

트랜잭션의 복제 일관성을 위한 갱신 전파 알고리즘 설계

A Design of Update Propagation Algorithm for Replica Consistency of Transaction

이 병 욱*
Byung-Wook Lee

요 약

자료 복제는 분산 시스템과 데이터베이스 시스템에서 가용성과 성능을 향상시켜 주지만, 자료를 갱신할 때 일관성을 엄격하게 유지하는 것은 쉽지 않다. 기존 알고리즘들은 엄격하게 일관성을 유지하지만, 비용이 많이 들며 시간이 지연된다는 문제점이 있다. 본 논문에서는 원본 자료에 대해서 즉시 갱신을 허락하고 다른 복사본들에 대해서는 지연 갱신을 전파하는 혼합 갱신 전파 알고리즘을 소개한다. 혼합 갱신 전파 알고리즘은 또한 소유권에 있어서 그룹 갱신을 허락한다. 이 알고리즘은 그룹 갱신 일관성을 제어하기 위해 복제 버전을 관리한다. 그룹의 소유권을 가지고 즉시 갱신과 지연 갱신을 결합함으로써 일관성과 성능이 향상된다. 응용과 실행 환경에 따른 트랜잭션 처리량과 응답시간의 향상을 모의실험을 통해 보여준다.

Abstract

Data Replication makes distributed system and database system improved in availability and performance. But it is difficult to keep strict consistency in data update. Existing algorithms keep strict consistency, but have problems of cost a great deal and delay time. It is to introduce hybrid update propagation algorithm which permits eager update for original copy and propagates lazy update for other copies. Hybrid update propagation algorithm also permits group update in ownership. This algorithm manages replica version in order to control group update consistency. Consistency and performance is improved by combining eager update and lazy update with group ownership. Simulation shows improvement of transaction throughput and response time depending on application and execution environment.

Keyword : data replication, hybrid update propagation algorithm, serialization, consistency

1. 서 론

성능과 가용성을 위해 분산 시스템과 데이터베이스 시스템에서 자료를 복제한다. 복제된 데이터베이스 시스템은 복제 일관성을 만족하기 위해 2단계 완료 프로토콜을 사용하거나 지연 갱신 프로토콜을 사용한다. 2단계 완료 프로토콜은 복제된 자료 일관성과 트랜잭션의 직렬성을 보장한다는 장점이 있다. 그러나 많은 메시지 전송과 완료될 때까지 오랜 지연으로 인한 데드락 가능성이 있다는 단점이 있다. 지연 갱신 프로토콜은 단지 원본 자료를 갱신한 후에 트랜잭션을 완료하고

다른 사본들에게 비동기적으로 갱신을 전파한다. 이 방법은 대기 시간이 없기 때문에 더 빠른 응답 시간을 준다. 그러나 비 직렬 스케줄로 인한 충돌 방지를 위한 메커니즘이 필요하다[2].

자료를 갱신하는 것에 대한 소유권은 자료 갱신 권한에 따라 원본 갱신과 그룹 갱신 알고리즘으로 나뉘어진다. 갱신은 원본 갱신 알고리즘에 대해서는 원본 사이트 상에서 가능하며 복제 사이트는 읽기만 가능하다. 이것을 다중 읽기/단일 쓰기 프로토콜이라 부른다. 그룹 갱신은 어떤 사이트에서도 읽기와 갱신이 가능하다. 이것을 다중 읽기/다중 쓰기 프로토콜이라 부른다. 그룹 갱신은 복잡한 절차 대신에 모든 사이트에 대해 같은 코드를 사용하는 장점이 있다[10]. 원본 사이트에

* 종신회원 : 경원대학교 소프트웨어대학 교수
leebw@kyungwon.ac.kr(제1저자)

서는 동기적으로 즉시 갱신을 수행하고 다른 복사본에서는 비동기적으로 지연 갱신을 처리하는 혼합 갱신 전파 알고리즘을 제안한다. 일관성 정도를 완화하기 위해 지연 갱신 전파를 사용하고, 신속한 갱신 전파를 위해 복사본 트리 경로를 사용한다. 그룹 갱신 경우에 복제 일관성을 관리하기 위해 버전번호를 사용한다.

약 일관성 프로토콜을 사용함으로써 트랜잭션 완료 시간이 절감될 수 있다. 갱신 충돌로 인해 발생하는 트랜잭션 오류는 버전번호를 사용함으로써 방지할 수 있다. 또한 복사본 스페닝 트리 경로를 줄임으로써 신속한 지연 갱신을 수행한다.

