□특 집□

실시간 데이타베이스의 개론

숭실대학교 인공지능학과 이 상 호*

목 차

I. 서 론
II. 실시간 시스템의 특성
III. 실시간 데이타베이스 시스템
IV. 실시간 데이타베이스 시스템의 접근방법
4.1 주기억 상주 데이타베이스 시스템
4.2 데이타베이스 기능의 실시간화
V. 결 론

I. 서 론

차세대 컴퓨팅(computing)으로 부각되는 실시간 처리(real-time computing)는 도전적인 연구분야이면서, 실세계 응용에서 중요한 역할을 한다. 최근들어 하드웨어 및 인공지능 기법의 발달로 실시간 처리에 대한 관심이 고조되고 있다[IEEE91, ACM88]. 한편, 시간적 제약을 다루는 실시간 시스템은 시스템 고장(failure)이 경제적으로 사회적으로 생태학적으로 큰 재앙을 초래할 수 있기 때문에, 이에 대한 연구 및 개발은 활발하여야 하며, 최근들어 실시간 시스템을 구축하기 위하여 산업계, 정부에서는 많은 투자를 하고 있다.

실시간 처리란 시스템의 연산 정확성이 계산의 논리적 결과에만 의존하지 않고 결과가 도출된 시간에도 의존하는 시스템을 말한다. 특히 실시간처리 시스템이 사용하는 데이타 양의 증가로 데이타의 효율적인 저장과 조작을 위하여 기존 데이타베이스 시스템과 결합하게 되어, 이를 실시간 데이타베이스 시스템(real-time database system)이라 한다.

실시간 처리 시스템의 응용분야로는 항공우주(aerospace)산업(비행물체 제어, 우주항공선), 국방(defense)시스템(command and control system, 무기 시스템), 프로세스 제어(공장 및 실험실 제어), 공장자동화, 원자력 공장제어, 로보트, 주식거래, 우주정거장, 해저탐색, 교통제어시스템 등등 광범위하다.

본 고에서는 실시간 데이타베이스 시스템에 대하여 개괄적으로 고찰한다. 제 2장에서는 실시간 시스템의 특성 및 속성에 대하여, 일반적으로 실시간 처리에 대한 잘못된 인식을 중심으로 살펴본다. 제 3장에서는 실시간 데이타베이스 시스템을 기존 데이타베이스 시스템과 개괄적으로 비교, 설명한다. 제 4장에서는 실시간 데이타베이스의 두 가지 접근 방법에 대하여 설명하며, 제 5장에서 결론을 맺는다.

II. 실시간 시스템의 특성

일반적으로 현재까지 실시간 시스템에 대한 관심이 학계 및 산업계에서 부족하였으며 그 원인의 하나는 실시간 처리에 대한 잘못된 인식에 기인한다[Stan88]. 잘못된 인식으로 인하여 실시간 처리 시스템은 많은 관심을 받지 못했으며, 그 결과로 대부분의 실시간 처리 시스템 설계 및 구현이 ad hoc방법에 의존하여 왔다. 이 장에서는 이러한 잘못된 인식[Stan88]을 살펴봄으로써 실시간 시스템의 특성을 고찰한다.

* 정회원

첫째, 실시간처리는 빠른 계산(fast computing)을 의미하는 것이 아니다. 빠른 계산이란 주어진 작업을 수행하는데 평균 응답시간을 단축하려고 노력하지만, 실시간 처리란 각각의 작업에 연관된 시간 제약(timing constraints)을 만족하려고 노력한다. 실시간 처리는 빠른 계산을 종종 요구하지만 실시간 시스템의 중요한 성질은 예견성(predictability)이다. 즉 시스템의 기능 및 시간적 행위가 시스템 요구사항에 충족하는가를 예견할 수 있어야 한다. 빠른 계산은 실시간 처리에 도움은 되지만, 빠른 계산만으로는 예견성을 충족시킬 수 없다.

