제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

원격 백업 시스템에서 2-Safe 방식을 위한 최적화된 트랜잭션 완료 프로토콜

이창희*, 조행래
영남대학교 컴퓨터공학과
e-mail:chlee74@yumail.ac.kr*, hrcho@yu.ac.kr

Optimized Transaction Commit Protocol for 2-Safe Approach in Remote Backup Systems

Changhee Lee*, Haengrae Cho
Dept. of Computer Engineering, Yeungnam University

요 약

원격 백업 시스템은 주 시스템에서 재해가 발생했을 때 주 시스템의 데이타를 안전하게 복구하고 서비스를 지속적으로 유지시켜주기 위해 원격지에 주 시스템의 복사본을 두는 시스템이다. 주 시스템과 백업 시스템의 데이타를 동일하게 유지하기 위한 정책으로 1-Safe 방식과 2-Safe 방식을 들 수 있다. 1-Safe 방식은 주 시스템에서 트랜잭션을 완료한 후 백업 시스템에 완료 트랜잭션의 결과를 반영하는 방식이고, 2-Safe 방식은 트랜잭션이 주 시스템과 백업 시스템에서 동시에 완료하는 방식이다. 1-Safe 방식은 주 시스템에서 트랜잭션 처리 시간이 빨라진다는 장점을 갖지만, 재해가 발생했을 경우 백업 시스템에 반영되지 않은 완료 트랜잭션이 존재할 수 있다는 단점을 갖는다. 이와는 달리 2-Safe 방식의 경우 트랜잭션 처리 시간은 증가하지만, 주 시스템과 백업 시스템 두 곳에서 동시에 완료 트랜잭션의 안정성이 보장된다는 장점이 있다. 본 논문에서는 완료 트랜잭션의 안정성을 보장하는 2-Safe 방식을 위한 최적화된 트랜잭션 완료 프로토콜인 O2PC를 제안한다. O2PC는 주 시스템과 백업 시스템간에 메시지 전송 오버헤드와 로그 기록 오버헤드를 최소화함으로써 2-Safe 방식의 성능을 개선할 수 있다는 장점을 갖는다.

1. 서론

온라인 트랜잭션 처리가 필요한 응용 분야의 증가와 더불어, 고장으로 인해 서비스가 중지되는 것에 대한 많은 손실 등으로 인해, 고장으로부터 안전하고 연속적인 작업을 제공하기 위한 방식이 요구되고 있다[4]. 고장으로부터 안전하기 위해서는 원격 백업 시스템을 두는 것이 한 방법이다[5, 6, 7, 8]. 원격 백업 시스템에서는 주 시스템의 데이터베이스를 백업 시스템에서 복사본으로 유지하고 있다. 복사본은 인위적이거나 자연적인 재해 등이 발생했을 때에도 연속적인 작업이 가능하도록 가능한 한 지리적으로 떨어진 곳의 처리 노드에 위치한다. 트랜잭션 처리는 주 시스템에서 이루어지며, 주 시스템에서 생성한 로그 레코드가 백업 시스템으로 전파된다. 백업 시스템에서는 전송 받은 로그 레코드를 이용하여 데이터베이스를 최신의 상태로 재구성한다. 주 시스템에서 재해로 인한 고장이 발생했을 때, 백업 시스템은 서비스가 중지되는 일이 없도록 주 시스템의 작업을 이어 받는다.

원격 백업 시스템의 구성은 주 시스템과 백업 시스템으로 이루어진다. 주 시스템은 여러 처리 노드로 구성되어 있으며, 각 처리 노드에 일대 일로 대응하는 처리 노드들이 백업 시스템에 존재하며, 각각 데이타베이스와 로그 디스크들을 가지고 있다. 주 시스템의 로그 레코드들은 Multi Stream 방식으로 백업 시스템으로 전파된다. 그리고 주 시스템과 백업 시스템의 컴퓨터는 각각 조정자(Coordinator)와 참여자(Cohort)로 구성되어 있다.