2. 복사본 갱신 알고리즘

지연 갱신 알고리즘은 비 직렬화 문제를 야기한다. 특별히, 활성화된 트랜잭션이 나중에 이전 트랜잭션의 갱신을 겹쳐 쓰게 되는 갱신 손실 문제가 야기된다. 몇 가지 알고리즘이 트랜잭션 스케줄을 직렬화하기 위해 개발되었다.

2.1 방향성 비순환 그래프 알고리즘

방향성 비순환 그래프(DAG) 알고리즘에서 time stamp를 사용하지 않는 알고리즘은 문제를 해결하기 위해 방향성 비순환 그래프로부터 구성된 트리의 에지(edge)를 따라 복사본 갱신을 전파한다. 일단 트랜잭션이 방향성 비순환 그래프에 의한 사이트 S_i 에서 실행될 때, 트랜잭션의 갱신 작업들은 복사본 트리 T 에서 S_i 의 자식들에게 전달된다. 임의의 사이트에서 발생된 초기 트랜잭션들을 주 트랜잭션이라 부르고 부모에 의해 전달된 트랜잭션들은 부 트랜잭션이라 부른다. 부 트랜잭션들은 도착 순서에 따라 완료되고 그들의 자식들에게 갱신 메시지를 전달한다.

마지막으로 한 사이트의 주 트랜잭션과 부 트랜잭션들의 모든 전송은 완료 순간에 자동적으로 실행된다. 그러므로 만약 어떤 사이트에서 T_i 가 T_j

전에 완료된다면 T_j 가 전달하기 전에 T_i 는 자식들에게 전달된다. 복사본 트리 T 에서 에지를 따라 순차적으로 부 트랜잭션들을 전파하는 경우에, 한 트랜잭션에 대한 부 트랜잭션이 실행될 때, 모든 선행 트랜잭션들이 그 사이트 전에 완료되었다는 것이 보장된다.

방향성 비순환 그래프 알고리즘이 복사본 트리 T 의 에지를 따라 갱신을 수행하므로, 부 트랜잭션들이 갱신이 수행되지 않은 사이트로 전달된다. 이것은 메시지 부담을 의미하며 불필요한 사이트를 따라 전파 시간이 지연된다는 것을 의미한다[3].

2.2 역에지 알고리즘

역에지(Backedge) 알고리즘은 에지를 역으로 연결하여 순환하는 기법을 이용하여 복사본 그래프가 순환 그래프라도 직렬화 스케줄을 보장하는 혼합 알고리즘이다. 어떤 경우에는 즉시 갱신 전파를 사용하고 또 다른 경우에는 지연 전파를 적용한다.

복사본 그래프에서 특정 에지를 삭제함으로써 모든 순환을 제거할 수 있는 에지의 집합을 역에지라 한다. 만약 G 가 순환을 포함하는 임의의 복사본 그래프이고 B 가 G 의 최소의 역에지 집합이라면 B 에 어떤 에지를 추가함으로써 생성되는 방향성 비순환 그래프는 순환을 포함한다. 집합 B 의 그래프는 깊이 우선 탐색을 사용해서 계산될 수 있다. G_{dag} 는 G 에서 B 에지 집합을 삭제하여 생성되는 방향성 비순환 그래프이다.

T 는 비순환 그래프를 만족하는 G_{dag} 에 의해 생성된 트리이다. 만약 사이트 S_j 로부터 사이트 S_i 까지의 역에지가 있다면 S_j 로부터 S_i 까지의 최소 경로가 존재한다. 그러므로 S_j 는 트리 T 에서 T 의 속성에 의해서 S_i 의 조상이 될 수 있다. 트랜잭션을 갱신하는 것은 T 에서의 역에지를 따라 선행자에 대한 즉시 갱신을 수행한다. 선행자에 대해 즉시 갱신을 완료한 후에 방향성 비순환 그래프에서 자식인 사이트들에 대해서는 지연 갱신을 수행한다.

다[3]. 역에지 알고리즘은 복사본 그래프에서 직렬성을 관리하는 장점이 있지만 록킹 시간이 길어 병행성을 감소시키는 단점도 있다.

