

전전자 교환기 시스템에서 write-through 캐쉬 일관성 프로토콜을 이용한 중복 데이터 일관성 유지 방안

원병재, 이규영, 조시철

LG 정보통신 중앙연구소 교환 SW 실

A Replicated Data Consistency Mechanism based on write-through cache coherence protocol for TDX system

Byong-Jae Won, Kyu-Young Lee, Shi-Chol Cho

Switching Software, LGIC R&D Lab.

bjweon@rex.lgic.co.kr

요 약

다중 프로세서 구조로 실시간 분산 처리를 하는 전전자 교환기 시스템은 그 특성상 2 개 이상의 프로세서에 동일한 값을 유지하는 중복 데이터의 사용이 필수적이다. 시스템의 자원 정보, 번호 번역 정보, 과금 정보 등이 중복 데이터로 사용된다. 이러한 중복 데이터에 대한 변경은 불일치 상태를 회피하기 위해 그 처리에 많은 비용과 제한이 따른다. 과도한 시그널 전송 및 로그 저장, 재전송 알고리즘은 데이터베이스 시스템의 성능을 저하시키고 때때로 순간적인 마비 상태까지도 유발할 수 있다.

본 논문에서는 기존 일관성 방안의 문제점을 분석하고 단일-버스 다중-프로세서 시스템에서 각각의 캐쉬들간의 일관성 유지를 위한 write-through 캐쉬 일관성 프로토콜을 사용하여 저 비용이며 효율적인 중복 데이터 일관성 유지 방안을 제시한다.

1. 서론

전전자 교환기 시스템은 복수개의 프로세서로 구성되며 상호 연동하여 기능을 수행하는 분산제어 방식으로 모든 소프트웨어가 프로세서들에 분산되어져 있다.[1] 데이터베이스 관리 시스템(DBMS) 역시 교환기의 분산 구조를 지원하기 위하여 분산구조의 형태로 되어 있으며, 따라서 하나의 프로세서에는 단지 완전한 데이터베이스의 일부만이 필요하고 이를

PLD(Processor Load Data)라 한다. [2] (그림 1)

이러한 분산 구조로 인하여 각 데이터를 접근 방법에 따라서 지역(local) 데이터와 전역(global) 데이터로 구분하고 물리적으로 존재하는 위치에 따라서는 단일(normal) 데이터와 중복(replicated) 데이터로 구분한다. 지역 데이터는 해당 프로세서의 PLD에 존재하는 데이터로 직접 접근이 가능하고 전역 데이터는 다른 프로세서의 PLD에 존재하며 이를 접

근하기 위해서는 IPC(프로세서간 통신: Inter Processor Communications)를 사용한 절차(protocol)가 필요하다. 단일 데이터는 전체 데이터 베이스에서 오직 한곳에만 존재하고 중복 데이터는 두 프로세서 이상의 장소에 존재하며 동일한 값을 유지해야 한다.

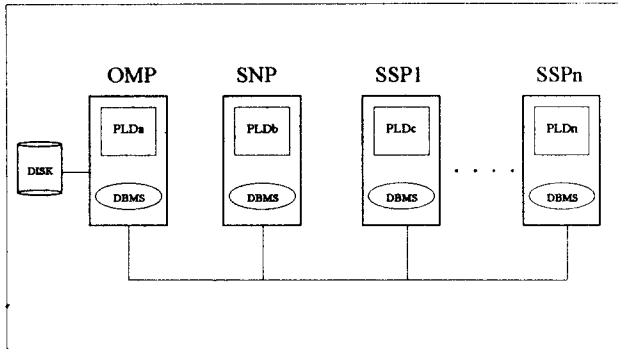


그림1. 전전자 교환기 DBMS 구조

단일 데이터에 대한 접근은 어느 경우든 명확하다. 해당 데이터가 존재하는 곳에서 필요한 처리가 이루어지며 중복 데이터에 대한 검색 역시 동일하다. 반면에 중복 데이터 변경은 중복된 모든 프로세서에서 해당 데이터의 변경이 이루어져야 하며 만약 중복 데이터 변경 이후 특정 프로세서의 데이터가 변경되지 않았다면 데이터 불일치 상태로 이 자체도 DBMS의 심각한(critical) 오류이지만 이 데이터를 사용하는 소프트웨어가 다른 오류를 발생시킬 가능성도 무시할 수 없다.