원격 백업 시스템은 주 시스템에서 백업 시스템으로 로그를 전파하는 정책에 따라 1-Safe와 2-Safe의 두 가지 방식으로 분류된다[3]. 1-Safe 방식에서는 주 시스템이 트랜잭션을 완료한 후 백업 시스템

으로 트랜잭션의 로그를 전파한다. 이 경우 트랜잭션 처리량과 응답시간에서는 좋은 성능을 보인다. 그러나, 주 시스템에서 완료된 트랜잭션이 고장이 발생하기 전에 백업 시스템으로 온전하게 전파된다는 것을 보장해 줄 수가 없다. 이와는 달리, 2-Safe 방식에서는 트랜잭션의 로그 레코드가 백업 시스템에 전파되어 백업 시스템에서 완전히 반영되기 전까지 주 시스템에서 트랜잭션의 완료가 지연되며, 이를 위하여 주 시스템과 백업 시스템간에 이 단계 완료(2PC) 프로토콜을 이용한다[3]. 그 결과 고장이 발생한 경우에도 트랜잭션의 영속성이 보장된다. 하지만 트랜잭션에 의해 획득된 로크와 같은 자원을 트랜잭션 완료 시점까지 유지하여야하므로 2-Safe 방식은 주 시스템에서 로크 보유시간이 증가하게 되고 이로 인해 시스템의 처리량이 감소한다.

본 논문에서는 2-Safe 방식의 단점을 개선하기 위하여 1PC 방식[2]과 2PC의 변형 형태인 Presumed Commit 방식[9]을 적용한 최적화된 트랜잭션 완료 프로토콜인 O2PC를 제안한다. O2PC는 2PC에서 발생하는 메시지 오버헤드와 로그기록 오버헤드를 줄임으로써 백업 트랜잭션의 빠른 완료를 지원할 수 있다는 장점을 갖는다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구에 대해 살펴보고, 3절에서는 본 논문에서 제안한 O2PC 기법을 설명한다. 그리고 4절에서는 메시지 오버헤드와 로그 기록 오버헤드 관점에서 O2PC의 성능을 분석하고, 마지막으로 5절에서는 결론 및 앞으로의 연구방향에 대하여 논의하기로 한다.

2. 관련 연구

[1]에서 제안한 *o2-Safe* 방식은 1-Safe 방식과 2-Safe 방식의 바람직한 속성들을 이용하여 주 시스템에서 트랜잭션의 로크를 일찍 해제함으로써 2-Safe 방식의 단점을 해결하고, 1-Safe 방식의 단점인 백업 시스템에서의 완료 트랜잭션 안정성을 2-Safe 방식의 이 단계 완료 프로토콜을 이용해 보장해 준다. 그리고 제어 정보 테이블(CIT)을 이용하여 기존의 다른 알고리즘들에서 언급하지 않은 시스템의 부분적인 고장까지 해결책을 제시하고 있다. 그러나 o2-Safe 방식에서는 주 시스템에서 로크를 일찍 해제하는 대신 백업 시스템에서도 주 시스템에서의 트랜잭션 실행 결과와 동일한 순서를 유지해야 한다. 이를 위해 o2-Safe 방식에서는 주 시스템의 참여자가 백업 시스템의 참여자에게 전파하는 준비 로그 레코드와 주 시스템의 조정자와 참여자가 각각의 백업 시스템의 조정자와 참여자에게로 보내는 Commit Subject to Backup(CSB) 로그 레코드의 타임스탬프를 비교하여 백업 시스템에서의 트랜잭션 완료 순서를 유지하고 있다(CP Rule). 주 시스템의 조정자가 백업 시스템에서 완료되어야 할 트랜잭션에 대해 CSB 메시지와 CSB 로그 레코드를 생성해서 CSB 메시지는 주 시스템의 참여자에게 보내며, CSB 로그 레코드는 백업 시스템의 조정자에게 전파하고 있다.