2.3 복제 일관성의 조건

이 논문에서 적용된 약 일관성은 1986년에 Dubois에 의해 정의되었다. 약 일관성의 조건은 다음과 같다. 1) 직렬성 변수들까지의 접근이 순차적인 일관성이 유지되도록 관리된다. 2) 트랜잭션들이 자료까지의 모든 접근을 실행하기 전에는 직렬성 변수에 접근이 허용되지 않는다. 3) 직렬성 변수에 접근된 트랜잭션들이 실행되기 전에는 자료까지의 접근이 허용되지 않는다. 그러므로 직렬성 변수의 접근은 울타리 역할을 한다[7].

분산 데이터베이스에서 복제 일관성은 약 일관성으로 직렬성을 관리할 수 있다. 일관성 기준을 완화하기 위해 이 논문에서는 모든 사이트에서 세마포어를 사용한다. 세마포어는 FIFO에서 트랜잭션을 동기화함으로써 트랜잭션 처리를 직렬화한다.

3. 혼합 갱신 전파 알고리즘

즉시 갱신은 일관성은 향상 시켰지만 성능을 감소시켰다. 혼합 갱신 전파 알고리즘은 즉시 갱신 내에서의 약 일관성과 지연 갱신에 의한 성능 향상을 이끌어 냈다. 또한 사용자의 편의를 위해서 그룹 사용자의 소유권을 사용한다.

3.1 알고리즘의 구성

원본을 저장하고 관리하는 사이트를 원본 사이트라 하고, 원본이 다른 사이트에 복제된 자료를 사본이라 하며 사본을 저장하고 관리하는 사이트를 사본 사이트라고 한다. 트랜잭션이 같은 사이트에서 기동되고 실행되는 트랜잭션을 주 트랜잭션이라 하고, 주 트랜잭션에 의해 다른 사이트로

보내지는 갱신 메시지를 리플(ripple)이라 하며, 리플에 의해서 기동된 트랜잭션을 부 트랜잭션이라고 한다[4,5].

혼합 갱신전파 알고리즘의 갱신 연산의 특징은 다음과 같다. 1단계는 사본 사이트 i 의 클라이언트가 갱신을 요청하여 사본 사이트 i 에서 원본 사이트로 접근하는 것이다. 2단계는 사본 사이트 i 가 가진 자료의 버전과 원본 사이트가 관리하는 자료의 버전을 비교한다. 버전이 같으면 원본 사이트와 사본 사이트 i 는 동기적으로 즉시 갱신한다. 3단계는 직렬성을 유지할 수 있는 실행 순서를 결정하는 단계로서 원본 사이트에서 갱신 전파를 위한 사본 트리 경로를 변경한다. 이 단계에서, 원본 사이트의 갱신 전파를 위해서 사본 트리의 경로가 재배열된다. 4단계는 3단계에서 변경된 사본 트리 경로에 따라 다른 사본 사이트들에게 지연 갱신 전파하는 단계이다. 5단계는 사본 사이트들이 원본 사이트에 결과를 응답해 주는 단계이다[4].

3.2 복제 일관성의 직렬화

혼합 갱신 전파 알고리즘은 다중 읽기/다중 쓰기를 허용하기 때문에 갱신 유실 문제(lost update problem)를 야기할 수 있다. 갱신 유실 문제를 해결하기 위하여 각 자료마다 버전번호를 두어 관리하고, 원본을 갱신하는 트랜잭션은 버전번호를 1씩 증가시킨다. 사본 사이트로 갱신 메시지를 전파할 때 갱신된 자료 값과 함께 자료의 버전번호를 전파한다. 사본 사이트에서 사본을 즉시 갱신하기 전에 사본 사이트의 자료 버전번호와 원본 사이트의 자료 버전번호를 비교한다. 버전이 서로 같으면 즉시 갱신을 허용하고, 버전이 서로 다르다면 이미 다른 사본 사이트에서 원본을 갱신한 후 갱신 전파가 완료되지 않은 상태이므로 갱신 요청 트랜잭션을 취소하고 그 결과를 갱신을 요청한 사본 사이트로 전달한다[9].