따라서 교환기 DBMS는 중복 데이터 일치를 위해 특정 프로세서를 중복 데이터 변경 처리 및 일치성 제어를 위한 주 프로세서로 지정하여 이를 관리하도록 하였다. 이 프로세서가 OMP(Operation & Management Processor)이며 이곳에는 유일하게 디스크가 존재하여 변경된 데이터에 대한 정보를 디스크의 PLD 파일로 백업하는 기능도 수행한다. 이제 모든 중복 데이터의 변경은 어느 프로세서에서 발생했든 OMP로 전달된다. 즉, 중복 데이터에 대한 변경허가를 요구하는 과정이다. OMP에서 변경에 성공하면 요청한 프로세서로 이를 알리어 변경을 수행하도록 허락하며 또한 중복된 다른 프로세서로 중복 데

이터가 변경되었음을 알리는 시그널을 보내어 일치를 이루도록 한다. 만약 IPC 이상 또는 DBMS의 이상으로 OMP가 중복 데이터 변경 요구에 대한 응답을 받지 못하면 해당 프로세서에 대해 변경 로그를 저장하고 주기적으로 프로세서의 상태를 검사하여 정상 상태가 되면 로그를 재전송하여 데이터를 일치한다. 이러한 절차 외에도 프로세서로 PLD 로딩 시에 또는 주기적인 중복 데이터 감사를 통해 데이터 일치를 수행한다.[3]

그러나 이러한 중복 데이터 일치 방안은 특정 프로세서가 주 프로세서로 지정되어 처리의 대부분이 수행되기 때문에 해당 프로세서의 DBMS에 과부하가 발생되며 전체적으로 시스템의 성능 저하 및 신뢰도 감소를 유발한다. 또한 중복 데이터 변경에 대한 결과가 성공일 경우, 결과를 받은 시점에서 타 프로세서의 중복 데이터가 변경되었음을 확인할 수 없다. 즉 어떤 프로세서에서 중복 데이터를 변경한 후 성공이면 다른 프로세서에서 변경된 중복 데이터를 사용하여 어떠한 처리를 수행하고자 할 경우에 과연 얼마의 지연(delay)을 두고 대상 프로세서로 시그널을 보내어 처리를 요구해야 할지 알 수 없다는 것이다. 이 지연시간은 OMP 및 IPC의 상태 및 성능에 따라 결정되는 것이므로 결국 운용 프로그램은 이러한 방법으로 중복 데이터를 이용할 수 없다.

따라서 본 문서에서는 write-through 캐쉬 일관성 유지 프로토콜을 사용하여 열거된 문제점들을 제거하고 시그널 전송 횟수 및 전송량을 효과적으로 분산시켜 시스템의 안정성을 향상시킨 새로운 중복 데이터 일치 방안에 대해 기술한다.

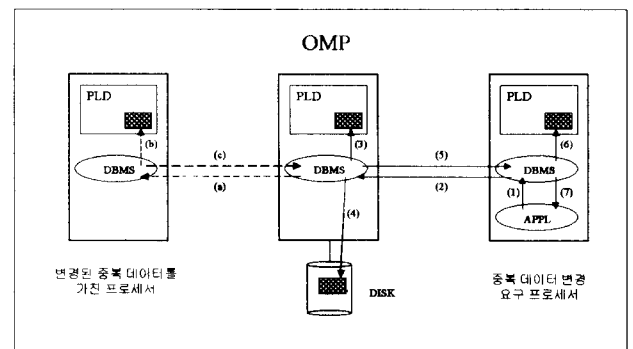


그림2. 기존의 중복 데이터 변경 절차

2. 기존 중복 데이터 일관성 유지 방안

현재의 중복 데이터 변경은 그림 2와 같이 이루어진다. (1) 중복 데이터 변경 요청은 해당 프로세서의 DBMS로 전달되고 이곳에서 조건에 맞는 대상 튜플을 찾는다. (2) 이후 이 요청을 OMP DBMS로 전달되고(3) 이곳에서 실제로 변경이 이루어진다.(4) 변경이 성공하면 디스크로의 백업이 수행되고(5) 결과를 중복 데이터 변경 요청 프로세서의 DBMS로 전송한다.(6) 성공을 확인한 최초 요구 프로세서의 DBMS는 실제로 중복 데이터를 변경하고(7) 운용 프로그램으로 복귀한다.