둘째, 슈퍼컴퓨터 하드웨어 발달이 실시간 처리 요구사항을 만족시킨다는 인식은 잘못되었다. 슈퍼컴퓨터는 다수의 프로세서를 사용한 병렬처리 기법을 사용하여 시스템 생산성(throughput)을 향상시키지만, 이것이 자동적으로 시간 제약을 만족하지는 못한다. 실시간 처리에 적합하도록 하드웨어 구조를 설계하지 않는 이상 프로세서와 그 부속 모듈이 시간 제약이 있는 작업을 마감시간(deadline) 내에 수행한다고 보장할 수 없다. 실제로 많은 하드웨어 장치를 사용할수록 실시간 작업 및 통신 스케쥴링이 복잡해진다. 현실적으로 실시간 처리는 공급을 능가하는 계산 능력을 항상 요구하였으며, 현재 슈퍼컴퓨터 하드웨어 발달은 실시간 시스템이 좀 더 향상된 기능과 성능을 추구하도록 영향을 미칠것이다.

셋째, 실시간 시스템 설계에는 체계적인 방법이 없다는 인식은 잘못되었다. 현재까지 실시간 시스템 설계가 ad hoc방법으로 하였지만, 이것이 실시간 시스템에 대하여 과학적인 접근이 불가능하다는 것을 의미하는 것이 아니다. 실제로 과학적인 접근방법을 필요로 하는 응용분야가 많이 있으며, 또한 체계적인 접근방법이 실시간 문제 해결에 도움을 준다.

넷째, 실시간 프로그래밍은 어셈블리 프로그래밍, 우선순위에 입각한 인터럽트 프로그래밍이라는 인식은 잘못되었다. 엄격한 시간 제약을 준수하기 위하여 현실적으로 실시간 처리 프로그래밍은 기계어 수준의 코딩 및 최적화에 많이 의존하여 왔으나, 이 방법은 시스템 개발 및 유지보수에 많은 시간과 노력을 필요로 한다. 이같은 접근방법은 실제로 대규모 실시간 처리 시스템 개발과 변경(modification) 과정에서 시스템 버그(bug)의 주 원인이었다. 실시간 처리 연구 목적중의 하나는 스케쥴링 이론과 최적화 기법을 활용하여 다수의 효율적인 프로그램과 자원 스케쥴링의 통합을 효율적으로 자동화하는 것이다.

다섯째, 실시간 처리 시스템은 정적 상황에서만 작동한다는 인식은 잘못되었다. 운용모드에 따라 실시간 시스템은 상이한 시간 제약을 만족시켜야 한다. 그러므로 실시간 처리 연구의 중요한 주제는 동적으로 변화하는 상이한 시간 제약조건에 적응력있게 작동하는 스케쥴링 기법을 고안하는 것이다. 실시간 처리 설계시 정적환경에만 시스템이 작동하도록 하는 접근방법은 바람직하지 않다.

종합적으로 실시간 처리 시스템의 특성을 정리하면 다음과 같이 요약될 수 있다.

- 시스템이 관리할 자원, 즉 시간 제약이 추가된다.
- 대규모이며 복잡성이 많다.
- 일반적으로 고성능 계산능력을 요구한다.
- 적응성(adaptability)이 있어야 한다.
- 예견성이 있어야 한다
- 신뢰성(reliability) 및 고장감뢰(fault tolerant) 기능이 강조된다.
- 동적환경에서 작동하여야 한다.
- 일반적으로 분산화된다.

III. 실시간 데이타베이스 시스템

실시간 데이타베이스 시스템은 트랜잭션(transaction)이 마감시간(deadline)이라 통칭하는 시간 제약(timing constraints)을 가지고 있어, 시스템의 정확성이 계산의 논리적 결과와 결과가 도출된 시간에 공히 의존한다. 실시간 데이타베이스 시스템의 트랜잭션은 연관된 마감시간내에 결과가 제공되도록 스케쥴링을 하여야 한다.

기존의 데이타베이스 시스템 설계시 주요 쟁점은 데이타의 무결성 유지, 시스템 비용 절감, 편리한 사용자 인터페이스 제공 등이다. 특히 질의어처리 단계에서는 디스크 접근 횟수를 최소화하도록 설계한다. 일반적인(conventional) 데이타베이스 시스템은 실시간 응용에 부적합하며 그 이유는 낮은 성능과 예견성(predictability) 부재이다. 기존의 데이타베이스 시스템에서는 대부분의 데이타가 디스크에 상주하므로, 트랜잭션 처리시간의 대부분을 디스크 접근시간에 할애한다. 디스크 접근시간은 보통 18~25 ms이므로 고성능 실시간 처리 응용분야에는 일반적으로 적합하지 않다. 또한 기존 트랜잭션 처리의 정확성 기준인 순차성(serialization)은, 제한시간이 결부되어 있는 실시간 트랜잭션에 적용하기에는 너무나 엄격한 조건이기도 하다. 순차성에 의한 트랜잭션 스케쥴링의

목적은 데이타 일치성(consistency)을 유지하면서 높은 동시성(concurrecny)을 제공하므로써, 트랜잭션의 평균 응답시간을 단축하는데 있다.

