 기억된다. Prepare 메시지를 받은 참여자는 조정자에게 Yes메시지를 보내고 결정을 기다린다. 트랜잭션을 완료할 때는 조정자는 Commit 레코드를 안전기억장치에 기록하고, 그때 Commit 메시지를 참여자에게 보낸다. 참여자는 결정 메시지를 받으면, Commit 레코드를 하고 결정에 대한 Ack 메시지를 보낼 필요 없이 트랜잭션을 완료한다. 트랜잭션을 철회할 때는 조정자는 로그에 철회 결정을 기록하지 않고, 반면에 Abort 메시지를 보낸다. 참여자는 결정 메시지를 받으면, Abort 레코드를 안전기억장치에 기록하고 결정에 대한 Ack 메시지를 보낸다

## 3. NB-EPC 규약·알고리즘

3PC 규약은 블록킹은 해결하였지만 Pre-commit 단계가 추가되어 규약을 수행하는 데 걸리는 시간이 증가하여 효율성을 저하시킨다. 따라서 규약의 수행 시간을 줄이고, 블록킹을 해결할 수 있는 새로운 규약을 제안한다. 가정으로 이 규약은 통신 고장은 고려하지 않고 사이트 고장인 경우만을 고려하였다. NB-EPC

Trans ID	Stable	State	{Participant Vote	{ACK}	종류	
	Yes No	Prepared Read-only Pre-commit Aborted Committed		Abort-vote(No) Commit-Vote(Yes) Pre-commit Commit	Yes No	Update Read-only

표1 규약 테이블

규약은 EPC 규약처럼 읽기전용 트랜잭션에 대한 최적화는 똑같은 방법으로 이루어진다. 그러나 규약 테이블에 기록되는 내용은 다르다. 규약 테이블에는 표1과 같은 내용이 저장되는 데 이것은 규약 수행시 발생하는 Pre-commit 메시지 때문이다. 왜냐하면 규약 수행시 트랜잭션 처리에 참여하는 모든 사이트의 전역 트랜잭션 관리자는 서로 Pre-commit 메시지를 주고 받는다. 다음으로 NB-EPC 규약의 수행은 그림 1과 같다.

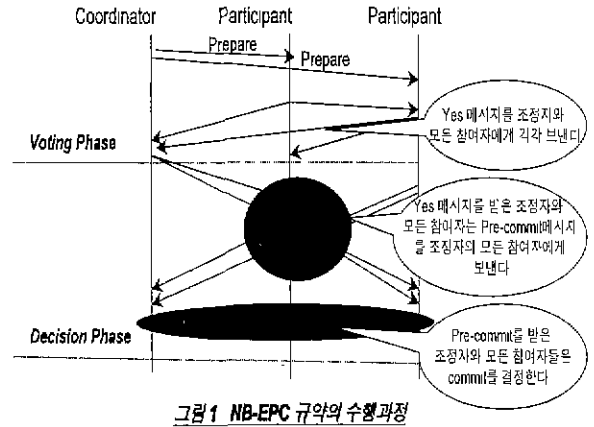
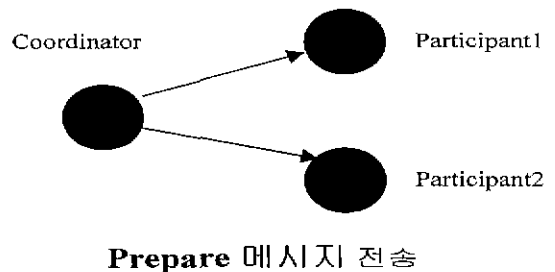


그림 1 NB-EPC 규약의 수행과정

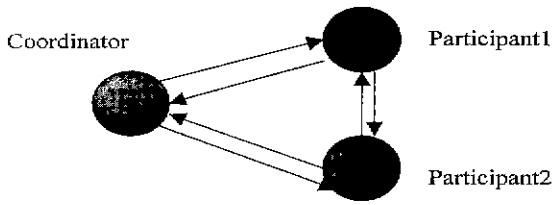
### ● 단계 1

조정자는 트랜잭션이 읽기전용인 경우에 로그에 Initiation 레코드하지 않고 참여자들에게 Read-only 메시지를 보내고 트랜잭션을 잇는다. 갱신 트랜잭션인 경우에는 로그에 Initiation 레코드를 하고 안전장치로 옮긴 후 Prepare 메시지를 참여자들에게 보내고 Yes메시지를 기다린다. 이때 Prepare 메시지에 완료 규약에 참여하는 모든 참여자들을 포함하여 보내게 되는 데 이 정보는 단계 2와 단계 3에서 완료 규약에 참여하는 조정자와 참여자들 사이에 서로 메시지를 주고 받는데 이용된다



### ● 단계 2

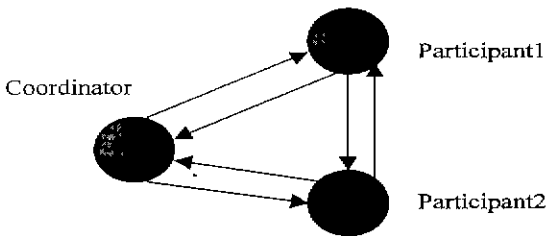
참여자들은 Read-only 또는 Prepare 메시지를 받으면 Read-only 메시지인 경우는 참여자는 완료규약을 끝내고 트랜잭션을 잇는다. 각각의 참여자들이 받은 메시지가 Prepare 메시지인 경우는 Prepared를 로그 레코드하고 안전장치로 옮긴 후 Yes 또는 No 메시지를 조정자와 모든 참여자들에게 보낸다. No 메시지를 보내는 경우에는 트랜잭션을 철회한다.