OMP에서 변경에 성공하면 (a) 중복된 모든 프로세서로 변경을 요구하는데 OMP와 최초 변경을 요청한 프로세서는 제외한다. (b) 중복 데이터 변경 요구를 받은 프로세서는 변경을 수행하고(c) 결과를 OMP의 DBMS로 전달한다. 이때, 성공하지 못했거나 결과를 보내오지 않은 프로세서에 대해서는 로그를 남기고 수시로 프로세서의 상태를 검사하여 정상동작을 하면 로그를 재전송한다. 모든 중복 프로세서가 변경되면 로그를 삭제한다.

기존 중복 데이터 변경 절차에 의한 시그널 전송횟수와 메시지 전송량을 측정하기 위해 위의 과정을 시그널 전송 순서로 보면 그림 3과 같다. (그림에서 각 번호는 그림 2의 변경절차 순서를 의미한다) 중복 데이터 변경 과정은 많은 시그널과 메시지 전달을 요하는 절차로 변경된 중복 데이터를 가진 프로세서의 개수가 많을수록 이 양은 증가한다.(표 1, 표 2 참조) 실제 운용에 있어서 15 내지 20 개의 프로세서는 일반적이다.

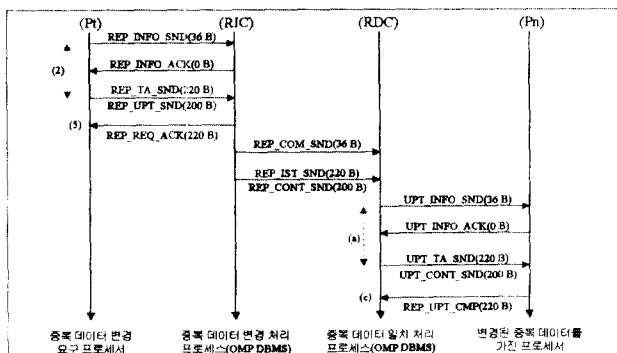


그림 3. 시그널 순서도

	Pt -> RIC	RIC -> RDC	RDC -> Pn
update	4/456	2/236	4/456
insert	4/476	2/256	4/476
delete	3/256	1/36	3/256

표 1. 각 명령별 처리비용 (전송횟수/전송량)

Processor #	5	10	15	20	30
시그널 개수	18	38	58	78	118
전송량(byte)	2060	4538	6620	9098	13460

표 2. 프로세서 개수에 따른 시그널 전송 횟수 및 전송량 (update 수행 예)

3. 개선된 중복 데이터 변경 처리기 구조

3.1. Write-Through Cache Coherence Protocol

단일버스-다중프로세서 구조에서 공유된 주기억 장치가 존재하고 각 프로세서 마다 별도의 캐쉬를 가진 경우, 캐쉬 내의 데이터 상태가 변경되면 주기억 장치 및 다른 프로세서의 캐쉬 상태가 일치 되도록 하여야 하며 이를 캐쉬 일관성 유지라고 한다.[4]

Write-through 쓰기 방식은 이러한 캐쉬 일관성 유지 프로토콜 중의 한 방식으로 프로세서가 캐쉬의 데이터 블록 내용을 변경하는 즉시 그 내용이 주기억 장치에도 갱신된다. 따라서 이 방식이 사용되는 시스템에서는 버스 감시기라는 별도의 하드웨어가 존재하여 타 프로세서가 주기억장치에 대한 쓰기 동작을 할 때, 그 주소를 입력으로 받아들여서 자신의 캐쉬에 있는 태그와 비교하여 일치하는 지를 검사한다. 만약 일치하는 경우(즉, 쓰기 동작에 의하여 내용이 변경되는 기억장치 블록이 자신의 캐쉬에도 존재할 경우)에는 그 블록을 무효화(invalidate)시킨다.[5](그림 4) 그렇게 되면, 프로세서가 그 데이터 블록을 접근할 때 캐쉬 적중(cache hit)이 되더라도 데이터를 캐쉬로부터 읽어오지 못하고 주기억장치로부터 다시 읽어와서 사용해야 한다.(그림 5)

캐쉬에 저장된 데이터 블록이 가질 수 있는 상태는 다음 두 가지 이다.

- 유효 상태(Valid state: V)

현재 캐쉬에 있는 데이터 블록은 주기억 장치의 내용과 동일하며 유효하다.