실시간 트랜잭션 처리에서는 각 트랜잭션에 결부되어 있는 마감시간 및 중요성(criticality)에 기초한 우선순위(priority)가 트랜잭션 스케쥴링에 반영되며, 낮은 우선순위 트랜잭션이 높은 우선순위 트랜잭션을 기다리게 하여 높은 우선순위 트랜잭션이 마감시간 내에 완료하지 못하는 현상을 피하여야 한다. 실시간 트랜잭션 스케쥴링에서는 적절한 우선순위 관리와 우선순위 상충(conflict) 해결이 실시간 데이타베이스 시스템의 예견성과 응답성(responsiveness)을 제공하는데 중요하다. 다시 말하여, 실시간 데이타베이스 시스템에서는 세 가지 제약에 즉, 데이타일치성, 트랜잭션 정확성, 트랜잭션 마감시간에 의거하여 동시성과 자원 효율성을 극대화하는 실시간 스케쥴링 및 동시성 제어 프로토콜을 개발하는 것이 중요하다.

실시간 데이타베이스 시스템의 트랜잭션은 크게 두 가지(soft 트랜잭션, hard 트랜잭션)로 구분될 수 있다. Hard 트랜잭션은 시간적 제한이 반드시 준수되어야 하는 경우를 지칭하며 만약 시간적 제약을 지키지 못하면 중대한 손실을 초래한다. 예를 들어, 비행 물체를 추적하여 대공 미사일을 발사하는 국방 시스템은 자연히 이 경우에 속한다. 한편 soft 트랜잭션은 시간적 제약이 존재는 하나, 시간제약을 지키지 못하면서 트랜잭션을 완료하여도 어느 정도의 가치가 있는 경우를 말한다. 일반적으로 soft 트랜잭션의 마감시간을 지키지 못해도 중대한 손실은 초래하지 않는다. Soft 트랜잭션은 시간적 제한을 고려하여 스케쥴링을 하지만 마감시간내 연산의 완료를 보장하지는 않는다. 각 트랜잭션의 구분은 각각의 응용분야에 의하여 대체적으로 결정되지만, 두 가지 트랜잭션을 공히 필요로 하는 실시간 시스템도 많이 있다.

IV. 실시간 데이타베이스 시스템의 접근방법

현존하는 데이타베이스 이론 및 기술에 의거하여 실시간 데이타베이스 시스템을 설계 및 구현하는 방법은 크게 두 가지로 요약될 수 있다. 첫째, 전통적인 데이타베이스 시스템의 낮은 성능의 주 원인인 디스크를 접근시간이 빠른 다른 매체로 대치하는 것이다. 주기억 상주 데이타베이스 시스템(main-memory database system)이 이 분류에 속한다. 둘째, 성능 향상을 위하여 트랜잭션 순차성의 일부를 무시하면서 트랜잭션을 실행시키는 등, 기존 데이타베이스 시스템 기능을 실시간 처리에 적합하도록 변경하는 방법이다.

4.1 주기억 상주 데이타베이스 시스템

일반적인 데이타베이스 시스템에서는 데이타를 디스크에 저장하므로 데이타 입출력시 디스크 접근을 하여야 한다. 디스크 접근은 대략 18~25 ms이므로 빈번한 디스크 접근은 고성능 데이타베이스 시스템을 구현할 때 큰 장애요인이다. 디스크 접근 횟수를 감소하여 데이타베이스 연산을 빠르게 하는 가장 보편적인 방법은 대다수의 데이타를 주기억장치에 상주시키는 주기억 상주 데이타베이스 시스템이다. 전체 데이타 전부를 주기억장치에 항상 상주하느냐, 대다수의 데이타를 상주하느냐에 의하여 main memory database system 또는 large main memery database system으로 구분하기도 한다.