혼합 그룹 갱신은 갱신들이 저장-전진(store-and-

forward) 방식으로 최종 목적지까지 전파되는 신뢰성 있는 방법으로 아래와 같다. 원본 사이트 S_i 의 주 트랜잭션 T_i 는 원본 자료 항목 d_i 를 갱신하고 완료한 후 S_i 의 자식 사이트인 S_j 에게 갱신을 전파한다. 갱신을 전파 받은 사본 사이트 S_j 에서는 T_i 의 부 트랜잭션 T_{ij} 를 실행하여 사본을 갱신하고 자신의 자식 사이트에게 다시 갱신을 전파한다. 따라서 사이트 S_k 에서 부 트랜잭션 T_k 가 실행되는 것은 그 앞의 모든 트랜잭션 T_k 가 완료되었음을 의미한다. T_k 후의 복사본 트리의 모든 서브 트랜잭션들이 트리를 따라 순차적으로 갱신을 실행한다. 이것이 지연 갱신 알고리즘이며 직렬성과 동기화 일관성 조건을 만족한다[7]. 사본 사이트 S_k 에서 사본을 갱신하는 주 트랜잭션 T_k 는 원본 사이트 S_i 가 관리하고 있는 원본 자료 항목 d_i 의 버전과 사본 d_k 의 버전이 동일해야 즉시 갱신을 허락한다. 원본과 사본이 갱신된 후에, 원본 사이트의 자식 사이트인 S_j 로 지연갱신을 전파한다.

자료를 갱신하는 트랜잭션은 자료의 사용을 배제하기 위한 임계 영역의 진입을 필요로 한다. 트랜잭션 T_k 가 동기화된 변수들의 접근을 가지고 있다는 것은 그 이전에 자료 항목을 갱신한 모든 트랜잭션들이 트리 T 에서 복사되기 전에 갱신을 완료한 것을 의미한다. 동기화 변수가 다른 트랜잭션을 위해서 사용될 때, 대기 큐에서 FIFO 순서로 대기하며 순차 일관성을 유지하여 처리된다. 이것은 약 일관성의 첫 번째 조건이다. 두 번째 안전한 조건은, 모든 자료에 대한 갱신 트랜잭션이 수행되기 전에는 동기화 변수에 접근을 허용하지 않는다. 약 일관성의 세 번째 안전 조건은, 동기화 변수에 대한 접근이 수행되기 이전에는 자료에 대한 접근을 허용하지 않는다[7].

3.3 혼합 갱신 전파 알고리즘

혼합 갱신 전파 알고리즘은 4개의 갱신 알고리즘으로 이루어져 있다. 첫 번째 주 트랜잭션을 위

한 원본 갱신 알고리즘과 복제 갱신 알고리즘이다. 두 번째 부 트랜잭션을 위한 복제 지연 갱신 알고리즘과 즉시 원본 갱신 알고리즘이다. 이들은 사본 갱신 시에 일관성을 유지하기 위하여 그룹 단위로 갱신되므로 그룹 갱신 알고리즘의 특성을 가진다.

즉시 갱신 알고리즘은 사본 사이트의 클라이언트가 갱신을 요청함으로써 수행된다. 사본 사이트 S_j 에서 사본을 갱신하기 위하여 주 트랜잭션 T_j 를 가동하고 원본 사이트에 부 트랜잭션 T_i 를 호출하여 동기화를 요청한다. 원본 사이트의 버전 번호가 사본 사이트의 버전 번호와 같으면 자료를 갱신하고 완료한다. 만약 그렇지 않으면 기동된 트랜잭션 T_j 를 취소하고 다른 트랜잭션의 지연 갱신 절차가 완료된 후 다시 시도한다.

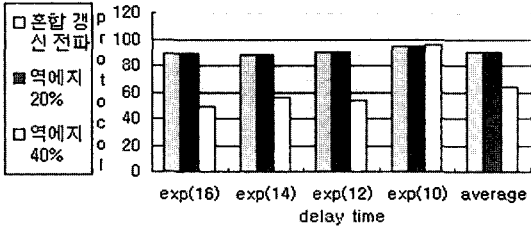
부 트랜잭션을 위한 즉시 원본 갱신 알고리즘은 사본 사이트의 갱신 요청에 의해서 실행된다. 두 버전의 번호가 같으면, 자료를 갱신하고 결과를 사본 사이트에 전달한다. 갱신을 요청한 사본 사이트를 사본 트리에서 제거하여 사본 트리를 재구성한 후 재구성된 사본 트리의 첫 번째 자식 사이트들에게 리플을 전송한다. 버전번호가 다르다면 트랜잭션을 취소하고 결과를 사본 사이트에게 전달한다.