- 무효 상태(Invalid state: I)

현재 캐쉬에 있는 데이터 블록의 내용이 다른 프로세서의 캐쉬에서 변경되었으므로 더 이상 사용할 수 없는 상태이다.

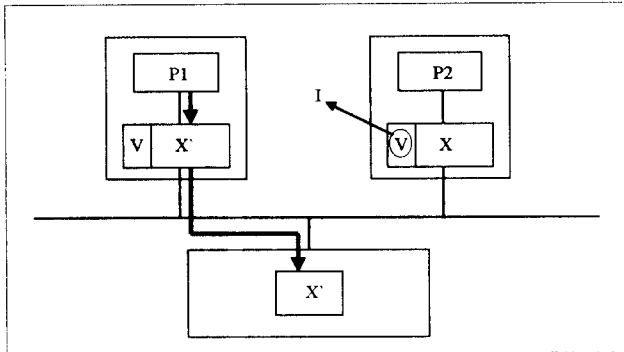


그림4. 일관성 유지 과정(1)

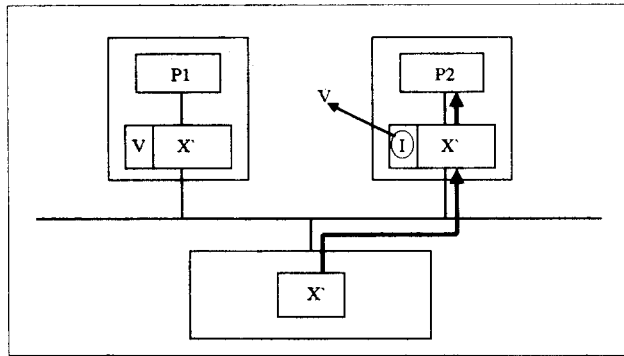


그림5. 일관성 유지 과정(2)

3.2. 중복 데이터 일관성 유지 방안

디스크가 OMP에만 존재하고 각 프로세서의 메모리에 PLD를 가진 전전자 교환기의 중복 데이터는 캐쉬 일관성 유지와 동일한 문제를 갖는다. 새로운 일관성 유지 방안은 OMP PLD를 주기억 장치로, 그 외 프로세서의 PLD를 캐쉬로 대체하여 write-through 쓰기 방법을 적용한 것으로 각 중복 데이터에는 상태 정보가 추가된다. 상태는 다음 두 가지 이다.

- 유효 상태(Valid state: V)
현재 PLD 내의 중복 데이터는 OMP의 내용과 동일하며 유효하다.
- 무효 상태(Invalid state: I)
현재 PLD 내의 중복 데이터의 내용이 다른 프로세서에 의해 변경되었으므로 더 이상 사용할 수 없는 상태이다.

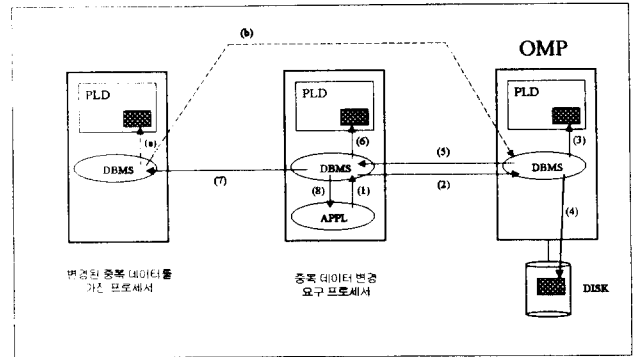


그림6. 개선된 중복 데이터 변경 절차

Write-through 캐쉬 일관성 프로토콜에서는 주기억 장치에 대해 쓰기 동작을 감시하는 하드웨어가 별도로 존재하여 다른 프로세서에 의해 변경되는 데이터를 갖고 있는 경우, 이를 무효화 한다. 그러나 전전자 교환기에서는 이러한 하드웨어가 존재하지 않으므로 이를 시그널을 이용한 소프트웨어로 구현하여야 한다. 즉, 중복 변경을 요구한 프로세서에서 OMP로부터 변경 성공을 수신하면 변경된 중복 데이터를 가진 프로세서로 중복 데이터 무효화 시그널을 전송하여 무효화 하도록 한다. 이 시그널은 단지 데이터의 위치 정보만을 갖는다.(그림 6)