주기억 상주 데이타베이스 시스템에서는 입출력으로 의한 지연이 거의 없으므로 빠른 응답을 사용자에게 제공할 수 있다. 또한 최근의 고집적 회로 기술의 발달과 칩 가격 하락으로 주기억 상주 데이타베이스 시스템 실현이 용이해졌다. 주기억 상주 데이타베이스 시스템은 기존의 데이타베이스시스템과 비교하여 다음과 같은 새로운 문제점을 제기한다.

휘발성인 주기억장치는 시스템 파손이 되었을 때 기억된 내용을 모두 잃어버리므로, 주기억 상주 데이타베이스 시스템에서는 전체 데이타베이스를 파괴하는 결과를 초래한다. 그러므로 주기억 상주 데이타베이스 시스템에서도 데이타 및 로그 저장용으로 안정된(stable) 기억장치로서 디스크가 필요하다. 데이타베이스가 주기억장치에 상주함으로 갱신 연산의 결과가 디스크에 있는 데이타베이스에 반영되지 않을 수 있으며, 이 경우에는 시스템 파손이 되었을 때 신속한 회복이 가능한 회복기법이 필요하다. 일반적인 데이타베이스시스템에서는 시스템 파손시 디스크에 있는 데이타베이스를 주기억 장치에 적재한 후 로그의 내용을 참조하여 데이타베이스 상태를 최신의 내용으로 회복하는데, 이러한 회복기법[HaRe83]은 실시간 처리에는 일반적으로 적합하지 않다.

주기억 상주 데이타베이스 시스템에 적합한 다양한 회복기법이 제안되었다. 대부분의 회복기법은 주기억장치의 일부분이 밧데리를 사용하여 안정적임을 가정하고 있으며, 이 부분에 주로 로그의 내용을 기록한다. 주기억 상주 데이타베이스 시스템에 적합한

회복기법으로 신속한 회복을 위하여 여러가지 방법이 제안되었다. Non-interfering checkpoint 및 Log compression 방법[Son87], Overlapped checkpointing 및 Reduced logs[DKO84], Working set개념을 이용한 방법[LeCa86] 등이 회복시간을 단축시키기 위하여 제안되었다.

일반적인 데이타베이스 시스템에서는 데이타구성(organization), 접근방법, 질의어 처리를 위한 알고리즘이 중앙처리장치(CPU) 사용시간보다 디스크 접근시간을 단축하는 방향으로 설계되어 왔다. 이는 일반적인 데이타베이스 시스템은 평상 소량의 데이타를 시스템에 버퍼에 유지하기 때문이다. 기존 데이타베이스 시스템에 적합하게 고안된 알고리즘이 주기억상주 데이타베이스 시스템에서는 좋은 성능을 기대하기 어려울 경우가 있다. 또한, 주기억 상주 데이타베이스 시스템에서는 한정된 주기억 장치에 대량의 데이타를 유지하여야 함으로 기억공간 사용 효율성이 중요한 문제가 된다[Bitt86]. 즉, 주기억 상주 데이타베이스 시스템에서는 처리시간 단축과 기억공간 사용 효율성을 모두 고려하여 알고리즘이 고안되어야 한다. 일반적인 데이타베이스 시스템에 적합한 접근방법 및 죠인 알고리즘 등이 주기억 상주 데이타베이스 시스템에서는 비효율적이라는 모의 실험 연구 결과도 문헌에서 발견할 수 있다[Shap86].

Lehman과 Carey[LeCa87]는 질의어 처리 관점에서 주기억상주 데이타베이스 시스템의 선택(seclection), 죠인, 투영(projection) 알고리즘 성능 연구를 하였으며, AVL트리와 B트리 구조를 결합한 "T트리"가 선택과 죠인 연산에서 성능이 우수함을 보였다. 또한 질의어 최적화 단계에서는 밀집화(clustering)과 투영 크기 축소 현상이 고려되지 않음으로, 일반적인 데이타베이스 시스템보다 주기억상주 데이타베이스 시스템이 일반적으로 더 간단하다.