o2-Safe 방식에서는 주 시스템의 트랜잭션에 대한 로크 보유시간이 단축되지만, 트랜잭션의 일관성을 유지하기 위해 백업 시스템에서도 주 시스템의 트랜잭션 처리 순서와 일치하도록 유지 하여야한다. 그 결과, 백업 시스템에서 트랜잭션 처리량이 감소할 수 있으며, 백업 시스템에서 주 시스템으로 트랜잭션 완료 메시지를 보내는 시점 또한 지연될 수 있다. 즉, 로크 보유시간은 단축되지만, 트랜잭션 실행 후 완료하기까지의 시간은 트랜잭션 의존성 여부에 따라 상대적으로 늘어날 수도 있으므로 최종적으로 주 시스템에서의 트랜잭션 완료시점이 지연될 수 있다. 또한 트랜잭션 일관성 유지를 위해 전파되는 로그 전송 및 기록 오버헤드가 증가하는 단점이 있다.

3. 트랜잭션 완료 프로토콜

본 절에서는 완료 트랜잭션의 안정성을 보장하는 2-Safe 방식을 위한 최적화된 트랜잭션 완료 프로토콜인 O2PC를 제안한다. O2PC는 주 시스템과 백업 시스템간에 메시지 전송 오버헤드와 로그 기록 오버헤드를 최소화함으로써 o2-Safe의 문제점인 백업 시스템에서의 트랜잭션 처리량 감소를 개선할 수 있다는 장점을 갖는다.

3.1 완료 과정

O2PC에서 트랜잭션 완료 과정이 그림 1에 나타난다. o2-Safe에서 준비 단계를 거치는 것에 비해 O2PC에서는 준비 단계 이전에 트랜잭션이 실행이 완료하는 시점에서 이미 트랜잭션 실행 완료 로그 레코드를 백업 시스템으로 전파하고, 준비 단계를 생략한다. 즉, O2PC는 2PC의 준비 단계를 생략한 1PC[2]의 개념을 채택함으로써 메시지 전송 및 로그 기록 오버헤드를 줄인다.

주 시스템의 참여자는 트랜잭션 실행완료 메시지를 조정자에게 보내면서 그 로그 레코드를 백업 시스템

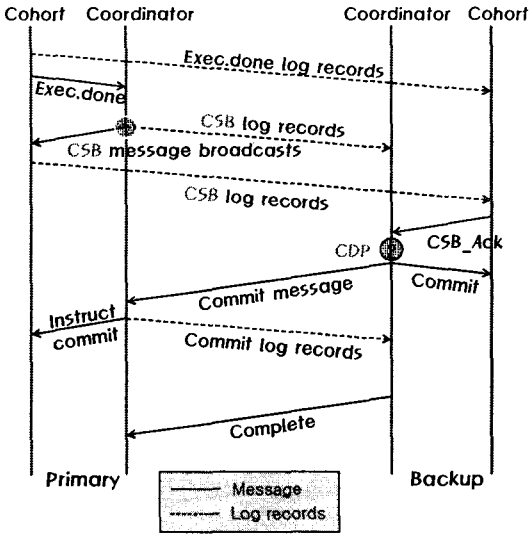

그림 1. 트랜잭션 완료 프로토콜

의 참여자로 보낸다. 이 로그 레코드는 o2-Safe에서의 준비 로그 레코드와 같은 역할을 하게 된다. 주 시스템의 조정자가 모든 참여자로부터 실행완료 메시지를 받으면, CSB 메시지를 모든 주 시스템의 참여자에게 보내고, 로그를 기록한 다음 트랜잭션에 관련된 로크를 해제하고, CSB 로그 레코드를 백업 시스템의 조정자에게 보낸다. CSB 메시지를 받은 주 시스템의 참여자는 로그를 기록한 다음 트랜잭션에 관련된 로크를 해제하고 각각에 해당하는 백업 시스템의 참여자에게 CSB 로그 레코드를 전파한다. 주 시스템의 참여자로부터 CSB 로그 레코드를 받은 백업 시스템의 각 참여자는 CSB에 대한 응답을 백업 시스템의 조정자에게 보낸다.