4. 평가 분석

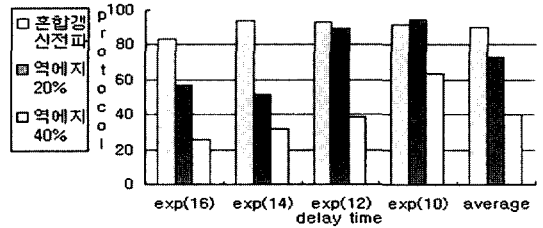
이 알고리즘의 성능은 모의실험에 의해서 평가된다. 모의실험 모델은 분산 LAN 환경에서의 클라이언트/서버 모델로 설계하였다. 성능은 처리 시간과 반환 시간을 통해 평가된다.

4.1 모의실험 모델

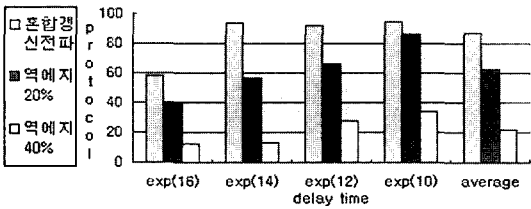
모의실험은 혼합 갱신 전파 알고리즘과 역에지 알고리즘을 비교함으로써 수행된다. 모의실험 환경은 MS Windows 2000 Server 환경을 설정하였고, 모델링 도구는 Visual Basic 6.0의 언어와 ARENA



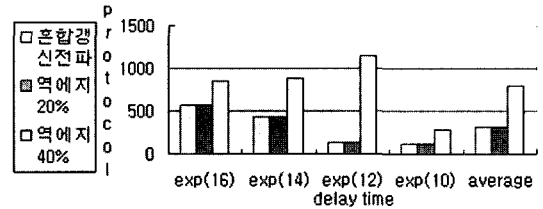
(그림 1) 사이트의 수가 4일 때 트랜잭션처리



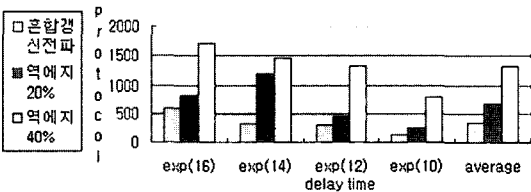
(그림 2) 사이트의 수가 7일 때 트랜잭션 처리



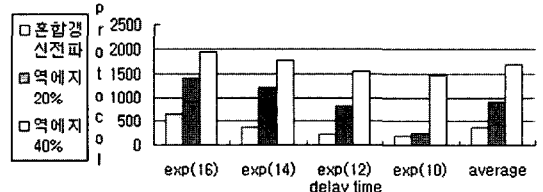
(그림 3) 사이트의 수가 10일 때 트랜잭션 처리



(그림 4) 사이트의 수가 4일 때 트랜잭션 반환시간



(그림 5) 사이트의 수가 7일 때 트랜잭션 반환시간



(그림 6) 사이트의 수가 10일 때 트랜잭션 반환시간

3.01이다. 모의실험 모델은 4개의 분산된 사이트와 2개의 역에지의 상태를 포함한다.

모든 사이트는 갱신 트랜잭션을 생성할 수 있고 갱신 트랜잭션을 위해 다른 사이트들에 부 트랜잭션을 만든다. 원본 사이트는 데이터 소유자가 주재하는 사이트를 의미한다. 그러므로 데이터에 의존하는 많은 원본 사이트들이 있다. 단지 소유자의 사이트는 사본 트리를 생성하는 함수들을 가지고 있다. 사이트들의 네트워크 통신 속도는 통신에 장애가 없고 모두 같은 것으로 가정한다. 혼합 갱신 전파 알고리즘은 전체 트랜잭션에서 역에지 비율에 따라 역에지 알고리즘의 두 가지 경우와 비교된다. 하나는 20% 역에지, 다른 하나는 40% 역에지이다. 역에지 비율이 더 많을 수록 갱신율도 증가한다. 만약 역에지 비율을 올린다면, 갱신 트랜잭션을 수행하기 위한 시간을 필요로 할 것이다.

4.2 성능 평가

모의실험 결과들은 평균 트랜잭션 처리시간과 평균 반환 시간을 포함한다. 그림 1,2,3을 보면 역에지 알고리즘의 트랜잭션 처리가 사이트의 수가 증가함에 따라 급격하게 감소되고 있는 것을 볼 수 있다. 그것은 사이트의 수가 증가함에 따라 역에지의 수가 증가하는 사실 때문이다. 역에지 처리는 또한 갱신이 증가함에 따라 록킹을 사용함으로써 감소시킨다. 반면에, 혼합 갱신 전파 알고리즘에서 처리는 사이트들의 수에 어떠한 주목할 만한 점 없이 변화 없는 낮은 비율을 유지하는 것을 볼 수 있다.