주기억상주 데이타베이스 시스템에서는 디스크 접근시간이 거의 없으므로 트랜잭션이 동시성 제어를 위한 록킹시간이 비교적 짧으며, 결과적으로 블록킹에 의한 지연시간이 짧게 된다. 그러므로 주기억상주 데이타베이스 시스템에서는 록킹 단위(granularity)를 크게 할 수도 있다. 큰 록킹 단위는 충돌을 빈번이 발생시킬 수 있으나 전체적으로 부대비용은 감소한다. 또한 주기억상주 데이타베이스 시스템에서는 기억공간이 중요함으로, multiversion에 의한 동시성 제어 및 임시적으로 데이타 공간을 생성하는 동시성 제어 방법 등은 적합하지 않을 수 있다.

4.2 데이타베이스 기능의 실시간화

일반적인 데이타베이스 시스템의 성능을 높이기 위한 다른 방법은 데이타베이스 시스템의 기능을 축소, 완화 또는 변경하여 실시간처리에 적합하게 하는 것이다. 개괄적으로 축소, 완화, 변경하는 기능으로는 두 가지로 구분할 수 있다. 즉 편리한(friendly) 사용자 인터페이스 기능축소와 트랜잭션의 직렬성 완화이다.

일반적으로 실시간 데이타베이스 시스템은 특수 목적에 맞게 설계된 실시간 응용시스템과 직접 인터페이스를 하게 되므로 성능향상을 위하여 사용자 인터페이스의 편리성을 축소할 수 있다. 즉, 사용 편리성보다 신속한 처리 및 효율적인 구현에 적합한 데이타 모델링 또는 데이타 조작언어를 실시간 데이타베이스 시스템에서는 선택할 수 있다.

다수의 사용자가 동시에 데이타베이스를 사용할 때 데이타베이스의 일치성을 위하여 동시성 제어를 하는데, 직렬성이 가장 널리 사용되는 정확성 검사기준이다[BHG87]. 트랜잭션을 직렬성에 의거하여 수행하면 트랜잭션의 블록킹 또는 재시작(restart) 등으로 인하여 성능이 저하된다. 그러므로 실시간 데이타베이스 시스템에서는 직렬성 완화로 성능을 향상시키는 방법이 제안되었다. 일반적으로 트랜잭션의 직렬성은 실시간 처리에는 매우 엄격한 조건이며, 비직렬성 트랜잭션 수행이 허용되는 실시간 응용분야가 있다. 실시간 응용분야에서는 경우에 따라 부분적으로 맞는 데이타를 얻는 것이 제한시간 이후에 정확한 데이타를 접근하는 것보다 필요할 수가 있다. 즉 실시간 트랜잭션 스케쥴링의 목적은 크게 두 가지로, 데이타의 일치성을 유지하는 것과 시간적 제약을 준수하는 것이며, 이 두 가지 목적은 결합되기가 일반적으로 어렵다[Son90]. 일반적으로 실시간 데이타베이스에서 동시성 제어 접근방식은 기존의 방법(특히 2단계 록킹)을 사용하면서 긴급한 트랜잭션에 우선순위를 높게 부여하는 스케쥴링을 적용하는 방식이다[AbGa88, SRL88].

통상 응용분야에 대한 사전지식(prior knowledge)이 트랜잭션의 직렬성을 완화하는데 사용된다. 예로서 화학공장을 제어하는 실시간 데이타베이스 시스템을 가정해 보자. 즉 갱신 트랜잭션은 주기적으로 온도, 압력, 속도 등을 갱신하며, 다른 읽기 트랜잭션은 이 값을 읽어 임의의 조건을 만족하면 적당한 조치(즉, 밸브개폐, 경고음 발생 등)를 취한다고 하자. 이런 경우, 직렬성은 필요이상으로 강한 조건이 된다. 갱신

트랜잭션이 갱신 작업을 수행중이라도 읽기 트랜잭션에게 데이타를 접근시켜 필요한 조치를 취하게 할 수 있기 때문이다. 일반적으로 범용 비직렬성 동시성 제어 기법을 고안하기는 어려우나, 다수의 비직렬성 동시성 제어가 여러 실시간 응용분야에서 데이타의 일치성을 유지하며 적용될 수 있다. 실시간 데이타베이스 시스템은 특수목적에 적합하게 설계되고, 이런 시스템에서는 트랜잭션의 수가 제한됨으로, 데이타 및 트랜잭션에 대한 의미적인 성질을 미리 분석하여 비직렬성 트랜잭션 수행이 가능하다. Garcia-Molina [Garc83]는 트랜잭션의 의미를 이용하여 데이타베이스 일치성을 유지하면서 비직렬성 동시성 제어 방법을 제안하였다.