모든 백업 시스템의 참여자로부터 CSB에 대한 응답을 다 받게 되면 트랜잭션은 백업 시스템에서 Commit Decision Point(CDP) 상태로 변경된다. CDP에 이르면 백업 시스템의 조정자는 트랜잭션의 완료를 위해 완료 메시지를 백업 시스템의 각 참여자 뿐 아니라 주 시스템의 조정자에게도 보낸다. 완료 메시지를 받은 주 시스템의 조정자는 이 사실을 주 시스템의 참여자에게 알린다. 그리고, 트랜잭션 완료 로그 레코드를 백업의 조정자에게로 보낸다. 이때 주의할 것은 주 시스템의 참여자는 트랜잭션 완료 로그 레코드를 백업 시스템으로 보내지 않는다는 것이다. 즉, Presumed Commit 정책을 적용하여 백업 시스템의 참여자는 주 시스템의 참여자로부터 트랜잭션 완료 로그 레코드가 없더라도 주 시스템에서 트랜잭션이 완료한 것으로 간주한다.

트랜잭션의 준비단계를 생략하는 1PC를 적용할 경우 주 시스템에서의 트랜잭션 실행 완료가 곧 트랜잭션의 완료 준비 상태임을 나타낸 것으로 간주한다. 이를 위하여, 지역 트랜잭션의 실행을 마친 후에는 자발적인 트랜잭션 철회를 허용하지 않는 동시성 제어 기법을 사용하여야만 한다. 이러한 동시성 제어 기법으로는 Strict Two-Phase Locking(S2PL) 기법과 Strict Timestamp Ordering(STO) 기법을 들 수 있다. S2PL과 STO를 사용할 경우 1PC의 환경에서도 online-serializability와 cascadelessness 속성을 만족할 수 있음이 [2]에서 증명된 바 있다.

3.2 고장 시 회복 과정

본 논문은 재해로 인한 주 시스템이나 백업 시스템의 전체적인 고장 뿐 아니라 부분적인 고장까지도 회복을 보장한다.

3.2.1 백업 시스템의 부분적인 고장

백업 시스템의 처리 노드 중 하나 또는 그 이상에서 고장이 발생할 경우, 고장이 발생한 처리 노드의 역할을 이미 지정되어 있는 다른 처리 노드가 대신 수행하는 방법을 사용한다. 처리 노드의 고장이 탐지되면 그 처리 노드와 대응하는 주 시스템의 처리 노드는 이미 지정된 백업 시스템의 처리 노드로 로그 레코드를 보낸다. 그리고 로그 레코드를 받은 백업 시스템의 처리 노드는 그 로그 레코드를 저장하고 있다가 고장이 발생했던 처리 노드가 회복하게 되면 고장발생동안 수행되었던 작업을 넘겨받는다.

3.2.2 주 시스템의 부분적인 고장

주 시스템에서 하나 이상의 처리 노드가 고장이 발생할 경우, 고장이 발생한 처리 노드(P_j)에 대응하는 백업 시스템의 처리 노드(B_j)가 주 시스템 처리 노드의 역할을 이어받아 하게된다. 이때 새롭게 주 시스템의 역할을 하는 처리 노드의 백업 역할을 할 처리 노드가 필요한데 이는 이미 지정된 주 서버의 처리 노드(P_k)가 B_j의 백업 역할을 수행 하게된다. 그리고 고장난 처리 노드가 되살아나면 그동안 수행되었던 작업에 대한 로그 레코드를 보낸다.

3.2.3 전체적인 고장

백업 시스템에서 전체적인 고장이 발생할 경우 주 시스템은 싱글 시스템 모드로 전환하게 되고, 더 이

상 로그 레코드를 전송하지 않는다. 그리고 백업 시스템이 재 가동 될 때 고장 중 생성된 로그 레코드를 보낸다.

주 시스템에서 전체적인 고장이 발생할 경우 백업 시스템에서 주 시스템의 역할을 이어받아서 한 다음 주 시스템이 재 가동되면 백업 시스템과 통신하여 고장동안 발생했던 트랜잭션의 완료, 취소 여부를 결정한다.