Yes 메시지 전송

● 단계 3

모든 참여자에게서 Yes 메시지를 받은 조정자와 참여자들은 모든 참여자들과 조정자에게 Pre-commit 메시지를 보낸다 이때 조정자와 참여자들은 Pre-commit를 로그 레코드하고 안전장치에 옮긴 후 모든 참여자에게서 Pre-commit 메시지를 받은 참여자는 로그에 commit 기록을 하고 트랜잭션의 완료를 결정한다. 조정자 경우에는 로그에 commit 기록을 하고 안전장치로 옮긴 후 완료를 결정한다.



Pre-commit 메시지 전송

위의 NB-EPC 규약 수행에서 보는 바와 같이 단계 1은 선출 단계(Voting phase)에 속하고 단계 2와 단계 3은 결정단계(Decision phase)에 속한다. 선출단계에서는 조정자가 규약을 조정하지만 결정단계에서는 조정자는 참여자의 역할을 참여자들은 조정자와 같은 역할을 한다. 사이트 고장이 발생하였을 경우 NB-EPC 규약의 종료 알고리즘을 간단하게 기술하면 다음과 같다 조정자와 참여자들은 로그에 기록된 메시지가 committed 이면 트랜잭션을 완료하고, Prepared이면 트랜잭션을 완료할 수 없는 불확실한 상태(Uncertain)이므로 트랜잭션을 철회한다. 또한 로그 레코드에 aborted가 없으면 트랜잭션을 철회하고, 로그 레코드에 Pre-commit이 있으면 다시 모든 참여자에게 Pre-commit 메시지를 보내고 다시 받은 메시지가 Pre-commit 이면 트랜잭션을 완료한다.

4. NB-EPC 규약과 3PC 규약의 비교

이 절에서는 정상적으로 완료규약이 수행되었을 경우, NB-EPC 규약과 3PC 규약의 장단점을 비교한다. NB-EPC는 3PC의 마찬가지로 블록킹을 방지하기 위하여 Pre-commit 메시지를 이용하지만 3PC 규약처럼 Pre-commit 단계를 두는 것이 아니라 2PC 규약과 마찬가지로 선출 단계와 결정 단계로 완료 규약을 수행한다 하나의 참여자는 모든 참여자들로부터 Yes 메

시지를 받으면 결정단계에서 모든 참여자들에게 동시에 Pre-commit 메시지를 보내고 받은 메시지가 모두 Pre-commit 이면 트랜잭션을 완료할 수 있기 때문에 수행시간이 3PC 규약보다 빠르다. NB-EPC의 단점은 참여자들은 Prepare 메시지를 받고 Yes 메시지와 Pre-commit 메시지를 분산 트랜잭션 처리에 참여하는 모든 사이트에 보낸다는 것이다 그러나 이것은 조정자 또는 참여자의 고장에 대해서 미리 서로가 완료 규약 진행 상태를 알고 있기 때문에 고장이 발생했을 때 조정자를 선출하는 선출 규약을 수행할 필요없이 3PC 규약보다 간단하고 빠르게 고장을 복구할 수 있는 장점이 있다. 또한 정상적으로 규약을 수행할 경우 결정단계에서 메시지를 동시에 규약에 참여하는 사이트로 보내기 때문에 통신 장애가 크지 않는 한 3PC 규약보다 빠르게 규약을 수행할 수 있다

5. 결론 및 향후 연구 방향

본 연구에서는 기존 연구에서 2PC 규약 보다 메시지 교환 횟수와 로그 레코드를 줄인 EPC 규약에 2단계 규약으로 블록킹을 해결하기 위한 NB-EPC 규약을 제안하였다.

향후 연구과제로는 NB-EPC 규약의 좀 더 구체적인 알고리즘을 제시하고 NB-EPC 규약과 2PC 규약, 3PC 규약을 모의 실험을 통하여 성능 평가를 해야 하겠다.

참 고 문 헌

[1] J. Gray, Notes on Data Base Operating Systems In Operating Systems: An Advanced Course, R Bayer, R Graham and G. Seegmuller(Eds.), Lecture Notes in Computer Science, Vol. 60, pp. 393-481, Springer-Verlag, 1978.  
 [2] B Lampson, Atomic Transactions, In Distributed Systems: Architecture and Implementation-An Advanced Course, B.Lampson(Ed.), Lecture Notes in Computer Science, Vol. 105, pp 246-265, Springer-Verlag, 1981  
 [3] D. Skeen, "Nonblocking Commit Protocols", Proce. of ACM SIGMOD Conf., June 1981.  
 [4] T Chandra and S. Toueg. "Unreliable failure detectors for reliable distributed systems", Journal of the ACM, 1996  
 [5] T. Chandra and S. Toueg. "Unreliable failure detectors for asynchronous systems", Proceedings of 10th ACM Symposium on Principles of Distributed Computing, 1991  
 [6] 안인순, 김정창, "PC(Presumed Commit Protocol)규약을 위한 읽기 전용 트랜잭션의 최적화", '98 정보과학회 추계학술발표회