Lin과 Son[LiSo90]은 우선순위에 의거한 록킹방식과 낙관적인 방식을 결합한 방식을 제안하였다. 이 방법은 높은 우선순위 트랜잭션을 먼저 실행함으로써, 높은 우선순위 트랜잭션이 완료(commit)되지 않는 낮은 우선순위 트랜잭션에 의하여 블록되지 않으며, 또한 낮은 우선순위 트랜잭션이 상충에 의해서도 반드시 철회(abort)되지 않는다. 일반적인 상황에서는 록킹에 의한 동시성제어가 낙관적인 방식보다 우월하다고 알려져 있으나[ACL87], 실시간 응용에서는 상기와 같은 결합된 방식이 우수한 성능을 제공할 수도 있다[Son90].

실시간 트랜잭션 처리에서 중요한 문제중 하나는 정확한(correct) 수행에 대한 새로운 개념이 필요하다. 실시간 데이타베이스에서는 시간적 제약이 데이타 일치성보다 더 중요할 수 있다는 특성을 강조하여, 데이타 일치성 조건을 임시적으로 위반하면서 시간적 제약을 준수하려는 새로운 시도가 있었다[Lin89, VrLi88]. 이 방법은 내부적 데이타 일치성보다 외부적 데이타 일치성을 우선적으로 유지하는 새로운 데이타 일치성 모형에 기반을 둔다. 아직까지 직렬성보다 엄격하지 않으면서 실시간 데이타베이스에 적합한 일치성 모델(model)은 아직 제안되지 않고 있다[Son90]. 문제는 임시적 데이타 비일치성이 활동중인 트랜잭션의 결과에 영향을 미칠 수 있으며, 이러한 트랜잭션 완료는 데이타의 비일치성이 제거될 때까지 연기되어야 하며, 그렇지 않으면 완료된 트랜잭션을 철회하여야 하는 경우가 발생한다. 그러나 실시간 응용에서는 이미 완료된 트랜잭션의 결과를 무효화할 수 없는 경우도 있다.

경우에 따라 부분적으로 맞는 데이타를 얻는 것이 제한시간 이후에 정확한 데이타를 접근하는 것보다 필요한 실시간 응용분야에서는, 데이타 일치성을 파괴하는 트랜잭션을 철회(rollback)하는 것이 비효율적일 수가 있다. 특히 트랜잭션의 결과를 원상 복귀할 수 없는 경우에는 기존의 트랜잭션 철회 및 재시동에 의존하는 회복기법은 적합하지 않다. 실시간 데이타베이스에서는 기존의 backward 회복기법보다 forward 회복기법이 적합할 수 있다. 실시간 데이타베이스 시스템에서 비직렬성 트랜잭션 수행에 의한 데이타의 비일치성을 감수하면서도 시스템의 성능 향상을 하는 또 다른 이유는 데이타의 유효시간이 실시간 응용에서는 매우 짧은 경우가 많다. 예를 들면, 레이다가 비행물체를 추적하는 시스템에서 데이타의 유효성이란 매우 적은 시간이므로 비일치성 데이타가 전체 시스템에 널리 퍼지지 않으며, 이런 경우 시스템 파손시 유효시간이 지난 데이타 회복을 하는 것은 비효율적이다.

V. 결 론

본고에서는 실시간 데이타베이스 시스템의 개략적인 사항에 대하여 고찰해 보았다. 먼저 실시간 처리 시스템의 정의 및 특성을 살펴보았고, 실시간 데이타베이스 시스템을 구현하는 두 가지 접근방법에 대하여 살펴보았다.

실시간 데이타베이스 시스템은 제어시스템, 군 장비 시스템 등 실세계에서 광범위하게 사용되고 있으며, 실시간 처리에 대한 과학적인 연구가 요망된다. 실세계에서 사용되는 실시간 데이타베이스 시스템 설계 및 개발을 위해서는 보다 많은 노력이 요구되며, 분산시스템, 고장감뢰 시스템, 운영체제 등 연관있는 분야의 공동 노력이 필요하다.