4. 성능 비교

본 절에서는 o2-Safe와 O2PC 방식에 대해 메시지 전송 오버헤드와 로그 기록 오버헤드의 관점에서 성능을 비교한다. 주 시스템과 백업 시스템이 각각 n 개의 처리 노드로 구성된다고 가정할 때, 표 1에서와 같은 오버헤드를 보인다.

	o2-Safe		O2PC	
	준비 단계	완료 단계	준비 단계	완료 단계
메시지 전송	$2(n-1)$	$5n-3$	0	$2(2n-1)$
로그 레코드 전송	$n-1$	$2n$	$n-1$	$n+1$
로그 기록	$n-1$	$4n$	$n-1$	$3n+1$

표 1. O2PC와 o2-Safe의 메시지, 로그 전송, 로그 기록 오버헤드 비교

O2PC의 경우 1PC 방식의 적용으로 트랜잭션 준비 단계가 없으므로 준비 단계에서의 메시지 전송이 없다. 그리고 Presumed Commit 방식의 적용으로 백업 시스템의 참여자가 조정자에게 보내는 트랜잭션 완료 응답 메시지가 없고, 주 시스템의 참여자가 백업 시스템의 참여자에게 보내는 트랜잭션 완료 로그 레코드가 없으므로 트랜잭션 완료 단계에서 메시지 전송, 로그 레코드 전송 오버헤드가 o2-Safe 방식보다 줄어든다. 로그 레코드의 전송이 줄어듦으로 인해 로그 기록 또한 o2-Safe 방식보다 $n-1$번 줄어든다.

5. 결론 및 향후과제

2-Safe 방식은 1-Safe 방식에 비해 주 시스템과 백업 시스템간에 데이터가 항상 동일하게 유지되는 장점이 있지만, 이를 위해 두 시스템에서 동시에 트랜잭션을 완료해야 하는 단점이 존재한다. 이런 단점이 o2-Safe 방식에서는 상당히 보완이 되었지만, 트랜잭션 직렬화 오버헤드로 인하여 백업 시스템에서 트랜잭션 처리량이 감소하고 2-Safe보다 메시지 전송과 로그 레코드의 전송이 늘어나게 되었다. 본 논문에서 제안한 O2PC 기법은 o2-Safe의 장점을 그대로 살리며, 메시지 오버헤드와 로그 레코드의 전송 수를 줄임으로써 좀 더 나은 트랜잭션 처리량을 보일 수 있다는 장점을 갖는다. 본 논문의 향후 과제는 시뮬레이션을 이용하여 o2-Safe와 O2PC의 성능을 정량적으로 비교하는 것이다.

참고 문헌

[1] K. Hu, S. Mehrotra and S. Kaplan, "Failure Handling in an Optimized Two-Safe Approach to Maintaining Remote Backup Systems," in *Proc. Symp. on Reliable Distributed Syst.*, pp. 161-167, 1998.

[2] M. Abdallah, R. Guerraoui and P. Pucheral, "Dictatorial Transaction Processing: Atomic Commitment Without Veto Right," *Distributed and Parallel Databases*, pp. 239-268, 2002.

[3] J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann, 1993.

[4] J. Lyon, "Tandem's Remote Data Facility," in *Proc. IEEE Compcon.*, pp. 562-567, 1990.

[5] C. Polyzois and H. Garcia-Molina, "Evaluation of Remote Backup Algorithms for Transaction Processing Systems," in *Proc. ACM SIGMOD Conference*, pp. 246-255, 1992.

[6] H. Garcia-Molina, C. Polyzois and R. Hagmann, "Two Epoch Algorithms for Disaster Recovery," in *Proc. VLDB*, pp. 222-230, 1990.

[7] R. King, N. Halim, H. Garcia-Molina and C. Polyzois, "Management of a Remote Backup Copy for Disaster Recovery," *ACM TODS*, 16(2) pp. 338-368 1991.

[8] R. King, N. Halim, H.Garcia-Molina and C. Polyzois, "Overview of Disaster Recovery for Transaction Processing Systems," in *Proc. ICDCS*, pp. 286-293, 1990.

[9] C. Mohan, B. Lindsay and R. Obermarck, "Transaction Management in the R* Distributed Database System," *ACM TODS*, 11(4), pp. 378-396, 1986.