평균 반환 시간은 트랜잭션의 활성화와 마지막 사이트에서 트랜잭션 수행 사이의 평균 경과 시간으로 측정한다. 변환된 자료의 활성화 사이의 그림 4,5,6을 보면 비록 사이트의 수가 증가하였

어도 혼합 갱신 전파 알고리즘에서는 역에서 비교된 것 보다 더 적은 변화를 유지하는 것을 볼 수 있다. 40%의 역에서는 20%의 역에서와 혼합 갱신 전파 알고리즘보다 더 오랜 기간 지속되었다.

5. 결론

혼합 갱신 전파 알고리즘은 복제된 문서들의 일관성을 해결한다. 그리고 트랜잭션 스케줄의 직렬화를 보장하고 그룹 소유권에 대해 다중 읽기/다중 쓰기를 허락한다. 그것은 또한 원시 사이트의 일관성과 갱신을 위해서 즉시 버전 제어를 관리한다. 그것은 록킹 시간을 감소시킨다. 복제 갱신의 경우에 그것은 복제를 위해 갱신을 늦게 수행하고 갱신을 위해 경로의 길이를 감소시키는 사본 트리 경로로부터 사이트의 갱신을 삭제한다.

사본 사이트가 원시 사본에 갱신을 요구할 때, 혼합 갱신 전파 알고리즘은 동시에 두 문서를 갱신하고 수행한다. 그리고, 그 다음에 그것은 다른 사본 사이트들의 갱신을 수행한다. 그것은 록킹 시간을 감소시키기 때문에 데드록 가능성을 줄일 수 있다. 그것은 또한 평균 트랜잭션 처리율을 증가시키고 반환 시간을 감소시킴으로써 시스템 성능 향상에 기여한다.

참고문헌

[1] 이병욱, “복제 일관성을 위한 혼합 그룹 갱신 프로토콜 설계”, 한국인터넷정보학회논문지, 한국인터넷정보학회, pp. 71~80, 2001.
 [2] Parvathi Chundi, Daniel J.Rosenkrantz, S.S.Ravi, “Deferred Updates and Data Placement in Dis-

tributed Databases,” 12th Inter. Conference on Data Engineering, IEEE, 1996.

- [3] Yuri Breitbart, Raghavan Komondoor, Rajeev Rastogi, S. Seshadri, Avi Silbershatz, “Update propagation Protocols For Replicated Databases,” In Procs. of ACM SIGMOD Inter. Conference On Management of Data, Philadelphia, 1999.
 [4] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, G. Alonso, “Understanding Replication in Databases and Distributed Systems,” In Procs. of 20th Inter. Conference on DCS. IEEE, 2000.
 [5] Yuri Breitbart, Henry F. Korth, “Replication and Consistency: Being Lazy Helps Sometimes,” In Procs. of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Tucson, Arizona, 1997.
 [6] W, David Kelton, Randall P. Sadowski, Deborah A. Sadowski, “Simulation With ARENA,” McGrawHill, 1998 .
 [7] Dabid Mosberger, “Memory Consistency Models,” Operating System Review, ACM Jan. 1993.
 [8] Todd Anderson, Yuri Breitbart, Henry F. Korth and Avishai Wool, “Replication, consistency and practicality: Are these mutually exclusive?,” In Procs. of the ACM SIGMOD International Conf. on Management of Data, Seattle, WA, 1998.
 [9] M. Colton, “Replicated Data in a Distributed Environment,” In Procs. of ACM SIGMOD International Conf. on Management of Data, Washington, DC, 1993.
 [10] Jim Gary, Pat Helland, Patrick O’Neil, Dennis Shasha, “The Danger of Replication and a Ip Conf. on Management of Data, Montreal, Canada, 1996.

◎ 저 자 소개 ◎



이 병 옥

1973년 연세대학교 공학사

1984년 George Washington Univ. 대학원 전자계산학과 졸업(석사)

1994년 중앙대학교 대학원 전자계산학과 졸업(박사)

1985년~현재 : 경원대학교 소프트웨어대학 교수

관심분야 : 데이터베이스, 분산시스템, 전자거래학

E-mail : leebw@kyungwon.ac.kr