참 고 문 헌

[AbGa88] R. Abbott and H. Garcia-Molina, "Scheduling Real-Time Transactions; A Performance Study," Proceedings of the International Conference of Very Large Data Bases, pp. 1~12, 1988.

[ACM88] ACM SIGMOD Record, Special Issue on Real-Time Database Systems, Vol. 17, No. 1, 1988.

[ACL87] R. Agrawal, M.J. Carey, and M. Linvy, "Concurrency Control Performance Modeling: Alternatives and Implications", ACM Trans. on

Database Systems, Vol. 12, No. 4, 1987.

[BHG87] P. Bernstein, V. Hadzilacos and N. Goodman, Concurrency Control and Recovery in Database Systems, Addison-Wesley, 1987.

[Bitt86] D. Bitton, "The Effect of Large Main Memory on Database Systems," Proceedings of ACM SIGMOD, pp. 337~339, 1986.

[CJL89] M. Carey, R. Jauhari, and M. Livny, "Priority in DBMS Resource Scheduling," Proceedings of the International Conference of Very Large Data Bases, pp. 397~410, 1989.

[ChDe85] H-T. Chou and D. DeWitt, "An Evaluation of Buffer Management Strategies for Relational Database Systems," Proceedings of the 11th International Conference on Very Large Data Bases, pp. 127~141, 1985.

[DKO84] D. DeWitt, R. Katz, R. Olken. L. Shapiro, M. Stonebraker and D. Wood, "Implementation Techniques for Main Memory Database Systems," Proceeedings of ACM SIGMOD, pp. 1~8, 1984.

[Garc83] H. Garcia-Molina, "Using Semantic Knowledge for Transaction Proceesing in a Distributed Databases," ACM Trans. on Database System, Vol. 8, No. 2, pp. 186~213, 1983.

[HaRe83] T. Haerder, and A. Reuter, "A Principles of Transaction-Oriented Database Recovery," ACM Computing Surveys, Vol. 15, No. 4, pp. 287-317, 1983.

[IEEE91] IEEE Computer, Special Issue on Real-Time Systems, Vol. 24, No. 5, 1991.

[Leh86] T. J. Lehman, Design and Performance Evaluation of a Main Memory Relational Database System, Ph/D Dissertation (also as Technical Report No. 655), Dept. of Computer Science, University of Wisconsin at Madison, 1986.

[LeCa86] T. J. Lehman and M. Carey, "Query Processing in Main Memory Database Management Systems," Proceedings of ACM SIGMOD, pp. 239~250, 1986.

[Lin89] K. Lin, "Consistency Issues in Real-Time Database Systems," Proceedings of Hawaii International Conference on System Sciences, 1989.

[Sing88] M. Singhal, "Issues and Approaches to Design of Real-Time Database Systems," ACM SIGMOD Record, Vol. 17, No. 1, pp. 19~33, 1988.

[SRL88] L. Sha, R. Rajkumar, and J. Lehoczky, "Concurrency Control for Distributed Real-Time Databases," ACM SIGMOD Record, Vol. 17, No. 1, pp. 82~98, 1988.

[Shap86] L. Shapiro, "Join Processing in Database Systems with Large Memories," ACM Trans. on Database Systems, Vol. 11, No. 3, pp. 239~264, 1986.

[Son87] S. H. Son "A Recovery Scheme for Database Systems with Large Main Memory," Proceedings of the 11th Computer Software and Applications Conference, pp. 422~427, 1987.

[Son90] S. H. Son, "Real-Time Database Systems: A New Challenge," Data Engineering, Vol. 13, No. 4, 1990.

[Stan88] J. Stankovic, "Misconceptions about Real-Time Computing: A Serious Problem for Next-Generation Systems," IEEE Computers, Vol. 21, No. 10, 1988.

[VrLi88] S. Vrbsky, and K.J. Lin, "Recovering Imprecise Transactions with Real-Time Constraints," Symposium on Reliable Distributed Systems, pp. 185~193, 1988.

이 상 호

1984 서울대학교 컴퓨터공학과 학사
1986 Northwestern University 전산학 석사
1989 Northwestern University 전산학 박사
1990 ~1992 한국전자통신연구소 선임연구원
1992 ~현재 숭실대학교 인공지능학과 조교수
관심 분야 : 데이터베이스 시스템, 자동연역(Automated Reasoning), 실시간 처리