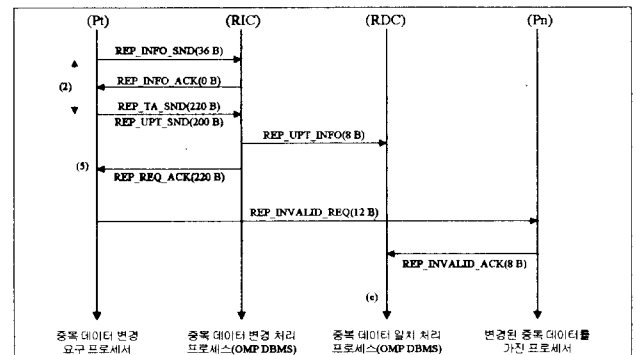


그림7. 새로운 일관성 유지 방안 시그널 순서도

중복 데이터 무효화 시그널을 전송한 변경 요청 프로세서는 결과를 기다리지 않고 DBMS 처리를 마친다. 시그널을 수신한 프로세서는 PLD에서 해당 데이터를 찾아 무효화 시킨 후 OMP로 응답을 보낸다.(그림 7) 프로세서가 비정상 상태이거나 IPC의 이상으로 시그널을 수신하지 못한 경우, 일정 시간 경과 후 OMP DBMS는 응답이 없는 프로세서에 대해 로그를 저장하고 재전송을 수행하게 된다. 이때도 단지 변

경된 중복 데이터의 위치 정보만을 전송한다.

새로운 방법에 의한 시그널 전송 횟수 및 전송량은 표 3, 표 4 와 같다.

	Pt -> RIC	RIC -> RDC	Pt->Pn->RDC
update	4/456	1/8	2/20
insert	4/476	1/8	2/20
delete	3/256	1/8	2/20

표 3. 각 명령별 처리비용 (전송횟수/전송량)

Processor #	5	10	15	20	30
시그널 개수	11	21	31	41	61
전송량(byte)	524	624	724	824	1024

표 4. 프로세서 개수에 따른 시그널 전송 횟수 및 전송량 (update 수행 예)

4. 결론

개선된 중복 데이터 일관성 유지 방법은 변경 데이터를 송수신하는 대신 무효화 시키는 방법으로 시그널 전송 횟수와 전송량을 감소시켰고 OMP에서 주관하던 중복 데이터 일치 작업을 각 프로세서로 분담하여 부하를 조절하였다. 또한 중복 데이터 변경을 요청한 프로세서에서 중복 데이터 무효화 시그널을 전송한 후 리턴하므로 운용 프로그램은 곧바로 변경된 중복 데이터를 사용하는 처리를 타 프로세서로 요청할 수 있다.

운용 프로그램이 무효화된 중복 데이터를 접근할 경우, 데이터를 OMP로부터 가져와야 하는데 이 비용은 시그널 2회에 256 바이트의 전송량이다. 이 비용을 포함하더라도 개선된 방법은 비용면에서 우수하며 타 프로세서에서 무효화된 데이터에 대한 접근이 변경 직후가 아닌 실제로 사용하는 시점에서 이루어 지므로 부하가 시간적으로 분산된다.

그러나 운용 프로그램이 무효화된 중복 데이터를 접근하여 DBMS가 OMP로부터 가져오는 중에 다른 프로그램이 동일한 데이터를 접근하면 역시 무효화되어 있으므로 동일한 절차로 동일 데이터를 전송 받게 된다. 이 문제는 상태 정보를 추가하여 방지할 수 있으며 시험상 의도적인 경우 외에는 발생하지 않았다.

5. 참고문헌

[1] H.K. Kim, et al., "System Concepts and Implementation of TDX-10 Digital Switching System," "Proceedings of Chinese-Korean Telecommunication Technology Symposium, April 1989

[2] 김영시 외 23인, TDX-10 소프트웨어, 한국전자통신연구소, 1994, pp.29-42.

[3] 이규영 외 3인, STREX-TX1 DBMS 사용자 설명서, LG 정보통신, 1995, pp.15-21.

[4] Emil Gustafsson and Bruno Nilbert, "Cache Coherence in Parallel Multiprocessors, Uppsala 24th February 1997, 1997.

[5] 김종현, 병렬 컴퓨터 구조론, 생능출판사, 1994, pp.142-148.